Received: February 1, 2024. Revised: March 26, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Dwindling DCNN Training Time and Improving Authentication by Spurring Algorithm in Cloud Computing

Kandasamy Gowri[1]* **Banthumy Lingam Shivakumar[1]**

[1]*Department of Computer Science, Sri Ramakrishna College of Arts and Science, Coimbatore - 641006, Tamil Nadu, India*
*Corresponding author's Email: gowridhilipkphd@gmail.com

Abstract: Cloud Computing (CC) is a solution that efficiently maps cloud tasks to resources, ensuring high-performance utilization in computing. But, the challenge lies in managing attacks while managing a large data flow. For this reason, Lightweight deep learning model to secure authentication (LDLSA) was developed using Deep Convolutional Neural Network (DCNN) and Homomorphic Encryption (HE) is developed for deep face recognition authentication, enhancing privacy protection in CC. However, these models are computationally expensive, require extensive training data and lack information on cloud-stored authentication data for privacy preservation. In this paper, DL with Cloud Authentication (dCAuth) is proposed to resolve the above mentioned issues to provide efficient cloud authentication with lesser complexity. This method employs a spurring algorithm for dwindling DCNN with recall (dDCNNr) to handle the number of parameters in DCNN to handle a large number of parameters for cloud authentication. dDCNNr model reduces execution time and improves classification accuracy, but it necessitates faster training time. To solve this, Fully Connected (FC) layers of DCNN is replaced with Hopfield Neural Network (HNN). The model utilises a DCNN for feature extraction and HNN for pattern recognition highly focusing on authentication, both the process have similar training phase with known patterns by adjusted weights. The proposed authentication method stores encrypted Neural Network (NN) weights in the cloud, eliminating the need for a verification table. The weights for unknown input patterns are generated during pattern recall. The HNN weight matches input indicating a known or legal pattern, aiming to identify a known pattern that best fits the input with minimum processing resources. The proposed authentication model integrates DCNN and HNN to rapidly and precisely recall the legitimate user ID and face image (password) information. The model effectively manages the large training data and parameters, ensuring privacy for cloud-stored authentication data, simultaneously reducing the time duration of registration and password changes (old to new face image). Finally, an extensive simulation reveals that the proposed model achieves accuracy of 94.15%, 94.53% and 94.38% on Georgia Tech face (GTF), Labelled Faces in the Wild (LFW) and Biometric Signature (BS) database respectively.

Keywords: Cloud computing, Deep learning, Hopfield neural network, Pattern recall, Cloud authentication.

1. Introduction

CC is a popular on-demand model that integrates parallel and distributed computing, using shared resources like software and hardware [1]. It allows users to access and pay for resources with a web connection, promoting coherence and economic growth [2]. CC allows businesses and consumers to store and process data in private facilities or third-party data centers, reducing organizational effort [3]. The end-user information will be cached in the data centers of the Cloud Service Provider (CSP), raising concerns about privacy [4]. However, managing user identity and providing adequate privacy and protection remains a significant challenge in CC.

The CC has implemented security measures like authentication, access control and threat detection models to enhance the security of cloud-based information [5]. The performance of these parameters is impressive, but additional security measures are needed to protect information, as cloud infrastructure using standard IP and virtualization techniques may

be susceptible to attacks like IP spoofing, insider threats and DoS [6]. Conventional prediction and avoidance systems struggle to detect new attack types or their signatures, and are insufficient for handling large data volumes.

Recently, DL has shown outstanding performance in a variety of applications, including image recognition, pattern matching and even cyber-security [7]. DL models are increasingly used in cloud security to protect user information, preventing from data vulnerabilities, providing fast problem-solutions, large automation, high-quality results, cost reduction, data labelling and complex interaction identification [8]. CNN, Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) are the types of DL algorithms which offers significant prediction results in cloud security, enhancing detection rates, identifying known and unknown attacks, improving data privacy, predicting unauthorised users and restricting them [9].

By utilizing the DL concept, LDLSA model was developed [10] using DCNN and HE for face recognition authentication in mobile CC. This model utilizes the information of already registered user's and compares their images to previously categorised images on a cloud slide after receiving their registration. If the outcome is affirmative, the user can access the application server using partially HE. If the result was negative, the cloud side sends an illegal access request to the user. In the cloud side, DCNN model was used to extract the features and for classification for efficient authentication. But, this model fails to address the computational cost of DCNN due to large training data and parameters and lacks details on cloud storage for privacy of authentication data.

In order to resolve this, dCAuth is developed to resolve the above mentioned issues to provide efficient cloud authentication with lesser complexity. This method employs a spurring algorithm for dDCNNr to efficiently handle large parameters in DCNN for cloud authentication with less execution time and efficient classification accuracy but requiring faster training time. To solve this, FC layers of DCNN is replaced with HNN. This model combines a DCNN and an HNN for feature extraction and pattern recognition, highly focusing on authentication using pattern recognition. The proposed authentication method uses no verification table and stores encrypted NN weights in the cloud, enhancing its security. Pattern recall involves creating a weight for unidentified input patterns and determining whether the resultant matches the input when the pattern has been determined to exist or is valid. Pattern categorization strives for a well-known pattern that significantly fits the input, whereas pattern recall uses less processing power. The suggested authentication technique employs the DCNN and HNN to rapidly and correctly ensure information for an authorized user's ID and facial image (password). The system effectively maintains an extensive amount of training data and parameters, protecting the confidentiality of cloud-stored authentication details and reducing time to registration and password changes (old to new face image). This lowers the omplexity in handling the large number of training data and parameters of DCNN with respect to the privileged security for cloud storage authentication data.

The remaining article is prepared as follows: Section II addresses several DL-based cloud security techniques that have appeared in recent years. Section III outlines the proposed dCAuth algorithm whereas Section IV evaluates its effectiveness. Section V is a synopsis of the whole text.

2. Related works

A multi-device continuous authentication approach, AuthCODE was developed [11] using Artificial Intelligence (AI)-based privacy preservation method. The model improves single-device solutions by incorporating behavioral data from various devices and combining features to enhance user authentication. However, this model was susceptible to over-fitting difficulties which reduce accuracy.

