Suganya and Sasipraba
Journal of Cloud Computing (2023) 12:74
https://doi.org/10.1186/s13677-023-00442-6

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Stochastic Gradient Descent long short-term memory based secure encryption algorithm for cloud data storage and retrieval in cloud computing environment

M. Suganya[1*] and T. Sasipraba[2]

Abstract

With the increasing rise of distributed system technologies, one of the most pressing problems facing the digital world is ensuring the security of sensitive and confidential data during transport and storage, which is also regarded as one of the most critical difficulties facing cloud computing. Numerous techniques exist for enhancing data security in the cloud computing storage environment. Encryption is the most important method of data protection. Consequently, several accessible encryption strategies are utilized to provide security, integrity, and authorized access by employing modern cryptographic algorithms. Cloud computing is an innovative paradigm widely accepted as a platform for storing and analysing user data. The cloud is accessible via the internet, exposing the data to external and internal threats. Cloud Service Providers (CSPs) must now implement a secure architecture to detect cloud intrusions and safeguard client data from hackers and attackers. This paper combines Stochastic Gradient Descent long short-term memory (SGD-LSTM) and Blow Fish encryption to detect and prevent unauthorized cloud access. User registration, intrusion detection, and intrusion prevention are the three phases of the planned system. The SGD-LSTM classifier predicts cloud data access and prevents unauthorized cloud access. In the data access phase, cloud data access is managed by authenticating the authorized user with the Blowfish encryption algorithm. Comparing the proposed classifier to existing classifiers demonstrates that it detects abnormal access accurately. The experimental outcomes enhanced data security, which can be utilized to protect cloud computing applications. The experimental results of the suggested SGD-LSTM algorithm indicated a high level of protection, as well as a considerable improvement in security and execution speed when compared to algorithms that are often used in cloud computing.

Keywords Spatial data, Cloud computing, Cryptosystem, Cipher texts, Multi authority, Data correctness

Introduction

To a new computer paradigm known as cloud computing, many services are available on demand and at a minimal cost. Cloud computing's main objective is to offer quick, simple data and compute data storage. Cloud computing and other contemporary computer designs provide a wide range of immediately available properties. The main idea of cloud computing is to offer specific, quick functions for data processing and storing in a cloud architecture [1]. The computing industry is adept at controlling the dangers and hazards that the environment of

*Correspondence:
M. Suganya
suganyam02@gmail.com
[1] Research Scholar, Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India
[2] Vice Chancellor, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India

cloud computing poses. One method for improving cloud computing security is to use cryptography, which is the main element of cloud security. Furthermore, it is a technique for converting the user's letter into a cipher text, a coded phrase only the intended receiver can decipher and expose the concealed message. Secrecy is achieved throughout the transmission of data or communications. A mathematical method for handling authentication, such as encryption, and data integrity, is known as cryptology. It is conceivable to provide these security services since cryptography offers a variety of reliable possibilities [2]. Private data signified by cryptography (Hybrid Algorithms, Symmetric Methods, and Asymmetric Algorithms) is encrypted and decrypted using encryption protocols, digital signatures, and hashing algorithms of numerous well-known encryption algorithms [3]. For each of these current techniques, there are security issues. These techniques also take a long time to generate cryptographic keys, retrieve keys, encrypt data, and decrypt data.

A new cloud computing paradigm integrates high-level IT services alongside low-level resources abstractly. As a result, the user can quickly access numerous applications using an abstract cloud. The cloud-based outsourcing information storage and delivery service is one of the most likely uses. Some examples of these cloud storage services are Azure from Microsoft, Simple Storage Service, and CDNs like Cloud Front from Amazon Web Services. A third-party information center, a provider of cloud-based services, serves a significant role as an information management entity in a cloud storage system (CSP). The CSP can investigate data items kept in cloud storage without the consent of the data owners because it is the authority controlling the data items stored in the system. Unfortunately, most commercial cloud storage systems we are aware of offer essential storage services with plaintext data content storage. Even when the data owners do not want to disclose anything, there is a privacy issue involving the outsourced data due to either the CSP's malfunction or abuse for illegal profit. Some sensitive information, including medical records, financial graphs, or company reports, should be protected against exposure to unauthorized parties, according to the perspective of the data owners. As a result, the literature is paying more attention to the security and privacy concerns with this CSP and the potential illicit users. In other words, additional mechanisms like cryptographic methods are required, and data access management should be included in the hands of the CSP. By using cryptographic techniques, the information owner can encrypt the content of their data before outsourcing rather than leaving it in plaintext. The CSP could not recover encrypted items from a plaintext query without the decryption keys, so standard encryption would not be appropriate for cloud info retrieval systems.

Cloud storage for multimedia data reduces users' need for local storage space and privacy breaches. Still, it also introduces search issues, makes it easier for clients to share files, and causes data security issues [4, 5]. The rapid advancement of the Internet, cloud storage, and other technologies and the expanding use of multimedia collection technology worsen these problems. Speech info must be encrypted before being sent to the cloud because it comprises sensitive information. It is challenging to extract encrypted speech because of the significant modifications in encrypted speech characteristics and the ongoing expansion of speech information. Consequently, various research organizations and academics have become interested in researching encrypted speech retrieval devices.

Speech perceptual hashing technology is used in conventional encrypted speech detection methods to retrieve the perceptual components of speech [6–8]. The core of the retrieval process is the extraction of speech features, and the achievement of function expression directly impacts the outcomes of following retrievals. Since the speech features used by these perceptive hashing-based encrypted voice retrieval algorithms were already developed, redesigning the speech feature necessitates extensive research and testing. CNN is the most complex network structure for deep learning. CNN has succeeded in various artificial intelligence fields thanks to its strong generalization abilities and local information mining. Compared to CNN and CNN, LSTM can procedure the period sequence and simulate changes in the time sequence. The LSTM neural network, ideally predicated on the LSTM neural network, can handle the apparent data redundancy brought on by longer transmitting delays. The speech encryption mechanism is a vital component of the encrypted speech retrieval system that protects voice privacy in the cloud. Advanced Encryption Standard (AES), Multimedia data encryption using RSA, and Data Encryption Standard (DES), have replaced older encryption techniques like Rivset-Shamir-Adleman. Furthermore, hyperchaotic systems with initial parameter sensitivity, randomness, and ergodicity are frequently used for multimedia data encryption [9, 10].

Long-short-term memory (SGD-LSTM) based on stochastic gradient descent with BlowFish encryption. The three steps of the suggested model are the phases of intrusion prevention, intrusion detection, and user registration. The SGD-LSTM classifier forecasts cloud data access to weed out unauthorized cloud access. The Blowfish encryption algorithm is also used to verify the authenticated person during the data access process to restrict data in the cloud environment. It is empirically

shown that the recommended classifier correctly detects abnormal access by comparing it to the present classifiers. The experimental findings improved data security that can be utilized to safeguard cloud computing applications. Compared to widely used existing cloud computing methodologies, the practical consequences of the projected SGD-LSTM procedure exhibited a high level of safety and an apparent increase in safety and execution speed.

The following sections make up the remaining text: The analysis of linked studies is provided in Section II. The projected procedure has been provided in Unit III. The experimental validation of the encrypted voice recovery approach and a comparison to other methods are shown in Section IV. In Section V, the findings and future studies are covered.

Related work

Wei Song et al. [11] described a full-text retrieval Bloom-based tree index in a cloud service provider. It took a lot of work to isolate each word from the emails due to the abundance of data on the cloud. The user's privacy was protected, and utilizing index word membership entropies, it was found that the query and the encrypted data were identical. The difficulty of full-text recovery in cloud computing was overcome, and users' privacy was protected. But the filter-based tree index failed to increase the data security to the desired level.

Laurence T. Yang et al. [12] suggested a cryptography-based cloud-based security system. They contrasted the findings of their experiments to their concept of the RSA method for offering secure public-key cryptography techniques. RSA technology has been frequently employed in cloud computing for possible data protection. The difficulties of successfully factoring huge integers contribute to the RSA algorithm's security. Cloud computing allowed for the implementation of the General Number Field Sieve (GNFS) method, which at the time, was the most excellent and effective way of dividing numbers with more than 110 digits. They investigated the GNFS algorithm on the cloud while focusing on RSA security research. They notably presented current research on resolving various and sparse linear systems over General Field (GF), one of the GNFS algorithm's most time-consuming components.

Chris J. Mitchell et al. [13] investigate the vulnerability of the two-key triple DES. This technique provides 80-bit security, which is not possible in standard systems. The frequently changing keys have also been criticized as an extreme security measure. The author has proposed that the security margin of the two-key triple DES can be increased by replacing it with a three-key system because it is currently at its lowest possible level.

Tang et al. [14] investigated group activity motion-level features with different coherence restrictions to present a novel Coherence Constrained Graph LSTM (CCG-LSTM) for collection activity acknowledgment. Create an improved encryption technique that uses a three-layered dynamic approach to give moving targets dynamic definitions. This technique, which has the advantage of a partial critical update and a lower level of complexity, depends on network coding and DES. The implementation results showed that the proposed technique's running time is significantly less than the triple DES method.

