International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I11P102

Volume 74 Issue 11, 18-31, November 2026
© 2026 Seventh Sense Research Group®

Original Article

Improving Access Control in Cloud System by Novel Adversarial Training of Deep Learner

K. Gowri[1], B.L. Shivakumar[2]

[1,2]*Department of Computer Science, Sri Ramakrishna College of Arts and Science, Coimbatore, Tamil Nadu, India.*

[1]*Corresponding Author: gowridhilipphd@gmail.com*

Received: 17 April 2026 Revised: 18 August 2026 Accepted: 18 September 2026 Published: 25 November 2026

Abstract *- Cloud computing has evolved as a viable method for meeting the computing requirements of high-performance applications through the optimum mapping of cloud workloads to resources. But it is challenge to handle the adversarial attacks that poses a significant threat to DL system, as adversaries may get illegal access by tricking the network with changed samples that mimic normal ones. To resolve this, dCloud Access Control (dCAC) model is developed in this paper to mitigate the adversarial attacks during access control in cloud environment. In this model, the access control for each user is maintained as training data. The learned Deep Convolutional Network Model (DCNN) model is used to allocate the access data level for every authenticated user in the cloud. In DCNN based Access Control (DCNNBAC), the access control systems make decisions in accordance with the user/resource metadata, making it crucial to understand the reasons behind these decisions and the influencing metadata. Integrated Gradients (IG) and Knowledge Transferring (KT) are employed to gain the internal details of DCNN. IG attributes provide the decisions to input features of prediction samples, while Knowledge Transferring helps to identify relationships among metadata and the network's logic. Moreover, Adversarial training with optimization over randomized weights of DCNN is suggested to improve the robustness and accuracy of adversarial attack prediction. For this, Taylor series is utilized to expand the objective function over DCNNBAC weights, effectively replacing it with Taylor terms like zero, first, and second, thereby reducing time consumption. This model effectively prevents adversarial attacks during access control and reduces the complexity of access control updating in cloud systems. Finally, an extensive simulation reveals that the proposed model achieves accuracy of 96.93% on amazon-uci dataset than the other cloud access control models.*

Keywords *- Cloud computing, Deep Learning, Taylor series, Access control adversarial training.*

1. Introduction

The on-demand feature of cloud computing has made it immensely important in many businesses, since it provides resources such as network connections, storage, services, applications, data, and processors whenever they are needed [1]. It combines decentralized and parallel computing with shared facilities, including software and hardware. All users require to ingress and contribute for their usage; no platform or software ever needed. It relies only on the internet and allows for the rapid distribution of resources with nothing in the way of administrative overhead [2]. Data may be stored and processed by users and businesses in remote third-party data centres or in independently held premises. By pooling resources, as in a utility computing model, the cloud promotes consistency and boosts the economic growth [3].

User information is stored in the data processing centres of the CSP rather than on the user's personal systems [4]. Users' privacy will certainly pose a major issue. Furthermore, user privacy and security would be compromised when cloud services are centralized. The user takes precautions to prevent unauthorized parties or harmful actions on the cloud servers from gaining access to sensitive information [5]. To protect sensitive data stored in the cloud from harmful threats such denial-of-service assaults, internal attacks, and equipment failures, cloud security must adhere to the three tenets of Confidentiality, Integrity, and Availability (CIA) [6].

One major obstacle to cloud computing is the control over user identifications and the provision of sufficient privacy and security. To improve the data security stored in the cloud, cloud computing has integrated many security measures, including authentication, access control and threat detection model [7]. More security measures are required to safeguard data, even if these criteria give substantial performance. Because attacks such as IP tampering, internal assaults and Denial of Service (DoS) may affect cloud systems that uses conventional Internet Protocols (IP) and virtualization methods [8]. When it comes to handling assaults and massive amounts of data, conventional detection and mitigation methods fall short, and they have trouble detecting new, undiscovered attack variants or their signatures [9].

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

Image detection, matching patterns and cybersecurity are just a few of the recent areas where Deep Learning (DL) has shown exceptional performance [10]. To protect private user data and forestall security holes in massive datasets, DL models are finding more and more applications in cloud security. Data categorization, complicated interface verification, rapid problem-solving, large-scale automation, low-cost outcomes, and high-quality results are further benefits [11, 12]. To improve cloud security, increase identification rates, recognize identified and unidentified incidents, improve data privacy, forecast unauthorized individuals, and restrict them, DL algorithms like Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Long Short - Term Memory (LSTM) provide significant forecasting outcomes [13].

In accordance with the DL concept, a lightweight approach using Deep CNN (DCNN) and homomorphic encryption was developed for integrating deep face recognition authentication in mobile cloud computing [14]. This model uses a DCNN model to extract features and classify users for efficient authentication. Once registered, the cloud slide compares the user's image to previously categorised images, allowing authorized access to the application server using partially homomorphic encryption. If the result is negative, an illegal access request is sent [14].

1.1. Problem Statement

Existing access control systems are susceptible to adversarial attacks, in which malicious actors attempt to gain unauthorised access by manipulating user data. These attacks degrade the reliability of cloud security and may compromise sensitive data. Hence, there is a need for an innovative access control framework that can accurately identify legitimate users while effectively protect against various attacks.

1.2. Main Contributions

To solve the above-mentioned issues, a dCloud Access Control (dCAC) is proposed in cloud environment. In this method, DCNN with a Fully Connected (FC) layer and softmax classifier is employed to oversee the authorisation of cloud consumers. The DCNN model learns to assign the access levels to authorized users in the cloud by using access control rules as training data. In the constructed DCNNBAC, the access control systems make decisions in accordance with the information for the user and that resource, making it crucial to understand the reasons behind these decisions and the metadata that influenced them.

Moreover, IG and KT approaches is employed to learn the internal details of DCNN. IG model assigns the decisions of a DCNN for the sample prediction of input features, but it fails to demonstrate the connection between metadata and the logic of the network, despite determining attribution scores for each decision. To solve this, KT is used to identify the relationships in DCNNBAC by extracting a decision tree to understand traditional rules. In addition, optimizing adversarial training over randomized DCNN weights can make adversarial attack prediction more robust and accurate. To broaden the goal function over DCNNBAC weights, this approach uses Taylor series, effectively replacing it with Taylor terms like zero, first, and second, thereby enhancing efficiency and reducing time consumption. This model effectively mitigates adversarial attacks during access control and simplifies the process of updating access control in cloud systems.

The subsequent article is organised in the following manner: Section 2 examines various deep learning-based cloud security techniques that have developed in recent years. Section 3 outlines the proposed dCAC method, while Section 4 assesses its effectiveness. Section 5 encapsulates the complete body of work.

2. Literature Survey

Nobi & Krishnan [15] investigated the adversarial attack difficulty in ML-based access control, focusing on manipulating user and resource information for unauthorized access. Encoding the categorical metadata with a single pass, and the MinMaxScaler were used to pre-process and normalize the continuous data from the adversarial assault. Finally, the ResNet was used for the final prediction of adversarial attacks. But only limited resources were utilized to prove the security efficiency.

Zhang et al. [16] introduced a TIKI-TAKA framework for identifying adversarial manipulations in cloud systems. This system has a defensive strategy to increase resilience against hostile assaults. Then, multiple adversarial attack methods were chosen to spoof three prominent malicious traffic detectors using the Neural Network (NN) structure. On the other hand, more parameters were required to enhance the efficiency and security.

Nobi et al. [17] developed a DL Based Access Control (DLBAC) in cloud environment. In this method, synthetic data and a real-world dataset which constitutes to quantity of users/resources. The Decision Engine (DE) was employed to predict access permissions, which retrieve metadata from internal repositories, encode them, and feed them into a NN to determine authorization. However, this model was susceptible to adversarial attacks, which deceive unauthorized access with modified samples resembling genuine data. Vijayanand and Saravanan [18] developed a Secure Access Control Scheme with Detecting Anomalous in Cloud Servers (SACS-DACS) system. In this method, Multidimensional Feature Misbehaviour Server Detection method (MFMSD) was proposed to recognize anomalies in multiple cloud servers. The large - scale System Call Identifier Sequences (SCIS) was retrieved from host logs to detect anomalies using CNN-Hidden Markov Model (CNNHMM) for maximum accuracy. However, the model training time was high.