A Face Recognition system based on Cloud Authentication (FRCA) was developed [12] using Tree-based Deep Neural Network (TDNN) for automated face verification in a cloud. The tree was characterized by its limb length and weight with remaining ability consisting of a two-stage layer, array game plan and indirect competence. However, this model results in lower F1-Score.

A secure E-Education framework was constructed [13] for cloud environments, enabling user authentication through multiple factors based on user roles and activities. The VGG2F-CNN was utilized to improve the accuracy of biometric face authentication by retrieving accurate biometric face attributes. But, this model results in high training time and required more manual inception.

A Biometric Authentication framework in Mobile Cloud (BAMCloud) was developed [14] using various dynamic signatures. This framework uses Principle Component Analysis (PCA) for pre-processing and Feed Forward Back-propagation Neural Network (FFBPNN) method stored in cloud. But, this model result in high time complexity issues.

A Cryptography- and Machine Learning (ML)-based Authentication Protocol (CMAP) was presented [15] to protect the data interchange in CC. The ensemble voting classifier was utilized to provide efficient online data sharing through registration, key agreement and password change phases for authentication. But, this method obtain lower accuracy for authentication due to problems like privacy leakage.

A Face Feature Ciphertext Authentication Scheme (FFCAS) was developed [16] using HE and FaceNet model. The initial extraction of face image features was done using the FaceNet model, followed by packaging them into ciphertext using HE. However, lower accuracy was resulted as the parameters of DL needs to be optimized.

A flexible multi-layered authentication model was developed [17] in cloud platform. This authentication mechanism uses multiple factors like user length, validity, and value, geolocation and browser confirmation to enhance identity verification in cloud users to protect the data. However, abundant memory and time cost were determined form this model.

A model was developed [18] to improve data protection in CC environment by employing the mutual authentication method with DL-based hybrid encryption technique. This technique offers a flexible approach to cloud data security by fusing the advantages of CNN with hybrid encryption. On the other hand, this model has high complexity in terms of memory and time.

3. Proposed methodology

The complete working module of the suggested dCAuth method is shown extensively in this section. Fig. 1 depicts the suggested dCAuth paradigm. Table 1 lists the notations used in this study.

3.1 Feature extraction using DCNN

Here, features from the collected dataset are retrieved using the DCNN, which was constructed by stacking different convolutional and pooling layers. At first, convolution layer will be initialized. The numerical neuron value n_{xy}^{a} associated with the y^{th} feature map in the x^{th} layer of the position a is defined as follows in Eq. (1) and Eq. (2)

$$n_{xy}^{a} = G\left(B_{xy} + \sum_{p} \sum_{z=0}^{Z_x - 1} W_{xyp}^{z}\, n_{(x-1)p}^{a+z} \right) \quad (1)$$

$$G(a) = \tanh(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}} \quad (2)$$

Where z_a represents the kernel dimensions of the face image scale, B_{xy} denotes the bias of y^{th} feature map in the x^{th} layer, W_{xyp}^{z} denotes the weight of point z linked to the p^{th} feature map, p is the feature map in the preceding layer $(y-1)^{th}$ layer that is coupled with the current feature map. By reducing the density of the feature maps, pooling might potentially give invariance. Every subsequent pooling layer is subsequently similar for its preceding convolutional layer. A patch from $N \times 1$ convolution layer is combined by a neuron in the pooling layer.

Table 1. Lists of notations

Notations	Description
n_{xy}^{a}	Neuron value with y^{th} feature map and x^{th} layer of the position a
z_a	Kernel dimensions of face image scale
p	Feature map in the preceding layer
W_{xyp}^{z}	Weight of point z linked to the p
N_{epoch}	Number of training data and parameter
E_t	Ejection rate
S_t	Block Threshold
$\mathcal{R}^{1*c}$	Class Vector
u_x	Testing Point
e_x	Minimized error rate
N	Testing samples
c	Data classes
$\mathbb{M}$	commencing epoch with all activation parameters
$\Upsilon \in [0,1)$	Weight variable
s_a	Neuron in state a
$V \in \mathcal{R}^{q*N}$	Forward-Propagation Output
ptn_{max}	Maximum number of patterns in DCNN
W_{ab}	Weight link from node a to b
i_a^s	Dimension instantiation sequence with s
$L.$	Number of instances
H	Hard limiting function
ID	user identity
UID_R	User identity of Registered user
UFI_R	Extracted feature from face image of UID_K by DCNN for authentication
U_R	Binary value of encrypted UFI_R and UID_R
UID_K	User identity Authenticated user
UFI_K	Extracted feature from face image of UID_K by DCNN for authentication
U_K	Binary value of encrypted UFI_K and UID_K

