AN EFFICIENT MODEL FOR USER IDENTIFICATION AND DATA SECURITY USING He-RNN AND SKD-ECC WITH USER AUTHORIZATION

Badari Narayan V S
Research scholar, Computer Science and Engineering, SSIT, Sri Siddharth Academy of Higher Education, Tumakuru

Dr. Siddappa P
Professor and Head, Dept of CSE, SSIT, Sri Siddharth Academy of Higher Education, Tumakuru

ABSTRACT

Cloud Computing (CC) is an Internet-centered, advanced technology that is rapidly growing. It offers numerous options for resource sharing, storage, and several features. Security concerns have emerged as one of the most significant difficulties faced by the cloud environment in recent years. Using He-Recurrent Neural Network (He-RNN) and Secret Key Double-Elliptic Curve Cryptography (SKD-ECC) with user authorization, an effective model for user identification and data security was proposed in this paper to address such problems. The training phase and the secure data transmission phase were the two vital phases in the proposed work. The following steps - data owner registration, IP spoofing, User profile database, and user identification phase occur in the training phase. The He-RNN network was trained centered on the registration details. Using Double Public Key- Digital Signature Algorithm (DOUPK-DSA), the signatures are engendered during the secure data transmission phase. Using SKD-ECC, the data encryption is executed along with spoofing the data owner's IP address. The data's authorization is verified with the altered data identification phase before the data is transmitted to the user. Moreover, the signature matching process verifies the data user's authentication. The data is sent to the data user if the signature matches; otherwise, the data transfer is rejected. And the proposed method's outcomes are analogized with prevailing algorithms. The experimental outcomes exhibited that the proposed framework outperformed existing methodologies regarding accuracy, sensitivity, and specificity, and executed the privacy-preserving process more accurately.

Keywords: *Recurrent neural network, He initialization, Elliptic curve cryptography, Digital signature algorithm, Privacy-preserving in the cloud.*

DOI Number: 10.14704/nq.2022.20.9.NQ44599 **Neuro Quantology 2022; 20(9):5180-5191**

1. INTRODUCTION

Digital data security has grown to be a very severe issue owing to the elevated utilization along with sharing of digital information in today's networking along with the media technology world [1]. Huge information is saved together as of distinct sources in the CC technology [2]. With the invention of this technology, stored data can be accessed via the web browser using the internet. From the Cloud Server (CS), the services or computing resources are contributed by the Cloud Service Provider (CSP) to the users [3, 4]. The document's owner is the individual who uploads the document [5]. Only the Authorized Data users can access the document as of the CS. As a result, resources are provided at a lower cost by CC [6]. Additional services offered by the cloud environment incorporate Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS) [7].

Despite the advantages, some problems and hurdles were encountered by CC that prevents the cloud's wide utilization [8]. The most vital issues, for instance, were data security and privacy protection. The majority of CC's security concerns are internal and external threats [9]. From the lack of isolation within cloud hosts, internal threats originated whereas external ones originate from all directions in the cloud environment [10]. Typically, the data user's confidential or valuable information is comprised in the cloud. Therefore, protecting user data's privacy is crucial [11]. The technique of protecting the sensitivity of cloud-stored data is known as privacy preservation. The data's privacy was ensured by the use of a few security measures like key generation, encryption, and decryption [12]. For ensuring sensitive data privacy, various privacy preservation strategies are engendered in conventional works [13]. However, multiple user access, no maintenance issues, cost-effectiveness, and a lack of massive data storage were a few drawbacks [14].

There have been a lot of studies done recently on cloud storage privacy protection techniques to tackle these issues. However, too large a public key, enormous ciphertext expansion rate, and more time for cipher text calculation are the drawbacks exhibited by the prevailing methods [15]. As Homomorphic Encryption (HE) allows a third party to execute mathematical operations on encrypted texts without decrypting them, it is a widely used technique [16]. A Homomorphic operation betwixt cipher texts encrypted with different keys cannot be executed in the HE scheme; however, only data from users utilizing the identical key can be shared in CC. It is challenging to maintain data privacy among users if the identical key is utilized by them all [17, 18]. Consequently, an effective model for modified data identification and data security utilizing He-RNN and SKD-ECC with user authorization is proposed here, to alleviate the aforementioned issue. Thus, the encryption process's performance is elevated by the proposed scheme along with delivering higher security with less complexity

The remaining parts are sorted as: the associated works are surveyed in Section 2. Section 3 explains the proposed efficient model for USER identification and data security utilizing He-RNN and SKD-ECC with user authorization. Section 4 depicts the results and discussion grounded on performance metrics. Finally, the paper is concluded in section 5 with future work.

2. LITERATURE SURVEY

Sarala V *et al.* [19] presented a Deep Learning Network-centered Frequent Pattern Mining scheme (DLFPM) merged with Single Sign-On linked with Paillier Encryption technique (DLFPMSSO-PE) grounded secure data storage model in the cloud. Association rules achieved intermediary data generation for appropriate data security. Subsequently, the features considered here were the retrieved patterns. A long short-term memory DL network processes the features along with the sensitive information identified by the network. By the SSO user authentication process, the information was then saved in the cloud. Next, for the data encryption, the homomorphic-centered private and public key was engendered. Minimal computation time, cost, along with maximum data security accuracy was attained by the developed scheme than the prevailing research.