An access control framework was created by Alharbe et al. [19] to evaluate the potential security concerns associated with storing critical and dynamic information in the cloud. The model uses risk assessment indicators from subject, resource, and environment attributes, ensuring consistency in access request characteristics. The Fuzzy Analytic Hierarchy Process (FAHP) method calculates weight and creates access control policies using XACML to protect cloud resources' security. But this method was coarse, lacking flexibility and accuracy, and lacking internal attack.

Saviour and Samiappan [20] developed an authentication and access control model utilizing the Interplanetary File System (IPFS) within a cloud framework, employing a Deep Belief Network (DBN). There are several steps involved in the process, including setting up, login, initialisation, verification, evaluation, cryptography of data and storage, control of access, and decoding were employed to execute the authentication and access control processes. A subsequent transmission of the data log file was made to the feature fusion component, which integrates DBN and hybrid correlation factors for security tasks within the cloud model. But this method constitutes to severe problems like risk of privacy leakage.

Paulraj et al. [21] developed an admission control policy and key agreement in cloud computing using the Multi-Agent Reinforcement Learning (MARL) model. This model builds trust between cloud providers, facilitating cross-cloud data migration and boosting provider trust. The model uses pre-processing techniques, feature selection, and dimensionality reduction to identify and classify anonymity in the cloud, requiring privacy protections and binding agreements. But this model was computationally expensive and necessitated more manual input.

Jodeiri Akbarfam et al. [22] suggested DL with Block-Chain Access Control (DLBCAC) for automated and secure access control in cloud. This model allows for the identification of malicious activities by storing malicious activities in a storage system and using a verification algorithm to cross - reference them with the blockchain. This approach was particularly utilized to automate access control while detecting and preventing data breaches, enhancing security. But this model was characterized by high data management issues, resulting in low accuracy results.

Ayedh et al. [23] developed a Policy Decision-Making (PDM) and Access Control - based Supervised Learning (ACSL) algorithm to enhance cloud security during access control. The model converts Policy Decision Points (PDP) into binary classifications for security access control and attack mitigation, using a vector decision classifier for accurate, distributed, and dynamic PDP. This model ensures high privacy for access control policies and improves security by making it dynamic and distributed. However, this model provides lower efficiency on larger datasets and slower convergence rate.

Alkhowaiter et al. [24] suggested an Adversarial-Aware DL System (AADLS) based on a secondary classical ML verification approach. Initially, the primary image classification with ResNet model for detection and classification. The input is either real or adversarial-generated images from four attacks. The output of ResNet-34 would be the highest probability index, indicating the top labelled image's classification for adversarial attacks. However, these schemes were inevitable to include additional communication costs.

2.1. Research Gaps

Although several DL-based cloud access control methods have been proposed, they still suffer from several limitations. Many existing models are vulnerable to adversarial attacks and fail to prevent unauthorized access effectively. Some methods require high computational cost, long training time, and complex model architectures, reducing their scalability in cloud environments.

In addition, existing approaches provide limited interpretability of access control decisions and often lack effective optimization strategies to improve robustness and generalization. Therefore, an efficient and adversarial robust access control framework is required for secure cloud computing.

3. Proposed Methodology

In this section, dCAC model is developed to eliminate the adversarial attacks and assist the model's efficiency for cloud environment. In this framework, both authentication and access control task are processed to secure the data, which is briefly illustrated in below sections.

Figure 1 devises the pipeline for proposed model in which (1) revises the authentication by DCNN and (2) devises the access control by DCNN in cloud computing.

3.1. Access Control by DCNN in Cloud

The DCNNBAC model introduces a novel approach to access control by leveraging user and resource metadata alongside a trained neural network, distinguishing it from traditional methods. A significant distinction in this context is between metadata and attributes-a difference that, while largely semantic, holds practical value in cloud computing.

Unlike conventional access control frameworks, DCNNBAC replaces the general access control process with Deep Convolutional Neural Network (DCNN) that determines admit decisions. Furthermore, DCNN is trained directly on raw metadata from the organization, eliminating the need for manually crafted access control attributes. Figure 2 illustrates the decision-making process of DCNNBAC.

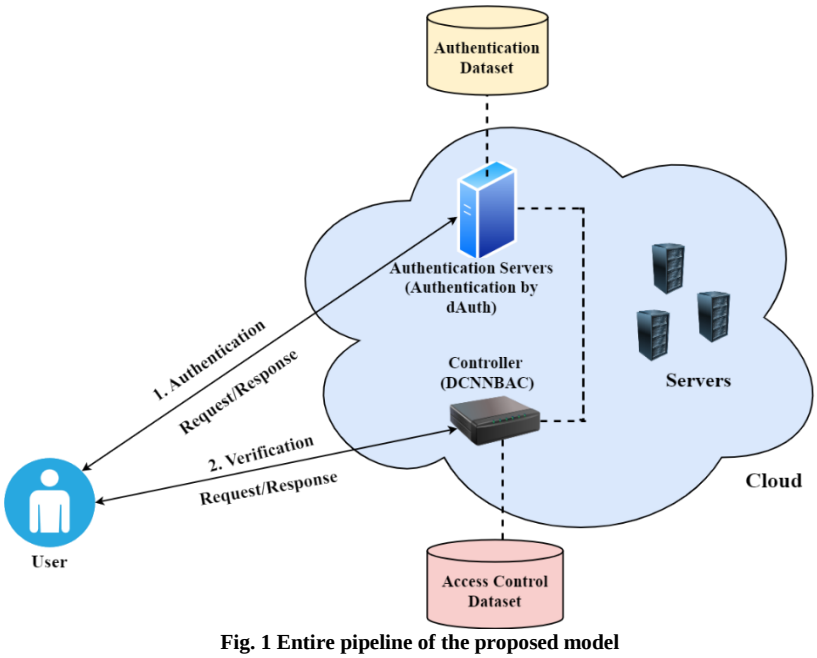


Fig. 1 Entire pipeline of the proposed model

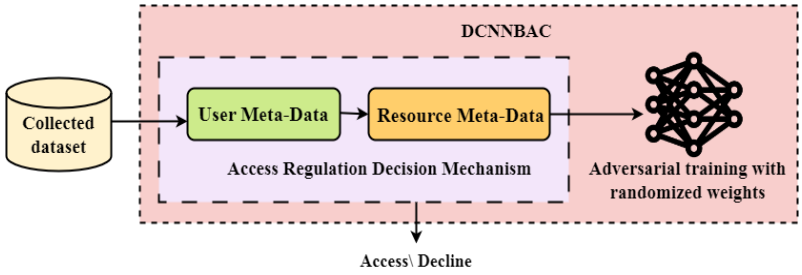


Fig. 2 General block structure of DCNNBAC

3.1.1. Training DCNN Model

In the DCNNBAC framework, The DCNN approach utilises user and resource information as input and incorporates a categorisation layer with a quantity of neurones that aligns with the accessible processes, $\mathcal{OP}$ each generating a chance for accessibility authorisation. For a specific feature vector a obtained from user and resource information, the DCNN is characterised as the detection function df such that $\hat{b} = df(a)$ where $\hat{b}$ is the detected label or authorization, i.e., Accept (1) or Decline (0) of $\boldsymbol{\mathcal{OP}}$ determined by evaluating the ratio of providing the access to the output of the DCNN with the threshold, which is adjusted to 0.5. The DCNN is trained to obtain the weight of the network in which a training authorization tuple $\mathbb{A}$ of size π is gathered, where (a_x, b_x) represents the x^{th} sample in $\mathbb{A}$ where the a_x is the feature vector and b_x will be respective $\boldsymbol{\mathcal{OP}}$ or the desired class. The user/resource information values in a tuple are encrypted using one-hot encoding to convert a definite number into a 2D binary array. The array comprises metadata, and each column holds the encoded binary representation of the relevant metadata value. Also, the binary encoding is utilized to improve memory efficiency at which each action is applied as target labels without further processing. The training data for DCNNBAC is encoded with user and resource information values, and permissions are applied to related actions without additional processing. The DCNNBAC, which has been trained, will be is utilized in the Access Regulation Decision Mechanism (ARDM) to make test data access decisions. Figure 3 depicts the training data construction of DCNNBAC.