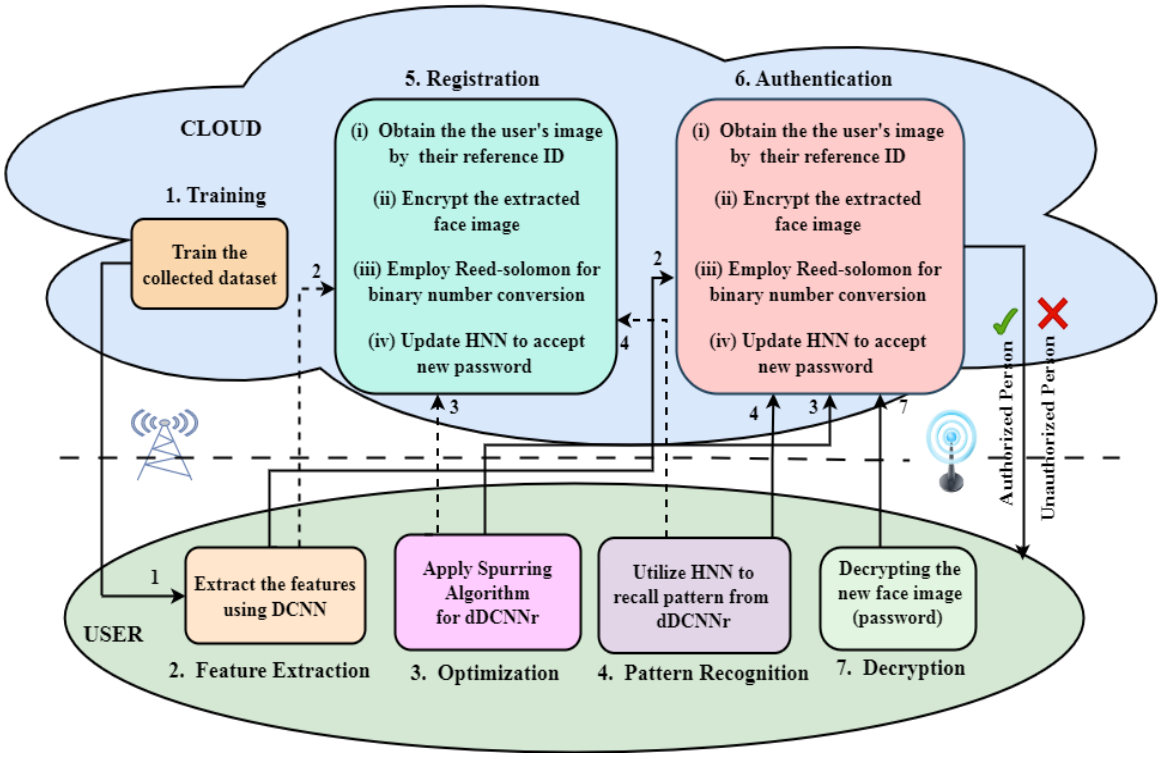

Figure. 1 Pipeline of the proposed dCAuth model

The simplest and popular pooling procedure i.e., max pooling is employed in this study is mentioned in Eq. (3)

$$i_y = \underset{N\times 1}{max}\left(i_x^{n\times 1}\,\mathbb{F}(n,1)\right) \tag{3}$$

Where in Eq. (3) i_y is the neighborhood maximum and $\mathbb{F}(n,1)$ denotes the window operation of convolution layer. By utilizing the concept of DCNN model, the feature extraction is carried out in this model.

3.2 Spurring algorithm for dwindling DCNN with recall (dDCNNr)

The spurring algorithm is a robust and reliable model to effectively fine-tune the parameters, removes the small error samples from training data and utilizing fewer data for training.

Let $v_x^{(0)} \in \mathcal{R}^{1*c}$ be the class indicator vector for the testing point $u_x \in \mathcal{R}^{q*1}$, $i = 1,2,..,N$. Here, N and c denotes the amount of testing samples and classes. To determine the category of this test extremities, v_x includes only 0s with the exception of .Assume $v_x \in \mathcal{R}^{1*c}$ be the resultant of the NN preceding the validation extremities u_x. v_x is the x^{th} row of V and has continuous values. The spurring algorithm necessitates two inputs, i.e., and block threshold S_t and ejection rate E_t. E_t represents the proportion indicating the number of training data and parameter to be eradicated with one epoch, S_t denotes the integers to delay exterminateing the training data and optimize the parameter when $N_{epoch} < S_t$, N_{epoch} is the present number of training data and parameter. The implicated array $\mathbb{M}$ is retained and applied on a subset of all training terms and parameter.

The commencing epoch, $\mathbb{M} = \{1,2,\dots,N\}$ is adjusted to encompass all activation parameters. $N_{epoch}E_t$ indexes of training data have been selected with the least error e_x preceding forward and backward propagation of all epoch. Then $\mathbb{J}$ tokens is eliminated from $\mathbb{M}$ and upgrade $\mathbb{M}$ which is defined as $\mathbb{M} = \mathbb{M} - \mathbb{D}$ in which there is no longer training data while considering $N_{epoch}S_t$ indesexes. The $N_{epoch}S_t$ indices of training data with the lowest error e_x are chosen. The threshold for excluding data changes across batches for the similar epoch. Consider there are N_{batch} per epoch, it is necessary to drop $N_{epoch}E_t \backslash N_{batch}$ samples on average in each batch. In batch x, the threshold $N_{epoch}E_t \backslash N_{batch}$ will remain as lowest error $(S_t)_x$ in batch $x+1$. The threshold $(S_t)_{x+1}$. $(S_t)_x$ and $(S_t)_{x+1}$ leads to large variations. The threshold for removing samples is configured using exponential smoothing with batch $x+1$ as the threshold shown in Eq. (4)

$$(S_t)_{x+1}{}' = \Upsilon(S_t)_x{}' + (1-\Upsilon)(S_t)_{x+1} \tag{4}$$

Where $\Upsilon \in [0,1)$ is a weight variable that regulates the significance of previous threshold integers and $(S_t)_1{}' = (S_t)_1$. The idea behind polynomial averaging is to preserve the criterion used for every epoch to be consistent. In batch $x + 1$, the samples with errors less than $(S_t)_{x+1}{}'$ will be deleted. With a realistic value of 0.6 - 0.7, the smoothing impact on threshold is not visible while Υ is near to 0 and diverges greatly from $(S_t)_{x+1}{}'$ when $(S_t)_{x+1}$ when is relatively closer within (0, 1).

The dDCNNr model compensates for the weight function W acquired with the sub-category of the total data sample. To use all of the training data, the number of active learning repetitions $N_{epoch} < S_t$ must be greater than $\mathbb{M} = \mathbb{M}_0$. In this model, the threshold S_t on the classification performances will be computed which effectively reduces the execution time and improves classification accuracy for cloud user authentication by service providers. However, the overall training time will be increased as it utilizes all the training data which provides the drift between the speedup and classification results.

