ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20240920002000
Original scientific paper

Enhanced Data Protection and Application Delivery in Cloud Computing Using Intelligent Optimization and Encryption

SATHISH D.*, K. R. VALLUVAN, S. JABEEN BEGUM

Abstract: Cloud computing has become a crucial component of modern IT infrastructure, offering scalability and cost-effectiveness. However, ensuring the security of data and applications in cloud environments remains a significant challenge. To enhance cloud computing security, we propose a novel design that optimizes application delivery using an Intelligent Deer Hunting Optimized Adjustable Long Short-Term Memory (IDAHO-ALSTM) model combined with Triple Data Encryption Standard (3DES) encryption. The IDAHO-ALSTM model adapts resource allocation and routing decisions by learning from historical data and user patterns, thereby maximizing application delivery. For robust security, 3DES is employed for data encryption and decryption, ensuring data integrity and confidentiality. Experimental results demonstrate that the integration of IDAHO-ALSTM and 3DES not only maintains superior security but also enhances the performance of cloud applications. The IDAHO-ALSTM model reduces latency and minimizes resource wastage by dynamically adjusting to changing workloads. A comparative performance analysis indicates that the proposed method outperforms traditional approaches, achieving an Accuracy of 98.2%, Precision of 95%, *F*-score of 95.6%, Recall of 96.7%, Encryption time of 4 seconds, and Decryption time of 2.6 seconds. These findings suggest that this hybrid solution is suitable for various cloud computing scenarios, providing strong data protection and optimal application delivery. This study contributes to ongoing efforts to enhance the capabilities of cloud computing services concerning application delivery and security. In real-world applications IDAHO-ALSTM model is capable of forecasting possible health problems or identifying unusual patterns in time-series data (such as heart rate, glucose levels, etc.). ALSTM is especially effective in understanding long-term relationships and trends in time-series data, which is crucial for healthcare predictions.

Keywords: application optimization; cloud computing; data security; encryption; long short-term memory (LSTM)

1 INTRODUCTION

Cloud computing has become a disruptive force in the changing data management and technology landscape, providing enterprises of all sizes with never-before-seen levels of scalability, flexibility and affordability [1-3].The traditional on-premises data centers have been redefined by cloud computing, which offers a variety of service models such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [4]. There are many benefits, including lower capital costs, more teamwork and faster reaction to market shifts for firms. There are hazards associated with using cloud services, so it is crucial for businesses to give security top priority in their cloud settings [5].

Data security is one of the main issues with the cloud. Modern businesses rely heavily on data and when that data is housed in a third-party cloud infrastructure, maintaining its availability, security and integrity becomes a challenging task. Modern security measures are invested by cloud providers, but the shared responsibility model requires that the customer bears the final duty for data protection [6]. This implies that companies need to be proactive in protecting their data when it is in transit and at rest. In addition, cloud environments are vulnerable to a variety of security risks, such as Distributed Denial of Service (DDoS) assaults and data breaches [7]. The dynamic nature of cloud services involve the quick provisioning and de-provisioning of resources, which makes it more difficult to manage security. Furthermore, if security flaws are not addressed, the very flexibility and ease that cloud computing provides could unintentionally lead to security breaches [8]. Another essential element of cloud environment security is identity and access management. Malicious activity, data loss and security breaches can result from unauthorized access to cloud resources [9]. There are different regulations in different areas and sectors when it comes to data security and privacy. To avoid legal repercussions, organizations must comprehend and abide by these requirements [10]. Although many cloud providers provide tools and certifications for compliance, it is the customer's obligation to make sure that their cloud deployment complies with these regulations [11]. Novel security solutions have emerged in response to the intricacy and magnitude of cloud systems [12]. We propose a novel design that uses intelligent deer hunting optimized adjustable Long Short Term Memory (IDAHO-ALSTM) to optimize application delivery.

2 RELATED WORKS

The study [13] enhanced cloud security by using encrypted Artificial Neural Networks (ANN) for voice and speech detection. For both privacy and accuracy, the research trains neural networks on encrypted data by employing Matrix Operation-based Randomization and Encipherment (MORE) that was based on a Fully Homomorphic Encryption (FHE). Scalability and real-time processing can be hindered by the computational cost of FHE. The research [14] presented a novel hybrid method that seeks to enhance the security of cloud data through the utilization of an encrypting technique. The research integrated homographic encryption and blowfish encrypting to improve the level of security in cloud computing. The research [15] created a block-chain architecture for provenance, preservation and evidence gathering in IaaS cloud computing. The suggested system included user registration, authentication, data encryption, data storage, activity tracking for users and controller data mining. They have implemented the most recent method that yields optimal outcomes with minimal computing overhead. The article [16] offered a more robust security framework to protect cloud users' data in the cloud setting. This new security architecture consists of techniques for digital signatures, encryption, decryption and access control. Here, a novel key generation method based on elliptic curve cryptography was put forth to provide secure

keys. The study [17] aimed to investigate the implementation of requirement-oriented health information security through the utilization of the Modular Encryption Standard (MES) inside a layered modelling framework for security measures.