3.1.2. Decision Making Task in DCNNBAC

The ARDM (decision model) is a DCNNBAC component that handles the acquiring and authorization of any access request. In DCNNBAC, the DE operates using three inputs: user, resource, and operation. It retrieves metadata for both users and resources from internal databases and converts them into a corresponding binary representation. This encoded data is then processed by a DCNN, which predicts access permissions for the given requests. The network generates access predictions for all operations, and the DE finalizes the actual authorization. For instances, the specific user (C) needs to access $\mathcal{OP} - 2$ in project 1 resource. The resultant of the DCNN for the $\mathcal{OP} - 2$ will be 1 indicating the user has $\mathcal{OP} - 2$ permission on project 1. Therefore, the DE will grant this specific request during the access control. Figure 4 illustrates the decision-making task in DCNNBAC.

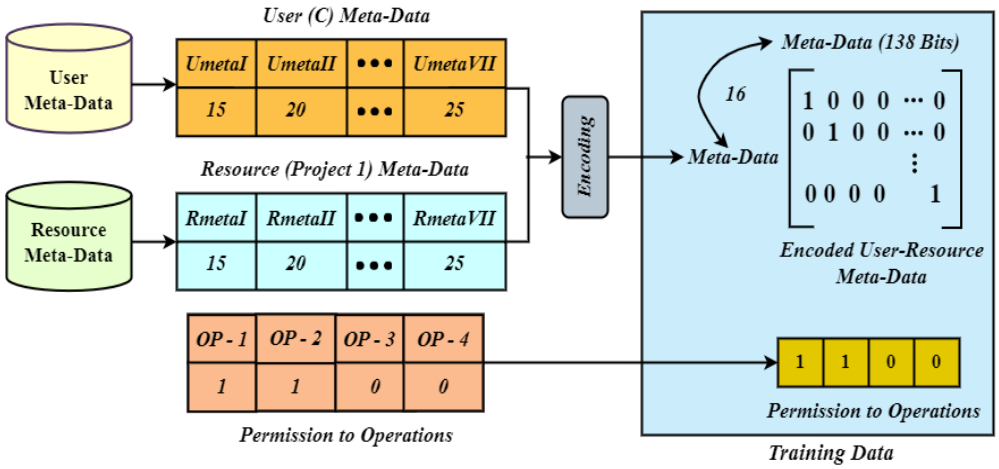


Fig. 3 Devising training data for DCNNBAC

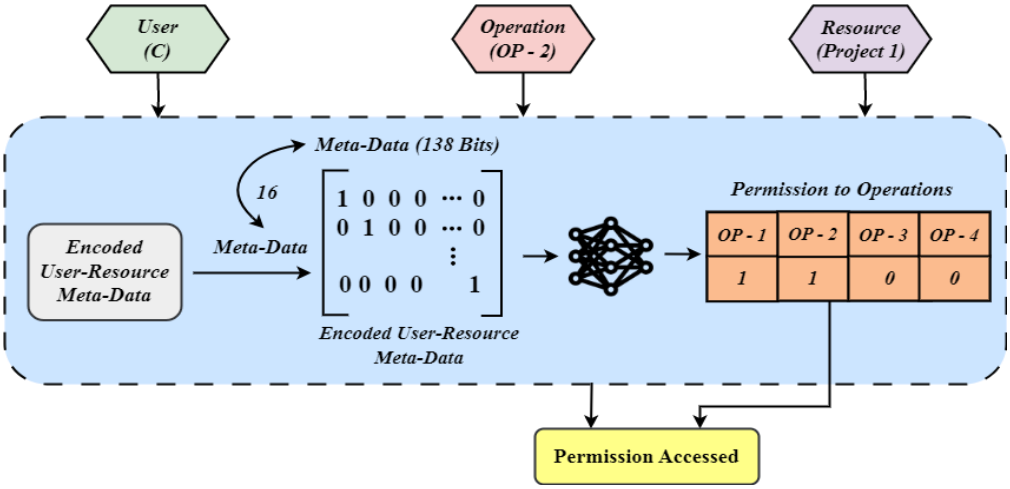


Fig. 4 Decision making process in DCNNBAC

3.1.3. Understanding the DCNNBAC Decisions

In this model, IG and Knowledge Transferring approaches have been used to understand the perspectives of internal details of DCNN model for providing efficient access control task in cloud systems.

Integrated Gradients (IG): IG is a technique that assigns NN decisions to the characteristics of prediction samples' input for effective interpretation. The gradient assigns the network's selection of its input characteristics identifying the most useful components for a choice. To comprehend tuple decision of DCNNBAC, user/resource metadata values, decision and the DCNN must be inputted to the IG. The IG output will have considered as the attribution rating for input metadata normalized between 0 and 1, indicating the effect on decisions, which is termed to be local interpretation. This local interpretation helps understand each decision of a DCNN whereas the global perspective of a network clarifies its general understanding but typically requires a set of decisions. The DCNNBAC is trained on the collected dataset. This model examines the ability of DCNNBAC for particular access control selections and network knowledge acquisition. The DCNNBAC is trained on the collected dataset. The operation 1 ($\mathcal{OP} - 1$) accessed to project resources for users, and the DE allows the request. Local interpretation is performed using metadata, decision and DCCNNBAC to acknowledge the reason behind the decision. In order to process the global perspective for the allowed access to $\mathcal{OP} - 1$, the $\mathcal{OP} - 1$ access details for a random sample set with finite access will be considered from the collected dataset. Every user resource metadata is sent with corresponding decisions for $\mathcal{OP} - 1$ operation to IG in order to ascertain the global perspective.

The output shows $RmetaII$ as the most important metadata; the user's $UmetaII$ is found as the next most important information and attributing parameters of supplementary information. The fundamental initiative is to change the metadata integers of important metadata for a pertinent judgment during the access control choreography. The metadata for all tuples with deny access on $\mathcal{OP} - 1$ will be modified sequentially, based on their implication tier in the

global analysis. For instance, the significance of $RmetaII$ metadata, followed by $UmetaII$ metadata, and so forth. The metadata values for user\resources are derived from tuple 1. The types are initially not accepted without modifications. However, in the initial metadata modification of the first metadata ($RmetaII$), approximately 5% of tuples are granted access permission. Modifying the second metadata value results in access being granted to approximately one-third of the tuples. A comparable increase persists in securing grant access for more than 50% of all tuples prior to a decline correlated to sixth metadata ($RmetaV$) modification, falling underneath 40%. This suggests that specific tuples with extensive metadata alterations are classified inside an alternative distribution, resulting in a 'deny' access status. This technique enables a system administrator to assess the influence of metadata values in relation to dependent accesses. This model facilitates grant access for more than fifty percent of all tuples, with a noted decrease for the sixth metadata modification. This technique enables system administrators to assess the impact of metadata values in relation to dependent accesses by identifying tuples that exhibit multiple metadata modifications within a distinct distribution characterized by 'deny' access.