Sunitha Pachala *et al.* [20] designed a hybrid approach for augmenting security along with cloud data privacy with the multi-cloud hosting environment. '3' modules were comprised in this hybrid method. Byzantine protocol was the first module that tolerant security breaches to CS failures. The second module named DepSky architecture elevated the reliability along with preserving data secrecy in the cloud with encoding as well as decoding techniques. Shamir's secret sharing process elevated reliability along with the privacy of data storage. Regarding memory utilization, Encryption/Decryption Time (ET&DT), along with total authentication time, the experiential outcomes suggested that the hybrid approach's performance was elevated when analogized with prevailing works.

G. Prabu Kanna *et al.* [21] developed a new privacy conservation mechanism named the Fully Homomorphic– ECC (FH-ECC) algorithm. The ECC algorithm helped in encrypting the original data in this method. Before saving the encrypted data on the cloud, the FH operations were applied to it. The restricted access to the data was enabled by CSP verifying the user's control policy while a data request was provided by the user to the cloud. The encrypted data was fed to the user if the access policy was validated. The ciphertext was extracted from that. Then, the original text was engendered by applying the

ECC decryption along with FH operations. The FHE technique's effectiveness was demonstrated by the outcomes.

Danish Ahamad *et al.* [22] generated the privacy conservation scheme with the artificial intelligent technique's enhancement in the cloud. Sanitization and restoration were the '2'vital phases in the scheme. Two better algorithms like Shark Smell Optimization (SSO) and Jaya Algorithm (JA) were merged into a hybrid meta-heuristic algorithm named J-SSO that executed a sanitization process that was grounded on the optimal key generation. A multi-objective function was derived that encompassed the metrics like degree of modification, hiding ratio, along with information preservation ratio, for executing optimal key generation. Consequently, a better efficacy over the prevailing models was proved by the model's performance analysis in a cloud environment.

Suyel Namasudra [23] implemented an efficient along with secure access control model by utilizing Attribute-Centered Encryption (ABE), Distributed Hash Table (DHT) network, along with Identity-Centered Timed-Release Encryption (IDTRE) for resource and knowledge sharing in the CC environment. By utilizing the user's attributes, the data were encrypted initially along with getting partitioned into the encapsulated ciphertext and extracted ciphertext. The decryption key was encrypted by utilizing IDTRE algorithm. The key's ciphertext was combined with the retrieved ciphertext by the scheme for the ciphertext shares creation. Finally, the DHT network was disseminated with the ciphertext shares along with saving the encapsulated ciphertext on the CS. The experiential analysis exhibited the scheme's proficiency over the prevailing approaches in a CS.

As security problems turn out to be more complicated in a cloud environment, the quick development of numerous applications cannot be equaled by the privacy protection techniques generated in the prevailing approaches. Since the security perimeter-centered mechanism is focused mainly on the defense against external onslaughts, it may not be appropriate for cloud storage. But, a large portion is potentially taken by internal threats in security accidents. Thus, more attention is requisite to secure sharing, access, along with data authorization among user groups. Therefore, an efficient model utilizing He-RNN and SKD-ECC with user authorization is proposed here for USER identification and data security. Designing an efficient model for USER Identification, Authorization, and Enhancing Data Security using He-RNN, DOUPK-DSA, and SKD-ECC is the ultimate goal of this paper. The following enlists the research work's objectives:

- To avert unauthorized access to data, a strong authorization technique called DOUPK-DSA is utilized.
- To assure secure data transmission to the CS, the novel SKD-ECC algorithm is produced.
- To uphold the confidentiality of the system IP address, the **2SLC** binary conversion technique is employed.
- To authenticate the proposed system design by utilizing several parametric comparisons.

3. PROPOSED METHODOLOGY

Owing to the several benefits like efficient utilization of computing resources, CC offers an increasing interest. However, the vital obstacle facing the new technology's wide utilization is privacy along with security issues. So, an efficient model for user identification and data security utilizing He-RNN and SKD-ECC with user authorization was proposed here. Centered on the data owner's personal details, timestamps, and the spoofed IP address, the training process is executed initially. Subsequently, the secure data transmission process commences. The signatures are engendered by DOUPK-DSA and the encryption is executed by SKD-ECC to ensure security. By utilizing the user identification phase along with the signature matching process, the data owner, as well as the data user's authorization, is validated before the data is sent to the user. The encrypted data is transmitted if both the data owner as well as the data user, are authorized. The data is accessed by the user by decrypting it. Figure 1 exhibits the proposed methodology's block diagram.