3 METHODOLOGY

The IDAHO-ALSTM model optimizes application delivery by adjusting resource allocation and routing decisions based on historical data and user behavior. To ensure robust security protocols for safeguarding sensitive data, the Triple Data Encryption Standard (3DES) algorithm is employed for the encryption and decryption processes. Fig. 1 displays the framework for enhancing security in a cloud environment.

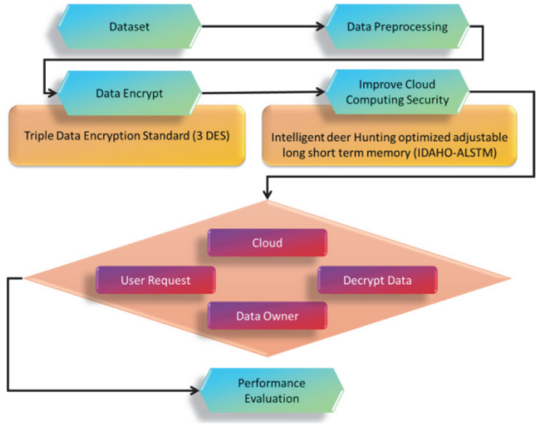

Figure 1 Overall framework for enhancing security in cloud environment

3.1 Dataset

The "Network Security Layer-Knowledge Discovery Database (NSL-KDD)" was created utilising the KDD99 dataset. The dataset presented here represents an extensive compilation of TCP/IP connections obtained from the intrusion detection system assessment data set. The NSL-KDD dataset is composed of TCP/IP connection records, where each record consists of 41 attributes that provide a description of the connection. Furthermore, a property exists that categorizes the connection into five unique classes, encompassing the "normal" class as well as four specific kinds of assaults. The assault categories listed above are presented in the preceding section.

DOS: The attacker's main goal is to overload the system, server or network that they are targeting to prevent services from being provided. The system assaults outlined above result in the immediate consequence of obstructing the flow of network traffic.

U2L: The objective of this category of attacks is to elevate privileges from a standard user account to that of the system administrator (Root) by leveraging vulnerabilities.

R2L: The attacker, operating from a remote system, transmits packets to a specific machine within the target network subsequent to exploiting weaknesses present in such machine.

PROBE: The attacker of this class initiates a first investigation phase, sometimes referred to as a scan that includes scanning each IP port to gather information about the victim's system, such as the operating system, network topology and implemented security measures.

3.2 Data Pre-Processing

The procedure of data pre-processing for an attack dataset encompasses the necessary steps to address various data quality issues, such as missing values, duplicate records and inconsistencies.

3.2.1 Data Cleaning

Due to the nature of attack data, there are unique concerns while cleaning an attack dataset. The cleaning procedure takes care of data drift, time synchronization problems and missing or incorrect sensor readings. Missing data can be handled using interpolation or imputation techniques as well as outliers can be found and dealt by using statistical thresholds or machine learning algorithms.

3.2.2 Min-Max Normalization

The process of min-max normalization involves transforming numerical data to a specific range through rescaling. This process is denoted as feature scaling or data normalization. The objective of normalization is to standardize the attributes or variables in a dataset in order to facilitate their comparison or analysis.

$$A_{\text{new}} = \frac{A - \min(A)}{\max(A) - \min(A)} \tag{1}$$

A_{new} = New value derived from the outcomes of normalization.
A = Former value.
$\max(A)$ = The dataset's maximum value.
$\min(A)$ = The dataset's minimum value.

3.3 Triple Data Encryption Standard (3DES)

Using a three-stage encryption technique, Triple Data Encryption Standard (3DES) improves security in cloud environments and makes it extremely resistant to attackers. To secure data in cloud-based services and apps, it offers strong security for sensitive data and guarantees confidentiality during data transfer and storage.

The selected data is subjected to input encrypted using the 3DES method. Frequently used open method of encryption, 3DES provides 114 bits of encrypted quality. Symmetric keys are used by the 3DES to read, write and transmit data. The 3DES technique is the most efficient one since it produces keys symmetrically and offers superior security. The information is preserved in enormous quantities after the procedure of encryption. Eq. (2) provides a general definition of the 3DES approach.

$$F^1 = F^3 = F,\ F^2 = C \tag{2}$$

where C is the equivalent of decryption, F^1 is the one encrypting feature in 3DES, F^2 is the dual encrypting feature in 3DES and F^3 is the feature of ternary encrypting in 3DES. The DES-3EES algorithm employs three distinct types of keys to implement the three DES. For each of its three operations encryption, decryption and verification, DES uses a different key. Different keys are used by the DES-EEE2 and DES-EDES2 algorithms to facilitate secondary decryption processes. The encryption along with decryption of the DES for a given input J is performed using the key 1, denoted as F_1 (J) and C_1(J) respectively. The 3DES encrypting and decrypting operations are a composite operation that combines the encrypting as well as decrypting processes of DES. These operations are employed as described below. The encrypting process of 3DES involves converting a 64-bit input block, denoted as block J , into a 64-bit output block and referred to as block 0. This conversion is represented by Eq. (3). The decrypting procedure of 3DES involves reversing the encrypting process to recover the original unencrypted from the encrypted text. The initial block of "64 bits is transformed into block 0 of 64 bits", as represented by Eq. (4): There exist three established methods for employing the amalgamation of subkeys inside the framework of encrypting and decrypting using 3DES. These methods shall be explained as follows:

$$P = F_{l_3}\left(C_{l2}\left(F_{l_1}\left(J\right)\right)\right) \tag{3}$$

$$P = C_{l1}\left(F_{l_2}\left(C_{l3}\left(J\right)\right)\right) \tag{4}$$

The use of three subkeys is the crucial option l, denoted as l_1, l_2 and l_3. These subkeys are treated as distinct keys and can be combined in various ways, such as in the case of 3K3DES. The important option 2 to examine is the utilization of three subkeys, namely l_1, l_2 and l_3. These subkeys can have independent or distinct combinations, with l_1 and l_2 having various combinations, while l_1 and l_3 have a comparable combination, as exemplified by the (2K3DES) scheme. One of the key options, referred to as Key Option 3, involves the utilization of three subkeys denoted as l_1, l_2 and l_3. These subkeys are characterized by having similar combinations, as represented in Eq. (5).

$$l_1 = l_2 = l_3 \tag{5}$$

In addition to the three primary alternatives for deploying the subkeys, the initial method is deemed superior due to the fact that the three subkeys encompass distinct combinations that possess an effective "key length of 168 bits". Because 3DES procedures make use of compatible keying options, they are a reliable substitute for single DES. In particular, the 3DES operation is utilized in the decryption process of a single DES to produce encrypted plaintexts. Using a single 3DES and the proper DES technique for decryption, the encrypted plaintexts are generated. Choosing 3DES (Triple Data Encryption Standard) over more contemporary encryption methods like AES (Advanced Encryption Standard) can be justified in specific contexts, despite AES being generally considered more secure and efficient. 3DES is still in use in older systems and applications, particularly in legacy financial services infrastructure and some government systems. These systems may not be easily updated or may require backward compatibility with existing encryption protocols. For such systems, 3DES may be preferred due to its familiarity and the extensive support that exists for it in older technologies.

3.4 Intelligent Deer Hunting Optimization Algorithm (IDHOA)

A metaheuristic optimization method called the Intelligent Deer Hunting Optimization method (IDHOA) was developed with inspiration from deer hunting behaviour. It is intended to simulate the communication and foraging habits of deer in the wild in order to solve challenging optimization problems. IDHOA is utilized to improve cloud security. This algorithm adjusts to changing threats, decreasing weaknesses and strengthening the overall defense system.

The deer's unique capabilities enable it to avoid predation, resulting in its successful avoidance of harvesting attempts by animals. This procedure can be determined in the following manner:

$$Y = \left[y_1,\ y_2, \ldots, y_n\right],\ 1 < j \le n \tag{6}$$

In the present instance, the variable "n" denotes the size of the hunting populations, whereas "Y" refers to the overall population. The method initiates by initializing the essential elements of the algorithm, including the wind direction and the angle of the deer's positioning. The technique requires that the search space is circular in shape. Hence, the wind angle can be determined based on the dimension of the circle.

$$\theta_j = 2\pi\lambda \tag{7}$$

$$x_j = \pi + \theta \tag{8}$$

In this instance, the symbol λ denotes an arbitrary number that falls inside the interval [0, 1], whereas the variable j denotes this particular repetition. Moreover, the location angle of the deer can be expressed as follows: where θ is the angle of the wind. The application of position development, which includes the successor position W^T representing the subsequent position of the hunter and the leader position W^K denoting the the hunter's initial, perfect location was utilized by reality.
The leadership location:

Following the initialization and implementation of the initial iteration, hunters make an effort to acquire the optimal location through the process of updating. Therefore, the behavior of encircling can be described as follows:

$$W_{j+1} = W^K - Z \times T_x \times \left|K \times W^K - W_j\right| \tag{9}$$

The current positions W_j and the next ones W_{j+1} are denoted in this equation. The variable T_x represents a random value that is determined by the wind speed, which falls in the range of 0 to 2. The coefficient vectors K and Z can be expressed in the following manner:

$$K = 2 \times \tau \tag{10}$$

$$Z = 0.25 \times \log\left(J + \frac{1}{J_{\max}} \right) \beta \tag{11}$$

In this scenario, the symbol τ denotes a generated value within the range of 0 and 1. The variable $J_{\max}$ represents the maximum number of iterations, whereas β is a generated parameter within the interval of −1 and 1. The updated position is depicted in Fig. 2. The hunter's initial location is shown by the coordinates (W, Y) in the picture, and it is adjusted based on the location of the prey. The process of location update will persist until the optimal position (W^*, Y^*) is attained, as determined by the variables K and Z. The hunters attempt to advance towards the path of the leader. After the leader experiences a failed activity, the hunters will maintain their present location. The adjustment of the location is determined by Eq. (10) under the condition of $T_x < 1$. In other words, the hunters are able to move in various directions without considering the position angle as specified in Eq. (9) and Eq. (10).