Knowledge Transferring: IG assigns attribution ratings to every metadata, but it doesn't set relationships between metadata or the network's functionality. KT assists to detect such inter-relation. KT allows users to create a DT to realize the decision of DCNNBAC using established rules. Although a precise replication of representation specifics is unachievable, the produced DT will provide a close interpretation of the foundational details in DCNNBAC. Although DT categorization assumptions are easily recognized, DCNN outperforms them in terms of generalization. The generalization capabilities of DCNN may be translated to a DT via distillation. DCNNBAC outputs the probability of granting access to resources, allowing for the determination of possibilities of every dataset tuple and training a DT. These probabilities represent the DCNN knowledge and transform it to a DT. The KT technique is explored by training DCNNBAC on the IG approach by accessing the $\mathcal{OP}-1$ probabilities for all samples. The Decision Tree (DT) is applied on the same dataset, providing the probability outputs of DCNNBAC, indeed, the ground-truth permissions. The DT is developed with a maximum depth of eight, enabling the retrieval of underlying rules for specific decisions. The tree serves global interpretation, while a rule represents local interpretation. The Dave metadata values is applied to retrieve the permission rules for $\mathcal{OP}-1$ function by preceding rule, $\langle\langle\langle Umeta0 > 31\rangle \Lambda\langle Umeta0 < 61\rangle\Lambda\langle UmetaVI < 50\rangle\rangle\Lambda\langle\langle Rmeta0 < 72\rangle \Lambda\langle RmetaII < 18\rangle\Lambda\langle Rmeta5 < 111\rangle\rangle\rangle$. For more exact rules, the DT should be built with infinite depth which is crucial to recognize that KT and IG methods have orthogonal observations into DCNN for eliminating the complexity while performing the access control process in cloud systems.

3.1.4. Adversarial Training in DCNN Model

In the pre-determined DCNN model with weight $\mathcal{W}$, an additional small Gaussian noise δ is determined to smooth $\mathcal{W}$. The efficient tuned issues can be improved by the following Equation (1),

$$\min_{w} \mathcal{E}_{\delta}\left[\underset{I}{\mathcal{E}_z(\ell(f_{w+\delta}(z),v))} + \underset{II}{\max_{z'\colon \|z'-z\|_{q\le\beta}} \mathcal{E}_{z'} \mathcal{KL}(f_{w+\delta}(z)||f_{w+\delta}(z')/\epsilon}\right] \quad (1)$$

The I-term aids in virtue-based accuracy, whereas the II-term, with hyper-parameter ϵ, is the regularization for adverse performances with weights $\mathcal{W}$ and tiny Gaussian noise, which is modest enough to keep generalization performance on organized data. The Equation (1) is extended by displacing $KL(.\,||.)$ with a multi-labelled adaptive loss $(.,.)$ is provided in Equation (2),

$$\min_{w} \mathcal{E}_{\delta}\left[\begin{matrix}\mathcal{E}_z(\ell(f_w(z),v)) \\ +\mathcal{E}_{z\ z'\colon \|z'-z\|_{q\le\beta}} \max \ell(f_{w+\delta}(z), f_{w+\delta}(z')/\epsilon\end{matrix}\right] \quad (2)$$

It is difficult to address a mini-max efficiency issue precisely because it is intertwined with the expected of stochastic weights. Rather, this issue can be solved in alternative way by switching among internal maximizing and external minimization.

The internal maximization problem is solved using a gradient-based technique, i.e., PGD, generates the adversarial perturbation which is iteratively produced at the t^{th} iteration by allowing Equation (4) until the maximum iterations are achieved.

$$z'_{T+1} \leftarrow \prod(z'_T + \mathcal{N}_z sn\,(\nabla_{z'_T} \ell(f_{w+\delta'}(z), f_{w+\delta'}(z'_T)))) \quad (3)$$

In Equation (3), Π is the estimation function, $\mathcal{N}_Z$ is the step size and δ' is a sample of δ. In order to eliminate the complexity, an adversarial sample is developed for every mini-batch by employing a single instances δ'. It is still possible to generate enough samples of δ' in one epoch, even with the many mini-batches. The adversarial instances that unified model starts may be improved in various strategies by using the minimization approach with stochastic weights.

By solving the internal maximizing equation for the optimal perturbed sample s', we may solve the external minimization issue by Equation (4),

$$\min_{w} \mathcal{E}_z[\ell(f_w(z),v) + \mathcal{E}_{\delta}\,\ell(f_{w+\delta}(z), f_{w+\delta}(z')/\epsilon] \quad (4)$$

The interaction among the arbitrary perturbation δ and the predictable weight $\mathcal{W}$ makes the optimization issue difficult and necessitates extensive gradient updates.

The issue is tackled by separating gradient updating and model averaging by decomposing the two components. Consider, $f_{\mathcal{W}+\delta}(z) = g_z(\mathcal{W}+\delta)$ using the Taylor extension of $g_z(\mathcal{W}+\delta)$ at $\mathcal{W}$, which is illustrated as follows in Equation (5),

$$g_z(\mathcal{W}+\delta) = g_z(\mathcal{W}) + g'_z(\mathcal{W})^T\delta + \delta^T\frac{g''_z(\mathcal{W})}{2!}\delta + o_z(\|\delta\|^2 \quad (5)$$

Here, $g'_z(\mathcal{W}) = \frac{\partial g_z(\mathcal{W})}{\partial \mathcal{W}} \in \mathcal{R}^{d*1}$ and $g''_z(\mathcal{W}) = \frac{\partial^2 g_z(\mathcal{W})}{\partial \mathcal{W}\partial \mathcal{W}} \in \mathcal{R}^{d*d}$ stands for the first and second derivative order of g_z operation with respect to the weight $\mathcal{W}$ accordingly and $o_z(\|\delta\|^2)$ will be similar to zero in accordance to $\|\delta\|^2$.

The gradient calculation will be limited to the predetermined component $\mathcal{W}$, while model smoothing using the mean arbitrary element δ will be operated individually. The perception is defined as follows,

The Taylor approximation for the randomized model $\mathcal{W}+\delta$ with a specific input z includes the definitive model $g_z(\mathcal{W})$, the extension of its gradual is connected with the arbitrary orientation $g'_z(\mathcal{W})^T\delta$ and estimate of its curve is computed with subspace stretched by $\delta^T\frac{g''_z(\mathcal{W})}{2!}\delta$. This captures high-dimensional data of the loss operation with relation to δ.

The upper limit of Equation (4) is lowered using Jensen's inequality and local convexity of the loss landscape to simplify the calculation as give in Equation (6),

$$\ell = \left(\sum_x i_x \sum_y j_y\right) \le \sum_x \ell(i_x, j_y) \quad (6)$$

Thus, the second term in Equation (4) has an upper bounding which is computed as

$$\mathcal{E}_\delta\, \ell(f_{\mathcal{W}+\delta}(z), f_{\mathcal{W}+\delta}(z')) \le \ell\left(g_z(\mathcal{W}), g_{z'}(\mathcal{W})\right) + \mathcal{E}_\delta(\ell\left(g'_z(\mathcal{W})^T\delta, g'_{z'}(\mathcal{W})^T\delta\right)) + \frac{1}{2}\,\mathcal{E}_\delta\,(\ell(\delta^T g''_z(\mathcal{W})^T\delta, g''_{z'}(\mathcal{W})^T\delta)) \quad (7)$$

In this case, the right-hand side is known as the 0th, 1st or 2nd order of the Taylor formula. The upper limit in Equation (7) is decreased rather than immediately minimizing Equation (4), which is simpler to calculate in actual models. This model aims to maintain the training by limiting the noise variance of δ to a negligible value, such as $\varphi = 0.01$. The 0th term hyperparameter $1/\epsilon$ is adjusted to 6 in this model. The number of internal refinement cycles is predefined as 10. The approach is customizable by utilizing $\mathcal{N}$ and $\mathcal{N}^2$ as 1st and 2nd term hyper-parameters. The default values are $\mathcal{N}$ are 0.3 for 1st term optimization and 0.2 for 1st + 2nd term optimization. In comparison to the single method, this proposed method involves a model fused with deterministic weight $\mathcal{W}+\delta$ to slay the weight upgrade during adversarial training. The end product is a model that can withstand possible weight perturbations and has a smoothed classification border. Additionally, the technique allows for fast computation of gradient updates by disentangling gradient from random weight perturbation. By utilizing the 0th, 1st, and 2nd Taylor term approximate the cross-entropy loss function and apply the deterministic component to the mean weight w. Because of this, the deterministic and probabilistic elements of $\mathcal{W}+\delta$ may be independently decayed and computed. The below Algorithm 1 defines adversarial training with randomized DCNN weights.