To resolve this, FC layers of DCNN is replaced with a HNN which is trained with known patterns by adjusting the weights by recalling the pattern. Multiple trainable weights (parameters) in the FC layer may be replaced while retaining efficiency in accordance to the utilization of a correlated memory bank from the HNN. Algorithm 1 defines the stepwise process of dDCNNr

Algorithm 1: Dwindling DCNN with Recall

Input: Data matrix $U \in \mathcal{R}^{q*N}$, class matrix $V^{(0)} \in \mathcal{R}^{N*c}$, elimination rate E_t (%), stop threshold S_t

Output: Adjusting the DCNN parameter

1. Pre-processing training data
2. Adapt training data indices $\mathbb{M} = \{1, 2, \dots, N\}$, $\mathbb{M}_0 = \mathbb{M}$
3. for $epoch = 1, 2, \dots$ do
4. Execute forward-propagation on $\mathbb{M}$
5. Calculate forward-propagation output $V \in \mathcal{R}^{q*N}$
6. Accomplish back-propagation
7. Reform the weight W using $W^{epoch+1} = W^{epoch} - \delta\nabla(W^{epoch})$
8. if $N_{epoch} < S_t$ then
9. Determine the error using Eq. (4)
10. Adjust $\mathbb{D}$ with indices $N_{epoch}S_t$ of lowest $e_x$values
11. Remove all samples in $\mathbb{D}$ and upgrade $\mathbb{M}$, $\mathbb{M}$= $\mathbb{M} - \mathbb{D}$
12. Else
13. Emply all data for training task
14. end if
15. end for
16. Quantify the classification error V and $V^{(0)}$

3.3 Hopfield neural networks

In HNN, each of the n neurons are individually and sequentially updated in their particular activation value. The neuron a is categorized by its state $s_a = \pm 1$. The HNN assists to preserve the binary sequences in the form of $\{+1, -1\}^n$ and then it employs Hebb's rule to learn the weight patterns W of dDCNNr. Patterns are anticipated throughout the testing phase using a noisy input vector, which is very important in a variety of application because to its resistance to noise. The potential described in the Hopfield networks is depicted as,

$$HNN\ (\mathcal{E}) = -\frac{1}{2}\sum_{ab}^{n} s_a s_b W_{ab} \tag{5}$$

In Eq. (5), weight related to neurons a and b is denoted by W . The s_a represents the position of neuron a. This amount is a Lyapunov function that stays steady or diminishes when network configuration changes. The HNN converges to a limited optimum in the energy operation, which necessitates the use of optimum weight values that reduce the energy function. Eq. (6) gives the potential storage ability of the HNN assuming pattern consistency.

$$ptn_{max} = \frac{n}{4\ ln\ n} \tag{6}$$

Here, ptn_{max} is the maximum number of independent patterns recorded in the DCNN. Every encoded sequence represents a local minimum of energy as described in Eq. (2). The HNN is utilized for pattern detection in this model which comprises g nodes and each node's output is fed into other nodes $b = b(b = 1, \dots, a-1, a+1, \dots, N)$ via weights W_{ab} . The weights are identical for $W_{ab} = W_{ba}$. Every node computes a weighted sum of $g - 1$ inputs and implements a quadratic function derived for the output resultant generated on each node. The subsequent relationship weights are modified throughout the training phase, as follows in Eq. (7)

$$W_{ab} \begin{cases} \sum_{s=1}^{L} i_a^s i_b^s & a \neq b \\ 0 & a = b\ \ (1 \leq a, b \leq z) \end{cases} \tag{7}$$

Where, W_{ab} is the weight connection from node a to node$b$; i_a^s is the dimension an of the instantiation

sequence with regard to s and L is the amount of instances. When calculating W_{ab}, i_a^s is converted from a binary integer to a bipolar value, i.e., $+1$ for 1 or -1 for 0. In the sequence recall stage, a simulation is carried out after allocating the node's result to the unknown input stack using Eq. (8).

$$j_b(S_t + 1) = \mathcal{H}\left[\sum_{a=1}^{n} W_{ab} j_a\ (S_t)\right]\ (1 \le b \le n) \qquad (8)$$

Where H is the variability constant values. Consider the pattern recall problem is handled by this study using HNN. There is a single execution of Eq. (8) throughout pattern recall, and the comparison is accomplished by comparing the input and output. Put simply, the learned HNN checks if the input and output are matching immediately after executing Eq. (8) once, before doing any additional repetitions. Handling discontinuous binary information is where HNNs shine compared to layered NNs. With a small amount of training data (weight distribution), HNNs can recall a recognized pattern instantly and effectively.

In Eq. (8), intensely inhibiting uncertainty operator is represented by H. To address pattern recall, this study uses HNN. The pattern recall process involves a single execution of Eq. (8) with a matching comparison of the relevant input and output. The obtained HNN checks whether the input and output are consistent after just one repetition of Eq. (8), When it comes to discontinuous discrete data, HNNs outperforms better than multilayer NNs. For large datasets, HNNs needs little training time (weight allocation) and provide rapid, precise recall of established patterns.

3.4 HNN based pattern recognition

In this model, pattern recall-based cloud data authentication is provided. The pattern recall is not the same as pattern recognition. When pattern identification categorizes previously unseen patterns into established classes, pattern recall checks for prior observations of a pattern. The training challenge is same for pattern recognition and pattern recall. Although the process for pattern recall is different, the training phase for categorization and HNN is comparable; both use weight modification to learn identified patterns. The HNN can execute Eq. (8) concurrently, which is great for pattern recall. Assuming the output of the HNN is identical to the input (a valid and authorized pattern throughout the context of this investigation), then the link has previously been identified. In such case, the pattern is not legitimate. By iteratively running Eq. (8) until convergence is reached (i.e., no change in the output j), pattern identification attempts to discover a known pattern that closely matches the input. The recall quality of a HNN is significantly influenced by its informational capacity, which refers to the number of patterns it can store.

This capacity increases with the number of nodes and weights, assuming all other factors remain constant When the capacity is exceeded, the HNN only remembers primary patterns, leading to information saturation and pattern decay, resulting in unpredictable recall behavior near saturation. To enhance memory intention, the algorithm can increase data ability and encoding sequences sparingly by employing configurations of three binary digits with at least three vertices. This model uses Reed-Solomon Coding (RSC) [19] to maximize