The concept of position angle:

The improvement of the space for solution efficiency can be achieved by integrating the calculation of the location angle into the modification method. The methodology for determining the graphical angle for the deer is as the following:

$$b_j = \frac{1}{8} \times \pi \times \lambda \tag{12}$$

By considering the disparity between the visual angles of the target with the angle of the wind, it is possible to establish a variable that aids in the upgrading of the position angle.

$$c_j = \theta_j - b_j \tag{13}$$

The wind angle, denoted as θ, is a parameter that specifies the direction of the phenomenon of winds. The measurement of the position angle can be altered through updates.

$$x_{j+1} = x_j + c_j \tag{14}$$

Utilizing the acquired location angle, the new location is attained in the following manner:

$$W_{j+1} = W^k - T_x \times \left| \cos\left(x_{j+1} \right) \times W^K - W_j \right| \tag{15}$$

The deer is unable to perceive the hunter due to its location outside the deer's field of perception.

$$W_{j+1} = W^T - Z \times T_x \times \left| T \times W^K - W_j \right| \tag{16}$$

The variable W^T denotes the future location of the hunters relative to the current people. Using the best answer, the program modifies the hunters' position. The best result is obtained from this process when the value of $|K| \geq 1$. The selection of the hunter will be conducted using a random process, given that the number of participants $|K| < 1$. One limitation of the DHOA is its tendency to prematurely converge. In the subsequent section, a novel methodology is introduced to mitigate this limitation. One notable limitation of the DHOA is its susceptibility to early convergence, which can be attributed to the presence of numerous random parameters (λ, T_x, τ, and β). In order to address this limitation, the present work utilizes a methodology to adjust the fundamental parameters of the deer hunting optimization algorithm with the aim of enhancing its rate of convergence. The primary factors influencing the optimization of deer hunting algorithms are the parameters (λ, T_x, τ, and β). Chaos theory is a scientific discipline that investigates random processes. The objective of chaos theory is to examine the behavior of complex and sensitive dynamic systems that are susceptible to even minor perturbations. With the previously cited justification in mind, using chaos theory to DHOA can increase generational variation and accelerate convergence. The subsequent section presents a conventional manifestation of chaos theory.

$$\mathbb{DN}_{i+1}^{l} = e\left(\mathbb{DN}_i^{l} \right) \tag{17}$$

In this case, the variable n indicates the dimension of the map, whereas the term $e\left(\mathbb{DN}_i^{l} \right)$ denotes the generating function of the chaotic model. In the Intelligent Deer Hunting Optimization Algorithm (IDHOA), the parameters λ, T_x, τ, and β are represented using a model that depends on the sinusoidal chaotic map.

$$\begin{aligned} \lambda &= b_1 \theta_l^2 \sin\left(\pi \theta_l \right) \\ T_x &= b_2 \theta_l^2 \sin\left(\pi \theta_l \right) \\ \tau &= b_3 \theta_l^2 \sin\left(\pi \theta_l \right) \\ \beta &= b_4 \theta_l^2 \sin\left(\pi \theta_l \right) \\ T_x &= b_2 \theta_l^2 \sin\left(\pi \theta_l \right) \end{aligned} \tag{18}$$

In this context, the variable 1 represents the iteration number, while the condition $\theta_0 \in [0, 1]$, $b_i \in (0, 4]$, $i = 1, 2, 3, 4$ indicates that the value of $0,4$ should be less than 1, 2, 3, 4. Fig. 3 displays the block diagram of the IDHOA.

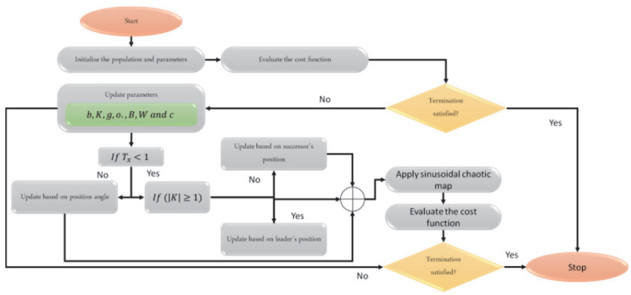

Figure 3 The IDHOA's flow design

3.5 Improved Long Short-Term Memory (ILSTM)

The utilization of Improved Long Short-Term Memory (ILSTM) in cloud systems presents notable advancements in data protection and anomaly detection, therefore enhancing overall security measures and decreasing the potential risks associated with cyber threats. This strategic decision mitigates the issue of overfitting that can arise during the training process. Algorithm 1 presents the pseudocode for the ILSTM. The framework of an ILSTM is illustrated in Fig. 4. The architecture of the model comprises a network that exhibits complete connectivity among its input, hidden and output layers. The data received by the input layer is referred to as "one-dimensional time series data." Two LSTM neural network layers make up the hidden layer.

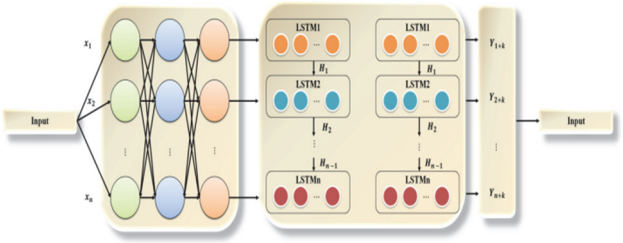

Figure 4 Structure of ILSTM

Algorithm 1: ILSTM pseudocode:

Step 1: Define LSTM cell with peephole connections

Step2: function

$$\text{LSTM\_Cell}(\text{input}, \text{prev\_hidden\_state}, \text{prev\_cell\_state}, \text{weights}, \text{biases}):$$