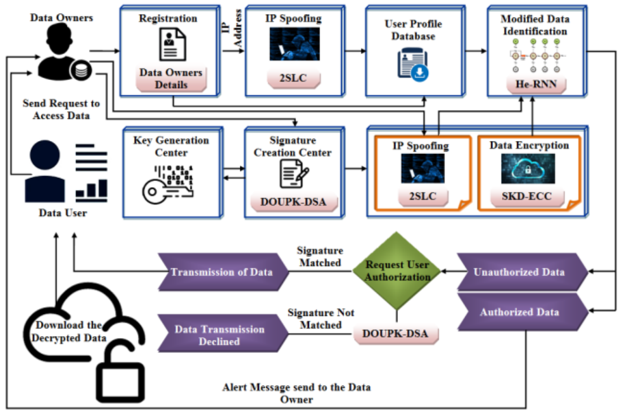


Figure 1: *Block diagram for the proposed methodology*

I. Training phase

On the basis of data owner information, spoofed IP addresses, uploading data files name, and the time at which the data files are uploaded, the training process is executed initially. The following steps are included in the training process.

3.1. Registration

By entering their details, the data owner registered to the CS gets access to uploading the data files.

3.2. IP Spoofing

The data owner's IP address is fetched during the registration time along with altering into the spoofed format by utilizing the 2SLC technique. The technique by which the IP address is altered into binary format (in the 0's and 1's form) is termed 2SLC. Two times left-shifted was gained by binary number along with identifying the shifted binary number's 2's complement.

3.3. User profile database

In the user profile database, the data owner's information, as well as the spoofed IP address, is stored for further use.

3.4. USER Identification

The user Identification (MDI) phase is the training phase, where the data as of the User Profile Database along with the timestamp are provided. The He-RNN Algorithm performs the training in this work. Normally, the weights are initialized by RNN randomly. The training process's efficiency is degraded by this random updation of weights. So, He-initialization is utilized by research methodology for weight updation. The training process's accuracy is elevated drastically by this alteration. The gradient issues along with the slow convergence issues are mitigated efficiently by the He-initialization.

The output as of the prior step is given as the next step's input in the RNN. The hidden state is the RNN's critical feature for remembering little information about a sequence. Each Hidden Layer (HL) has its own set of weights and biases in Conventional Neural Networks (CNN). For example, the weights and biases are (W_1, B_1) in HL 1, the weights and biases are (W_2, B_2) for the second HL and (W_3, B_3) for the third HL. This indicates none of these layers were dependent on each other. So, the prior outputs were not memorized by them. By providing similar weights and biases to every layer, the independent activations are converted into dependent activations by the RNN. Thus, the increasing parameter's complexity is diminished. Also, each prior output is memorized by providing each output as the next HL input. Figure 2 illustrates the RNN's general architecture.

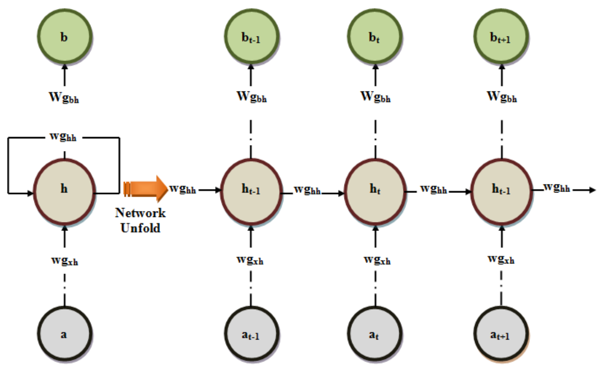


Figure 2: *General structure of the RNN*

The He-RNN steps are given as follows.

Step 1:The He-RNN is inputted with the data owner's information, spoofed IP address, data files names, and timestamps. By iterating the following sequence from $t = 1 to T$, the He-RNN is executed on the input data $a_{IN} = \{a_1, a_2, a_3, ..., a_t\}$ comprising of a hidden vector sequence $h_{Lr} = \{h_1, h_2, h_3, ..., h_t\}$ together with output vector sequence $b_{OT} = \{b_1, b_2, b_3, ..., b_t\}$. The HL is calculated as follows.

$$h_{Lr} = \Im_{act}[wg_{ah}a_t + wg_{hh}h_{t-1} + Ba] \tag{1}$$

Here, the He-initialization scheme is utilized for the weight updation process rather than the random weight initialization. The He-initialization expression is defined by,

$$wg = \frac{2}{N} \tag{2}$$

Where, the weight matrices are indicated as wg (e.g. the input-hidden weight value is modeled as wg_{ah} and the HL weight value is signified as wg_{hh}), the number of incoming neurons in the He-RNN is signified as N, and the bias vectors are denoted by Ba and the Relu activation function is illustrated as $\Im_{act}$.

Step 2:The HL's output is further fed into the output layer. The sigmoid activation function (σ_S) activates the output layer, which is determined by,

$$b_{OT} = \sigma_S[wg_{ab}h_t + Ba] \tag{3}$$

$$a = wg_{ab}h_t + Ba \tag{4}$$

Step 3: By using equation (5), the sigmoid activation function is computed as

$$\sigma_S(a) = \frac{1}{1+e^{-a}} \tag{5}$$

Step 4:Later, by calculating the diversity betwixt the actual value a_{ac} and the predicted value $\hat{a}_{pr}$, the loss value is computed. The error value can be computed as.

$$Error = (a_{ac} - \hat{a}_{pr})^2 \tag{6}$$

The data is more effectively trained by the model if the model's loss value or error value is zero $(Error = 0)$. The back-propagation occurs by updating the weight values if the loss value $Error \neq 0$. Thus, the data is efficiently trained by the He-RNN whose pseudo-code is illustrated in figure 3.