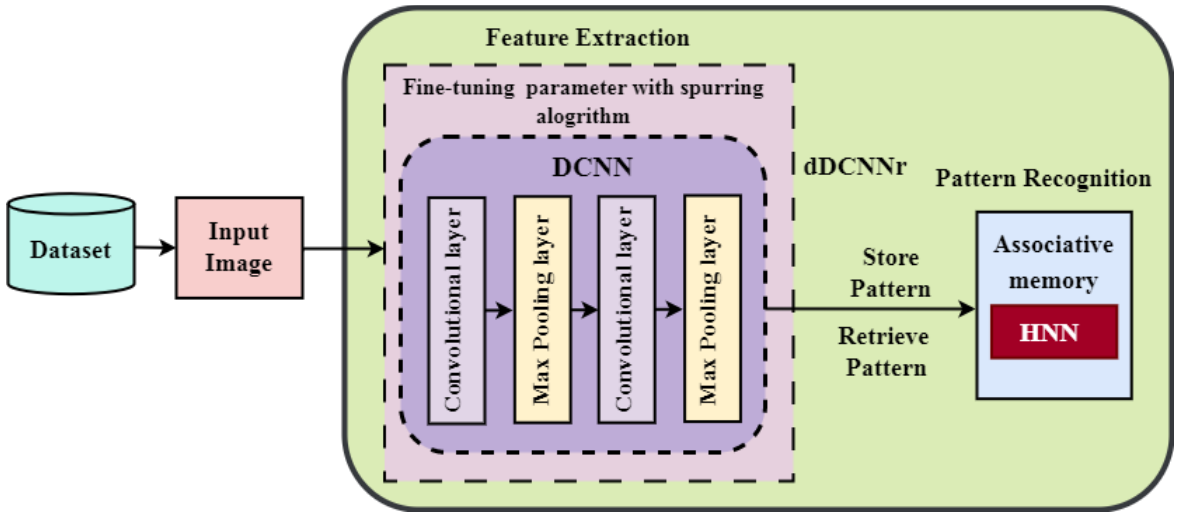

Figure. 2 Integration of HNN and dDCNNr for pattern recognition

the minimum distance between encoded patterns, maximizing data ability and memory efficiency for easy pattern recognition. The Fig. 2 defines the integration of HNN and dDCNNr model.

3.5 Deep face recognition scheme in cloud

In order to prove the authentication method can be employed to validate a user's face without jeopardizing its generalizability, a novel password verification technique based on HNN has been devised. This authentication scheme is employed for any authentication of computing resources including authorization to connect multiple servers or role-based security control. The three main steps of the authentication process are password change, authentication, and registration.

3.5.1. Algorithm 2:registration process

The step-wise process of the registration process is illustrated below.

1. The user chooses the user identity UID_R and password UFI_R (features extracted from face images by DCNN)
2. Encrypt the password UFI_R
3. The encrypted UFI_R and UID_R are converted to a $q - bit$ binary value (bn_R).
4. RSC paradigm is used to convert bn_R to $N -$ bit binary number U_R by satisfying $N \geq 2q$.
5. U_R is employed to train the $N -$node HNN by changing the weights in respect to Eq. (8).
6. The HNN is updated if U_R as input. The outcome of HNN whether accept U_R, are ask another password.
7. The preceding steps are repeated until the NN verify all UID_R and its corresponding password UFI_R.

3.5.2. Algorithm 3:authentication process

The following procedures are part of the authentication process

1. The user delivers the user ID (UID_K) and password UFI_K(Feature extracted by DCNN from face image).
2. Encrypted password UFI_K
3. The UFI_K and UID_K are transformed to a $q -$ bit binary integer (bn_K)
4. The RSC model employed in the registration will be applied here to change bn_K to $N -$ bit binary number (U_K).
5. The HNN's weight is updated using the Eq. (8).
6. U_K is given as input. If the output of HNN is similar to U_K, the access of UID_K is granted, otherwise, authentication is denied.

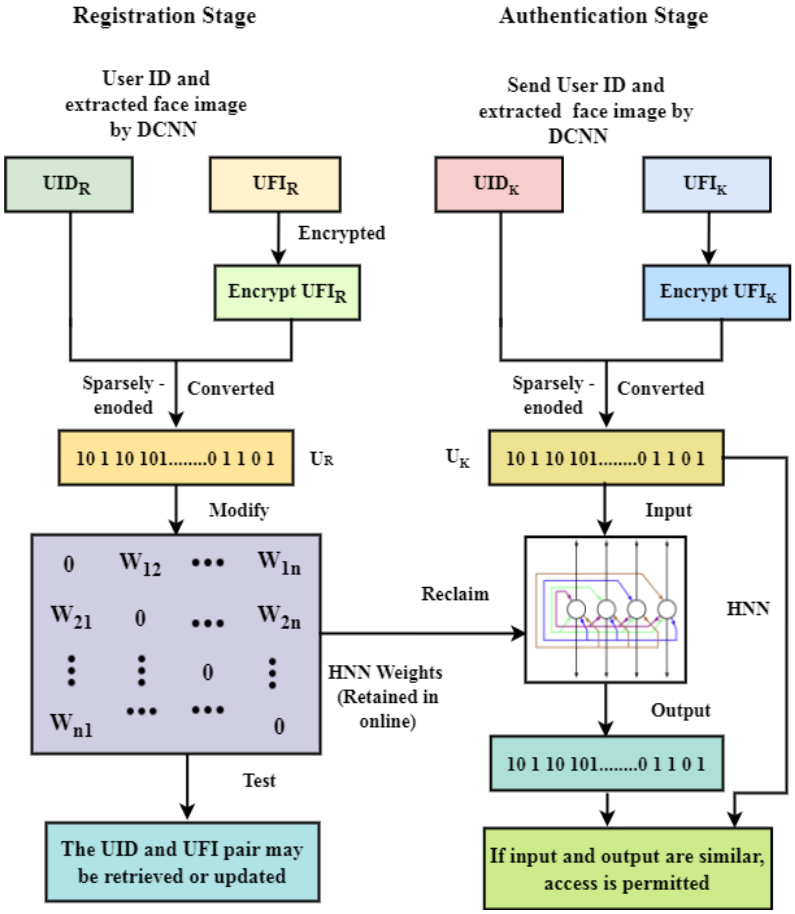

Figure. 3 Registration and Authentication Task in dCAuth

3.5.3. Decrypting the new face image (Passwords)

By combining the aforementioned steps to remove the previously saved face image as password makes up the face image change process. Access to the system is granted after the authentication procedure is carried out whenever a user needs or plans to change their password with a new facial images. Once the authorization is given, the method starts the registration procedure to store the new password as a face image. As shown in Eq. (5), when the new password has been registered, the old ones is removed by eliminating the HNN weights that are based on U_K. After entering the new password, the system will provide access. Furthermore, The HNN recall is error-proof and self-reliant, as its registration method ensures a perfect match between any legal combination of UID and UIF as a password. Fig. 3 depicts the registration and authentication procedures of dCAuth model. This proposed dCAuth provides robust and reliable solution for cloud authentication and train large cloud data with lesser complexity and better computing resources.

