EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

Trust and Security Enhancement in Cloud Networks Using a Multiscale-Dilated Depthwise Separable Convolutional Network

Anita Sachin Mahajan[1], Deepti Varshney[1], Shruti Gupta[2], Surjeet[3], Vivek Srivastava[4,*], Abhishek Bhatt[5]

[1]Department of Computer Engineering, Ajeenkya D Y Patil School of Engineering, Lohegaon, Pune, Maharashtra, India
[2]Department of Computer Science and Engineering, ABES Engineering College, Ghaziabad, Uttar Pradesh, India
[3]Department of Electronics & Communication Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India
[4]Department of Computer Science and Engineering, Pranveer Singh Institute of Technology, Kanpur, Uttar Pradesh, India
[5]Department of Computer Science and Engineering, Uttar Pradesh Textile Technology Institute, Kanpur, Uttar Pradesh, India

Abstract

In cloud networks, maximum resource utilization, high trust, and security remain basic issues. To overcome such inherent drawbacks, this article proposes a novel deep learning framework to optimize trust and security in cloud networks. Initially, hierarchical attribute-based encryption is employed here to encrypt personal data and hide users' locations and reputations, thereby effectively preventing re-identification attacks. For resource allocation, a multiscale-dilated depthwise separable convolutional network is employed, which dynamically adapts to varied workloads and user request demands. Also, the model leverages key input parameters, such as available resources, job size, and time limit, to generate optimally adjusted resource distribution plans that avoid resource waste and ensure high service efficiency. To further enhance the trustworthiness of cloud services, an orangutan optimization algorithm-based trust assessment module is incorporated. It considers both trust scores and service performance metrics as input parameters to evaluate trustworthiness, thereby supporting more accurate and responsive trust judgments among nodes in a distributed network. The proposed framework is experimentally tested on the Integrated Cloud Security 3Datasets, which includes real-world cloud attack and performance information. The suggested framework not only enhances cloud operational transparency and security but also provides a robust, stable cloud environment. The model is implemented in Python and evaluates performance metrics including accuracy, recall, F-measure, precision, execution time, and mean absolute error. It demonstrates improved accuracy, improved precision, and lower execution time than other comparable schemes.

Keywords: Cloud Security, Trust Assessment, Attribute-Based Encryption, Depthwise Separable Convolution, Resource Allocation, Metaheuristic Optimization

Received on 14 September 2026, accepted on 30 October 2026, published on 13 November 2026

Copyright © 2026 Anita Sachin Mahajan *et al.*, licensed to EAI. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetiot.15122

*Corresponding author. Email: vivek.srivastava1@psit.ac.in

1. Introduction

With digital transformation, cloud computing (CC) has emerged as a core technology for data storage, computation, and service delivery [1]. Additionally, CC enables an organization to use scalable and on-demand resources without direct control over physical infrastructure [2]. However, this increasing dependence on the cloud environment creates significant security and trust concerns. The root causes of these are issues such as

unauthorized access, leakage of data, insider threats, and misuse of services, which have become more frequent due to the shared and distributed nature of the cloud resources [3]. Therefore, assuring data integrity, confidentiality, and availability has become a big issue for CC providers and users. Most of the traditional security features, like firewalls, encryption, and authentication models, are unable to handle dynamic behaviors of cyber threats in CC [4]. Besides this, attackers exploit vulnerabilities with sophisticated models, and static defense measures are ineffective. Moreover, trust between cloud providers and users cannot be established due to insecurity and the potential for manipulation or disclosure of data [5]. Hence, adaptive security models that can track malicious activities and counter them reliably in cloud environments are most needed at present [6].

Nowadays, deep learning (DL) models have emerged as a robust antidote to overcome these cloud network security challenges [7], [8]. Here, these models can automatically learn high-level patterns from large-scale, heterogeneous data to accurately detect threats and calculate risks. When trust management functionality is integrated with the DL model, cloud networks will be able to dynamically analyze user activity, authenticate requests, and detect intrusions in real time [9]. In this paper, we establish a trust and security model within cloud networks using a DL-based approach to ensure a strong and adaptive defense mechanism that provides secure data transmission, efficient utilization of resources, and ensures better reliability in cloud services.

Even with the several advantages of CC, maintaining trust and security in cloud networks remains the greatest challenge, given the increasing sophistication of attacks and the lack of transparency in service operations. Conventional security approaches are generally unable to address dynamic, large-scale cloud environments where data leakage, denial-of-service attacks, and insider misuse can occur at any level of the infrastructure. Furthermore, the decentralized and multi-tenant nature of cloud platforms makes the measurement of consumer- and provider-generated trust difficult. This, therefore, demands an adaptive, intelligent, and automated security system capable of detecting anomalies in the services offered, forecasting threats, and ensuring secure interaction between consumers and cloud service providers. The proposed framework leverages the sophisticated pattern recognition and decision-making capabilities of DL to create an adaptive trust management and intrusion detection (ID) model that securely strengthens cloud security and user trust in ever-evolving threat scenarios. The major contributions of the manuscript are described below.

- The presented framework incorporates Hierarchical Attribute-Based Encryption (HiAbE) to safeguard user locations, reputations, and data, efficiently eliminating re-identification threats in cloud networks.
- The Multiscale-dilated Depthwise Separable Convolutional Network (MDS-CNet) dynamically adapts to varying workloads, resource availability, and time urgency, allocating computing resources optimally for maximum service efficiency with minimal wastage.
- The Orangutan Optimization Algorithm (OOA)-based trust evaluation module provides a biologically inspired optimization method to assess trustworthiness based on both trust scores and performance indicators of services, resulting in more trustworthy and accurate trust management among distributed cloud nodes.
- The model is tested on the Integrated Cloud Security 3Datasets (ICS3D), which comprises heterogeneous cloud performance and attack information, assuring the functionality of the model.
- A combination of DL with optimization-based trust mechanisms yields a robust framework that significantly enhances the aspects of transparency, reliability, and security of cloud networks, thereby motivating the development of a trustworthy and robust cloud ecosystem.

The remaining sections of the manuscript are ordered as follows. Section 2 explains the existing works related to trust and security enhancement in cloud networks. Section 3 details the proposed approach. The results are presented in Section 4, and the manuscript concludes in Section 5.

2. Recent Research Work: A Brief Review

This section emphasizes recent works on trust and security enhancement in cloud networks. Foundational studies have identified the security and privacy challenges inherent in cloud service delivery models and multi-tenant cloud environments [10], [11], and ciphertext-policy attribute-based encryption established the basis for fine-grained access control over shared cloud data [12]. Saini et al. [13] presented a hybrid approach that provides advanced privacy protection, trust estimation, and efficient resource assignment for the safe sharing of cloud-based data. It utilizes k-anonymity for anonymizing participants, the Time-aware Modified Best Fit Decreasing (T-MBFD) algorithm for dynamic resource assignment, and an integrated Optimized Levenberg–Firefly Algorithm (LFA) for precise trust estimation. This mechanism efficiently weighed privacy preservation, trust precision, and computational cost in changing cloud scenarios. The method exhibited better performance in accuracy, precision, and scalability as opposed to current practices. Nevertheless, it relied on the use of big datasets and algorithmic sophistication at the cost of somewhat higher computational overhead under load conditions.

Lv *et al.* [14] presented a Digital Twin (DT)-based network security system combining DL for improved ID and trust management. It employed a Deep Belief Network (DBN) for network ID and a Keyed-Hashing-based Self-

Synchronization (KHSS) model to predict and maintain trust in cloud service systems. The approach virtually modeled the network environment, optimized the ID data, and forecasted high-precision cloud security states. Here, the DT-DBN-KHSS model reached an improved detection rate, reduced error margins, and speedier performance compared to traditional Support Vector Machine (SVM)-based techniques. However, its efficiency depended on good modeling of network dynamics and required large computational resource consumption due to its large-scale implementation.