Aarushi et al. [15] created a new RSA-based security technique considering four prime numbers. Instead of transmitting public keys directly, their method leverages the utilization of two positive integer values. It applies that the user would be provided these two public keys. This method offers a substantial gain in speed compared to the CRT-based RSA-based decryption approach.

Yu et al. [16] suggested a new encryption technique constructed on the learning parity with noise problem using single-bit and multi-bit strong values. They switched from using only one public essential encryption technique to a multi-bit one. This technique accurately resolves the decryption error issue and provides security against selective plaintext attacks. In addition, the minimum range of ciphertext and computational overhead is also enhanced by applying this technique to provide high-level security.

Tyagi M et al. [17] proposed Secure Management of Hybrid Cloud-Edge Environments. The researchers of this work have proposed a hybrid model. The inter-module communication security in this work was accomplished by monitoring all system operations. Digital certificates with timestamp capturing have been used for tracking. A hybrid encryption model that uses symmetric and asymmetric keys has been used to secure the environment. The message-digest algorithm and the PK infrastructure are the digital signature's foundation. AES (a symmetric-key algorithm), RSA-1024 (public-key cryptography with a 1024-bit key), and SHA-1 are the three encryption techniques used (a cryptographic hash function that generates a 160-bit message digest).

Tieyu Zhao et al. [18] suggested a novel data encryption technique with DNA (Deoxyribonucleic Acid) as it is an emerging field to enhance data security for cloud environments. A 1024-bit secret key was generated here, allowing the system to withstand multiple security attacks. The Media Access Control (MAC) addresses, DNA bases, American Standard Code for Information Interchange (ASCII) values, and rules like decimal encoding and complementary rules were used to construct the secret key. Theoretical analysis, as well as experimental results, proved the efficiency of this scheme over a few popular existing methods.

According to Aieh et al. [19], a contemporary innovation, cloud computing is a computerized service that clients use online. The information grouping indicates that the structure as it is achieved achieves security. This system's CSP chooses the renowned server using the Markov chain technique, Levy's flight, and the cuckoo strategy. After receiving server approval, the client encrypts its information using the Elliptic Curve Integrated Encryption Scheme (ECIES) before transferring it to the CSP for storage. After conducting a second encryption using the Advanced Encryption Standard (AES) on the cloud side, the CSP saves the data. The categorization is delivered on both sides via this dual encryption. This system combines server-determination access, authentication, and encryption technologies to achieve integrity and privacy while maintaining efficient computing.

Tieyu Zhao et al. [20] recommended a new encryption and decryption technique. They stared into a brand-new double variational encoding-based image encryption technique (DRPE). Therefore, the RSA public-key method was suggested. The system's main feature was that it adhered to the One-Time Pad (OTP) cryptography feature, which meant that every encryption process generated a new decryption key (even for the same plaintext). The system's extra features, such as its fingerprint key, could only be obtained legally, and only then was actual decryption possible. Otherwise, the decoding would produce jittery images. The suggested method could be used to determine whether the attackers faked the ciphertext. Combining the Public-key approach and the Asymmetric Cryptosystem's basic consensus further secured the system (ACS). The simulation results demonstrated how robust the encryption method was against recent attacks.

The DNA encryption system developed by Aich et al. [21] produces and distributes a private key utilizing Diffie-Hellman technology, which is subsequently encoded in a DNA sequence. The short text is transcribed into DNA sequences, then prefixes are added, and binary polymerization occurs. The primary code is then produced using the transcript in the subsequent DNA hybridization procedure. The codes table was modified in this section.

Cimi Thomas et al. [22] established a DNA-based safety scheme in which binary information is treated as DNA. This symmetrical structure is based on DNA. This method uses an encryption block size of 128 bits or 64 nucleotides. This coding system, which uses the same 16-round Cryptographic structure as DES and AES and provides the procedure with access to random DNA, is yet to be appropriate for any applications.

Proposed system

With the rise of data servers, cloud computing has evolved into a potential adversary capable of a) fundamentally altering the entire data technology culture, b) deploying programs, and c) managing upgrades. However, data privacy remains the most dangerous to cloud security. Data loss, information theft, or data omission in the cloud infrastructure are the main security threats that raise the alarm. Because the sophistication of these attacks will only increase, it is crucial to evolve defense technology to match the dangers they pose. This study offers an Integrated Intrusion Detection and Prevention System (IDPS) for cloud data based on SGD-LSTM and signatures access control policy to safeguard and protect client information from hackers and intrusions. Integrated Intrusion Detection and Prevention Systems (IDPS) are used to detect and prevent unauthorized access, attacks, and other security threats to computer systems and networks. In cloud computing, IDPS can play a critical role in ensuring the security of cloud environments. IDPS can help identify potential security threats in real-time in cloud computing environments. This includes detecting malicious activity from users, applications, and networks. IDPS can be used to secure cloud networks and detect any suspicious network activity. This can help prevent unauthorized access to cloud resources and data. IDPS can be used to enforce security policies in cloud computing environments. This includes ensuring compliance with regulatory requirements and preventing unauthorized access to sensitive data [23]. IDPS can help protect cloud environments from malware attacks. This includes detecting and preventing the spread of viruses, Trojans, and other malicious software. IDPS can be used to prevent intrusions into cloud environments. This includes detecting and blocking unauthorized access attempts, such as brute force attacks, password guessing, and SQL injection attacks. IDPS can help respond to security incidents in cloud computing environments. This includes identifying the source of the incident, containing the damage, and restoring the system to its normal state. It is critical to apply a competent design approach to construct a system that provides precise and accurate outputs. The three phases of the proposed method are intrusion detection, authentication, and registration. These stages are discussed further below. Figure 1 demonstrates how these phases progress and shows how each block functions Fig. 2.

Registration phase

Data access is restricted by registration by permitting only authorized users. The users registered their data to the cloud for admission during this phase. Users must request administrator consent to utilize a system. As a result, the user can send a request message to the cloud server with a signature, that is a unique Number (u_{Id}) and passwords $(u_{pass})(u_{Id}, u_{pass})$