4. Simulation setup and results evaluation

4.1 Dataset details

For the experimental purposes, GTF [20] and LFW [21] and BS Dataset is used. GTF Database: GTF Database constitutes of 750 face images of 50 individual, each individual have 15 pictures including various facial expression with and without inclined face.

LFW Database: This dataset contains 13,233 images of 5,749 people, centered by the Viola Jones face detector, collected from the web. 1,680 individuals have two or more distinct photos, with four different sets of LFW images and three types of aligned images.

BS Database: It is created by digital signature of users. The digitizer is used to record signatures and capture features of signature like pen-pressure, time stamp and velocity. Totally, 1000 individuals provided their signature with each user providing 500 genuine signatures and 500 forged signatures.

4.2 Stimulation Setup and Performance Metrics:

In this study, the efficiency of both existing and proposed models is stimulated in Python 3.11.4, using GPU-enabled TensorFlow on a 64-bit Windows 8 machine with Intel Xeon 3.60 GHz CPU and 16 GB of RAM. Table 2 presents the simulation environment and parameter values utilized for both existing and proposed model to evaluate their performance.

Table 2. Stimulation Environment and Parameters for Existing and Proposed Model

Parameters	Values
Cloud configuration	
No. of. users	100
No. of. Servers	5
Key size	256
Encryption Memory	70 MB
Decryption Memory	65 MB
HNN	
Neuron Combination	100
Total No. of patterns	6
Activation Function	Tangent
No. of. HNN nodes	95
Initial weight and bias terms	Between 0 and 1
DCNN [10], TDNN [12], FFBPNN [14], FaceNet [16], HNN	
No. of. Convolutional Layers	2 (5*5 kernels; 3*3 kernels)
No. of. Max-pooling layers	2 (2*2 kernels; 2*2 kernels)
Flatten	1
Fully connected	2 (2048 neurons; 200 neurons)
Stride & Padding	2 & 3
Batch Size	50
Activation function	ReLU
Learning Rate	0.001
Optimizer	Adam
Epochs	60
Loss function	Mean Square Error
TDNN [12]	
No. of. Trees	8
Tree Depth	2
Channel Dimension	168
FFBPNN [14]	
Hidden Layers	2
Maximum Iteration	10000
Network Weights	Adjusted between -0.05 to +0.05
Maximum Error	0.0001
FaceNet [16]	
L2 normalization	1*1*128
Inspection Depth	2

For experimental purpose each dataset is randomly divided into training data (50%) and testing data (50%). The comparative analysis is conducted between the proposed dCAuth and existing algorithms including LDLSA [10], FRCA [12], BAMCloud [14] and FFCAS [16]. The accuracy, precision, recall, F1-score, time cost, and memory cost are some of the performance criteria that are used to evaluate these models.

In the literature, LDLSA [10] and BAMCloud [14] utilized GTF and BS database respectively. Similarly, FRCA [12] and FFCAS [16] utilized LFW databases. But, this work evaluate LDLSA, BAMCloud FRCA, FFCAS and proposed dCAuth model for GTF, LFW and BS datasets by using the

parameters as per Table 2. DL models DCNN, TDNN, FFBPNN, FaceNet are used in LDLSA, FRCA, BAMCloud and FFCAS respectively. The proposed dAuth utilized both DCNN and HNN.

Accuracy: This computation is determined by dividing the total number of successful authentications by the number of attempts. The higher accuracy values indicates its reliability and effectiveness in identifying legitimate users and minimizing unauthorized access. The Eq. (9) defines the accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

The number of classes in which the system successfully verified the identities of valid individuals is known as True Positives (TP). The number of categories in which the system properly rejected authorized users is shown by True Negatives (TN). False Positives (FP) are the number of instances in which unauthorized users were granted access by the system due to an inaccurate authentication. The quantity of instances when the system wrongfully refused access to authorized users is known as False Negatives (FN).

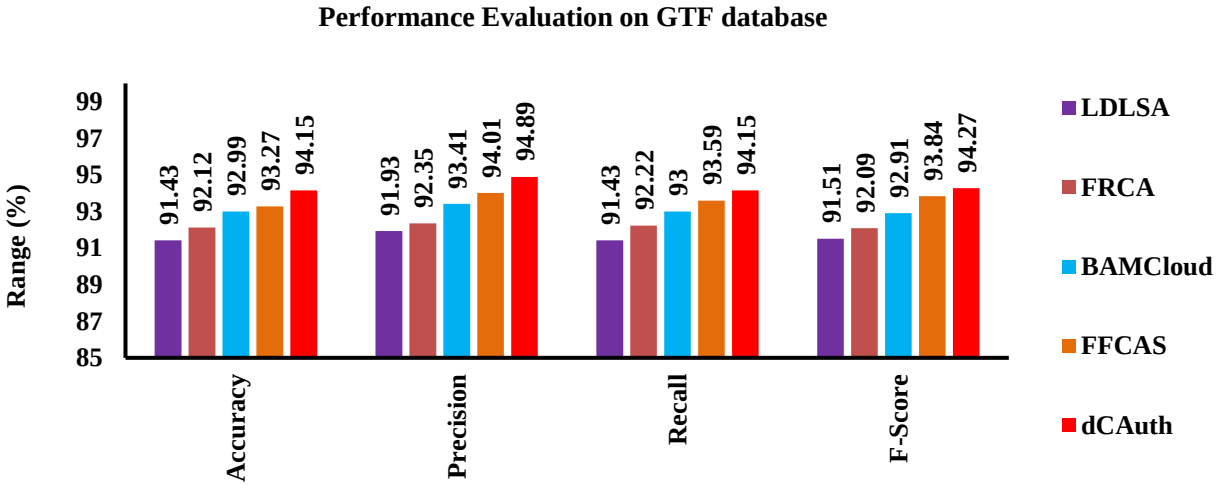

Figure. 4 Evaluation of proposed and existing cloud authentication models on GTF Database