Input: Data owner's information, spoofed IP address, data files names, and timestamps
Output: Accurate training of data

Begin
  Initialize weight $(wg_{i,j})$, bias $(Ba_{i,j})$, features, $\max_{itr}$
    For $t = 1 to T$
      Generate the hidden layer by using,
        $h_{Lr} = \Im_{act}[wg_{ah}a_t + wg_{hh}h_{t-1} + Ba]$
      Generate the output layer by using,
        $b_{OT} = \sigma_S[wg_{ab}h_t + Ba]$
    End for
  If $(a_{ac} = \hat{a}_{pr})$
    Accurately trains the data
  Else
    Perform back propagation by updating weights
  End if
End

Figure 3: Pseudo-code for He-RNN

II. Secure data transmission phase

A request is sent to the data owner if the data from the data owner is needed to be accessed by the data user. Data transmission is initiated after the request is accepted by the data owner. If not, the transmission is declined. The following steps take place to ensure secure data transmission.

3.5. Key generation centre

The keys are engendered to ensure secure data transmission. The data is encrypted and decrypted by utilizing these keys. For both the data owner along with data user, the key generation centre is accountable for engendering the public key as well as the private key. The public key is a sharable one while the private key is non-sharable.

3.6. Signature creation centre

Assuring whether the message was sent by an authorized data owner, is made by electronic signatures named Digital signature. It is unique for every user and hence cannot be forged. The message's authenticity along with the integrity is guaranteed by the signature. By utilizing a DOUPK-DSA, the signature is engendered in this work. By merging the data owner's public key along with the data user, the signatures are engendered in the DOUPK-DSA. Figure 4 exhibits the DOUPK-DSA algorithm's architecture.

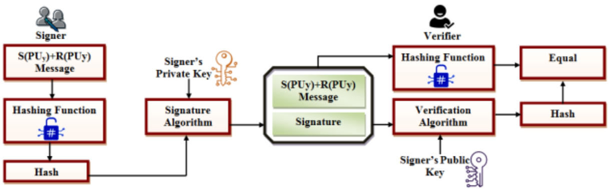


Figure 4: Architecture of the DOUPK-DSA

Normally, key generation, signature generation, and signature verification are the '3' phases in the DSA algorithm. A public key and private key generation are the DSA algorithm's initial part and it is described as,

(a) Key generation

- x and y are the two prime numbers chosen initially. Here, the prime divisor is notated as y and x is the prime modulus, such that $x-1 \bmod y = 0$
- An integer q is chosen, such that $1<q<x$, that satisfies the two conditions, $q^{**}y \bmod x = 1$ and $q = h^{**}((x-1)/y) \bmod x$. y is also called q's multiplicative order modulo x.
- An integer PR_X is randomly chosen by each users as the private key such that from $0 < PR_X < y$.
- The public key PU_Y is engendered corresponding to PR_X, by using,

$$PU_Y = qPR_X \bmod x \qquad (7)$$

- Thus, the private key package is indicated as $\{x, y, q, PR_X\}$, and the public key package is illustrated as $\{x, y, q, PU_Y\}$.

Signature generation and signature verification is the DSA algorithm's second part which can be described as.

(b) Signature Generation

- The message (Msg) is formed as the combination of the sender's public key $S(PU_Y)$ as well as the receiver's public key $R(PU_Y)$. To get the hash digest (H), the message is passed via the hash function $(\#h)$.
- Random number k is generated afterwards, such that $0<k<y$.
- To engender the signature (L,S) for the message (Msg), the signer is accountable.
- The signature's first part L is computed by using,

$$(q^{**}k \bmod x) \bmod y \qquad (8)$$

- The signature's second part S is computed by using,

$$S = I(H + L * PR_X) \bmod y \qquad (9)$$

Where, the modular multiplicative inverse of k modulo y is called as I .

The user with the public key PU_Y and private key PR_X, signed the signature of the message M that is represented by the pair (L,S).

(c) Signature verification

- By utilizing the following equation, the signature pair (L,S) is verified

$$V = (((q^{**}u_1) * (PU_Y{}^{**}u_2)) \bmod x) \bmod y \qquad (10)$$

Where, $u_1 = H * w \bmod y$, and $u_2 = L * w \bmod y$, the modular multiplicative inverse of S modulo y is notated as w. The digital signature is verified if $V == L$.

3.7. Data encryption

The data encryption process commences after the signatures are generated. The requested data are altered into another format named cipher text. A possibility for the external attack and unauthorized tampering of data is may occur during the data transmission. The encryption process is executed to tackle it along with transferring the encrypted data to the user. By utilizing the 2SLC technique, the data owner's IP address is also spoofed during the data encryption. The original data can be read by the user by encrypting it. Using SKD-ECC, the encryption is executed here.

The method that encodes the data files along with permitting the authorized users to decode such data is named ECC, which is grounded on the mathematics of the elliptic curve over finite fields. To encrypt and decrypt information, it utilized the location of points on the elliptic curve. To assure the strong authenticity of data transmission, double encryption is eventuated here. The SKD-ECC algorithm's algorithms are elaborated as,

The mathematical expression for the elliptic curve is defined by,

$$E^2 = Q^3 + pQ + s \tag{11}$$

Where, the integers are symbolized as p, s, the functions are defined by utilizing the standard variables E, Q.