Algorithm 1: Implementation of adversarial training utilizing randomized weights in Deep Convolutional Neural Networks (DCNN)

Input: Minibatch $\{z_x\}_{x=1}^N$ network structure is optimized by $\mathcal{W}$. The learning rate is denoted as $\mathcal{N}_L$. The step size is represented by $\mathcal{N}_Z$. Hyper-parameters include $\mathcal{N}, \epsilon$, Gaussian δ. The number of iterations in internal optimization is P.

Output: Aligning Randomized weights in DCNN model $f_{\mathcal{W}}$

1. Initialize network with pre-trained arrangement or initialize network $f_{\mathcal{W}}$ randomly
2. Reiterate
3. Construct the adversarial instances:
4. Data Sample δ' from δ
5. For $x = 1$ to N do
6. $z'_x = z_x + 0.001.n(0, X)$
7. For $q = 1$ to Q do
8. $z'_x \leftarrow \prod(\mathcal{N}_Z\, sn(\nabla_{z'_x}\, \ell(f_{\mathcal{W}+\delta}(z_x), \ell(f_{\mathcal{W}+\delta}(z'_x))) + z'_x)$ which is related to $\|z_x - z'_x\|_q \le \boldsymbol{\lambda}$, Π defines the projection function
9. End for
10. End for
11. Optimization:
12. $\mathcal{W} \leftarrow \mathcal{W} - \mathcal{N}_L \frac{1}{N}\sum_{x=1}^N \nabla_{\mathcal{W}}[\,\ell(f_{\mathcal{W}}(z_x), j_x) + \ell\left(g_{z_x}(\mathcal{W}), g_{z'_x}(\mathcal{W})\right)/\epsilon +$ *Zeroth term* $\mathcal{N}\mathcal{E}_\delta\left(l(g'_{z_x}(\mathcal{W})^T\delta, g'_{z'_x}(\mathcal{W})^T\delta)\right) +$ *first term* $\frac{\mathcal{N}}{2}\mathcal{E}_\delta\left(l(\delta^T g''_{z_x}(\mathcal{W})^T\delta, \delta^T g''_{z'_x}(\mathcal{W})^T\delta)\right)]$ *second term*

Where different optimization methods can be employed like 0th term optimization, 0th +1st terms optimization and 0th +1st +2nd terms optimization

13. Continue until convergence achieved
14. Final sequencing arbitrary weights in DCNN

Thus, the developed dCAC substantially reduces adversarial assaults during access control and streamlines the process of updating access control in cloud systems.

4. Implementation Setup and Evaluation Criteria

4.1. Settings

The performance of the proposed and existing models is evaluated through simulations conducted in Python 3.11.4. The implementation utilizes GPU-accelerated TensorFlow operating on a 64-bit Windows 8 environment with an Intel Xeon 3.60 GHz processor and 16 GB of memory. For experimental purposes, Amazon in the UCI Machine Learning Repository [25] provided the Amazon-UCI dataset as the access control modules. The dataset comprises access information for over 36,000 users and 27,000 permissions. Less than 10% of the total consumers have submitted requests for access permissions. The dataset is extensively utilized in access control research, with experiments typically limited to the 5 to 8 most frequently requested permissions. The seven most sought-after permissions can be obtained, together with a roster of users granted admission to the selected requests. The data balancing task was performed on the experimental data, which was randomly divided into training data (50%) and testing data (50%) to compute the stimulation results.

4.2. Simulation Results

In this section, a comparison examination is performed between the suggested dCAC and current algorithms comprising DLBAC [17], SACS-DACS [18], IPFS-DBN [20], DLBCAC [22], PDM-ACSL [23]. The assessment of these models is conducted by accuracy, precision, recall, F1-score, time cost and memory cost and which is briefly illustrated below. The implementation details are illustrated in Section 4.1.

Accuracy: It is calculated based on the quantity of effective access control instances split by the overall amount of access control endeavours. A higher accuracy percentage indicates the respective model is more reliable and has effective access control system, as it correctly identifies legitimate users while minimizing incorrectly granting access to unauthorized users. Equation (8) defines the accuracy,

$$Accuracy = \frac{TP + TN}{TP+TN+FP+FN} \tag{8}$$

True Positives (TP): are the quantity of instances where the system correctly authenticated legitimate users. True Negatives (TN) refer to the number of occurrences in which the system correctly rejected unauthorized users. False Positives (FP) are the number of occurrences in which the system incorrectly authenticated unauthorized users, allowing them access. False Negatives (FN) are the number of occurrences in which the system incorrectly denied access to legitimate users.

Precision: It calculates the fraction of positive identifications (access control) that were indeed accurate by the model. It focuses on minimizing false positives in Equation (9),

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

Recall: It computes the proportion of actual positive cases (legitimate users) that were correctly identified by the model. It focuses on minimizing false negatives in Equation (10),

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

F1-score: It is a stabilized mean of precision and recall. It helps when legitimate and illegitimate users are unequally distributed in Equation (11),

$$F1 - score = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{11}$$

Receiver Operating Characteristic - Area Under the Curve (ROC-AUC): It graphs the TP Rate (TPR) compared to FP Rate (FPR) at various barrier levels. It assesses the framework capability to differentiate among good and bad categories. The TPR and FPR is defined in Equation (12) and in Equation (13),

$$TPR = \frac{TP}{TP+FN} \tag{12}$$

$$FPR = \frac{FP}{FP+TN} \tag{13}$$

Table 1 depicts the confusion matrix for the proposed model. Figure 5 illustrates the efficiency of various models applied on the Amazon-UCI dataset database to provide efficient cloud access control system.

From the analysis, it is observed that the efficiency of the suggested dCAC model in terms of Accuracy, Precision, Recall and F-Score is greater than the traditional models, as because the dCAC effectively mitigates the adversarial attacks during access control in cloud systems. So, this scrutiny indicates that the accuracy values obtained by the dCAC is 5.46% larger than the DLBAC, 4.37% larger than the SACS-DACS, 4.06% larger than the IPFS-DBN, 3.20% larger than the DLBCAC and 2.95% larger than the PDM-ACSL. Similarly, the precision values obtained by the dCAC is 4.82%, 4.45%, 3.55%, 2.32% and 1.39% larger than the DLBAC, SACS-DACS, IPFS-DBN, DLBCAC, and PDM-ACSL, respectively.

Table 1. Confusion matrix for the proposed DCAC model

Actual/Predicted	Predicted Active	Predicted Decline
Actual Active	4850 (TP)	120 (FN)
Actual Decline	187 (FP)	4843 (TN)

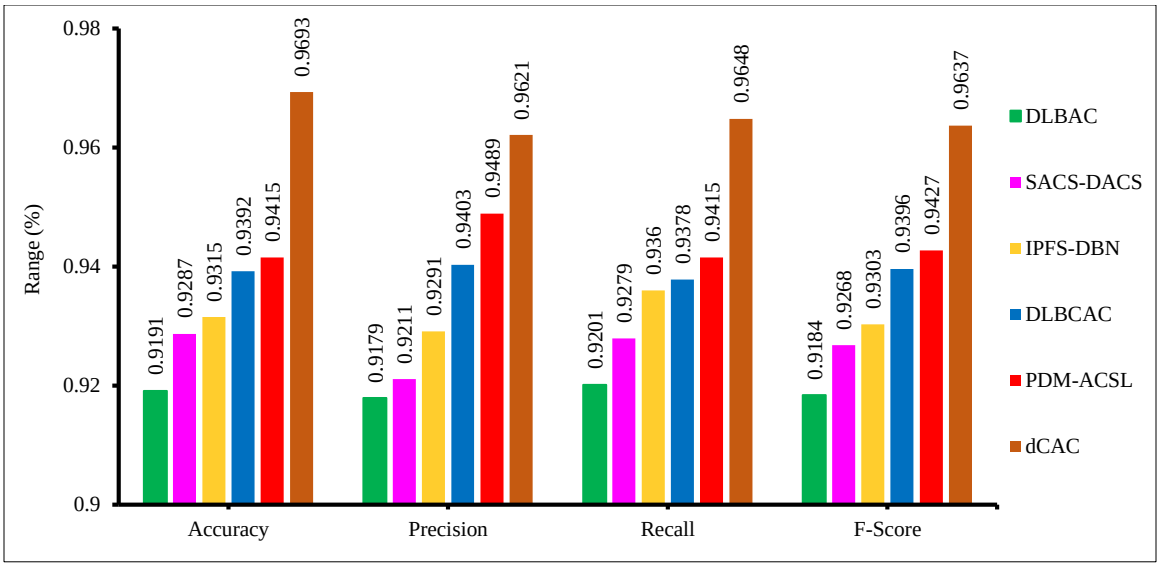


Fig. 5 Performance evaluation of proposed and existing cloud access control models