Saini *et al*. [15] presented a hybrid privacy-preserving scheme as spatial K-anonymity (CSKA) and privacy-preserving incentive and reward (PPIR), aimed at enhancing trust and security in CC systems. It used a hybrid approach combining future-proof privacy preservation methods and trust-based service mechanisms, and experimented with several datasets. Here, the approach proved to be superior in accuracy, precision, and F-measure, demonstrating its capacity to properly safeguard confidential information and guarantee the secure operation of cloud services. Nevertheless, it performed slightly worse under infinitely high data loads, which suggests an improvement is needed to preserve scalability under large-scale cloud conditions.

Mohammed and Rangu [16] presented a new Goat-based Recurrent Forensic Mechanism (GbRFM) to effectively detect malware and classify it in cloud applications. It transformed the input dataset through a hidden stage, obtained features without errors, and employed a continuous model that was trained on the hidden layer outputs for the correct detection of attack types. Here, the use of the wild goat optimization (WGO) model enhanced the convergence and feature selection of the model for improved detection accuracy. Experimental evaluation proved that the model exhibited high accuracy, proving to be better than current malware detection methods. The use of long feature training and optimization involved increased computational costs, thus making it less adaptable to real-time cloud large-scale environments.

Nizamudeen [17] presented an intelligent ID framework for securing multi-cloud Internet of Things (IoT) networks against network and application-based assaults. This model includes a data pre-processing step based on Integer-Grading Normalization (I-GN), efficient feature selection using the Opposition-based Learning–Rat Inspired Optimizer (OBL-RIO) model, and a 2D-Array-based Convolutional Neural Network (2D-ACNN) for classification. The mechanism, hence, identified normal and abnormal traffic with improved accuracy rate, diminished false positives, and stability in features. Thus, the model excelled in delivering superior performance on both accuracy and detection rate. However, the sophisticated optimization and 2D feature model introduced computational overhead, thus limiting real-time instantiation on resource-limited IoT devices.

Ahmad et al. [18] presented the Secure Policies based Cloud Security Framework (SPCSF) for improving application privacy and keys management in the cloud. Here, the method relied on a permission detection engine that provided secure application authentication for Representational State Transfer (REST) Application Programming Interface (API) consumption. Also, the approach efficiently protected the cloud resources from malicious usage and access. Thus, it provided greater compliance and security and managed application interactions across multiple cloud platforms, but it had higher computational complexity.

Tamizshelvan and Vijayalakshmi [19] proposed a hybrid machine learning-based privacy-aware and secure cloud data management for IoT environments. The approach utilized a Distributed Data Service (DDS) to collect and screen data, with supply chain quantum neural networks (QNN), multi-agent federated Bayes vectors, blockchain (BC), and partial homomorphic encryption utilized for enhanced privacy and security. Also, this approach effectively offered secure data access, integrity, and governance with scalability and low latency in distributed cloud systems. The model demonstrated robust protection of sensitive data and improved performance compared to traditional methods. However, the application of more than one sophisticated method increased computational complexity and implementation overhead.

3. Proposed Methodology

This paper presents a privacy and trust framework for enhancing privacy and trust within CC platforms. The proposed approach incorporates HiAbE for protecting sensitive user data, like identity and location, by encrypting every user's data and hence reducing the opportunities for re-identification. The second is the improved job allocation model, which utilizes the MDS-CNet to allocate the jobs to cloud providers effectively with regard to resource suitability and workload equilibrium. After that, to enhance the credibility of cloud services, a trust assessment module based on OOA is integrated. By incorporating effective resource management into privacy protection, the framework gives a full and realistic solution for secure and reliable operations in clouds. The block diagram representation of the presented approach is shown in Figure 1.

3.1. Input Acquisition

The ICS3D is a large-scale cybersecurity dataset [20] with a total of 18.9 million security records occupying 8.4 GB. It contains attack scenarios along with system performance measurements, making it sufficient for training and evaluating DL models in cloud and distributed environments. These security records are used for training, validation and testing purposes of the proposed model in an 80:10:10 ratio.

The records of the ICS3D dataset are prepared as follows before they are supplied to the proposed framework.

(i) Data cleaning
(ii) Encoding
(iii) Normalization: the numerical attributes, including the available resources, job size, request frequency, and time limit, are normalized before being given to the input module.
(iv) Labelling: each record is labelled as benign or malicious for the binary detection task evaluated in Section 4.
(v) Sampling and splitting: subsets of 20,000, 40,000, 60,000, 80,000, and 100,000 records are drawn for the comparative experiments in Section 4.2, and each subset is divided into training, validation, and testing sets in an 80:10:10 ratio.

3.2. Secure User Information using HiAbE

HiAbE is an advanced encryption technique that safeguards user data by stacking encryption keys and attributes in hierarchies [21]. Here, data owners are allowed to encrypt the information according to pre-defined attributes, such as identity, role, or location of users. This hierarchy allows higher authorities, like administration, to provide privileges for decryption to lower authorities without revealing the main private key. Furthermore, each user will be assigned a unique attribute-based key, and data decryption can only be done when a user's attribute satisfies the access policy set by the data owner. To begin with, HiAbE protects the data, including user identity and location information, by encrypting information based on more than one level of user attribute in a hierarchical manner. The very first step begins with the generation of a master key by a Root Private Key Generator (PKG), which then offloads key generation to lower entities, such as cloud servers, that further distribute attribute-based keys to individual users or devices. Every encrypted data block will be accessible to users whose attributes satisfy the specified access policy, providing fine-grained and distributed control of data access. By encrypting user data using such multi-layer encryption, HiAbE effectively mitigates unauthorized access and reduces user re-identification risks in cloud settings.

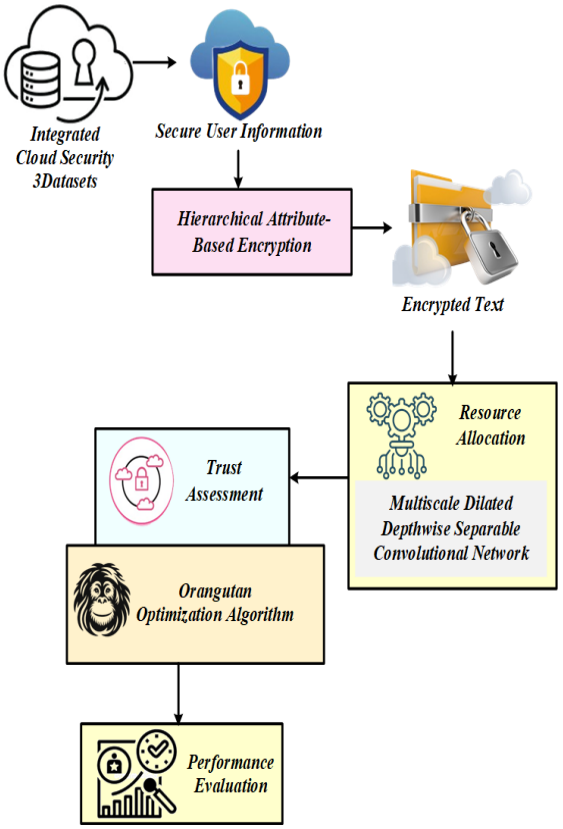


Figure 1. Block Representation of the Proposed Methodology

3.2.1. Key Generation and Setup

In this, HiAbE provides secure and fine-grained access control by organizing the encryption system in hierarchical layers: a root authority, cloud servers, and users. In the initial setup phase, the PKG picks random elements from a bilinear group B_0 of prime order P and the master secret key (MSK) $Msk_1^0 \in Z_p$, where Z_p denotes the set of integers modulo the prime P. Also, the master public key (MPK) is represented as $Mpk_1 \in B_0$, the set of attribute-based specific randomness (SR) is denoted as $SR = \{s_1, s_2, s_3, \ldots..,s_N\} \in Z_p$. Additionally, the hash functions of the model are denoted as $h_1:\{0,1\}^* \rightarrow B_0$ and $h_2:\{0,1\}^* \rightarrow B_T$ that are utilized to transform identity attributes into cryptographic keys, where $\{0, 1\}^*$ denotes the set of binary strings of arbitrary length and B_T is the target group of the bilinear pairing. These values are critical for key generation at the lower layers without leaking the master key.