Key Generation

The public keys as well as the private keys are engendered in the key generation process. With the receiver's public key, the message is encrypted by the sender along with the original message is decrypted by the receiver with its private key. The ciphertext private key (Pv_{key}) is randomly engendered by the ECC initially. After, the ciphertext public key (Pu_{key}) is computed along with the shared secret key (Sc_{key}) finally. The keys are generated by utilizing the following expressions,

$$Pv_{key} = rand\,(n) \tag{12}$$

$$Pu_{key} = Pv_{key} * v_d \tag{13}$$

$$Sc_{key} = S(Pu_{key})^{R(Pu_{key})} \tag{14}$$

Encryption

The point Pt on the curve C is comprised of the encrypted message along with randomly selecting R from $[1 to\, n-1]$. Along with the secret key, double encryption is executed. With the receiver's public key's aid, the original message M is encrypted initially which is depicted in equation (15). By utilizing the sender's private key, the data gets encrypted which is expressed in equation (16).

$$CT_1 = R * Pt \tag{15}$$

$$CT_2 = M + R * R(Pu_{key}) + Sc_{key} \tag{16}$$

$$CT_3 = CT_2 + R * S(Pv_{key}) + Sc_{key} \tag{17}$$

Therefore, the message's encrypted form is depicted as CT_1, CT_2 and CT_3.

The encrypted data packets are ready for transmission after the data encryption. The data packets are checked earlier with the User Identification phase. By checking with the spoofed IP address, this phase validates whether data packets are sent by the authorized data owner. The request for user authorization is executed only if the spoofed addresses are identical. Otherwise, the process gets declined. It sends an intimation message to the data owner.

3.8. Request user authorization

By utilizing the signature matching process, the request for user authorization is executed. The data owner's signature along with the data requested user is verified here. The data user is confirmed as an authorized one if the signatures are matched with each other and the data is transferred to them. The data can be utilized by the data users by encrypting it. The decryption processes mathematical relation is given by,

$$CT_2 = CT_3 - S(Pu_{key}) * CT_1 - Sc_{key} \tag{18}$$

$$M = CT_2 - R(Pv_{key}) * CT_1 - Sc_{key} \tag{19}$$

Here, the original information is represented as M.

The transmission gets discarded if the signatures are not matched. Thus the network's authenticity is ensured by the proposed framework along with preserving the cloud data against intruders and external attacks.

4. RESULTS AND DISCUSSION

Here, the detailed analysis of the proposed scheme's final outcome is explained. The performance analysis, as well as the comparative analysis, is performed to prove the work's effectiveness. The proposed methodology's implementation is executed by utilizing JAVA.

4.1. Performance Analysis of the proposed He-RNN

Regarding various performance metrics like accuracy, sensitivity, specificity, precision, recall, and F-Measure, the proposed He-RNN is validated with various prevailing models like Deep Neural Network (DNN), CNN, Deep Belief Network (DBN), and RNN. To state the model's effectiveness, the comparative analysis is also executed with the prevailing techniques.

Table 1: Performance analysis of proposed He-RNN based on accuracy, sensitivity, and specificity

Techniques	Performance metrics (%)		
	Accuracy	**Sensitivity**	**Specificity**
DNN	90.62	89.62	89.62
CNN	91.54	91.45	90.56
DBN	93.65	92.54	92.15
RNN	94.87	93.65	93.65
Proposed He-RNN	96.84	95.87	95.89

Regarding accuracy, sensitivity, and specificity, the proposed He-RNN's performance analysis with various existing techniques like DNN, CNN, DBN, and RNN are illustrated in Table 1. As of the analysis, the proposed technique achieves higher metrics rates, like 96.84% of accuracy, 95.87% of sensitivity, and 95.89% of specificity. But the prevailing schemes obtain the accuracy rate that overall ranges between 90.62%-94.87%, sensitivity rate those overall ranges between 89.62%-93.65%, and specificity rates that range between 89.62%-93.65%. Hence, superior performance metric rates were obtained by the He-RNN that preserves the network's security in a more efficient way.

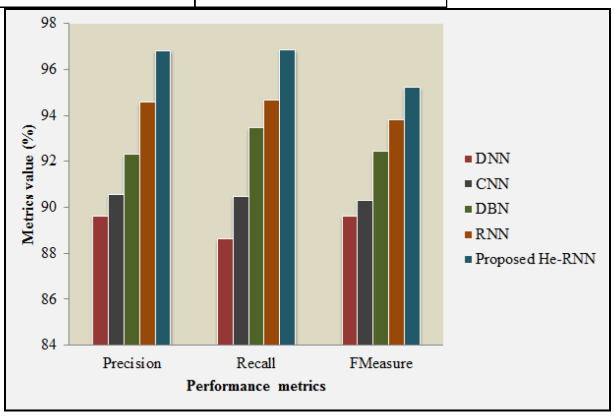


Figure 5: *Graphical representation of the proposed He-RNN based on Precision, Recall, and F-Measure*

Regarding precision, recall, and F-measure, the proposed He-RNN and the existing works' comparative analysis is exhibited in Figure 5. By the higher rate of precision, recall, and F-measure, the model's worthiness is identified. This comparative analysis states that a higher precision, recall, and F-Measure of 96.84%, 96.87%, and 95.23%respectively, were attained by the proposed He-RNN. Whereas, the metrics overall range lying between 88.65%-94.68% is attained by the prevailing techniques like DNN, CNN, DBN, and RNN, which is lower than the proposed He-RNN. Here, other state-of-art methods were outperformed by the proposed technique that delivers more prominent outcomes under various complex circumstances.