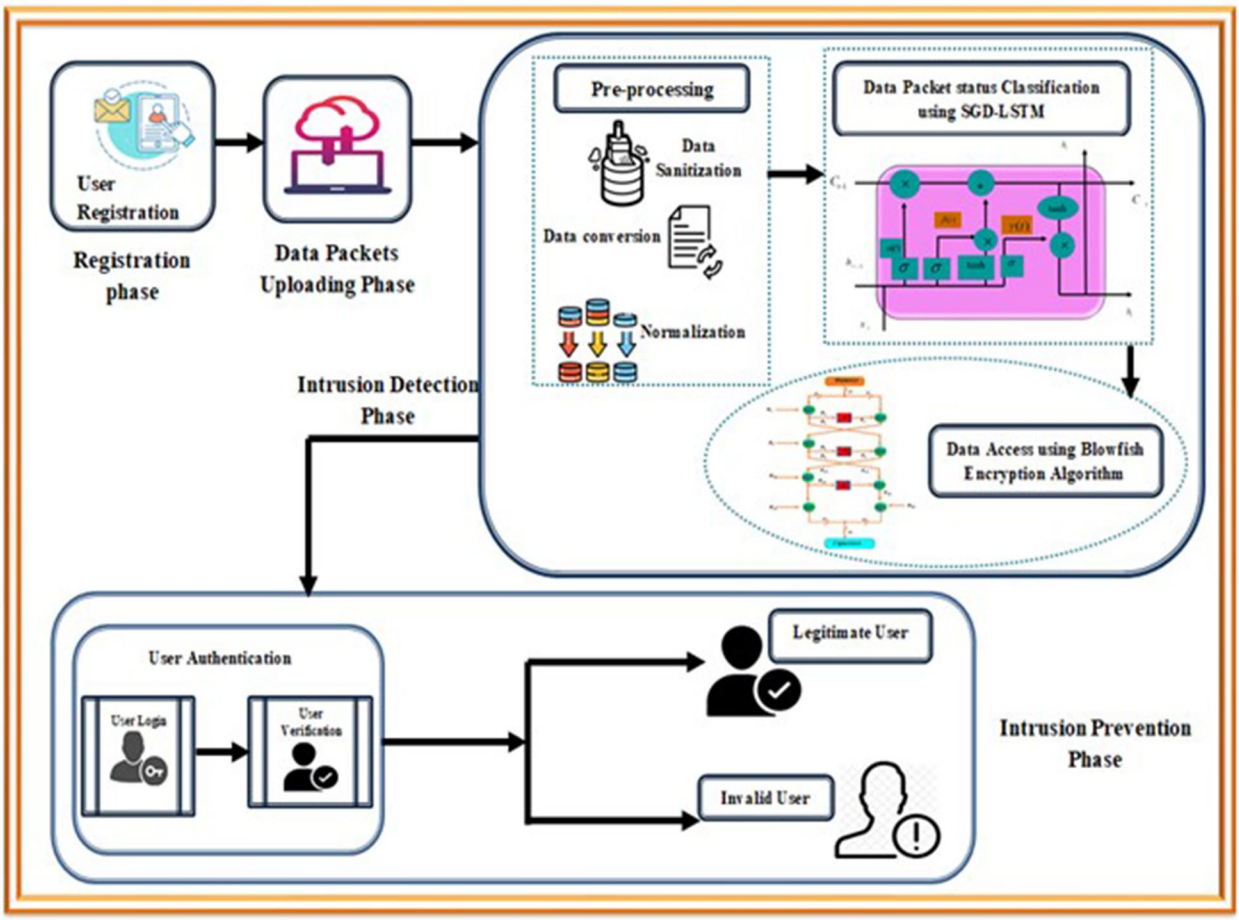

Fig. 1 Architecture of the proposed SGD-LSTM method

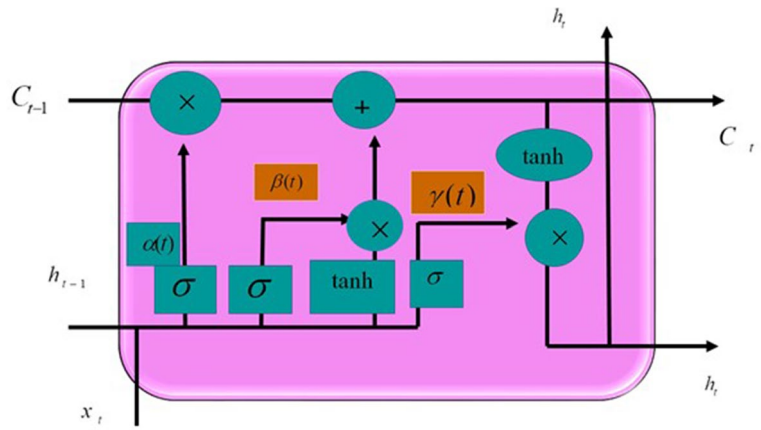

Fig. 2 The hidden layer of LSTM

$$user \xrightarrow{R\_REQ(u_{Id},u_{pass})} Cloud_{server} \quad (1)$$

where $Cloud_{server}$ stands for the cloud server and $R\_REQ\{\bullet\}$ for the registration request [24]. Following receipt of the user's registration request, the cloud server determines the hash code for the user's password.

Figure 1 displays the block diagram for the suggested SGD-LSTM technique. The proposed system's process

is illustrated in detail in the figure. The user first goes through the registration step, when login credentials are created and registered for uploading data packets. The user is directed to the intrusion detection phase, where data pre-processing and access occur after registering with their credentials [25]. The data pre-processing module does data conversion, data normalization, and Data sanitization. The pre-processed data is categorized and accessed using the SGD-LSTM and BEA algorithms. The intrusion protection phase is the final stage of the suggested model, and it involves user authentication via user login and identity verification status. Suppose the entered data is coordinated with the registered information [26]. In that case, the user is confirmed to be a legitimate user, and if the data is mismatched, the user claims to be an invalid user.

Data pre-processing

Massive volumes of unneeded, duplicated, and outlier values are present in the collected data. Due to the existence of outliers, learning algorithms do not yield better results. Pre-processing increases the prediction accuracy of the system by removing duplicate data and irrelevant or noisy information from user data. The four pre-processing phases are data cleansing, numerical conversion, normalization, and sanitization. The four pre-processing stages include data cleaning, sanitization, normalization, and numerical transformation. Consider the data provided as

$$(F_d) = \{F_{d1}, F_{d2}, F_{d3}, \ldots\ldots, F_{dk}\} \tag{2}$$

The input data are denoted by $(F_d)_i$, while the number of information points is denoted by F_{dk}.

(a)The process of eliminating outliers from data, replacing missing numbers, cleaning up noisy data, and repairing inconsistent data is known as data sanitization. The dataset is being cleaned up by removing duplicate and noisy data. (b) Numerical Conversion: The algorithms for learning can only operate on numbers [27]. The features in the obtained dataset are in string format, such as protocol type. Numerical data is created from the textual values.

(c) Normalization: Scaling the feature values into precise confidence intervals is done in this stage. The bias in the raw data is eliminated, which is the main advantage of this stage. Every feature value is positioned within a predetermined range based on Z-score normalization.

$$N(F_d) = \frac{X(F_d)_i - Y(F_d)_i}{\sigma_{X(F_d)_i}} \tag{3}$$

where $Y(F_d)_i$ stands for the data's standard deviation and $\sigma_{X(F_d)_i}$ for the $X_{(F_d)_i}$ data's mean. Pre-processing is characterised by,

$$(F_d)_i \xrightarrow{\text{preprocessing}} P(P_d)_i \tag{4}$$

In Eq. (4), $P(F_d)_i$ denotes the pre-processed data.

Stochastic gradient descent

The stochastic gradient descent (SGD) technique is a subtype of descent-based approach that finds ideal values by utilizing the gradient of functions. Numerous applications have used the SGD technique for cost optimization. Optimization issues involving the function f(x), where x is a constant vector, can be solved using SGD techniques [28]. An example of an optimization issue is as follows:

$$Minimize S_{TR}(w) where (w) = W_n, W_s, L_n, L_s, V_{th} \ldots \tag{5}$$

The SGD algorithm's fundamental form is as follows,

$$w_{n+1} = w_n - \lambda_n \nabla S_{TR}(w_n) \tag{6}$$

If w_n must be estimated, the value that reduces the objective function. The optimization model $S_{TR}(w)$ i's gradient or derivative with respect to w_n. The size of the fall is determined by λ, a user-defined quantity. Another name for it is the learning rate. Because n represents the number of iterations stages, the default value for λ is frequently set to 1 n or another, decreasing functionality concerning n. With a big λ, the process could become unstable, while a small λ would cause prolonged steps and a long convergence time [29]. While SGD and gradient descent are similar, SGD uses a part of the data vector randomly selected at each iteration stage to estimate the gradient of the optimization problem. $S_{TR}(w)$ The optimization process is sped up by predicting the gradient at each successive iteration, and calculation costs are significantly decreased. Due to this feature, the SGD is ideally suited for costly simulations and functions that are difficult to differentiate. The SGD works well for local optimization since it can get stuck at a local minimum [30]. We provide a method that repeatedly runs the process N times, memorizing the local minima discovered and the parameter range travelled. To avoid unnecessary searches, the algorithms want to ensure that each time it starts over with a different random area that has yet to be explored. The best point is chosen from the local minima group after N applications of the procedure.

Algorithm 1 displays the suggested algorithm. The strategy enhances an objective outcome S $S_{TR}(w)$ as a factor of the design requirements w by altering the standard SGD. The most significant number of iterations, called N, is used to determine the starting point for the optimization procedure. After each iteration step, a set of solutions and the route are stored in vector W (lines 4–8). The process is restarted or repeated via lines 4–16 until the maximal number of iterations has been reached, or the other end condition is satisfied. Before choosing the next random point, it checks to see if it has been searched. The optimized design aim

is selected from the vector w's minimal set of values at the algorithm's conclusion. By keeping track of the collection of random points in this algorithm, we increase efficiency by limiting the range of parameters chosen to only those whose pathways have not yet been travelled [31]. The optimization process can go more swiftly by removing unnecessary searches, search queries that would yield discarded results, and inquiries that would return already-stored optima.

Algorithm 1 Algorithm for stochastic gradient descent

1. $N \leftarrow Max\_Iter$
2. Choose random variable $w_0, w_0^{'}$
3. Calculate FoM $S_{TR}(w_0)$
4. **while** $\| S_{TR}(w_{n+1}) - S_{TR}(w_n) \| > \varepsilon$ **do**
5. Select a decreasing λ_n (generally $\frac{1}{n}$)
6. Estimation $\nabla S_{TR}(w_n)$ using $\nabla S_{TR}(w_n^{'})$
7. Compute $x_{n+1} = x_n - \lambda_n \nabla S_{TR}(x_n)$
8. **end while**
9. $W \leftarrow \{w_n, S_{TR}(w_n)\}$
10. Reset w_0, $w_0^{'}$
11. **if** ($w_0^{'}$) within range of W **then**
12. Reset
13. **else**
14. $N \leftarrow N-1$
15. Restart the search algorithm
16. **end if**
17. **repeat**
18. algorithm search
19. **Until** N=equals 0
20. **return** the lowest couple $w_n, S_{TR}(w_n)$ found

LSTM

RNN is the architecture of deep learning that is used for sequential data. Deep knowledge also has other architectures. LSTM is one of the many models of RNN, and it is distinguished from the others by its hidden layer containing three gates and cell memory. First, let's assume we are dealing with the long-term memory model (LSTM) output as $x_1 \ldots x_2 \ldots x_t$. At each t-step, data will be stored in the cell's memory [32]. Three gates from LSTM's hidden layer, as depicted in Fig. 4: forget gates, input gates, and output gates.