Input gate

$$i_t = \text{sigmoid}(\text{dot}(\text{input}, \text{weights}['W\_i']) + \text{dot}(\text{prev\_hidden\_state}, \text{weights}['U\_i']) + \text{dot}(\text{prev\_cell\_state}, \text{weights}['C\_i']) + \text{biases}['b\_i'])$$

Forget gate

$$f_t = \text{sigmoid}(\text{dot}(\text{input}, \text{weights}['W\_f']) \text{dot}(\text{prev\_hidden\_state}, \text{weights}['U\_f']) + \text{dot}(\text{prev\_cell\_state}, \text{weights}['C\_f']) + \text{biases}['b\_f'])$$

Cell state update

$$c_{tilde} = \tanh(\text{dot}(\text{input}, \text{weights}['W\_c']) + \text{dot}(\text{prev\_hidden\_state}, \text{weights}['U\_c']) + \text{biases}['b\_c'])$$

$$c_t = f_t * \text{prev\_cell\_state} + i_t * c_{tilde}$$

Output gate

$$o_t = \text{sigmoid}(\text{dot}(\text{input}, \text{weights}['W\_o']) \text{dot}(\text{prev\_hidden\_state}, \text{weights}['U\_o']) + \text{dot}(c\_t, \text{weights}['C\_o']) + \text{biases}['b\_o'])$$

Hidden state

$$h_t = o_t * \tanh(c_t)$$

return h_t, c_t

Step 3: Improved LSTM model with layer normalization and residual connections

Step4: function

$$\text{Improved\_LSTM}(\text{input\_sequence}, \text{weights}, \text{biases}):$$

$$\text{hidden\_state} = 0$$

$$\text{cell\_state} = 0$$

for input in input_sequence:

Layer normalization before feeding into the LSTM cell

$$\text{input\_norm} = \text{layer\_norm}(\text{input})$$

LSTM cell calculation

$$h\_t, c\_t = \text{LSTM\_Cell}(\text{input\_norm}, \text{hidden\_state}, \text{cell\_state}, \text{weights}, \text{biases})$$

Residual connection: adding the input to the LSTM output

$$\text{residual\_h\_t} = h\_t + \text{input}$$

Update the hidden state and cell state for the next iteration

$$\text{hidden\_state} = \text{residual\_h\_t}$$

$$\text{cell\_state} = c\_t$$

return hidden_state

Example usage

$$\text{input\_sequence} = [\text{input\_1}, \text{input\_2}, ..., \text{input\_n}]$$

List of input vectors

weights = {...} # LSTM model weights

biases = {...} # LSTM model biases

output=

$$\text{Improved\_LSTM}(\text{input\_sequence}, \text{weights}, \text{biases})$$

3.6 Intelligent Deer Hunting Optimization Algorithm (IDHOA)

The Intelligent Deer Hunting Optimization Algorithm (IDHOA) is an optimization technique that generates motivation from the hunting behavior of deer that hunt for food sources in nature. In order to enhance security measures, it is possible to include an Improved Long Short-Term Memory (ILSTM) model. The Improved Long Short-Term Memory (ILSTM) model boosts the performance of the conventional LSTM by augmenting its capacity to capture long-range correlations in data. This

improvement renders ILSTM more proficient in the tasks of anomaly identification and threat prediction.

The combination of the IDHOA and ILSTM techniques produces the IDHOA-ILSTM framework, presenting a novel and advanced methodology for enhancing cloud security. When used in large-scale cloud environments, the IDAHO-ALSTM system shows a number of scalability advantages and disadvantages, particularly when handling a variety of workloads. Elastic scaling, which the cloud environment offers, is essential to the ALSTM's functionality for managing massive data sets. In order to ensure effective use of compute and storage resources, cloud systems (such as AWS and Azure) can dynamically assign resources based on workload demands. This is especially crucial in situations where the system encounters surges in incoming data or varying processing requirements.

4 PERFORMANCE EVALUATION

Precision - It is defined as the ratio of true positive predictions to the overall number of positive predictions the model makes. It measures how successful a model is at making positive predictions.

$$\Pr ecision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{19}$$

Recall - In the context of binary classification, recall is a metric used in machine learning and statistics to assess the way a classification model is performing (when there are two classes: positive and negative). It measures a model's proficiency in accurately locating or capturing each and every pertinent occurrence of the positive class present.

$$\mathrm{Re}call = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{20}$$

F-Score The *F*-Score is a frequently used metric in machine learning and statistics that evaluates how well a binary classification model separates data into positive (usually represented by the letter "1") and negative (often represented by the letter "0") classes.

$$F - Score = 2 \times \frac{\Pr ecision \times \mathrm{Re}call}{\Pr ecision + \mathrm{Re}call} \tag{21}$$

Accuracy - The evaluation of classification models, particularly in the fields of machine learning and data science, frequently depends on the accuracy statistic. The metric quantifies the ratio of accurately classified examples to the total number of instances.

$$Accuracy = \frac{Number\ of\ correct\ \Pr edictions}{Total\ Number\ of\ \Pr ediction} \tag{22}$$

Evaluation of Encryption time:

Tab. 1 and Fig. 5 present a comparison examination of encryption time between the IDAHO-ALSTM methodology and current methods. The results of the research clearly show that the recommended strategy performed better than the ones that were already in use, the suggested method's and the current method's encryption times, respectively. RSA, AES, and DNA had values of 13.6 s, 13.8 s, and 8.5 s, respectively, when tested against the existing methods; however, the technique that was recommended (IDAHO-ALSTM) has a value of 4 s.