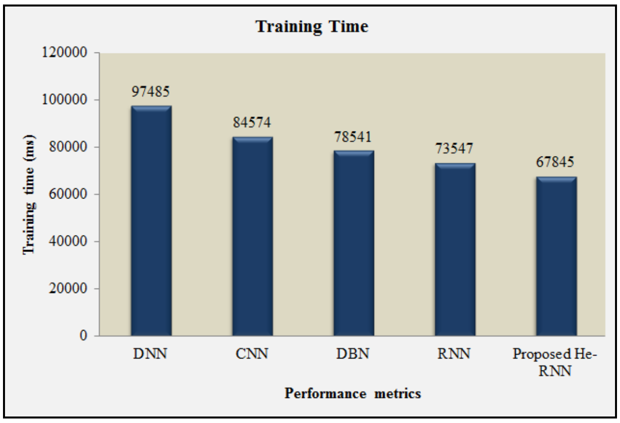
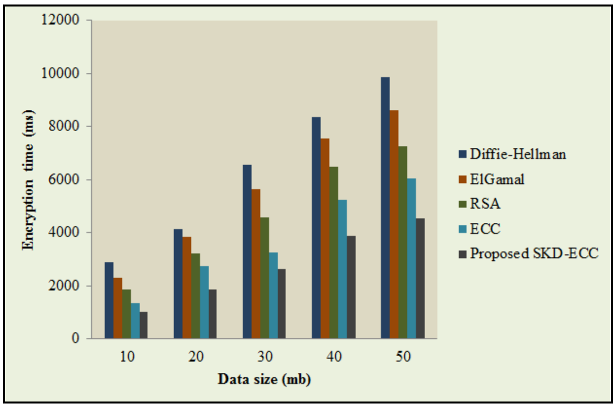


Figure 6: *Comparative analysis of the proposed He-RNN with respect to training time*

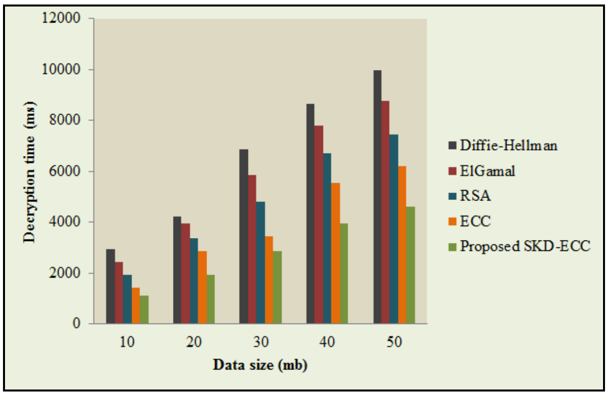
The proposed He-RNN's training time is analogized with the various prevailing ones in figure 6. It is evident from the comparative study that 67845 ms are taken by the classifier to complete the training process. The proposed technique takes a minimal amount of training time whereas the prevailing classifier like DNN, CNN, DBN, and RNN requires 97485 ms, 84574 ms, 78541 ms, and 73547 ms respectively to complete the training process. Owing to this, the total model's overall time can be elevated. But the total task was completed quickly as possible by the proposed model alleviating the work's time complexity

4.1. Performance Analysis of the proposed SKD-ECC

Regarding various performance metrics, like ET, DTmemory usage on encryption and decryption, and security level, the proposed SKD-ECC is evaluated and compared with various prevailing models like Diffie-Hellman, ElGamal, Rivest Shamir, Adleman (RSA), and ECC to state the model's worthiness.



(a)



(b)

Figure 7: *Comparative analysis of the proposed SKD-ECC in terms of encryption and decryption time*

The proposed SKD-ECC's (a) encryption time and (b) decryption time are analogized with the prevailing works like Diffie-Hellman, ElGamal, RSA, and ECC in figure 7. The data is encrypted by the proposed SKD-ECC algorithm in the least amount of time, according to the graphical analysis. The ETfor 10 mb, 20 mb, 30 mb, 40 mb, and 50 mb of data is 1012 ms, 1867 ms, 2612 ms, 3875 ms, and 4521 ms respectively. An average ETof 5085 ms for 10, 20, 30, 40, and 50 mb of data was obtained by the prevailing works. Furthermore, figure 7 (b) exhibits the proposed method's analysis regarding DT. For decrypting the data with the size of 10mb, 20 mb, 30 mb, 40 mb, and 50 mb, the proposed method requires 1098 ms, 1947 ms, 2847 ms, 3964 ms, and 4589 ms of DTrespectively. Whereas the prevailing works decrypt the 10mb, 20 mb, 30 mb, 40 mb, and 50 mb data with an average of 5254 ms. Thus, the proposed method's ET and DTare very less when analogized with prevailing works, thereby minimizing the proposed work's time complexity.