This hidden layer's first task is determining how much information from the preceding hidden layer will be stored in LSTM cells' internal memory. Using this mathematical method, one could get the following answer at the forget gate:

$$\alpha(t) = \sigma(W_{x^{x_t}} + W_h h_{t-1}) \tag{7}$$

In this example, W_h stands for without gate weight. The memory of the cell will retain how much of the new input text. Using the following formula, the Ct value will be calculated at the input gate:

$$\beta(t) = \sigma(R_{x^{x_t}} + R_h h_{t-1}) \tag{8}$$

where the R_x and R_h values represent the weights for the input gate, following the acquisition of new data, we will use the following formula to bring the cell memory C_t up to date:

$$C_t = C_{t-1}\alpha(t) + \beta(t).tanh(p_{x^{x_t}} + p_h h_{t-1}) \tag{9}$$

P_x and P_h represent the cell's memory weight [33]. That leaves figuring out how much of the cell memory C_t data will be used to generate an output for layer t, which is calculated using the formula below:

$$h_t = \alpha(Q_x x_t + Q_h h_{t-1}).tanh(c_t) \tag{10}$$

where the Q_x and Q_h values represent the weight of the output gate.

Login phase

In general, login refers to credentials that aid in user authentication. Any user can send a login request to the cloud to gain access to its data by providing the cloud with their unique ID u_{Id} and password u_{pass}.

$$user \xrightarrow{L\_REQ\{u_{ID}, u_{pass}\}} Cloud_{server} \tag{11}$$

$L\_REQ\{\bullet\}$ Here, the login request is indicated. User authentication is done after the user is asked to log in.

User authentication phase

During the user authentication stage, the system checks to see if the user is a legitimate user. After receiving the user's login request, the cloud requests a code from the user. The user follows that by sending a code to the cloud server.

$$Cloud_{server} \xrightarrow{ask\ code} user \xrightarrow{send\ code} Cloud_{server} \tag{12}$$

If the user sends a correct hash code, the server authorizes the request and confirms that the user can access the cloud information. This method is described as,

$$\mathrm{H}(u_{pass})_{matched} \rightarrow \left(Cloud_{server} \xrightarrow{Confirms} L_{user}\right) \tag{13}$$

$$\mathrm{H}(u_{pass})_{Not\ matched} \rightarrow \left(CS \xrightarrow{Confirms} I_{user}\right) \tag{14}$$

L_{user} stands for "valid user" and I_{user} for "invalid user" in this context. Cloud users only utilize the produced hash code once at a time [34]. Every time a user accesses the data, they generate a new hash code and password, preventing unauthorized individuals from accessing that user's data.

Data security using the blowfish encryption algorithm

A BEA is used to encrypt the information input after registration. The Blowfish Algorithm is a symmetric key cryptography algorithm (BEA). The 64-bit block's key length is 32–448 parts. There are four 32-bit S-boxes and a P-array available. The S-boxes can recognize 8-bit data while transmitting 32-bit yield. The key expansion and encryption phases are the two main stages of the BEA. To encrypt data, a 16-round FSTEL network is utilized. There are key-dependent substitutions and primary dependency permutations in every round. The only functionality in XOR and BA is the addition of 32-bit words. In Fig. 3, the BEA structure is shown.

Public key encryption phase

The data owner uses an elliptic algorithm to create a key for authorized access during this stage. The records and index are then encrypted with Blowfish.

Step 1: Conception of Keys ECC's starting point is the information provided below.

$$E : x^2 = y^3 + ay + b \tag{15}$$

$$4a^3 + 27b \neq 0 \mathrm{mod} r \tag{16}$$

Here, a and b are numbers that satisfy condition (2), r is a prime number, and r includes a point at infinity.

$$P = d^* r \tag{17}$$

d is a random number between 1 and n – 1. r is for the curve's point, d is for the private key, and P is for the public key.

Step 2: preserving personal data The Blowfish algorithm is used to address privacy concerns. It is a symmetric block cipher that works well for data encryption and preservation.

Algorithm 2: Blowfish encryption algorithm

Blowfish Encryption Algorithm
Split X into two 32-bit division: X_L, X_R
For k=1 to 16
$X_L = X_L\ XOR_{pk}$
$X_R = F(X_L) XOR_{XR}$
Swap X_L and X_R
Next k
Swap X_L and X_R (undo the last swap)
$X_R = X_R XOR_{p17}$
$X_L = X_L XOR_{p18}$
Re*merge* X_L *and* Y_R

Algorithm 2 represents the Blowfish encryption algorithm through the algorithm Function, which is determined below.

$$f_n(XL) = \left((S_0, i + S_1, j \mathrm{mod} 2^{32}) XORS_2, k\right) + S_3, l \mathrm{mod}\ 2^{32} \tag{18}$$

Partition X_L into 8-bit parts: i,j, k,l

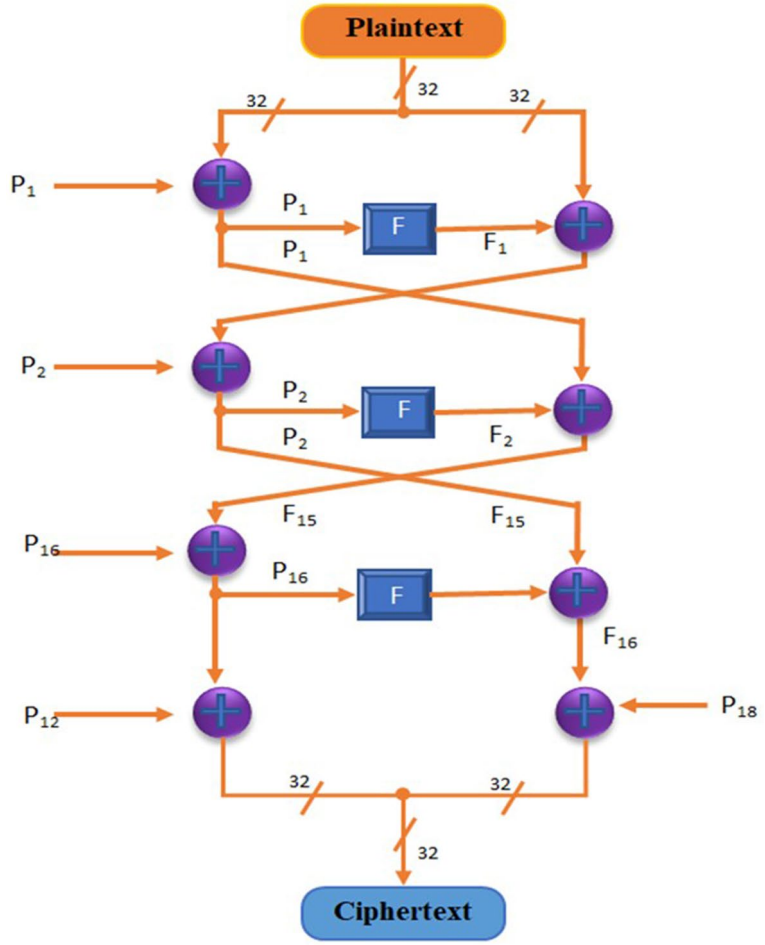

Fig. 3 Structure of BEA

S [0]: 243f6a88	S [9]: 38d01877
S [1]: 85a368d3	S [10]: be5466cf
S [2]: 13198a2e	S [11]: 34e90c6c
S [3]: 03,707,244	S [12]: c0ac24b7
S [4]: a4093822	S [13]: c97c50dd
S [5]: 279f31d0	S [14]: 3f87d5b5
S [6]: 082efa98	S [15]: b5470517
S [7]: ec4e6c89	S [16]: 9296d5d9
S [8]: 452821e6	S [17]: 8879fb1b

Step 3: Cloud Storage For the benefit of the information owner, the searchable index I and the system file collection C are kept in the cloud [35]. Since receiving the user's query request, the index locates the pertinent encrypted data and makes it accessible to users. When an authorised user wants to learn concerning the cloud data, they create a query request and send it to the cloud server. After obtaining the encrypted text, the cloud server processed the request and used the cloud's index to generate paired results. The authorised user then decrypts the database tools on the cloud server.

Result and discussion

Decryption time, Encryption, accuracy, f-score, f-score, precision, recall, f-score, accuracy, and RMSE are evaluated to determine how resilient the suggested coding scheme is. This is accepted to validate the recommended procedure. The proposed method was compared to other genetic data encryption and existing encrypted with symmetric keys approaches in terms of decryption time, precision, recall, decryption time, accuracy, and root mean squared error based on the scope of the plain text and the time essential for Encryption and decryption.

Experimental setup

Java programming is used to implement the suggested full-text retrieval approach. On a server running

Windows 7 with a 64-bit 2.9 GHz CPU and 4 GB of primary RAM, we conducted the experiments.