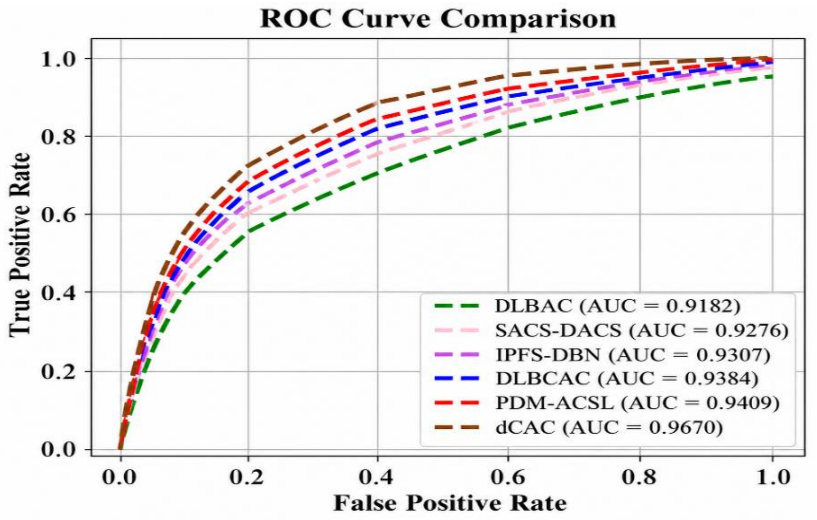


Fig. 6 AUC comparison of the proposed and existing models

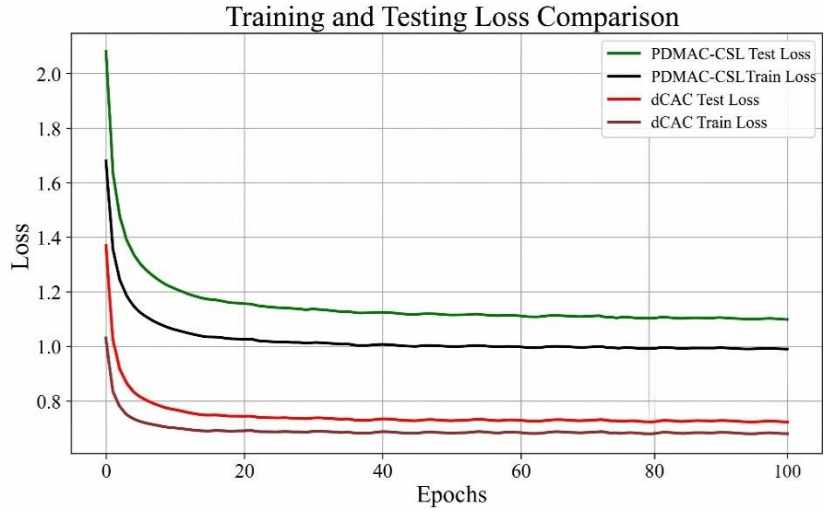


Fig. 7 Training and testing loss learning curve

Finally, the F-score values obtained by the dCAC is 4.93%, 3.98%, 3.59%, 2.56% and 2.23% larger than the DLBAC, SACS-DACS, IPFS-DBN, DLBCAC and PDM-ACSL accordingly. From the analysis, it is observed that the dCAC effectively mitigates the adversarial attacks during access control in cloud systems.

Figure 6 depict the AUCs of various classifiers applied to effective cloud access control. It represents that, the AUC of proposed dCAC model is 0.9670, which 5.31%, 4.25%, 3.90%, 3.05%, and 2.77% higher than the DLBAC, SACS-DACS, IPFS-DBN, DLBCAC and PDM-ACSL models, respectively. The reason behind the decision-making process of the dCAC model can be optimized that enhances generalization in a wide access patterns and threat conditions leading to increased TPR, reduced FPR and better AUC, with respect to traditional access control systems.

Figure 7 represents the training and testing curve of loss for the proposed dCAC and existing PDMAC-CSL models. The dCAC model has lower loss and smooth convergence compared to PDMAC-CSL. This indicates better learning efficiency, generalization capability. Finally, the proposed model is better than the already available method with reduced error and improved performance.

Figure 8 represents the training and testing accuracy curves for the proposed dCAC and existing PDMAC-CSL model. The dCAC model shows the best accuracy with stable convergence as compared to the PDMAC-CSL model. This shows an indication of learning and generalization performance. Overall, the proposed model is better than the existing method.

Figure 9 and Figure 10 show the number of requests that were served in single-thread and multi-thread. The time taken to process the access control mechanisms of the mentioned models were compared; the DLBAC, SACS-DACS, IPFS-DBN, DLBCAC, PDM-ACSL and the proposed dCAC model. The findings reveal that the models, including IPFS-DBN and PDM-ACSL, have much longer processing time with the growth in the volume of requests.

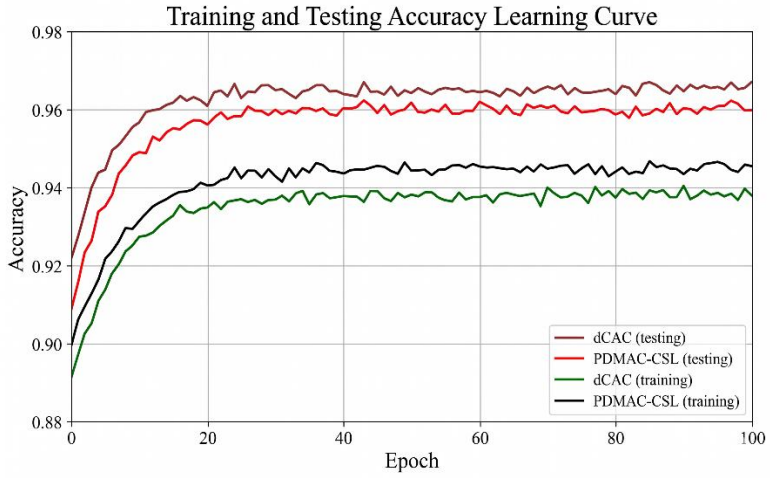


Fig. 8 Training and testing accuracy learning curve

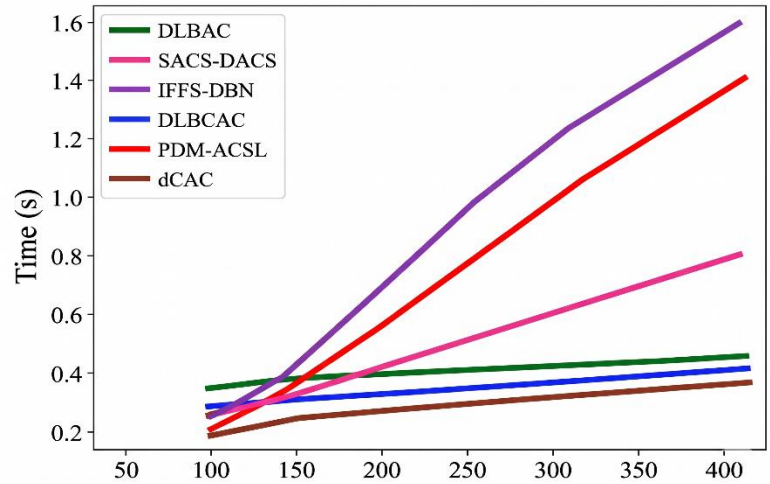


Fig. 9 Number of requests handled - single thread

This is mainly because of distributed validation processes and other calculation overheads of their structures. On the contrary, DLBAC and DLBCAC show more stable performance, with the increase in latency being rather marginal with the increase in workload. The proposed dCAC model has the shortest processing time in each request tier in the multi-thread. This shows it is more scalable and robust with high concurrency and thus is more applicable to real-world large-scale deployments.