Table 2 (a): Performance analysis of the proposed SKD-ECC with respect to memory usage on encryption

	10	20	30	40	50
Diffie-Hellman	6978457	7324515	7847544	8247445	8745874
ElGamal	6471245	6754875	7454756	7948542	8365884
RSA	5877464	6255478	6832457	7458845	7854841
ECC	5474485	5875421	6477845	6877452	7315484
Proposed SKD-ECC	4897844	5484755	5887452	6425884	6874452

Table 2 (b): Performance analysis of the proposed SKD-ECC with respect to memory usage on decryption

	10	20	30	40	50
Diffie-Hellman	6845745	7488547	7754812	8254784	8655845
ElGamal	6355487	6874451	7584454	7845875	8265844
RSA	5744856	6474854	6744854	7358447	7784542
ECC	5365784	5784542	6355847	6748547	7415784
Proposed SKD-ECC	4754841	5684754	5748542	6322544	6878455

Regarding memory usage of encryption and decryption process, the proposed SKD-ECC and the prevailing works like Diffie-Hellman, ElGamal, RSA, and ECC's performance analysis are depicted in Table 2 (a) and (b). The amount of memory space consumed by the model to execute the encryption and decryption process is named memory usage. The model is recognized as efficient if it consumes a minimal amount of memory. The proposed SKD-ECC utilizes 5914077 kb of memory for encrypting 30 mb of data on average and utilizes 5877827 kb of memory for decrypting 30 mb of data on average. While, other prevailing works like Diffie-Hellman, ElGamal, RSA, and ECC use an average of 7121945 kb, and 7085195 kb memory for encryption and decryption processes, respectively. When analogized with the prevailing models, the proposed model's memory consumption rate is literally low. Hence, a noteworthy performance is exhibited by the proposed scheme in the privacy-preserving framework.

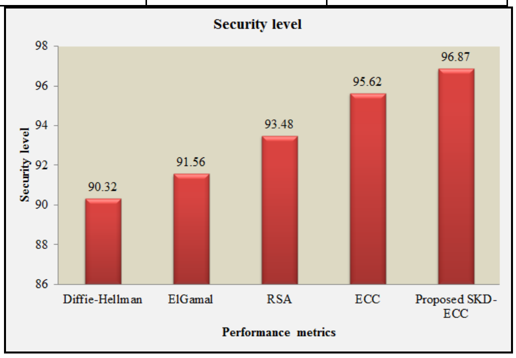


Figure 8: *Graphical analysis of the proposed SKD-ECC on the basis of security level*

Figure 8 depicts the proposed method and the existing method's security level. A high-security rate of 96.87% was attained by the proposed methodology as per the performance analysis. The security level at an average of 92.74% was obtained by the prevailing works which are low when analogized with the proposed work. The outcomes conclude that a high-security level encryption and decryption process was efficiently executed by the proposed work along with mitigating the external attack. Thus, the network is safeguarded by the proposed SKD-ECC against intruders.

5. CONCLUSION

Using He-RNN and SKD-ECC, an efficient model for user identification and data security with user authorization was proposed here. Registration with the CS, spoofed IP address, user identification, Key generation centre, signature creation centre, data encryption, and request user authorization are the various steps performed in the proposed methodology. Regarding some metrics, the experimental analysis is executed, which includes the proposed technique's performance analysis and the comparative analysis to validate the proposed algorithm's effectiveness. The outcomes exhibited that the network is safeguarded by the proposed model with a security level of 96.87%. Various uncertainties were handled by the proposed scheme along with rendering more promising outcomes. Overall, the prevailing state-of-art methods were outperformed by the proposed framework, which remains to be more reliable along with robust. The work will be extended in the future with a few advanced neural networks along with a major focus on preserving the privacy of complicated datasets.