Dataset description

Our experiments were built on the Enron Email Dataset, containing 200,399 interactions from 158 people. We use the Enron email dataset, a real-world dataset, as our corpus. We take a subset of emails from the Enron dataset. This dataset is suitable because it can represent businesses that occasionally want to search through encrypted emails that are stored on a remote server. Before testing, we pre-processed the dataset by creating keywords and permutations. We specifically cleaned up the corpus using the Porter stemming algorithm, removed content-unrelated words with a spelling checker, and generated an inverted index for the top 1500 most popular keywords for keyword generation. The permutation was created using the prefix method.

Evaluation parameters of the proposed algorithm

SGD-LSTM network using '7' quality metrics: precision, recall, F-Score, accuracy, encryption time, decryption time, and RMSE. These metrics are calculated using the four crucial parameters, true positive (TP), false positive (FP), true negative (TN), and false negative (FN), as follows:

Precision: It represents the fraction of data packet status correctly recognized as intruder packets concerning all packets detected as intruder packets. Precision is a measure of the correctness of the retrieved data. It is the ratio of the true positives to the total number of positive instances, where true positives are the number of correct results that were retrieved. In the context of cloud data storage and retrieval, precision is important because it ensures that the retrieved data is relevant and accurate, reducing the risk of errors or inconsistencies in data analysis or decision making.

$$\text{Pr}ecision = \frac{TP}{TP + FP} \tag{19}$$

Recall: It is the correctly detected segment of intruder packets. Recall is a measure of the completeness of the retrieved data. It is the ratio of the true positives to the total number of positive instances in the dataset. Recall is important in cloud data storage and retrieval because it ensures that all relevant data is retrieved, reducing the risk of missing critical information.

$$\text{Re}call = \frac{TP}{TP + FN} \tag{20}$$

F-Score: It is the sensitivity and precision's harmonic mean.

$$F - Score = 2 * \frac{precision * recall}{precision + recall} \tag{21}$$

Accuracy: It is the rate of data packets status that is correctly predicted. Accuracy is a measure of the overall correctness of the retrieved data. It is the ratio of the true positives and true negatives to the total number of instances in the dataset. In cloud data storage and retrieval, accuracy is important because it ensures that the retrieved data is both relevant and complete, reducing the risk of errors or inconsistencies in data analysis or decision making. precision, recall, and accuracy are important metrics in influencing the aim of cloud data storage and retrieval in a cloud computing environment. These metrics ensure that the retrieved data is both relevant and accurate, reducing the risk of errors or inconsistencies in data analysis or decision making.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

Encryption time: The quantity of time it takes to alter plain text into cipher text can be used to quantify encryption time. The quality of the encryption algorithm is inversely correlated with the encoding time; if the method requires less encryption time, it will be more effective. There are two ways to quantify this variable:

- Based on the input size, analyze the encryption time (20, 40, 60, 80, 100, and 120 KB).
- Based on the input's variable character count, we determine how long encryption and decryption will take.

The results revealed that although deciphering the process takes less time than decrypting it, it is still feasible that the time required for encryption may increase linearly simply as the number of letters or file size grows. It demonstrates how much less computationally complex the suggested job is.

The proposed method encrypts plain text more quickly than the symmetrical algorithm, a different genetic algorithm. Deoxyribonucleic acid (DNA), Rivest-Shamir-Adleman (RSA), Data Encryption Standard (DES), Advanced Encryption Standard (AES), and RC4-DNA (DNA).

Decryption time: The Decrypting Time shows how long it takes to decrypt text inputs and reveal their original contents. The time required by the decrypting algorithm is known as its time complexity. The following

formula can be used to determine the amount of time spent decrypting:

$$Timeconsumed = endtime - starttime \quad (23)$$

Comparing the suggested method to other encryption methods, the proposed way is faster to decode. Therefore, the presented approach can be used and is successful for safe communication.

Encryption time analysis

The encryption time of the SGD-LSTM methodology is compared to that of other methods in Table 1 and Fig. 4. The data demonstrate that the proposed method outperformed the different strategies in every way. For instance, with 20 nodes, the SGD-LSTM technique encrypts data in 3.652 ms, while other existing methods such as RSA, DES, AES, DNA, and RC4-DNA

Table 1 Encryption time analysis for SGD-LSTM method with existing systems

No of Nodes (msec)	RSA	DES	AES	DNA	RC4-DNA	SGD-LSTM
20	13.749	10.294	9.465	7.346	5.845	3.652
40	13.294	10.321	9.642	8.765	6.543	4.876
60	13.983	10.387	9.132	8.132	6.895	4.172
80	14.293	10.854	9.875	8.453	6.567	4.765
100	13.554	11.743	11.167	8.876	5.934	4.115
120	14.854	11.345	10.187	9.143	7.132	5.364

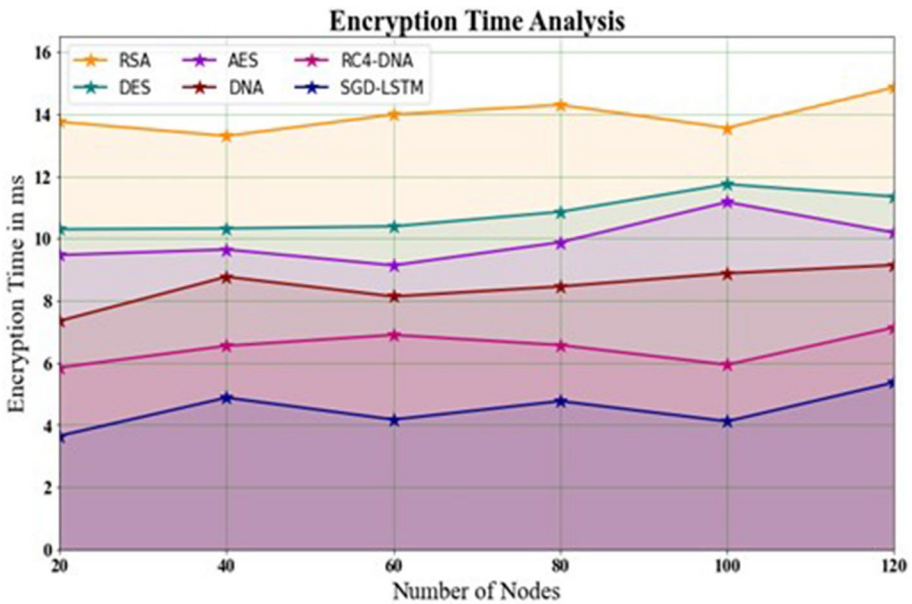

Fig. 4 Encryption time analysis for SGD-LSTM method with existing systems

Table 2 Decryption time analysis for the SGD-LSTM method with existing systems

No of Nodes (msec)	RSA	DES	AES	DNA	RC4-DNA	SGD-LSTM
20	10.763	9.632	7.543	6.345	4.156	2.125
40	11.542	9.321	7.652	6.123	4.276	2.325
60	11.621	9.487	7.145	5.765	4.854	2.638
80	11.984	9.865	8.346	6.954	5.123	3.143
100	12.321	10.543	9.567	7.321	5.654	3.453
120	12.876	10.672	8.862	7.432	5.876	3.654

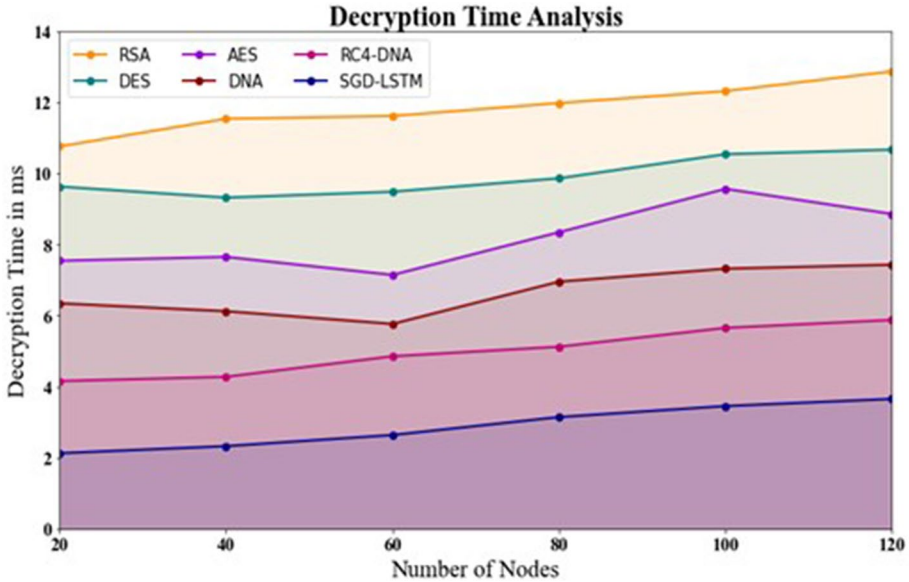

Fig. 5 Decryption time analysis for SGD-LSTM method with existing systems

take 13.749 ms, 10.294 ms, 9.465 ms, 7.346 ms, and 5.845 ms, respectively. Similarly, the SGD-LSTM method's encryption time for 120 nodes is 5.364 ms, compared to 14.854 ms, 11.345 s, 10.187 ms, 9.143 ms, and 7.132 ms for other existing techniques.