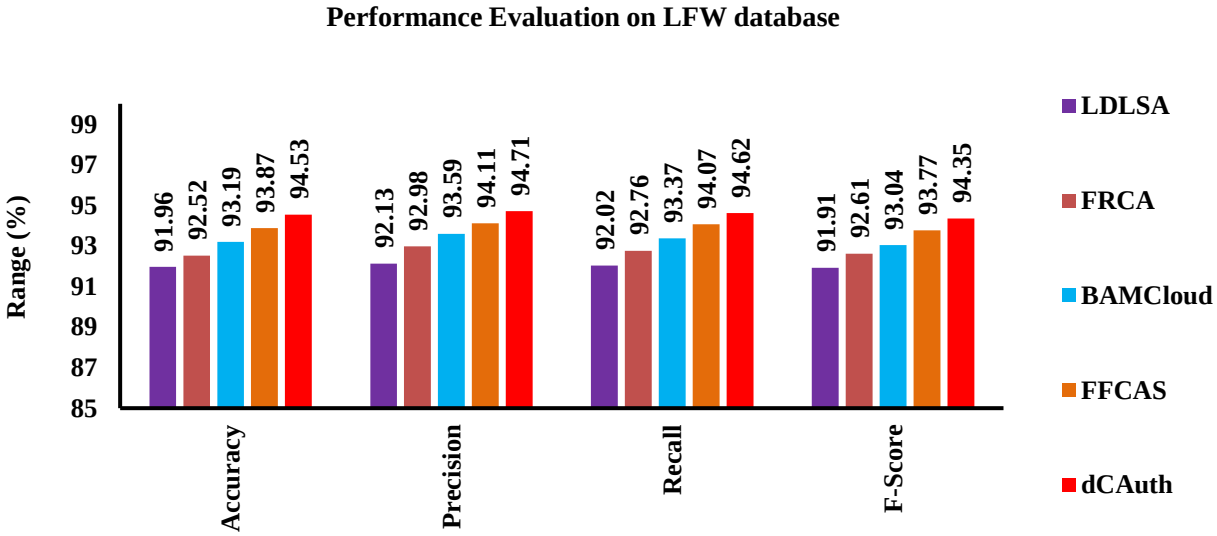

Figure. 5 Evaluation of proposed and existing cloud authentication models on LFW Database

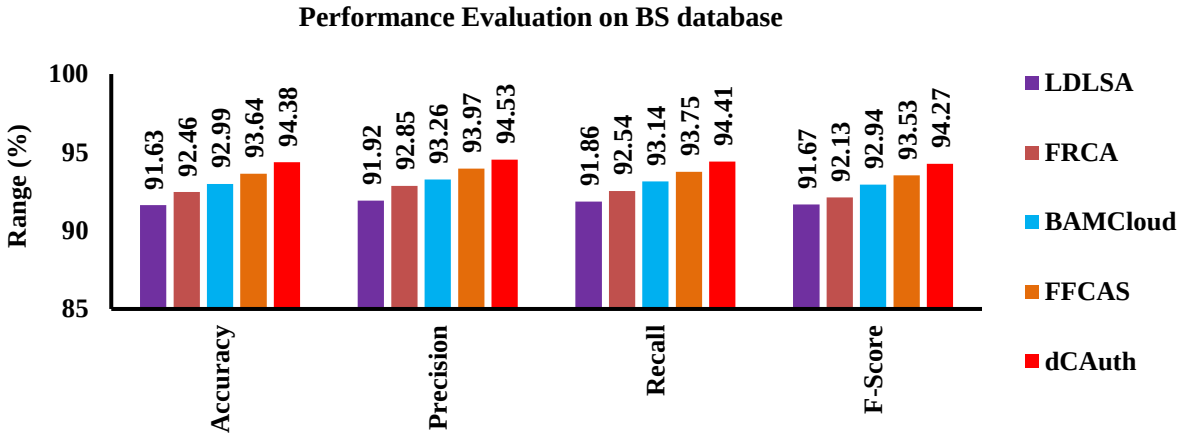

Figure. 6 Evaluation of proposed and existing cloud authentication models on BS Database

Table 3. Time and Memory Cost

	LDLSA	FRCA	BAMCloud	FFCAS	dCAuth
Time (ms)	85ms	68ms	53ms	31ms	22ms
Memory (mb)	154 mb	129 mb	113 mb	101 mb	95 mb

Precision: It calculates the ratio of positive authentications in Eq. (10) that were actually correct by the model. It focuses on minimizing the FP.

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

Recall: It computes the ratio of actual positive legitimate users in Eq. (11) that were correctly identified by the model. It focuses on minimizing false negatives.

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

F1-score: It calculates the partial average of precision and recall which is beneficial in cases of uneven distribution between legitimate and unauthorized users.. It is defined in In Eq. (12),

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (12)$$

Fig. 4 illustrates the efficiency of existing and proposed models applied on the GTF databases estimated using accuracy, precision, recall and F1-score to provide efficient cloud authentication system. This scrutiny indicates that the proposed dCAuth achieves 94.15%, 94.89%, 94.15% and 94.27% for accuracy, precision, recall and F1-score on GTF database which is higher than other authentication models.

Fig. 5 depicts the cloud authentication models evaluated on LFW database using different metrics. It is observed that the dCAuth obtains the accuracy, precision, recall and F1-score of 94.53%, 94.71%, 94.62% and 94.35% on LFW dataset relatively higher than the other existing models respectively.

Fig. 6 provides the analysis of existing and proposed models validated on BS dataset using various metrics. This analysis shows that dCAuth achieves 94.38% (accuracy), 94.53% (precision), 94.41% (recall) and 94.27% (F1-score) on BS dataset, comparatively greater than the other Cloud authentication models on BS dataset.

Hence, it is observed that the dCAuth performs better for all dataset which higher than other models. The usage of HNN in dCAuth and pattern matching by recall concepts ensuring higher accuracy under privacy preservation based cloud authentication.

Table 3 presents the time and memory costs of proposed and existing methods for cloud authentication task. It is observed that the time cost (ms) and memory cost (mb) of dCAuth model is 22ms and 95 mb which is lower than the other existing models. The above experimental observation shows that the suggested dCAuth method outperforms other existing cloud authentication methods due to its ability to reduce complexity and resource demands, resulting in lower time and memory costs and improved scalability.