3.2.2. Key Derivation based on Cloud and User Level

After the root node is initialized, the cloud server node produces its own secret based on a random value $r1 \in Zp$. It is used to compute the MPK as $Mpk_2 = h_1\left(ID_C \| ID_R\right)$, where ID_C and ID_R denote the identities of the cloud server

and the root authority, respectively, and ‖ denotes concatenation. The MSK of the cloud server, Msk_2, is then computed based on the root key, as given in equation (1).

$$Msk_2 = Msk_1 \cdot Mpk_2 \quad (1)$$

Moreover, this process computes the shared public variable for further distribution, which is denoted as S_v. After that, the user attains the secret and public keys from the server. The identity of the user is ID_U and the public key is Mpk_3; then, the secret key of this user, Msk_3, is calculated using equation (2).

$$Msk_3 = Msk_2 + r1 \cdot Mpk_3 \quad (2)$$

The random value $r1$, the new random value $r2 \in Z_p$ and the shared public value $Sv = SR.Mpk_2$ are created to ensure cryptographic diversity among users.

3.2.3. Process of Encryption and Decryption

Here, the random value $r \in Z_p$ is selected for encrypting a message $N = \{n_1, n_2, ...n_k\}$. Additionally, the content keys $Ck = Ck_1 + Ck_2 + ... + Ck_k$ are embedded using a hierarchical access tree as (T) and the encrypted ciphertext is computed using equation (3).

$$Ciphertext = \left\{T, C_i, C_i', C_{(x,y)}, C'_{(x,y)}, \vec{C}_{(x,y),j}\right\} \quad (3)$$

Here, (x,y) denote the nodes of the access tree T, and (j) denotes the index; C_i, C_i', $C_{(x,y)}$, $C_{(x,y)}'$, and the vector term are the ciphertext components generated for the i-th attribute and for node (x, y) of the access tree, respectively. For decryption, the right secret key and corresponding attributes must be held by the user. By applying bilinear pairing operations, decryption is carried out through equation (4).

$$Decry(Ciphertext, SK, (x,y)) = \frac{e\left(D_i, C_{(x,y)}\right)}{e\left(D_i', C'_{(x,y)}\right)} \quad (4)$$

Here, $e(\cdot,\cdot)$ denotes the bilinear pairing, SK is the secret key of the user, and D_i and D_i' are the secret-key components corresponding to the i-th attribute. This procedure guarantees that only users whose attribute set meets the policy inherent in the access tree (T) can decrypt the message successfully. Therefore, HiAbE is an efficient privacy-preserving as well as trust-enhancing solution for cloud systems. Through hierarchical user attribute-based encryption of data and anonymization of both data producers and consumers, it guarantees that only legitimate parties can access the private data. Thus, the multi-layered key management process not only ensures data re-identification prevention and illegal disclosure protection but also enhances confidentiality and trust in cloud interactions, thus being a secure solution for data sharing and access control.

3.3. Resource Allocation Using MDS-CNet

The MDS-CNet aims to achieve intelligent and adaptive resource allocation for CC environments [22]. The MDS-CNet architecture was originally introduced for audio classification [22]; in this work, only its convolutional structure is adopted, and its input and output layers are redefined for cloud workload and resource data, so that no audio, speech, or voice-command component is used. Here, the model self-tunes to different workloads and user request patterns through evaluation of the most important parameters, including available resources, time limits, and job size. In contrast to conventional static allocation methods, the MDS-CNet algorithmically learns intricate interdependencies between these parameters and makes optimized allocation plans that optimize computational load, reduce resource wastage, and ensure efficiency in services. Herein, dynamic responsiveness lets the cloud system deploy resources just where and when required, hence ensuring scalability and performance under a wide range of operational conditions. Also, the MDS-CNet architecture comprises four fundamental modules: input module, dilated depthwise separable convolution (DSC) module, multiscale embedding (ME) module, and output module. In this regard, the first input module takes workload and resource data and then processes it through the DSC module to achieve fine-grained spatial and temporal patterns using dilated convolutions, which expand the receptive field without extra computation cost. Moreover, the ME module merges multi-scale features so that MDS-CNet captures variability in demand patterns and intensity levels at different scales. At last, the output module provides optimal resource allocation decisions to supply balanced usage across cloud nodes. Such modular architecture allows MDS-CNet to solve challenging issues in cloud resource management with good computational efficiency and flexibility. Figure 2 depicts the architecture of MDS-CNet.

3.3.1. Input Module

The input module is the interface of the MDS-CNet framework, where it receives and preprocesses necessary parameters like available computation resources, workload size, frequency of user requests, and task deadlines. These inputs are normalized and formatted in a proper format for efficient processing by the network. The module accurately reflects the variations in cloud resource supply and demand to serve as the basis for the model's adaptive decision-making process in resource allocation. The input is fed into the first convolutional layer $("3\times3, con2d, 3, 32, 2,1")$ where the (3×3) kernel performs a two-dimensional convolution. Here, the number '3' represents the number of input channels, 32 indicates the number of output channels, 2 is the stride for spatial down-sampling, and 1 is the single repetition of these. This phase produces a strong

low-level feature map by boosting relevant workload and resource-usage patterns and filtering out noisy measurements, thereby prepping the input for hierarchical feature learning in the later DSC and ME modules.