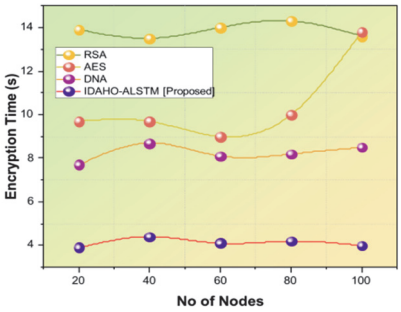

Figure 5 Encryption Time

Table 1 Outcomes of Encryption Time

No of Nodes	Encryption Time			
	RSA	AES	DNA	Proposed
20	13.9	9.7	7.7	3.9
40	13.5	9.7	8.7	4.4
60	14	9	8.1	4.1
80	14.3	10	8.2	4.2
100	13.6	13.8	8.5	4

Decryption Time Analysis:

Tab. 2 and Fig. 6 illustrate the comparative analysis of the decryption time between the IDAHO-ALSTM methods and other existing methods. The data presented in this study indicate that the proposed strategy exhibited superior performance compared to the existing techniques across all evaluated measures. The suggested method's and the present method's decryption times, respectively. RSA, AES, and DNA had values of 12.5 s, 9.4 s, and 7.2 s, respectively, when tested against the existing methods; however, the technique that was recommended (IDAHO-ALSTM) has a value of 2.6 s.

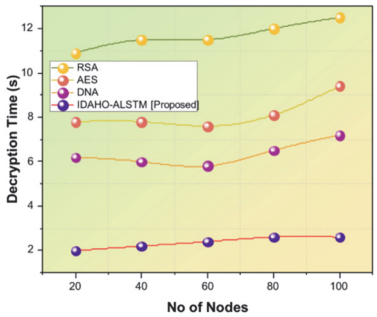

Figure 6 Decryption Time

Table 2 Outcomes of Decryption Time

No of Nodes	Decryption Time			
	RSA	AES	DNA	Proposed
20	10.9	7.8	6.2	2
40	11.5	7.8	6	2.2
60	11.5	7.6	5.8	2.4
80	12	8.1	6.5	2.6
100	12.5	9.4	7.2	2.6

Accuracy: The comparison of the IDAHO-ALSTM method's accuracy with that of other previous approaches is presented in Fig. 7 and Tab. 3. Based on the graphical

representation, it can be observed that the utilisation of the cloud technique has resulted in enhanced performance and increased accuracy. The accuracy of the proposed method and the existing method, respectively. RSA, AES, and DNA had values of 82%, 78%, and 74%, respectively, when tested against the existing methods; however, the technique that was recommended (IDAHO-ALSTM) has a value of 98.2%.

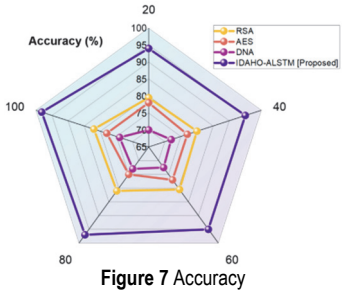

Figure 7 Accuracy

Table 3 Outcomes of Accuracy

No of Nodes	Accuracy / %			
	RSA	AES	DNA	Proposed
20	79.5	78	70	94
40	80	77	72	95
60	80.5	77	72.5	95
80	81	75	73	97
100	82	78	74	98.2

Precision: Fig. 8 and Tab. 4 present a comprehensive evaluation of the IDAHO-ALSTM approach in comparison to existing approaches, focusing on precision. The graph illustrates the enhanced performance and precision achieved through the utilisation of the cloud approach. The precision of the proposed method and the existing method, respectively. RSA, AES, and DNA had values of 77%, 81%, and 92%, respectively, when tested against the existing methods; however, the technique that was recommended (IDAHO-ALSTM) has a value of 95%.

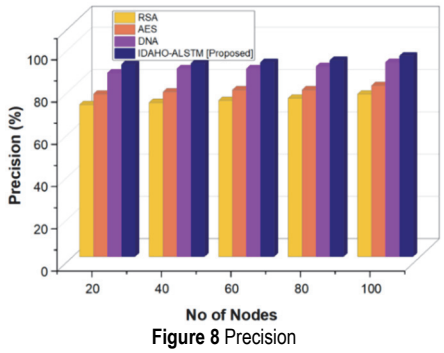

Figure 8 Precision

Table 4 Outcomes of Precision

No of Nodes	Precision / %			
	RSA	AES	DNA	Proposed
20	72	77	87	91
40	73	78	89	91.5
60	74	79	89	92
80	75	79	90	93
100	77	81	92	95

Recall: Fig. 9 and Tab. 5 present a comparative analysis of the recall performance between the IDAHO-ALSTM technique and alternative strategies. The graph depicts that the utilisation of the cloud technique resulted in an enhancement of recall performance. The recall of the proposed method and the existing method, respectively. RSA, AES, and DNA had values of 88%, 92%, and 90%, respectively, when tested against the existing methods; however, the technique that was recommended (IDAHO-ALSTM) has a value of 96.7%.