Decryption time analysis

The decryption time contrast of the SGD-LSTM methods with existing methods is characterized in Table 2 and Fig. 5. The data demonstrate that the proposed method outperformed the other techniques in every way. The SGD-LSTM process, for example, took only 2.125 s to decrypt the data with 20 nodes. In contrast, other existing methods, such as RSA, DES, AES, DNA, and RC4-DNA, took 10.763 s, 9.632 s, 7.543 s, 6.345 s, and 4.156 s, respectively. Correspondingly, for 120 nodes, the SGD-LSTM method decrypts in 3.654 s, while other existing techniques such as RSA, DES, AES, DNA, and RC4-DNA decrypt in 12.876 s, 10.672 s, 8.862 s, 7.432 s, and 5.876 s, respectively.

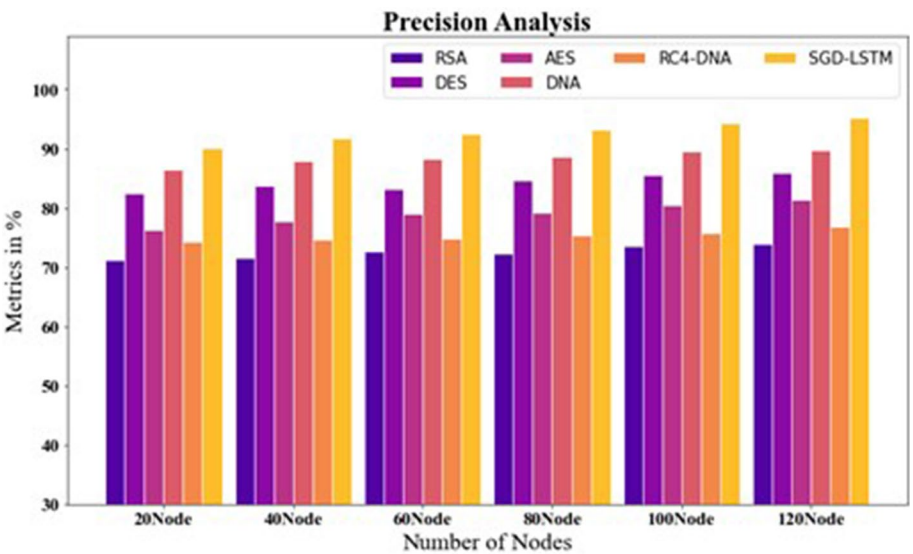

Fig. 6 Precision analysis of the SGD-LSTM method with existing systems

Table 3 Precision analysis for the SGD-LSTM method using existing systems

No of Nodes	RSA	DES	AES	DNA	RC4-DNA	SGD-LSTM
20	71.12	82.54	76.32	86.43	74.23	90.15
40	71.65	83.65	77.67	87.87	74.67	91.64
60	72.72	83.17	78.98	88.34	74.82	92.43
80	72.32	84.56	79.15	88.56	75.45	93.26
100	73.65	85.64	80.43	89.62	75.67	94.36
120	73.87	85.87	81.34	89.74	76.84	95.25

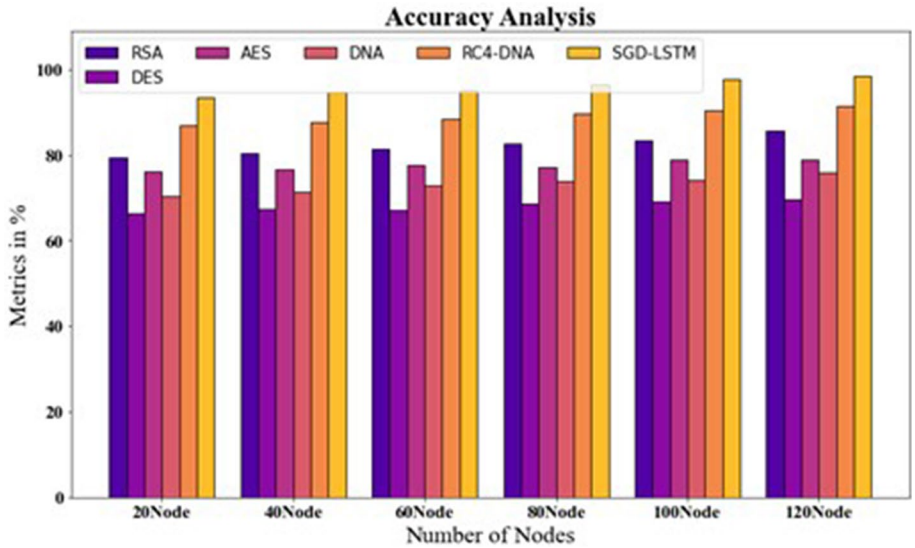

Fig. 7 SGD-LSTM method accuracy analysis with existing systems

Table 4 SGD-LSTM method accuracy analysis with existing systems

No of Nodes	RSA	DES	AES	DNA	RC4-DNA	SGD-LSTM
20	79.32	66.43	76.24	70.43	86.95	93.34
40	80.53	67.32	76.66	71.45	87.65	94.75
60	81.45	67.13	77.65	72.87	88.32	95.16
80	82.76	68.56	77.26	73.96	89.65	96.53
100	83.42	69.25	78.78	74.12	90.37	97.72
120	85.66	69.74	78.91	75.83	91.45	98.45

Precision analysis

A precision comparison of the SGD-LSTM strategy with other methods is shown in Fig. 6 and Table 3. The graph depicts how the cloud method has improved performance and precision. SGD-LSTM, for instance, has a precision value of 90.15% with 20 nodes, whereas RSA, DES, AES, DNA, and RC4-DNA have precision values of 71.12%, 82.54%, 76.32%, 86.43%, and 74.23%, respectively. However, the SGD-LSTM model performed best with a variable number of nodes. Similarly, under 120 nodes, the precision value of SGD-LSTM is 95.25%, whereas it is 73.87%, 85.87%, 81.34%, 89.74%, and 76.84% for the RSA, DES, AES, DNA, and RC4-DNA models, respectively.

Accuracy analysis

In Fig. 7 and Table 4, the accuracy of the SGD-LSTM method is contrasted with that of other earlier

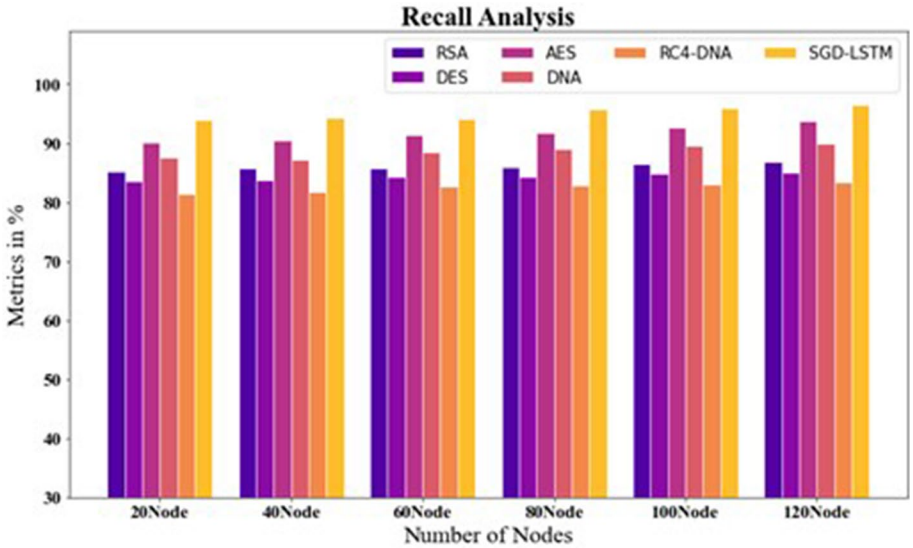

Fig. 8 Recall time analysis for SGD-LSTM method with existing systems

Table 5 Recall time analysis for SGD-LSTM method with existing systems

No of Nodes	RSA	DES	AES	DNA	RC4-DNA	SGD-LSTM
20	85.13	83.48	90.12	87.54	81.42	93.89
40	85.65	83.67	90.43	87.12	81.76	94.32
60	85.74	84.23	91.43	88.46	82.65	94.12
80	85.88	84.36	91.65	88.98	82.74	95.67
100	86.45	84.87	92.67	89.56	82.98	95.87
120	86.76	84.93	93.76	89.93	83.32	96.54

methods. According to the graph, the cloud technique has improved performance and accuracy. SGD-LSTM, for example, has an accuracy of 93.34% with 20 nodes, whereas RSA, DES, AES, DNA, and RC4-DNA have an accuracy of 79.32%, 66.43%, 76.24%, 70.43%, and 86.95%, respectively. The SGD-LSTM model, on the other hand, has shown excellent performance with various nodes. Similarly, under 120 nodes, SGD-LSTM has an accuracy value of 98.45%, whereas RSA, DES, AES, DNA, and RC4-DNA have accuracy values of 85.66%, 69.74%, 78.91%, 75.83%, and 91.45% respectively.