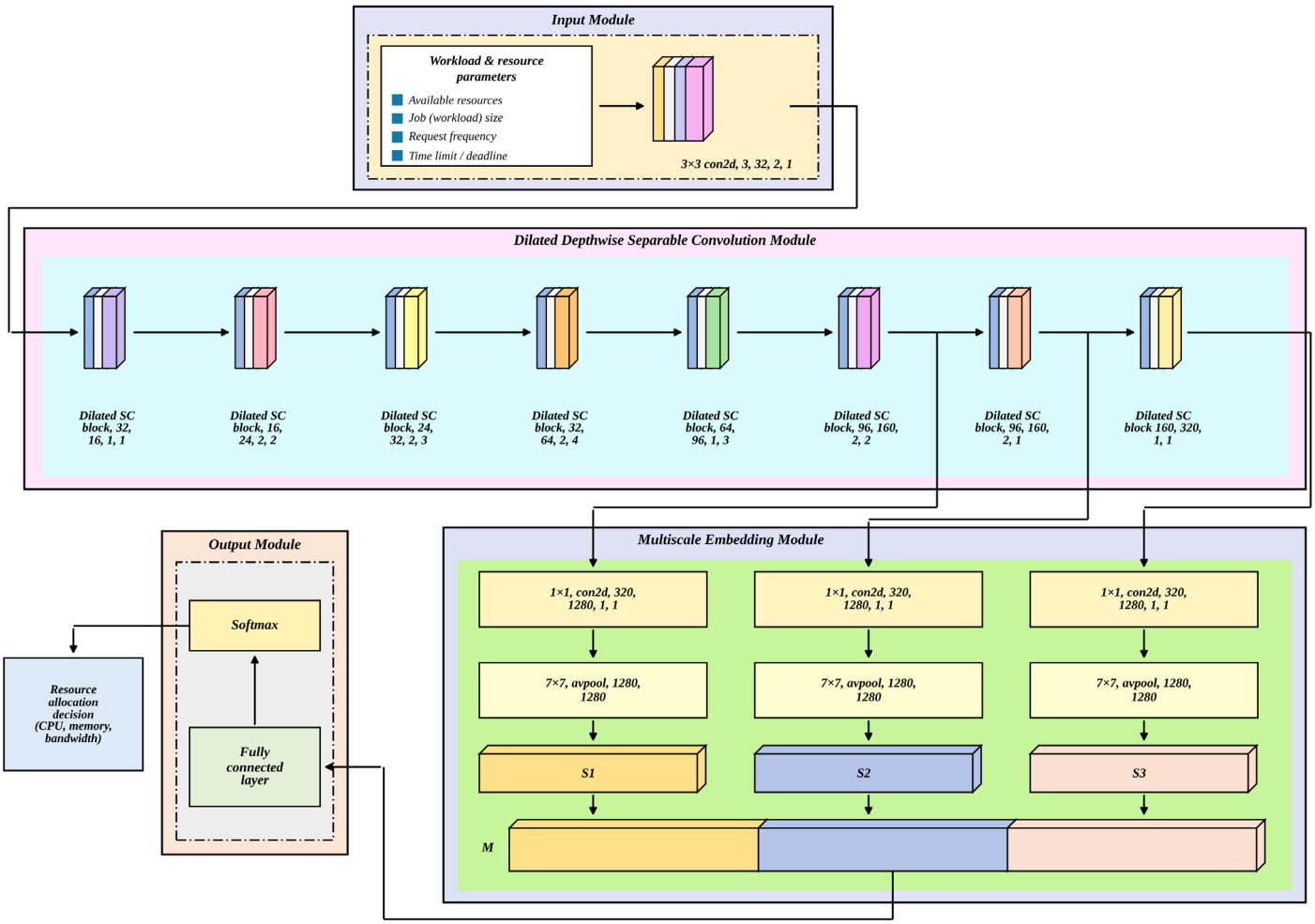


Figure 2. Architecture of the proposed MDS-CNet

3.3.2. DSC Module

The DSC module employs depthwise separable convolutions together with dilated filters to capture the spatial and temporal dependencies of the input data without adding significant computational overhead. Depthwise convolutions focus on applying filters to each channel individually, while pointwise convolutions mix these filtered results. Also, the dilation approach expands the receptive field, thus enabling the model to check resource and workload correlations at wider temporal and spatial scales. This helps feature extraction in an efficient manner for detecting intricate allocation patterns in varied cloud settings. The DSC module consists of 8 dilated separable convolution (SC) blocks. The numbers in each block, i.e., "Dilated SC block, 32, 16, 1, 1" refer to the number of input channels (32), output channels (16), stride (1), and repetitions (1). There are three identical dilated convolutional layers in each block. DSC is achieved by separating a normal convolution into depthwise convolution and pointwise 1×1 convolution for channel merging, with a substantial decrease in computation and network size. Batch Normalization (BN) and Rectified Linear Unit (ReLU) activation are used in all blocks. Depthwise convolution is expressed as in equation (5).

$$\widehat{H}_{a,b,c} = \sum_{d,e} \widehat{A}_{d,e,c} \cdot G_{a+d-1,b+e-1,c} \quad (5)$$

where $\widehat{A}$ represents the depthwise kernel of size $J_A \cdot J_A \cdot C$, $\hat{H}$ denotes the output feature map, G denotes the input feature map, (a, b) and (d, e) are the spatial position indices of the output and the kernel, respectively, c is the

channel index, C is the number of input channels, and J_A is the kernel size. The computational complexities for depthwise and pointwise convolution are denoted as $J_A \times J_A \times C \times J_G \times J_G$ and $P \times C \times J_G \times J_G$, respectively, resulting in a total computational cost of $J_A \times J_A \times C \times J_G + P \times C \times J_G \times J_G$ compared with $J_A \times J_A \times P \times C \times J_G \times J_G$ for a standard convolution, where J_G represents the feature map size and P is the number of output feature map channels. To handle long-range temporal and contextual dependencies in workload and resource-demand sequences, dilated convolution is used. A regular 2D convolution is shown in equation (6).

$$(G*A)(s) = \sum_{s=u+v} G(u)A(v) \tag{6}$$

where s, $u \in \mathbb{R}^2$, $v \in [-p, p^2] \cap \mathbb{R}^2$; $\mathbb{R}$ represents a set of integers, $*$ represents the convolutional operator, G is the discrete input function, A is the convolution kernel, and p determines the kernel extent. The dilated convolution with a dilation rate j is expressed by equation (7).

$$\left(G *_j A\right)(S) = \sum_{s=u+jv} G(u)A(v) \tag{7}$$

where j controls the separation of kernel elements, providing an extended receptive field without additional parameters. Regular convolution corresponds to dilated convolution with $j = 1$. This formulation holds for 1D convolution, with domain s, $u \in \mathbb{R}$, *and* $v \in [-p, p] \cap \mathbb{R}$.

3.3.3. ME Module

It is the responsibility of the ME module to merge the features extracted at different scales to improve the model's understanding of workload patterns. By integrating the fine-grained and coarse-grained feature maps, the ME module ensures that the network captures both local fluctuations, such as sudden spikes in demand, and global trends, such as consistent workload increases. Here, the multi-resolution analysis enables increased system robustness and flexibility, where it can efficiently allocate resources under different conditions and varied workload situations. The final three DSC blocks are passed through a 1×1 convolutional layer $(\text{e.g., } "1 \times 1, con2d, 320, 1280, 1, 1")$ for channel dimension adjustment and an average pooling layer (e.g., $"7 \times 7, avpool, 1280, 1280"$) for spatial dimension reduction and retaining salient features. 1×1 refers to the kernel size of convolution, $con2d$ is an operation, 320 and 1280 are the input and output numbers of channels, and the last two refer to the stride and repetition number. Likewise, in the pooling layer, 7×7 refers to the size of the pooling unit, pool refers to average pooling, and 1280 is the number of input and output channels. The aggregated output of the three scales (S_1, S_2, S_3) is concatenated together to create the multiscale embedding M, which retains both fine and coarse workload information. The embedding is forwarded to the Softmax layer to achieve the final output.

3.3.4. Output Module

The output module produces the final optimized resource allocation policy from the learned feature representations of the earlier modules. It converts the features extracted into concrete decisions, e.g., the level of central processing unit (CPU), memory, or bandwidth to allocate per task or user. The output is optimized to reduce wastage of resources without compromising on high service quality and response efficiency. It also updates its forecasts in real time as new data continuously come in, allowing adaptive and intelligent resource management across cloud nodes. Thus, the three-scale embedding M is then passed through a fully connected layer with 1,280 neurons to convert the aggregated features into a dense representation. This is then passed into a Softmax layer, where each neuron corresponds to a candidate resource allocation decision. For instance, if K candidate allocation options are available, the Softmax layer has K neurons, yielding a probability distribution over the potential allocation decisions.

3.4. OOA for Trust Assessment

In this, the OOA model is proposed, which is an optimization method inspired by the intelligent foraging and social actions of orangutans that imitate their behavior to assess and optimize trust between cloud nodes [23]. In this process, every node has to be simulated as an agent, having properties like service reliability, response time, and transaction history. Further, the OOA model iteratively enhances these trust parameters through the exploration and exploitation processes to find new trusted nodes and to adjust existing relationships, respectively. Based on an adaptive learning process, the OOA explores highly trusted nodes while minimizing malicious or low-performance ones. Therefore, the dynamic trust evaluation ensures the optimal data integrity and service reliability through scalability, flexibility, and protection of privacy in dynamic and sophisticated cloud systems. The problem of accurately estimating trust between dynamic and distributed nodes is solved using OOA in CC systems. Most of the conventional algorithms are doomed to failure in changing service conditions or when estimating malicious behavior suitably. Further, this shortcoming has been improved by OOA through the intelligent search and collective tendency of orangutans to enhance the trust score concerning service reliability, response time, and history. Thus, it updates the trust scores dynamically in real time, allowing only the trusted nodes to participate in cloud processes. The OOA model has high flexibility, high scalability, and strong optimization for the accurate estimation of dynamic trust. In addition, it improves security by detecting malicious nodes, saves evaluation time, and enhances overall network stability and reliability

in large-scale cloud systems. An OOA flowchart is shown in Figure 3, and its step-by-step process is stated as follows:

Step 1: Initialization

The process starts by initializing the population of orangutans at random within the specified search space, and every orangutan is a potential configuration of trust values for the cloud nodes. The initialization process is mentioned in a matrix form in equations (8) and (9),

$$Y = \begin{bmatrix} Y_1 \\ . \\ Y_2 \\ . \\ Y_P \end{bmatrix}_{P\times q} = \begin{bmatrix} y_{1,1} & \cdots & y_{1,d} & \cdots & y_{1,q} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{i,1} & \cdots & y_{i,d} & \cdots & y_{i,q} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_{P,1} & \cdots & y_{P,d} & \cdots & y_{P,q} \end{bmatrix}_{P\times q} \tag{8}$$

$$y_{i,d} = LB_d + r\cdot\left(UB_d - LB_d\right) \tag{9}$$

where the population matrix of OOA is denoted as Y, the candidate solution of the i^{th} orangutan is denoted as Y_i, the d-th dimension (decision variable) of the i-th candidate solution is denoted as $y_{i,d}$, the total number of orangutans is denoted as P, the number of decision variables is denoted as q, the random number is r in the range of [0, 1], the lower bound is denoted as LB_d, and the upper bound is denoted as UB_d, of the d^{th} decision variable.

Step 2: Generation of a Random Value

At the time of random generation, initial positions are distributed in a random manner to aid in diverse exploration of the solution space and avoid premature convergence.

Step 3: Evaluation of Fitness Function

Here, the position of each orangutan is assessed based on a specified trust fitness function considering factors like reliability, response time, and service consistency. The aim is to maximize the sum of overall trust while reducing malicious or low-performing node impact. This fitness value directs the choice of the highest-performing orangutans for the subsequent optimization iteration. The mathematical expression for this objective function is as follows in equation (10).

$$Maximum\ F\left(T_i\right) = w1\times R_i + w2\times S_i + w3\times C_i - w4\times M_i \tag{10}$$

where the overall trust fitness of the cloud node i is denoted as $F\left(T_i\right)$, the reliability score is denoted as R_i, the service performance is denoted as S_i, communication consistency is denoted as C_i, and the malicious behavior pattern is denoted as M_i, and the weighting coefficients are denoted as $w1$, $w2$, $w3$, *and* $w4$ respectively.

Step 4: Exploration Stage

In this exploration stage, the orangutans (nodes) adjust their locations for simulating the foraging character of the orangutans within the search space, seeking the food sources. Thus, the candidate trustworthy node set for the i^{th} node is computed using equation (11),

$$FB_i = \left(Y_t : F_t < F_i \ and\ t \neq i\right) \tag{11}$$

where FB_i includes every node with a better trust fitness value F_t than node i, Y_t denotes the position of the t-th node, and F_i is the trust fitness of node i. Here, every node discovers these greater candidates for enhancing the trust level. Here, the orangutan position (trust parameter vector) is updated based on equations (12) and (13).

$$y_{i,d}^{N1} = y_{i,d} + r\left(BFB_{i,d} - s\cdot y_{i,d}\right) \tag{12}$$

$$Y_i = \begin{cases} Y_i^{N1} & F_i^{N1} \leq F_i \\ Y_i & else \end{cases} \tag{13}$$

Here, the new position of the i-th orangutan (updated trust parameter value) is denoted as Y_i^{N1}, $BFB_{i,d}$ represents the trustworthy node equivalent parameter, random number is r in the range of [0,1], and s represents a random integer selected from {1, 2}, which provides movement diversity; F_i^{N1} is the trust fitness at the new position, and $y_{i,d}^{N1}$ is its d-th dimension. The new fitness is improved, and the trust parameters of the node are accordingly updated. Under this iterative exploration process, OOA allows cloud nodes to search out the best trust levels by learning from better-trusted nodes that provide improved accuracy, security, and network reliability for cloud nodes under changing cloud environments.

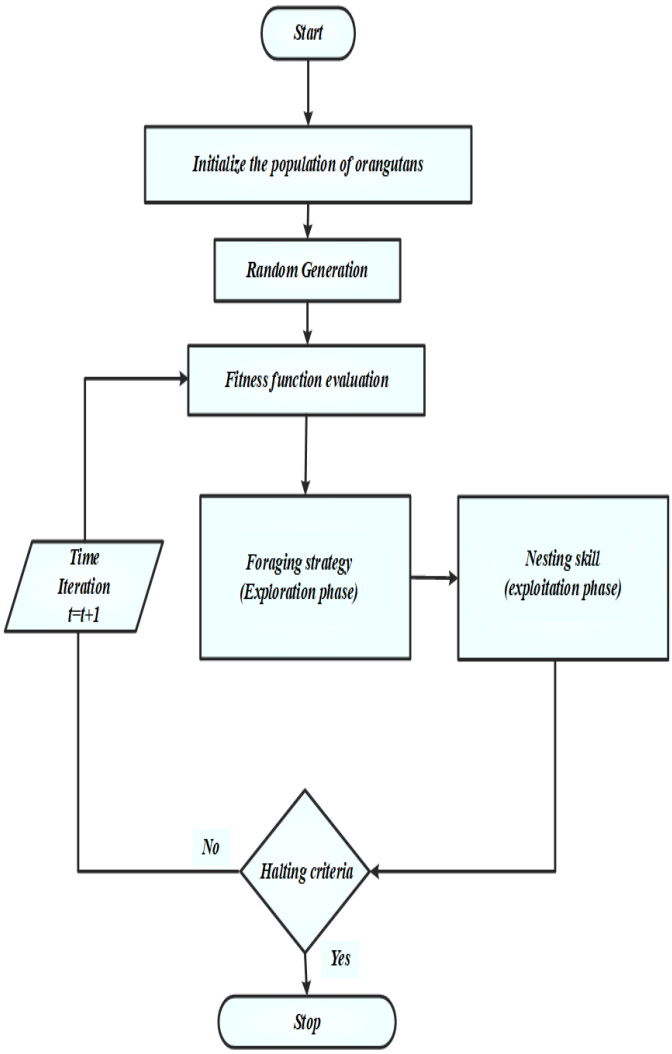


Figure 3. Flowchart representation of OOA

Step 5: Exploitation Stage

In the exploitation process within the OOA, an orangutan at each cloud node optimizes the search locally to exploit the maximum trust fitness. It is analogous to orangutans in the shifting of the trust parameters based on the results from the preliminary exploration. Thus, the new location of the orangutans is computed using equation (14). After the potential reliable nodes are identified during the exploration phase, each node will try to tune up the localized trust factors (reliability, service performance, communication coherence, and malicious activity) over the limited neighborhood such that the global maximum trust value is obtained. The movement toward the better nesting location (local optimum) is mathematically formulated using equation (15),

$$Y_{i,j}^{N2} = y_{i,j} + \left(1 - 2r_{i,j}\right) \cdot \frac{UB_j - LB_j}{t} \quad (14)$$

$$Y_i = \begin{cases} Y_i^{N2} & F_i^{N2} \le F_i \\ Y_i & else \end{cases} \quad (15)$$

where the new updated position is denoted as Y_i^{N2} of the i^{th} orangutan, its d^{th} dimension is denoted as $y_{i,d}^{N2}$, F_i^{N2} is the objective function value at the new position, and t denotes the iteration. In equation (14), j denotes the dimension index (equivalent to d), $r_{i,j}$ is a random number in the range [0, 1], and UB_j and LB_j are the upper and lower bounds of the j-th decision variable.