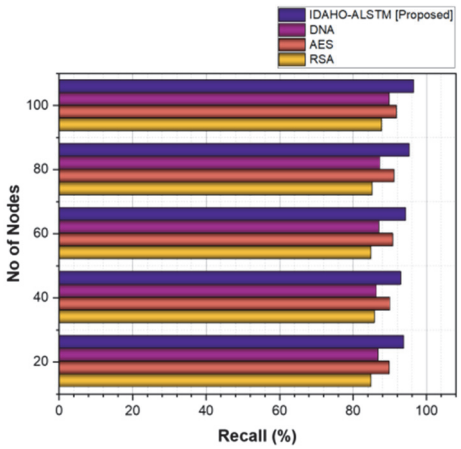

Figure 9 Recall

Table 5 Outcomes of Recall

No of Nodes	Recall / %			
	RSA	AES	DNA	Proposed
20	85	90	87	94
40	86	90.2	86.5	93.2
60	85	91	87.2	94.5
80	85.4	91.3	87.5	95.5
100	88	92	90	96.7

F-Score: Fig. 10 and Tab. 6 provide a comparison of the IDAHO-ALSTM methodology with various previous methods. The f-score measure in the graph indicates that improved performance was attained through the application of the cloud method. The f-score of the proposed method and the existing method, respectively. RSA, AES, and DNA had values of 89%, 85%, and 81.5%, respectively, when tested against the existing methods; however, the technique that was recommended (IDAHO-ALSTM) has a value of 95.6%.

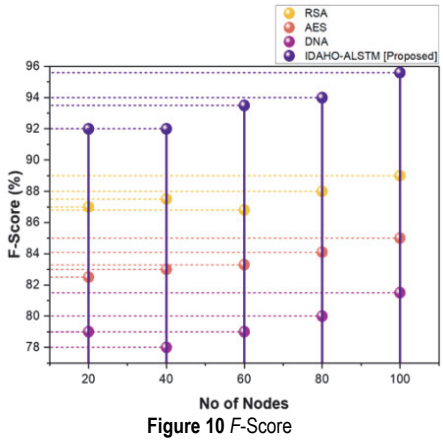

Figure 10 *F*-Score

Table 6 Outcomes of *F*-Score

No of Nodes	*F*-Score / %			
	RSA	AES	DNA	Proposed
20	87	82.5	79	92
40	87.5	83	78	92
60	86.8	83.3	79	93.5
80	88	84.1	80	94
100	89	85	81.5	95.6

Table 6 Outcomes of convergence rate is seconds

Number of epochs	PSO	ACO	IDAHO
20	21.7	25.2	11.2
30	21	25.4	12
40	21.2	24.9	12.1
50	21.4	24.5	11.9
60	21	25.7	11.7

The data in Tab. 6 displays the PSO, ACO, and IDAHO optimization algorithms' convergence rates (in seconds) over varying epoch counts. As the number of epochs rises, PSO shows a minor drop in convergence time, going from 21.7 seconds at 20 epochs to 21.4 seconds at 50 epochs, following which the rate stabilizes. This implies that PSO does not gain much from more epochs and reaches a comparatively quick convergence period. The convergence time for ACO, however, increases steadily with the number of epochs, rising from 21 seconds at 20 epochs to 25.7 seconds at 60 epochs. This suggests that ACO slows down as iterations increase, perhaps as a result of the optimization process's complexity.

5 CONCLUSION

The benefits of scalability and cost-effectiveness have made cloud computing an indispensable part of today's information technology architecture. We propose a novel design that uses intelligent deer hunting optimised adjustable Long Short Term Memory (IDAHO-ALSTM) with Triple Data Encryption Standard (3DES) encryption to optimise application delivery and enhance cloud computing security. The outcomes suggest that this hybrid approach can be applied with robust data protection and optimal delivery in a range of cloud computing applications. The computational complexity of IDAHO-ALSTM, which can require a lot of resources, is one of its limitations. In the case of fluctuating traffic, the method could load balancing techniques can be employed to distribute security processing across multiple servers or nodes, ensuring high data throughput does not affect security levels. Future research might concentrate on creating enhanced iterations of IDAHO-ALSTM and 3DES that utilize less computational power and memory. This could include simplifying the model or applying methods like quantization or pruning to streamline ALSTM, along with investigating lightweight encryption methods (such as elliptic curve cryptography) that demand lower processing resources compared to 3DES.