Within the single-thread, the total latency is more in sequential chain-based processing, but the suggested dCAC model is always better than the existing models. This ascertains its lower complexity of computation and greater efficiency. Figure 11 shows that the average waiting time increases for all models as the number of requests rises from 100 to 500. Among them, dCAC has the highest waiting time, which indicates lower efficiency under higher load conditions.

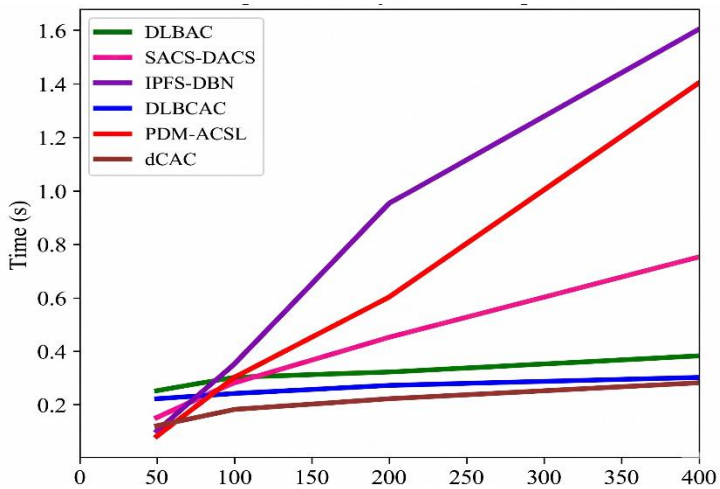


Fig. 10 Number of requests handled - multi thread

Meanwhile, DLBAC consistently keeps the lowest waiting time, showing better performance. The other models, including PDM-ACSL, DLBCAC, IPFS-DBN, and SACS-DACS, show moderate and stable performance with gradual increases. Overall, DLBAC is more efficient at handling a larger number of requests. Figure 12 shows that as the number of requests goes from 100 to 500, the response time for all models goes up. dCAC has the longest reaction time of the group, which means it doesn't work as well when there is a lot of traffic. DLBAC, on the other hand, always has the shortest reaction time, which means it works faster and better. The other models-PDM-ACSL, DLBCAC, IPFS-DBN, and SACS-DACS-are all doing well overall because they are all growing at a steady and small rate.

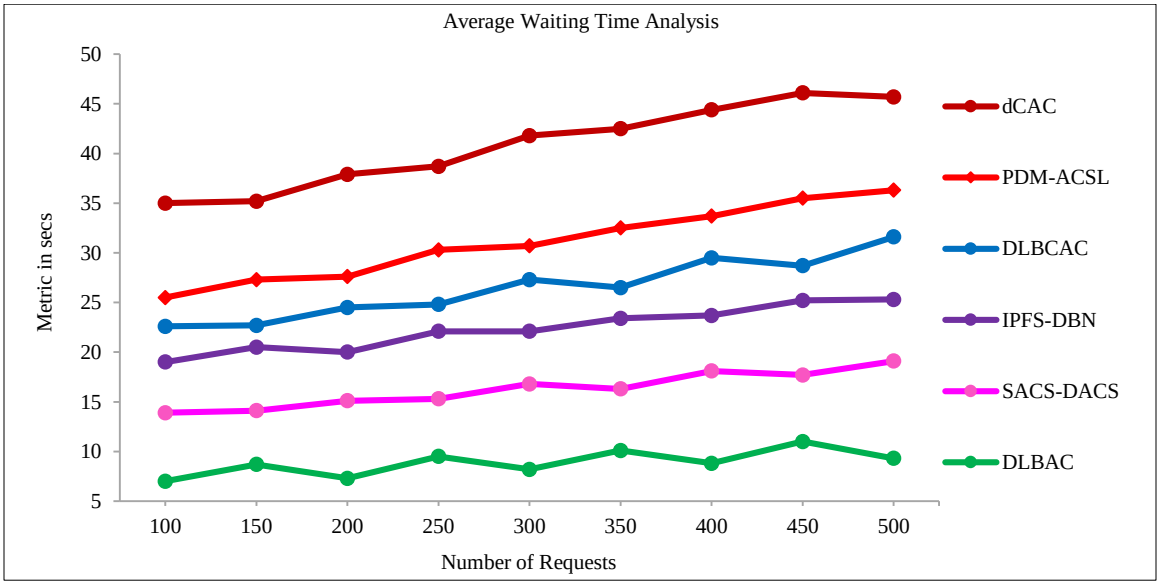


Fig. 11 Average waiting time analysis

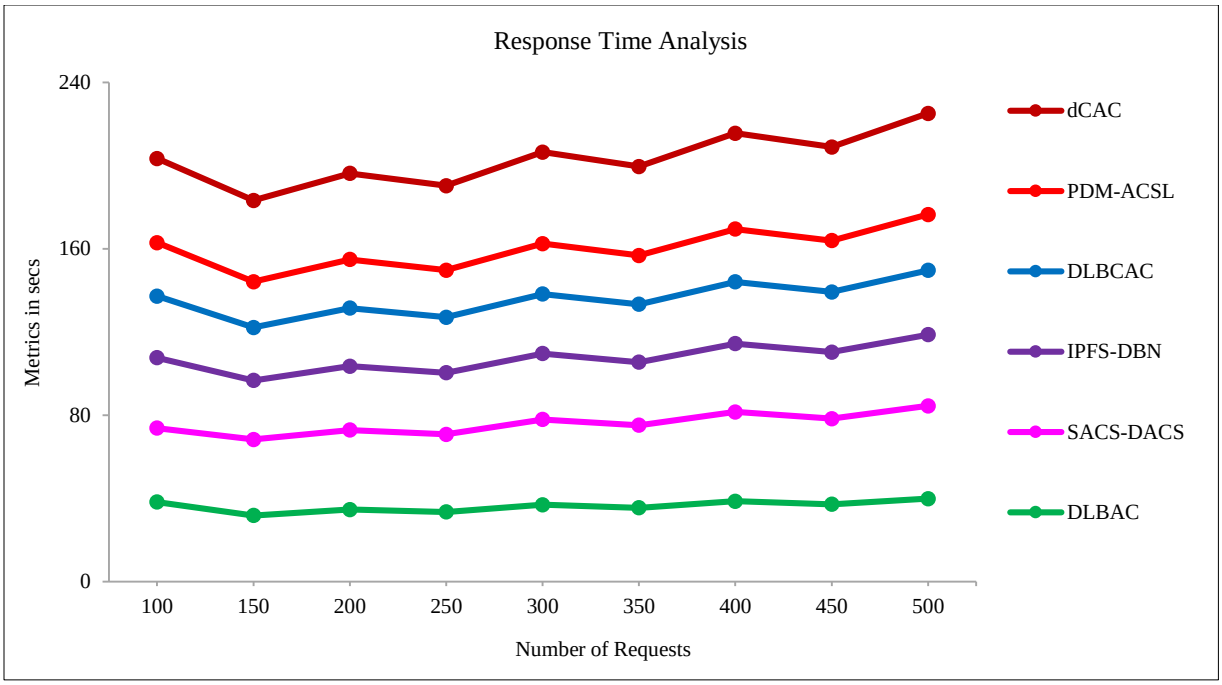


Fig. 12 Response time analysis

Table 2. Execution time and memory cost

	DLBAC	SACS - DACS	IPFS - DBN	DLBCAC	PDM - ACSL	dCAC
Time (ms)	75	64	59	46	22	**16**
Memory (mb)	141	136	128	113	95	**81**