Step 6: Return the Best Optimal Solution

Step 7: Termination Step

The algorithm repeats from Step 3 until termination conditions are fulfilled, and decision conditions are satisfied. Therefore, the OOA successfully improves trustworthiness assessment in CC by emulating the clever nesting and foraging patterns of orangutans. Through its exploration and exploitation phases, OOA maintains a dynamic balance between global search and local optimization to find the most reliable cloud nodes. Fine-tuning the trust fitness function integrating service performance, reliability, communication consistency, and malicious activity, taking into account OOA, presents a sustainable and dynamic trust evaluation. Hence, this leads to improved credibility, security, and the utilization of resources in the cloud networks. Hence, OOA offers a lightweight, scalable, and intelligent trust management system that is capable of bringing stability and privacy to decentralized cloud systems.

4. Results and Discussion

The implementation results of the proposed MDS-CNet-OOA approach are presented here. The approach was implemented in the Python programming language, and executed on DESKTOP-UA8M7M2, Intel(R) Core (TM) i5-5200U CPU @ 2.20GHz, 4.00 GB random access memory (RAM), and a 64-bit operating system. Along with this, performance measures such as the accuracy of detection, precision, recall, F-measure, execution time (ET), mean absolute error (MAE) of the predicted trust, and Trust Estimation Accuracy (TEA) of the suggested MDS-CNet-OOA approach are compared with some existing approaches that involve T-MBFD-LFA [13], DT-DBN-KHSS [14], CSKA-PPIR [15], and GbRFM-WGO [16] approaches for demonstration to determine the presented model's effectiveness.

4.1. Performance Measures

In this section, the evaluation metrics used to assess the effectiveness of the suggested strategy are described. Here, the metrics of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are employed to

better understand the proposed approach. The evaluation formulas are given below.

4.1.1. Accuracy (A)

It measures the model's overall predicted accuracy for detecting and classifying malicious threats. It is computed using equation (16),

$$A = \left(\frac{true\ positive\ + true\ negative}{true\ positive\ + true\ negative\ + false\ positive\ + false\ negative} \right) \quad (16)$$

4.1.2. Precision

The percentage of all positive forecasts that turn out to be accurate is the suggested model's precision. It is computed using equation (17),

$$Precision = \left(\frac{TP}{TP + FP} \right) \quad (17)$$

4.1.3. Sensitivity/Recall

Sensitivity measures the proportion of actual positive samples that the model correctly predicted. It is calculated by using equation (18),

$$\text{Sensitivity/Recall} = \left(\frac{TP}{TP+FN} \right) \quad (18)$$

4.1.4. F-Measure

The harmonic mean of recall and precision is shown by the F-Measure. The calculation is done with equation (19).

$$F-Measure = 2 * \left(\frac{Precision * Recall}{Precision + Recall} \right) \quad (19)$$

Figure 4 demonstrates the MDS-CNet-OOA model's training and validation performance across 100 epochs. This plot shows the training and validation performance of a DL model over 100 epochs. As training continues, both training and validation accuracies climb steeply and settle at approximately 1.0, showing that the model has high predictive efficacy. At the same time, training and validation losses reduce significantly and converge to values close to zero, showing successful learning and little overfitting. The fact that training and validation curves closely coincide assures that the model generalizes to unseen data.

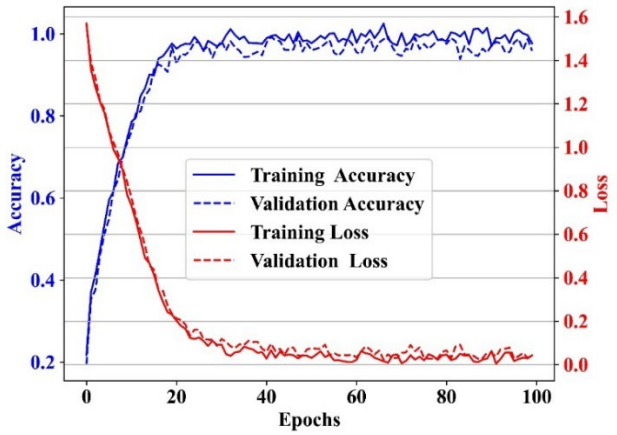


Figure 4. Training and Validation Accuracy and Loss

Figure 5 shows the confusion matrix for a binary classification model separating benign and malicious cases. It indicates that among all the samples of benign cases, 9,930 were correctly labeled as benign and 70 were mistakenly labeled as malicious. Likewise, among malicious cases, 9,910 were correctly labeled as malicious and only 90 were mistakenly labeled as benign. The large values on the diagonal and very few misclassifications off-diagonal suggest that the model is classifying with outstanding performance and high precision, recall, and accuracy for both classes. Taking the malicious class as positive (TP = 9,910, FN = 90, FP = 70, TN = 9,930), equations (16)–(19) give an accuracy of 99.20%, a precision of 99.30%, a recall of 99.10%, and an F-measure of 99.20%.

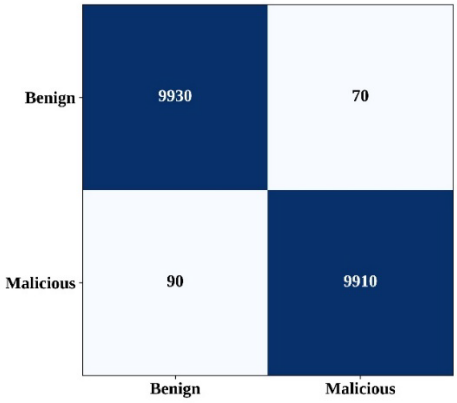


Figure 5. Confusion Matrix Analysis

4.2. Comparative Analysis of the Proposed Model

This section shows the simulation results of the proposed MDS-CNet-OOA approach. Here, the performance of the proposed strategy is compared with alternative techniques such as T-MBFD-LFA, DT-DBN-KHSS, CSKA-PPIR, and GbRFM-WGO respectively.

Figure 6 presents a comparative analysis of detection accuracy. Here, the proposed MDS-CNet-OOA approach has attained 32.89%, 30.98%, 28.78%, and 26.56% improved accuracy with 20000 data samples, 34.67%, 32.89%, 30.89%, and 28.67% improved accuracy with 40000 data samples, 35.78%, 32.89%, 30.79%, and 27.68% improved accuracy with 60000 data samples, 37.89%, 33.90%, 31.89%, and 27.68% improved accuracy with 80000 data samples, and 36.56%, 33.29%, 30.48%, and 27.57% improved accuracy with 100000 data samples than other methods like T-MBFD-LFA, DT-DBN-KHSS, CSKA-PPIR, and GbRFM-WGO respectively.

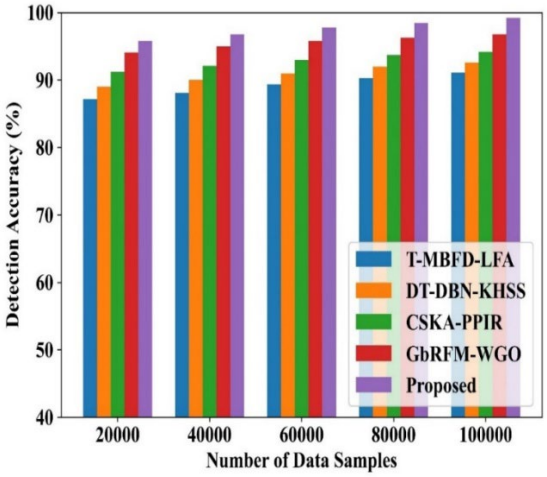


Figure 6. Accuracy Analysis

Figure 7 presents a comparative analysis of precision. Here, the proposed MDS-CNet-OOA approach has attained 30.78%, 28.76%, 26.57%, and 20.65% improved precision with 20000 data samples, 29.78%, 26.37%, 23.21%, and 20.78% improved precision with 40000 data samples, 31.90%, 27.67%, 23.54%, and 19.78% improved precision with 60000 data samples, 35.67%, 31.89%, 26.56%, and 23.08% improved precision with 80000 data samples, and 28.78%, 26.45%, 23.78%, and 18.67% improved precision with 100000 data samples than other methods like T-MBFD-LFA, DT-DBN-KHSS, CSKA-PPIR, and GbRFM-WGO respectively.