6 REFERENCES

[1] Ding, F.X., Liu, S.F., & Li, X.W. (2023). An innovative framework for sustainable and centralized material procurement management based on a full-domain set theory. *Advances in Production Engineering & Management, 18*(1), 5-18, https://doi.org/10.14743/apem2023.1.453

[2] Bajic, B., Suzic, N., Simeunovic, N., Moraca, S., & Rikalovic, A. (2020). Real-time Data Analytics Edge Computing Application for Industry 4.0: The Mahalanobis-Taguchi Approach. *International Journal of Industrial Engineering and Management, 11*(3), 146–156. https://doi.org/10.24867/IJIEM-2020-3-260

[3] Kim, J., Kim, E., Yang, J., Jeong, J., Kim, H., Hyun, S., Yang, H., Oh, J., Kim, Y., Hares, S., & Dunbar, L. (2020). lbcs: Intent-based cloud services for security applications. *IEEE Communications Magazine, 58*(4), 45-51. https://doi.org/10.1109/MCOM.001.1900476

[4] Mei, Z., Yu, J., Huang, J., Wu, B., Zhao, Z., Zhang, C., & Wu, Z. (2023). Enabling access control for encrypted multi-dimensional data in cloud computing through range search. *Technical Gazette, 30*(6), 1704-1716. https://doi.org/10.17559/TV-20230415000536

[5] Awaysheh, F. M., Aladwan, M. N., Alazab, M., Alawadi, S., Cabaleiro, J. C., & Pena, T. F. (2021). Security by design for big data frameworks over cloud computing. *IEEE Transactions on Engineering Management, 69*(6), 3676-3693. https://doi.org/10.1109/TEM.2020.3045661

[6] Hu, R. & Huang, P. (2024). Enhanced secure storage of big data at rest with improved ECC and Paillier homomorphic encryption algorithms. *Technical Gazette, 31*(2), 510-517. https://doi.org/10.17559/TV-20230618000745

[7] Ul Mehmood, M., Ulasyar, A., Khattak, A., Imran, K., ShehZad, H., & Nisar, S. (2020). Cloud-based IoT solution for fault detection and localization in power distribution systems. *Energies, 13*(11), 2686. https://doi.org/10.3390/en13112686

[8] Abbasi, M., Yaghoobikia, M., Rafiee, M., Jolfaei, A., & Khosravi, M. R. (2020). Energy-efficient workload allocation in fog-cloud based services of intelligent transportation systems using a learning classifier system. *IET Intelligent Transport Systems, 14*(11), 1484-1490. https://doi.org/10.1049/iet-its.2019.0783

[9] Aslan, Ö., Ozkan-Okay, M., & Gupta, D. (2021). A review of cloud-based malware detection system: Opportunities, advances and challenges. *European Journal of Engineering and Technology Research, 6*(3), 1-8. https://doi.org/10.24018/ejeng.2021.6.3.2372

[10] Valdez-De-Leon, O. (2019). How to develop a digital ecosystem: A practical framework. *Technology Innovation Management Review, 9*(8), 1-8. https://doi.org/10.22215/timreview/1260

[11] Kopalle, P. K., Kumar, V., & Subramaniam, M. (2020). How legacy firms can embrace the digital ecosystem via digital customer orientation. *Journal of the Academy of Marketing Science, 48*, 114-131. https://doi.org/10.1007/s11747-019-00694-2

[12] Alsamhi, S. H., Ma, O., Ansari, M. S., & Gupta, S. K. (2019). Collaboration of drone and internet of public safety things in smart cities: An overview of QoS and network performance optimization. *Drones, 3*(1), 13. https://doi.org/10.3390/drones3010013

[13] Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: A review. *Journal of Big Data, 6*(1), 12. https://doi.org/10.1186/s40537-019-0268-2

[14] Sana, M. U., Li, Z., Javaid, F., Liaqat, H. B., & Ali, M. U. (2021). Enhanced security in cloud computing using neural network and encryption. *IEEE Access, 9*, 145785-145799. https://doi.org/10.1109/ACCESS.2021.3115638

[15] Sajay, K. R., Babu, S. S., & Vijayalakshmi, Y. (2019). Enhancing the security of cloud data using hybrid encryption algorithm. *Journal of Ambient Intelligence and Humanized Computing, 10*(1), 1-10. https://doi.org/10.1007/s12652-019-01403-1

[16] Velmurugadass, P., Dhanasekaran, S., Anand, S. S., & Vasudevan, V. (2021). Enhancing Blockchain security in cloud computing with IoT environment using ECIES and cryptography hash algorithm. *Materials Today: Proceedings, 37*, 2653-2659. https://doi.org/10.1016/j.matpr.2020.08.519

[17] Prabhu Kavin, B., Ganapathy, S., Kanimozhi, U., & Kannan, A. (2020). An enhanced security framework for secured data storage and communications in cloud using ECC, access

control and LDSA. *Wireless Personal Communications, 115*, 1107-1135. https://doi.org/10.1007/s11277-020-07613-7

Contact information:

Sathish D.
(Corresponding author)
Department of Computer Science and Engineering,
Sri Ramanathan Engineering College,
Tiruppur, Uthukuli Main Road
Nadupatti, Tiruppur, 638 056, Tamil Nadu, India
E-mail: sathishcsephd@gmail.com

K. R. VALLUVAN, PhD, Professor
Department of Electronics and Communication Engineering,
Velalar College of Engineering and Technology,
Thindal, Erode - 638 012, Tamil Nadu
E-mail: valluvankr@velalarengg.ac.in

S. JABEEN BEGUM, PhD, Professor
Department of Computer Science and Engineering,
Velalar College of Engineering and Technology,
Thindal, Erode - 638 012, Tamil Nadu
E-mail: sjabeenbegum@gmail.com