Recall analysis

Figure 8 and Table 5 compare the recall of the SGD-LSTM strategy to that of other methods. The graph illustrates that the cloud technique improved recall performance. SGD-LSTM, for instance, has a recall value of 93.89% with 20 nodes, while the RSA, DES, AES, DNA, and RC4-DNA models have recall values of 85.13%, 83.48%, 90.12%, 87.54%, and 81.42%, respectively. However, the SGD-LSTM model performed best with a variety of nodes. Similarly, under 120 nodes, SGD-LSTM has a recall value of 96.54%, whereas RSA, DES, AES, DNA, and RC4-DNA have recall values of 86.76%, 84.93%, 93.76%, 89.93%, and 83.32% respectively.

F-score analysis

In Fig. 9 and Table 6, the SGD-LSTM strategy is compared to other earlier techniques. The graph demonstrates that the cloud approach improved performance as measured by the f-score. For instance, the RSA, DES, AES, DNA, and RC4-DNA models have f-scores of 87.43%, 75.45%, 81.76%, 78.75%, and 84.92%, respectively. SGD-LSTM, on the other hand, has 20 nodes and an f-score of 89.73%. However, the SGD-LSTM model performed best with a variety of nodes. Similarly, the f-score value of SGD-LSTM under 120 nodes is 93.76%, whereas it is 89.54%, 78.21%, 84.86%, 81.14%, and 86.95% for RSA, DES, AES, DNA, and RC4-DNA models, respectively.

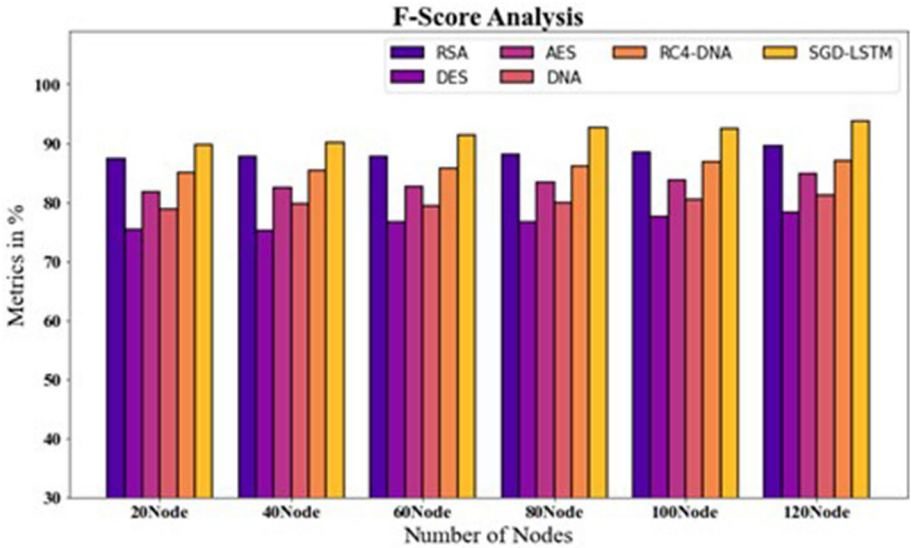

Fig. 9 F-SCORE analysis for SGD-LSTM method with existing systems

Table 6 F-SCORE analysis for SGD-LSTM method with existing systems

No of Nodes	RSA	DES	AES	DNA	RC4-DNA	SGD-LSTM
20	87.43	75.45	81.76	78.75	84.92	89.73
40	87.74	75.13	82.52	79.65	85.45	90.15
60	87.82	76.65	82.63	79.43	85.73	91.42
80	88.12	76.56	83.45	79.87	86.12	92.63
100	88.53	77.53	83.74	80.42	86.87	92.43
120	89.54	78.21	84.86	81.14	86.95	93.76

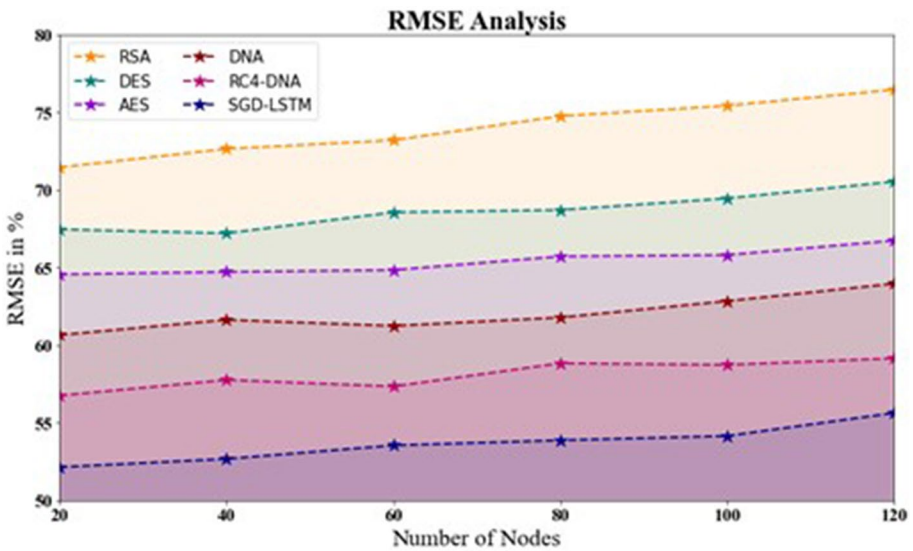

Fig. 10 RMSE analysis of the SGD-LSTM method with existing systems

Table 7 RMSE analysis of the SGD-LSTM method with existing systems

No of Nodes	RSA	DES	AES	DNA	RC4-DNA	SGD-LSTM
20	71.46	67.46	64.56	60.65	56.75	52.14
40	72.65	67.21	64.72	61.63	57.76	52.67
60	73.21	68.56	64.84	61.24	57.34	53.54
80	74.76	68.71	65.72	61.78	58.83	53.87
100	75.43	69.46	65.81	62.84	58.73	54.15
120	76.47	70.54	66.74	63.96	59.15	55.63

RMSE analysis

Comparative RMSE analyses of the SGD-LSTM strategy with other previous techniques are shown in Fig. 10 and Table 7. The graph shows that the cloud strategy produced an improved performance with a lower RMSE value. The RSA, DES, AES, DNA, and RC4-DNA designs all have slightly better RMSEs than SGD-LSTM, which has an RMSE of 52.14% with 20 nodes. The SGD-LSTM design, on the other hand, has shown maximum performance across a wide range of nodes while maintaining low RMSE values. Similarly, the RMSE value of SGD-LSTM under 120 nodes is 55.63%, whereas it is 76.47%, 70.54%, 66.74%, 63.96%, and 59.15% for RSA, DES, AES, DNA, and RC4-DNA models, respectively.

Conclusion

The ultimate consumers of the cloud are expanding with their use for personal and professional purposes due to the enormous increase in data and computing technology. Their efficient and adaptable computing and storage solutions tailored to the requirements of the applications are the cause of this achievement. On-demand, the cloud offers computational and data services. Concerning data security and privacy, numerous challenges remain to be overcome. The cloud makes data storage more accessible and adaptable, but unwanted attacks and operations still exist. Sensitive data could be covertly stored on the cloud server. Data security is essential as a result. Here, we combine Stochastic Gradient Descent long short-term memory (SGD-LSTM) with the Blow Fish encryption technique to identify and prevent unauthorized cloud access. The proposed system is divided into three phases: intrusion detection, user registration, and intrusion prevention. The SGD-LSTM classifier is used to forecast cloud data access in instruction to weed out unauthorized cloud access. The Blowfish encryption algorithm is also used to verify the authenticated person during the data access phase in instruction to limit access to information in the cloud environment. Comparing the recommended classifier to the current classifiers demonstrates empirically that it accurately detects abnormal access. The experimental results enhanced data security, which helps defend cloud computing applications. The experimental results show that the suggested strategy performs better than the current methods in terms of encryption time, decryption time,precision, recall, f-score, accuracy and RMSE. It also performs well in terms of retrieval efficiency and retrieval accuracy for SGD-LSTM algorithms.

Authors' contributions

Conceptualization, Author1. and Author 2.; Methodology, Author1 and Author2; Validation, Author 1; Data curation, Author 2; Writing—original draft preparation, Author 1 and Author 2; Writing—review and editing, Author 1,2 and Author 1 Supervision R.W and EOM. The author(s) read and approved the final manuscript.