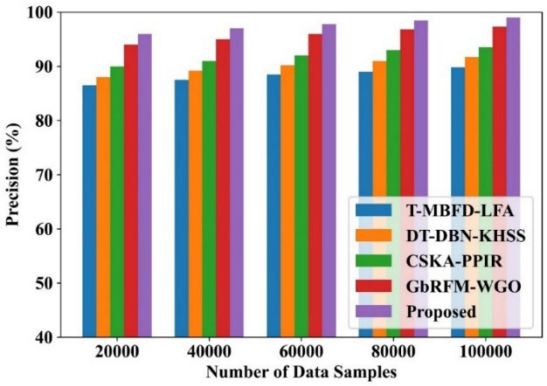


Figure 7. Precision Analysis

Figure 8 presents a comparative analysis of recall. Here, the proposed MDS-CNet-OOA approach has attained 29.78%, 25.46%, 21.09%, and 17.67% improved recall with 20000 data samples, 28.76%, 24.73%, 20.19%, and 16.59% improved recall with 40000 data samples, 27.56%, 23.94%, 21.89%, and 18.67% improved recall with 60000 data samples, 29.49%, 26.57%, 24.76%, and 21.98% improved recall with 80000 data samples, and 30.98%, 28.67%, 26.57%, and 20.68% improved recall with 100000 data samples than other methods like T-MBFD-LFA, DT-DBN-KHSS, CSKA-PPIR, and GbRFM-WGO respectively.

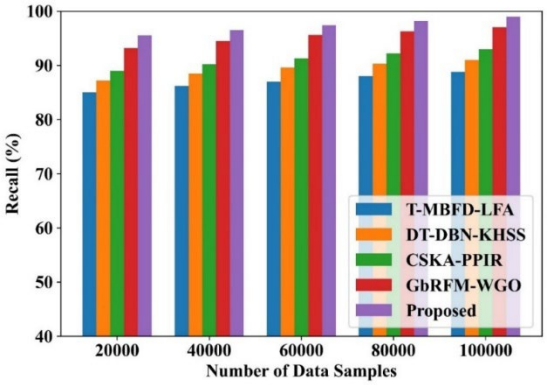


Figure 8. Recall Analysis

Figure 9 presents a comparative analysis of F-measure. Here, the proposed MDS-CNet-OOA approach has attained 27.7%, 23.67%, 20.78%, and 17.59% improved F-measure with 20000 data samples, 29.786%, 24.096%, 23.675%, and 20.493% improved F-measure with 40000 data samples, 28.49%, 21.6%, 19.67%, and 16.5% improved F-measure with 60000 data samples, 29.78%, 27.5%, 23.07%, and 21.89% improved F-measure with 80000 data samples, and 28.76%, 25.67%, 20.67%, and 17.67% improved F-measure with 100000 data samples than other methods like T-MBFD-LFA, DT-DBN-KHSS, CSKA-PPIR, and GbRFM-WGO respectively.

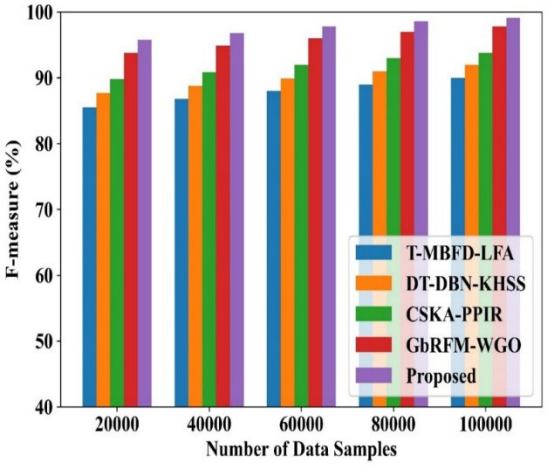


Figure 9. F-measure Analysis

Figure 10 presents a comparative analysis of ET. Here, the proposed MDS-CNet-OOA approach has attained 24.67%, 20.78%, 18.67%, and 11.08% lower ET with 20000 data samples, 26.57%, 21.09%, 19.87%, and 16.56% lower ET with 40000 data samples, 27.68%, 24.76%, 21.09%, and 18.78% lower ET with 60000 data samples, 26.06%, 21.7%, 18.67%, and 16.57% lower ET with 80000 data samples, and 28.76%, 26.75%, 22.39%, and 19.87% lower ET with 100000 data samples than other methods like T-MBFD-LFA, DT-DBN-KHSS, CSKA-PPIR, and GbRFM-WGO respectively.

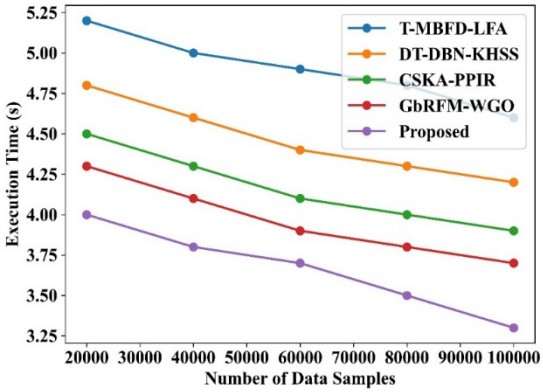


Figure 10. Analysis of Execution Time

Figure 11 presents a comparative analysis of TEA. Here, the proposed MDS-CNet-OOA approach has attained 32.67%, 29.87%, 26.57%, and 20.89% improved TEA with 20000 data samples, 30.89%, 27.67%, 25.78%, and 21.2% improved TEA with 40000 data samples, 29.87%, 26.57%, 21.98%, and 17.68% improved TEA with 60000 data samples, 29.76%, 25.23%, 19.87%, and 15.38% improved TEA with 80000 data samples, and 30.67%, 26.57%, 21.07%, and 17.57% improved TEA with 100000 data samples than other methods like T-MBFD-LFA, DT-DBN-KHSS, CSKA-PPIR, and GbRFM-WGO respectively.

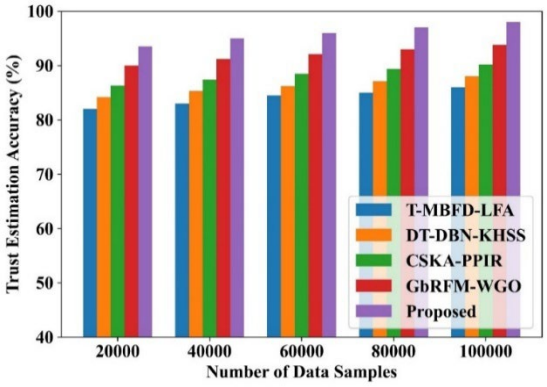


Figure 11. Analysis of Trust Estimation Accuracy

Figure 12 presents a comparative analysis of the MAE of the predicted trust. Here, the proposed MDS-CNet-OOA approach has attained 24.78%, 20.35%, 18.48%, and 15.49% lower MAE with 20000 data samples, 26.57%, 21.46%, 17.56%, and 14.59% lower MAE with 40000 data samples, 28.68%, 25.08%, 19.87%, and 16.46% lower MAE with 60000 data samples, 27.98%, 23.7%, 21.65%, and 17.68% lower MAE with 80000 data samples, and 29.76%, 25.67%, 21.08%, and 17.67% lower MAE with 100000 data samples than other methods like T-MBFD-LFA, DT-DBN-KHSS, CSKA-PPIR, and GbRFM-WGO respectively.