5. Conclusion

In this article, dCAuth is developed for efficient

cloud authentication with lesser complexity. The model uses a spurring algorithm for dDCNNr to handle numerous cloud authentication parameters. The complexity of DCNN is reduced by replacing FC layers by HNN which focus on pattern recognition. The model securely stores NN weights on the cloud, eliminating the need for a verification table. It matches input patterns indicating known or legal patterns, minimizing processing resources. The proposed authentication method employs DCNN and HNN to efficiently and reliably retrieve a user's ID and password. The model effectively manages large training data and parameters, ensuring privacy preservation and reducing registration and password alteration time. The proposed method achieves accuracy of 94.15%, 94.53% and 94.38% for GTF, LFW and BS databases, respectively. The result of dCAuth is higher than LDLSA, FRCA, BAMCloud, and FFCAS for all datasets. The implementation of dDCNNr in dCAuth increase the accuracy while reducing the running time and memory.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, methodology, software, validation, Gowri; formal analysis, investigation, Shivakumar; resources, data curation, writing—original draft preparation, Gowri; writing—review and editing, Gowri; visualization, supervision, Shivakumar.

References

[1] M. R. Prasad, R. L. Naik, and V. Bapuji, "Cloud computing: Research issues and implications", *International Journal of Cloud Computing and Services Science,* Vol. 2, No. 2, pp. 134-140, 2013.

[2] A. Poniszewska-Maranda, R. Matusiak, N. Kryvinska, and A. U. H. Yasar, "A real-time service system in the cloud", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 11, No. 1. pp. 961-977, 2020.

[3] S. A. Bello, L. O. Oyedele, O. O. Akinade, M. Bilal, J. M. D. Delgado, L. A. Akanbi, ... and H. A. Owolabi, "Cloud computing in construction industry: use cases, benefits and challenges", *Automation in Construction*, Vol. 122, No. 1, pp. 1-18, 2021.

[4] T. Halabi, and M. Bellaiche, "Towards quantification and evaluation of security of Cloud Service Providers", *Journal of Information Security and Applications*, Vol. 33, No. C, pp. 55-65, 2017.

[5] M. P. A. Saviour and D. Samiappan, "IPFS based storage Authentication and access control model with optimization enabled deep learning for intrusion detection", *Advances in Engineering Software*, Vol. 176, p. 103369, 2023.

[6] R. V. Patel, D. Bhoi, and C. S. Pawar, "Security hazards attacks and its prevention techniques in cloud computing: A detail review", *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, Vol. 6. No. 4, pp. 48 – 58, 2020.

[7] F. Jauro, H. Chiroma, A. Y. Gital, M. Almutairi, M. A. Shafi'i, and J. H. Abawajy, "Deep learning architectures in emerging cloud computing architectures: Recent development, challenges and next research trend", *Applied Soft Computing*, Vol. 96, pp. 106582. 2020.

[8] K. H. Al-Saedi, "Enhancing cloud security through the integration of deep learning and data mining techniques: A comprehensive review", *Periodicals of Engineering and Natural Sciences*, Vol. 11, No. 3, pp. 176-192, 2023.

[9] Z. Ghodsi, T. Gu, and S. Garg, "Safetynets: Verifiable execution of deep neural networks on an untrusted cloud", *Advances in Neural Information Processing Systems*, Vol. 30, No.1, pp. 4673-4682, 2017.

[10] A. Zeroual, M. Amroune, M. Derdour, and A. Bentahar, "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.

[11] P. M. S. Sánchez, L. F. Maimó, A. H. Celdrán and G. M. Pérez, "AuthCODE: A privacy-preserving and multi-device continuous authentication architecture based on machine and deep learning", *Computers & Security*, Vol. 103, No. 1, pp. 1-14, 2021.

[12] D. Chauhan, A. Kumar, P. Bedi, V. A. Athavale, D. Veeraiah and B. R. Pratap, "An effective face recognition D., & system based on Cloud based IoT with a deep learning model", *Microprocessors and Microsystems*, Vol. 81, No. 3, pp. 1-8, 2021.

[13] K. D. Priya and L. Sumalatha, "Secure Framework for Cloud based E-Education using Deep Neural Networks", In: *Proc. of 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 406-411, 2021.

[14] K. A. Shakil, F. J. Zareen, M. Alam, and S. Jabin, "BAMCloud: a cloud based Mobile biometric authentication framework", *Multimedia Tools and Applications*, Vol. 82, No. 25, pp. 39571–39600, 2022.

[15] A. K. Singh and D. Saxena, "A cryptography and machine learning based authentication for secure data-sharing in federated cloud services environment", *Journal of Applied Security Research,* Vol. 17, No. 3, pp. 385-412, 2022.

[16] D. Sun, H. Huang, D. Zheng, H. Hu, C. Bi and R. Wang, "Face security authentication system based on deep learning and homomorphic encryption", *Security and Communication Networks,* Vol. 2022, No. 1, pp. 1-9, 2022.

[17] A. M. Mostafa, M. Ezz, M. K. Elbashir, M. Alruily, E. Hamouda, M. Alsarhani and W. Said, "Strengthening Cloud Security: An Innovative Multi-Factor Multi-Layer Authentication Framework for Cloud User Authentication", *Applied Sciences,* Vol. 13, No. 19, pp. 1-24, 2023.

[18] A. A. Mohd, S. Kummarikunta, S. K. Thumboor Naga, V. R. Buthukuri, P. Chintamaneni, and R. Vatambeti, "Design of Mutual Authentication Method for Deep Learning Based Hybrid Cryptography to Secure data in Cloud Computing", *International Journal of Safety & Security Engineering,* Vol. 13, No. 5, pp. 1-15, 2023.

[19] P. Shrivastava and U. P. Singh, "Error detection and correction using Reed Solomon codes", *International Journal of Advanced Research in Computer Science and Software Engineering,* Vol. 3, No. 8, pp. 965-969, 2013.

[20] https://academictorrents.com/details/0848b2c9b40e49041eff85ac4a2da71ae13a3e4f

[21] https://www.kaggle.com/datasets/jessicali9530/lfw-dataset.