Author's information

Suganya M, Research Scholar, Department of Computer Science and Engineering,in Sathyabama Institute of Science and Technology and working as Assistant Professor, Department of Computer and Communication Engineering in Sri Sairam Institute of Technology. She has 6+ years of teaching and Industry Experience. Her Area of Interest is Cloud Security, Robotics process Automation and Satellite Technology. She has been awarded " Best Performing Professor" for contribution towards Nano Satellite Research Project, A stamp was released to honour her by Government of India Post. Also have been awarded "Young Research Engineer" for carrying out vibrant activities on small satellites under the banner of UNISEC India. She is one of the Project Head for UNITY SAT JIT-SAT launched from ISRO on Feb 28th 2021. She has also been awarded " Excellence in research and Innovation of the Year 2021. Dr.T. Sasipraba, obtained her B.E and M.E., from the University of Madras and Ph.D from Sathyabama University. She joined Sathyabama University in 1995 as a Lecturer and her 19 years of meritorious career in the same University has promoted her as Vice Chancellor of the university in the year 2020. During the course of her career at Sathyabama University Dr. T. Sasipraba has made exceptional contributions in the areas of research and developments, international linkages and Publications. For her outstanding contributions over the years, Dr. Sasipraba has received numerous awards from Sathyabama University and from Cognizant Technology Solutions. She has published more than 125 papers in refereed international journals and conference proceedings and has guided many Ph.D Scholars in the field of Computer Science and Engineering.

Funding

The authors received no specific funding for this study.

Availability of data and materials

The manuscript contains all of the data.

Declarations

Ethics approval and consent to participate
We confirm that our research does not involves a survey asking real human participants to give opinions, or animals data to make justifications.

Competing interests
The authors state that they do not have any conflicts of interest.

Received: 8 November 2022 Accepted: 11 April 2023
Published online: 09 May 2023

References

1. Groom F.M (2018) "The basics of cloud computing," in Enterprise Cloud Computing for Non-Engineers
2. Thabit, Fursan and Alhomdy, Sharaf Abdul-Haq and Alahdal, Abdulrazzaq and Jagtap, Sudhir B (2020) Exploration of Security Challenges in Cloud Computing: Issues, Threats, and Attacks with their Alleviating Techniques (December 02, 2020). J Chem Inf Comput, 12(10). Available at SSRN: https://ssrn.com/abstract=3749271
3. Sajid A, Abbas H, Saleem K (2016) Cloud-assisted IoT-based SCADA systems security: a review of the state of the art and future challenges. IEEE Access 4:1375–1384
4. Sudo H, Jimbo M, Nuida K, Shimizu K (2018) "Secure Wavelet Matrix: Alphabet-Friendly Privacy-Preserving String Search for Bioinformatics," IEEE/ACM Trans Comput Biol Bioinformatics 16(5):1675–1684
5. Kumar DVNS, Thilagam PS (2019) Approaches and challenges of privacy preserving search over encrypted data. Inf Syst 81:63–81
6. Wang H. X, Zhou L, Zhang W, Liu S (2013) "Watermarking-based perceptual hashing search over encrypted speech," in Proc. International Workshop on Digital Watermarking (IWDW), Springer, Berlin, Heidelberg, pp 423–434
7. He SF, Zhao H (2017) A retrieval algorithm of encrypted speech based on syllable-level perceptual hashing. Comput Sci Inf Syst 14(3):703–718
8. Zhang QY, Zhou L, Zhang T et al (2019) A retrieval algorithm of encrypted speech based on short-term cross-correlation and perceptual hashing. Multimedia Tools Appl 78(13):17825–17846
9. Wu, D, Dai Q et al (2019) "Deep incremental hashing network for efficient image retrieval," in Proc. the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Lang Beach, California, USA. pp 9069–9077
10. Ma XQ, Yu CC, Chen XX et al (2019) Large-scale person reidentification based on deep hash learning. Entropy 21(5):449–463
11. Song W, Wang B, Wang Q, Peng Z, Lou W, Cui Y (2017) A privacy-preserved full-text retrieval algorithm over encrypted data for cloud storage applications. J Parallel Distributed Computing 99:14–27 (Elsevier)
12. Yang LT, Huang G, Feng J, Li Xu (2017) Parallel GNFS algorithm integrated with parallel block Wiedemann algorithm for RSA security in cloud computing. Inf Sci 387:254–265
13. ChrisMitchell, J (2016) "On the Security of 2-Key Triple DES". IEEE Transact Inform Theory 62(11):6260–6267
14. Tang H, Sun QT, Yang X, Long L (2018) "A Network Coding and DES Based Dynamic Encryption Scheme for Moving Target Defense". IEEE Access 6:26059–26068
15. Yu ZM, Gao CZ, Jing zj, Gupta BB, Cai QR (2018) "A Practical Public Key Encryption Scheme Based on Learning Parity with Noise,". IEEE Access 6:31918–31923
16. Namasudra S, Devi D, Kadry S, Sundarasekar R, Shanthini A (2020) Towards DNA based data security in the cloud computing environment. Comput Commun 15(1):539–547
17. Tyagi, M, Manoria, M & Mishra, B (2019), 'A Framework for Data Storage Security with Efficient Computing in Cloud', Int Conference Advances in Intelligent Systems and Computing, 870:109–116. https://doi.org/10.1007/978-981-13-2673-8_13
18. Tieyu Zhao AN, Qiwen Ran A, Lin Yuan AB, Yingying Chi C, Jing Ma A (2016) 'Information verification cryptosystem using one-time keys based on double random phase encoding and public-key cryptography.' Optics Lasers Eng 83:48–58
19. A. Aieh, A. Sen, S.R. Dash, S. Dehuri, (2015) Deoxyribonucleic Acid (DNA) for a Shared Secret Key Cryptosystem with Diffie Hellman Key Sharing Technique. https://doi.org/10.1109/C3IT.2015.7060130
20. Cimi Thomas M, Sheeja S (2017) Secure symmetric encryption scheme using genetic algorithm. Int J Appl Eng Res 12(21):10828–10833
21. Asaduzzaman S, Ahmed MR, Rehana H, Chakraborty S, Islam MS, Bhuiyan T (2021) Machine learning to reveal an astute risk predictive framework for Gynecologic Cancer and its impact on women psychology: Bangladeshi perspective. BMC Bioinf 22(1):1–17
22. Akter T, Satu MS, Khan MI, Ali MH, Uddin S, Lio P, Quinn JM, Moni MA (2019) Machine learning-based models for early stage detection of autism spectrum disorders. IEEE Access 7:166509–166527
23. M. Sohal, S. Sharma, (2018) BDNA-A DNA inspired symmetric key cryptographic technique to secure cloud computing, J. King Saud Univ. - Comput. Inf. Sci. https://doi.org/10.1016/j.jksuci.2018.09.024
24. M. Tahir, M. Sardaraz, Z. Mehmood, S. Muhammad, (2020) CryptoGA: a cryptosystem based on genetic algorithm for cloud data security, Cluster Comput. 4 https://doi.org/10.1007/s10586-020-03157-4
25. S.K. Pujari, G. Bhattacharjee, S. Bhoi, (2018) A Hybridized Model for Image Encryption through Genetic Algorithm and DNA Sequence. https://doi.org/10.1016/j.procs.2017.12.023
26. C. Chunka, R.S. Goswami, S. Banerjee, (2019) A Novel Approach to Generate Symmetric Key in Cryptography Using Genetic Algorithm (GA). https://doi.org/10.1007/978-981-13-1951-8_64
27. S. Kalsi, H. Kaur, V. Chang, (2018) DNA cryptography and deep learning using genetic algorithm with NW algorithm for key generation, J. Med. Syst. https://doi.org/10.1007/s10916-017-0851-z
28. M. Thangavel, P. Varalakshmi, (2018) Enhanced DNA and elgamal cryptosystem for secure data storage and retrieval in cloud, Cluster Comput. 21 (2). https://doi.org/10.1007/s10586-017-1368-4
29. Narendren S, Yathish YB, C.M. B (2018) A cryptosystem using two layers of security - DNA and RSA cryptography. J Pure Appl Math 119(7):217–224
30. Shang L, Guo D, Ji Y, Li Q (2021) Discovering unknown advanced persistent threat using shared features mined by neural networks. Comput Netw 189:107937
31. Li L, Gu T, Chang L, Xu Z, Liu Y, Qian J (2017) A ciphertext-policy attribute-based encryption based on an ordered binary decision diagram. IEEE Access 5:1137–1145
32. Vempaty Prashanthi, Abdul-Rasheed Akeji Alhassan Alolo, S. Velmurugan (2022) "Chaotic Salp Swarm Optimization-Based Energy-Aware VMP Technique for Cloud Data Centers",. Comput Intell Neurosci 2022:9. https://doi.org/10.1155/2022/4343476. (Article ID 4343476)
33. Ezhumalai P et al (2023) Improved wild horse optimization with levy flight algorithm for effective task scheduling in cloud computing. J Cloud Comp 12:24. https://doi.org/10.1186/s13677-023-00401-1
34. Prakash M, Baburaj E (2023) An Efficient Hybrid Job Scheduling Optimization (EHJSO) approach to enhance resource search using Cuckoo and Grey Wolf Job Optimization for cloud environment. PLOS One 18(3):e0282600. https://doi.org/10.1371/journal.pone.0282600
35. Hatami-Marbini A, Kangi F (2017) An extension of fuzzy TOPSIS for a group decision making with an Application to Tehran stock exchange. Appl Soft Comput 52:1084–1097

Publisher's Note