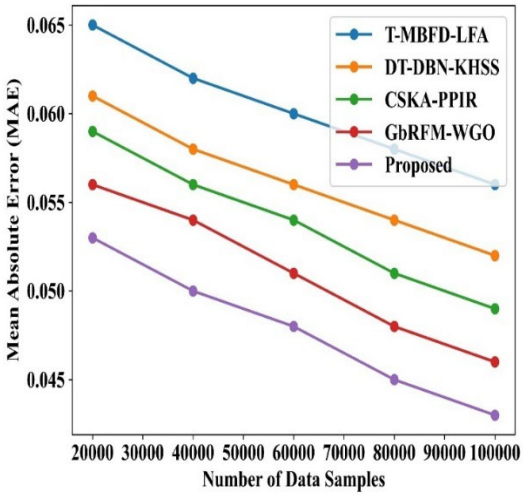


Figure 12. Analysis of MAE of Predicted Trust

5. Conclusions

In this manuscript, a new DL-based framework to optimize both trust and security in cloud networks is successfully implemented. Here, HiAbE is used to protect personal information and conceal users' positions and reputations, thus effectively reducing re-identification threats. For resource allocation purposes, the MDS-CNet is utilized, which is dynamically adaptive to different workloads and user request demands. To further improve the credibility of cloud services, a trust assessment module based on the OOA is integrated. Experimental validation of the framework is performed using ICS3D, which comprises real-world cloud attack and performance data. Also, the model developed in Python analyzes the performance metrics like accuracy, recall, F-measure, and precision. Thus, the proposed MDS-CNet-OOA approach has attained 28.76%, 24.73%, 20.19%, and 16.59% improved recall, 29.78%, 27.5%, 23.07%, and 21.89% improved F-measure, 29.87%, 26.57%, 21.98%, and 17.68% improved TEA, 28.68%, 25.08%, 19.87%, and 16.46% lower MAE than other methods like T-MBFD-LFA, DT-DBN-KHSS, CSKA-PPIR, and GbRFM-WGO, respectively. The experimental outcome verifies the method's excellent impact on improving resource management efficiency, enhancing trust reliability, and reducing potential security weaknesses in CC environments. In subsequent work, the framework presented here can further be enhanced by introducing adaptive encryption techniques. Additionally,

hybrid optimization methods can be introduced to minimize computation overheads in dynamic assessments. In addition, the introduction of explainable artificial intelligence (AI) mechanisms might help improve interpretability and transparency in trust-based decision-making in cloud systems.

Author Contributions

Anita Sachin Mahajan contributed to conceptualization, methodology, supervision, and writing original draft preparation, as well as review and editing. Deepti Varshney was responsible for software development, investigation, data curation, and writing original draft preparation. Shruti Gupta contributed to the investigation and data curation. Surjeet assisted in the investigation and data curation. Vivek Srivastava contributed to conceptualization, methodology, supervision, and writing review and editing. Abhishek Bhatt assisted in software development, investigation, and data curation.

All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The model is tested on the Integrated Cloud Security 3Dataset (ICS3D) which is a large-scale cybersecurity dataset. Available at https://www.kaggle.com/datasets/rogernickanaedevha/integrated-cloud-security-3datasets-ics3d

References

[1] Gnanajeyaraman R, Arul U, Michael G, Selvakumar A, Ramesh S, Manikandan T. VANET security enhancement in cloud navigation with Internet of Things-based trust model in deep learning architecture. Soft Comput. 2023:1-12.

[2] Moudoud H, Cherkaoui S. Empowering security and trust in 5G and beyond: a deep reinforcement learning approach. IEEE Open J Commun Soc. 2023;4:2410-20.

[3] El-Kassabi HT, Serhani MA, Masud MM, Shuaib K, Khalil K. Deep learning approach to security enforcement in cloud workflow orchestration. J Cloud Comput. 2023;12(1):10.

[4] Srilatha D, Thillaiarasu N. Implementation of intrusion detection and prevention with deep learning in cloud computing. J Inf Technol Manag. 2023;15(Spec Issue):1-18.

[5] Devi RA, Arunachalam AR. Enhancement of IoT device security using an improved elliptic curve cryptography algorithm and malware detection utilizing deep LSTM. High-Confid Comput. 2023;3(2):100117.

[6] Wang Y, Yang X. Research on enhancing cloud computing network security using artificial intelligence algorithms. arXiv [Preprint]. 2025. arXiv:2502.17801.

[7] Lin Y. Enhanced detection of anomalous network behavior in cloud-driven big data systems using deep learning models. J Theory Pract Eng Sci. 2024;4(8):1-11.

[8] Stutz D, de Assis JT, Laghari AA, Khan AA, Andreopoulos N, Terziev A, Deshpande A, Kulkarni D, Grata EG. Enhancing security in cloud computing using artificial intelligence (AI). In: Applying artificial intelligence in cybersecurity analytics and cyber threat detection. 2024. p. 179-220.

[9] Singh A, Mushtaq Z, Abosaq HA, Mursal SNF, Irfan M, Nowakowski G. Enhancing ransomware attack detection using transfer learning and deep learning ensemble models on cloud-encrypted data. Electronics. 2023;12(18):3899.

[10] Subashini S, Kavitha V. A survey on security issues in service delivery models of cloud computing. J Netw Comput Appl. 2011;34(1):1-11.

[11] Takabi H, Joshi JBD, Ahn GJ. Security and privacy challenges in cloud computing environments. IEEE Secur Priv. 2010;8(6):24-31.

[12] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP'07); 2007 May. IEEE; 2007. p. 321-34.

[13] Saini H, Singh G, Kaur A, Saini S, Wani NA, Chopra V, Akhtar Z, Bhat SA. Hybrid optimization machine learning framework for enhancing trust and security in cloud network. IEEE Access. 2024.

[14] Lv Z, Chen D, Cao B, Song H, Lv H. Secure deep learning in defense in deep-learning-as-a-service computing systems in digital twins. IEEE Trans Comput. 2023;73(3):656-68.

[15] Saini H, Singh G, Dalal S, Lilhore UK, Simaiya S, Dalal S. Enhancing cloud network security with a trust-based service mechanism using k-anonymity and statistical machine learning approach. Peer Peer Netw Appl. 2024;17(6):4084-109.

[16] Mohammed S, Rangu S. To secure the cloud application using a novel efficient deep learning-based forensic framework. J Interconnect Netw. 2024;24(1):2350008.

[17] Nizamudeen SMT. Intelligent intrusion detection framework for multi-clouds–IoT environment using swarm-based deep learning classifier. J Cloud Comput. 2023;12(1):134.

[18] Ahmad S, Mehfuz S, Urooj S, Alsubaie N. Machine learning-based intelligent security framework for secure cloud key management. Cluster Comput. 2024;27(5):5953-79.

[19] Tamizshelvan C, Vijayalakshmi V. Cloud data access governance and data security using distributed infrastructure with hybrid machine learning architectures. Wirel Netw. 2024;30(4):2099-114.

[20] Anaedevha RN. Integrated Cloud Security 3Datasets (ICS3D) [dataset]. Kaggle. Available from: https://www.kaggle.com/datasets/rogernickanaedevha/integrated-cloud-security-3datasets-ics3d.

[21] Sasikumar A, Ravi L, Devarajan M, Selvalakshmi A, Almaktoom AT, Almazyad AS, Xiong G, Mohamed AW. Blockchain-assisted hierarchical attribute-based encryption scheme for secure information sharing in industrial internet of things. IEEE Access. 2024;12:12586-601.

[22] Zeng Y, Li Y, Zhou Z, Wang R, Lu D. Domestic activities classification from audio recordings using multi-scale dilated depthwise separable convolutional network. In: 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP); 2021 Oct. IEEE; 2021. p. 1-5.

[23] Hamadneh T, Batiha B, Gharib GM, Montazeri Z, Werner F, Dhiman G, Dehghani M, Jawad RK, Aram E, Ibraheem IK, Eguchi K. Orangutan optimization algorithm: an innovative bio-inspired metaheuristic approach for solving

engineering optimization problems. Int J Intell Eng Syst. 2025;18(1):45-58.