REFERENCES

1. Ruixuan Li,Chenglin Shen, Heng He, Zhiyong Xu and Cheng-Zhong Xu, "A lightweight secure data sharing schemefor mobile cloud computing", IEEE Transactions on Cloud Computing,2014, http://doi.org,10.1109/TCC.2017.2649685.
2. Dongliang Lei, Ke Zhou, Hao Jin, Junping Liu and Ronglei Wei, "SFDS a security and flexible data sharing scheme in cloud environment", International Conference on Cloud Computing and Big Data, 12-14 Nov 2014,Wuhan, China, 2014.
3. Preeti Mishra, Vijay Varadharajan, Emmanuel S. Pilli and Uday Tupakula, "VMGuard A VMI-based security architecture forintrusion detection in cloud environment", IEEE Transactions on Cloud Computing, vol. 8, no. 3, pp. 957-971, 2018.
4. Simla MercySand Umarani Srikanth G, "An efficient data security system for group data sharing in cloud system environment", International Conference on Information Communication and Embedded Systems (ICICES2014), 27-28 Feb 2014, Chennai, India, 2014.
5. Manish H. Gourkhede and Deepti P. Theng, "Analysing security and privacy management forcloud computing environment", Fourth International Conference on Communication Systems and Network Technologies, 7-9 April 2014,Bhopal, India, 2014.
6. Lei Xu, JongHyuk Lee, Seung Hun Kim, Qingji Zheng, Shouhuai Xu, Taeweon Suh, Won Woo Ro and Weidong Shi, "Architectural protection of application privacyagainst software and physical attacks inuntrusted cloud environment", IEEE Transactions on Cloud Computing, vol. 6, no. 2, pp. 478-491, 2015.
7. Mazhar Ali,Saif U. R Malik and Samee U. Khan, "DaSCE data security for cloud environmentwith semi-trusted third party", IEEE Transactions on Journal Name, vol. 5, no. 4, pp. 642-655, 2015.
8. Li Wei, Li Xue-lian, Gao Jun-tao and Wang Hai-yu, "Design of secure authenticated key management protocol for cloud computingenvironments", Journal of Latex Class Files, vol. 14, no. 8, pp. 1-15, 2015.
9. UthayashangarS, DhaminiP, MahalakshmiM and MangayarkarasiV, "Efficient group data sharing in cloudenvironment using honey encryption", IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), 29-30 March 2019, Pondicherry, India,2019.
10. Abdulatif Alabdulatif, Ibrahim Khalil, Albert Y. Zomaya, Zahir Tari and Xun Yi, "Fully homomorphic based privacy-preservingdistributed expectation maximization on cloud", IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 11, pp. 2668-2681, 2020.
11. Ming-quan Hong, Wen-bo Zhao and Peng-yu Wang, "Homomorphic encryption scheme based on ellipticcurve cryptography for privacy protection of cloudcomputing", IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC)

eISSN 1303-5150 www.neuroquantology.com

and IEEE International Conference on Intelligent Data and Security (IDS), 9-10 April 2016, New York, NY, USA,2016.

12. SaxenaV. K and Shashank Pushkar, “Privacu preserving model in cloud environment”, Conference on IT in Business, Industry and Government (CSIBIG), 8-9 March 2014, Indore, India,2014.
13. Jingyu Wang, Dongyuan Shi, Jinfu Chen and Chen-Ching Liu, “Privacy-preserving hierarchical state estimation inuntrustworthy cloud environments”, IEEE Transactions on Smart Grid, vol. 12, no. 2, pp. 1541-1551, 2021.
14. Hong Rong, Huimei Wang, Jian Liu and Ming Xian, “Privacy-preserving k-nearest neighbor computationin multiple cloud environments”, IEEE Access, vol. 4, pp. 9589-9603, 2016.
15. Jaehyoung Park, Dong Seong Kim and Hyuk Lim, “Privacy-preserving reinforcement learning usinghomomorphic encryption in cloudcomputing infrastructures”, IEEE Access, vol. 8, pp. 203564-203579, 2020.
16. Mazhar Ali, Revathi Dhamotharan, Eraj Khan, Samee U. Khan, Athanasios V. Vasilakos, Keqin Li and Albert Y. Zomaya, “SeDaSC secure data sharing in clouds”, IEEE Systems Journal, vol. 11, no. 2, pp. 395-404, 2015.
17. Abdul Razaque, Saty Siva Varma Nadimpalli, Suharsha Vommina, Dinesh Kumar Atukuri, Dammannagari Nayani Reddy, Poojitha Anne, Divya Vegi et al, “Secure data sharing in multi-clouds”, International Conference on Electrical, Electronics and Optimization Techniques (ICEEOT), 3-5 March 2016, Chennai, India, 2016.
18. Jingwei Li, Dan Lin, Anna Squicciarini, Jin Li and Chunfu Jia, “Towards privacy-preserving storage and retrieval inmultiple clouds”, IEEE Transactions on Cloud Computing, vol. 5, no. 3, pp. 499-509, 2015.
19. Sarala V and Shanmugapriya P, “DLFPM-SSO-PE privacy-preserving and security of intermediate data in cloud storage”, Distributed and Parallel Databases, 2021, Doi: 10.1007/s10619-021-07352-z.

20. Sunitha Pachala, Rupa Chand Sumalatha L, “An improved security and privacy management system for data in multi-cloud environments using a hybrid approach”, Evolutionary Intelligence, vol. 14, no. 4, pp. 1117-1133, 2021.
21. Prabu Kanna Gand Vasudevan V, “A fully homomorphic–elliptic curve cryptography based encryption algorithm for ensuring the privacy preservation of the cloud data”, Cluster Computing, 2018, Doi: 10.1007/s10586-018-2723-9.
22. Danish Ahamad, Shabi Alam Hameed and Mobin Akhtar, “A multi-objective privacy preservation model for cloud security using hybrid Jaya-based shark smell optimization”, Journal of King Saud University Computer and Information Sciences, 2020, Doi: 10.1016/j.jksuci.2020.10.015.
23. Suyel Namasudra, “An improved attribute-based encryption technique towards the data security in cloud computing”, Concurrency and Computation Practice and Experience, vol. 31, no. 9, pp. 1-15, 2017.