Table 2 depicts the results of time and memory costs for proposed and existing methods to ensure a secure, cost-effective system and provide solutions to optimize access control processes in cloud. From the above analysis, it is observed that the time cost (ms) of proposed dCAC model is 78.67%, 75%, 78.88%, 65.22% and 27.27% larger than the DLBAC, SACS-DACS, IPFS-DBN, DLBCAC and PDM-ACSL, respectively. Similarly, the memory cost (mb) of proposed dCAC model is 42.55%, 40.44%, 36.72%, 28.32% and 14.74% larger than the DLBAC, SACS-DACS, IPFS-DBN, DLBCAC and PDM-ACSL respectively. From this analysis, it is proved that the proposed model attains a higher lower time and memory cost than the other existing cloud access control methods as the proposed dCAC reduces the complexity and resource demands which significantly reduces the time and improves the scalability while executing the access control task in cloud systems.

4.3. Discussion

The proposed dCAC framework achieves superior performance compared to existing cloud access control methods. It integrates robust adversarial training with an efficient DL-based access control mechanism. The DCNN effectively learns user and resource metadata, enabling accurate access control decisions without relying on manually engineered rules. In addition, the use of IG and KT improves the interpretability of the model by explaining the factors influencing access decisions and identifying the relationships between metadata. The incorporation of the Taylor series approximation simplifies the optimization process, resulting in faster convergence, diminished the level of computation, and decreased execution duration. Consequently, the dCAC model realises enhanced accuracy, precision, recall, and F1-score and AUC while requiring less execution time and memory than the existing methods. These improvements demonstrate that the proposed structure delivers a safe, efficient, and expandable solution for cloud access control.

5. Conclusion

This paper presents a dCAC model designed to prevent adversarial attacks during access control in cloud environments. The DCNN model is used to maintain user access control as training data, allocating access data levels for authenticated users in the cloud. IG and KT are employed to know the internal details of the DCNN with IG attributes providing decision-making features and KT determining the relationships between metadata and network logic.

Optimized adversarial training over randomized DCNN weights is suggested to enhance the robustness and accuracy of adversarial attack prediction. Taylor series expands the objective function over DCNN weights replacing the objective function with Taylor terms like zero, first, and second. The model effectively prevents adversarial attacks and reduces the

complexity of access control update in cloud systems. Finally, an extensive simulation reveals the model achieves an accuracy of 96.93% on the amazon-uci dataset.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

[1] Rashid Nazir et al., "Cloud Computing Applications: A Review," *EAI Endorsed Transactions on Cloud Systems*, vol. 6, no. 17, pp. 1-11, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[2] I. Odun-Ayo, B. Odede, and R. Ahuja, "Cloud Applications Management-Issues and Developments," *International Conference on Computational Science and its Applications*, Melbourne, VIC, Australia, pp. 683-694, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Zheng Li et al., "On Evaluating Commercial Cloud Services: A Systematic Review," *Journal of Systems and Software*, vol. 86, no. 9, pp. 2371-2393, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[4] Muthu Ramachandran, and Victor Chang, "Towards Performance Evaluation of Cloud Service Providers for Cloud Data Security," *International Journal of Information Management*, vol. 36, no. 4, pp. 618-625, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] Josep Domingo-Ferrer et al., "Privacy-Preserving Cloud Computing on Sensitive Data: A Survey of Methods, Products and Challenges," *Computer Communications*, vol. 140-141, pp. 38-60, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Zhifeng Xiao, and Yang Xiao, "Security and Privacy in Cloud Computing," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 843-859, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[7] Aized Amin Soofi, M. Irfan Khan, and Fazal-e-Amin, "A Review on Data Security in Cloud Computing," *International Journal of Computer Applications*, vol. 96, no. 2, pp. 95-96, 2017. [Google Scholar]

[8] T.K. Subramaniam, and B. Deepa, "Security Attack Issues and Mitigation Techniques in Cloud Computing Environments," *International Journal of UbiComp (IJU)*, vol. 7, no. 1, pp. 1-11, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Kazi Zunnurhain, and Susan V. Vrbsky, "Security Attacks and Solutions in Clouds," *Proceedings of the 1st International Conference on Cloud Computing*, pp. 145-156, 2010. [Google Scholar]

[10] Jain Jaya, *Application of Deep Learning in Cloud Security*, Deep Learning Approaches to Cloud Security, Wiley Publisher, pp. 189-205, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Sahil Garg et al., "A Hybrid Deep Learning-based Model for Anomaly Detection in Cloud Datacenter Networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 924-935, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Ammar Aldallal, and Faisal Alisa, "Effective Intrusion Detection System to Secure Data in Cloud using Machine Learning," *Symmetry*, vol. 13, no. 12, pp. 1-26, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] N. Rajkumar et al., "Deep Learning based user Identity Management Protocol for Improving Security in Cloud Communication," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 3, pp. 444-453, 2022. [Google Scholar] [Publisher Link]

[14] Abdelhakim Zeroual et al., "Lightweight deep Learning Model to Secure Authentication in Mobile Cloud Computing," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 6938-6948, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Mohammad Nur Nobi, and Ram Krishnan, "Adversarial Attacks in Machine Learning based Access Control," *Italian Conference on Big Data and Data Science (ITADATA2022)*, pp. 1-12, 2022. [Google Scholar]

[16] Chaoyun Zhang, Xavier Costa-Pérez, and Paul Patras, "Adversarial Attacks against Deep Learning-based Network Intrusion Detection Systems and Defense Mechanisms," *IEEE/ACM Transactions on Networking*, vol. 30, no. 3, pp. 1294-1311, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mohammad Nur Nobi et al., "Toward Deep Learning based Access Control," *Proceedings of the Twelfth ACM Conference on data and Application Security and Privacy*, pp. 143-154, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] S. Vijayanand, and S. Saravanan, "A Deep Learning Model Based Anomalous Behavior Detection for Supporting Verifiable Access Control Scheme in Cloud Servers," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 42, no. 6, pp. 6171-6181, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Nawaf Alharbe et al., "An Access Control Model Based on System Security Risk for Dynamic Sensitive Data Storage in the Cloud," *Applied Sciences*, vol. 13, no. 5, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Mariya Princy Antony Saviour, and Dhandapani Samiappan, "IPFS based Storage Authentication and Access Control Model with Optimization Enabled Deep Learning for Intrusion Detection," *Advances in Engineering Software*, vol. 176, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] D. Paulraj et al., "Admission Control Policy and Key Agreement based on Anonymous Identity in Cloud Computing," *Journal of Cloud Computing*, vol. 12, no. 1, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Asma Jodeiri Akbarfam et al., "Deep Learning Meets Blockchain for Automated and Secure Access Control," *arXiv preprint*, pp. 1-21, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Aljuaid Turkea Ayedh M, Ainuddin Wahid Abdul Wahab, and Mohd Yamani Idna Idris, "Enhanced Adaptable and Distributed Access Control Decision Making Model based on Machine Learning for Policy Conflict Resolution in BYOD Environment," *Applied Sciences*, vol. 13, no. 12, pp. 1-23, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Mohammed Alkhowaiter et al., "Adversarial-Aware Deep Learning System based on a Secondary Classical Machine Learning Verification Approach," *Sensors*, vol. 23, no. 14, pp. 1-18, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[25] Ara V. Nefian, Georgia Tech Face Database, Academic Torrents, 2015. [Onilne]. Available: https://academictorrents.com/details/0848b2c9b40e49041eff85ac4a2da71ae13a3e4f