Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Reconnoitering the significance of security using multiple cloud environments for conveyance applications with blowfish algorithm

S Shitharth[1*], Fahad S. Alotaibi[2], Hariprasath Manoharan[3], Adil O. Khadidos[4], Khaled H. Alyoubi[2] and Abdulrhman M. Alshareef[2]

Abstract

In recent years the process of transportation needs a highly effective traffic system in order to monitor all consumer goods as many goods are left out at different locations. To handle such moving cases cloud platform is highly helpful as with respect to geographical location the goods are mapped in correct form. However incorporation of single cloud platform does not provide sufficient amount of storage about all goods thus a multiple cloud platform is introduced in proposed system. As multiple cloud platform is provided the security features of each data base system is also checked and enhanced using encryption keys. Moreover for proper operating conditions of multiple cloud platforms an analytical model is designed that synchronizes necessary data at end system. The defined analytical model focuses on solving multiple objectives that are related to critical energy problems where demand problems are reduced. Further the encryption process is carried out using Improved BlowFish Algorithm (IBFA) by allocating proper resources with decryption keys. To validate the effectiveness of proposed method five scenarios are considered where all scenario outcomes proves to be much higher than existing models by an average of 43%.

Keywords: Security, Multiple cloud platform, Encryption, Transportation, Data storage

Introduction

Since all of the data is kept in the cloud, the development in the field of transportation applications offers strong support for wireless data transfer from diverse locations. To prevent the data of all customers, however, the process of data storage necessitates the activation of specific security elements. Additionally, as transportation applications are used in significant numbers across several nations, it is impossible to send, receive, and store a vast amount of data on a single cloud. As a result, many cloud systems can be combined to create a real-time data interchange platform with the best functionalities. Additionally, the integration of public and private data through several clouds qualifies a platform as a hybrid cloud. When a hybrid cloud platform is used, all uncertainties are eliminated and the delicate parts of the conveyance process are correctly identified. A central data service system is made possible by the fact that even the type of service can be automatically dispersed across all geographical locations in the cloud. Multiple cloud data sets are reported to a central station as part of the central data process, and at this step, security features are strengthened by encrypting and decrypting keys.

Any sort of assault on many clouds is prohibited by the adoption of such key management systems, and in this case, the data is routed through the proper channels. Figure 1 shows the block diagram of several cloud

*Correspondence: shitharths@kdu.edu.et

[1] Department of Computer Science, Kebri Dehar University, Kebri Dehar, Ethiopia
Full list of author information is available at the end of the article

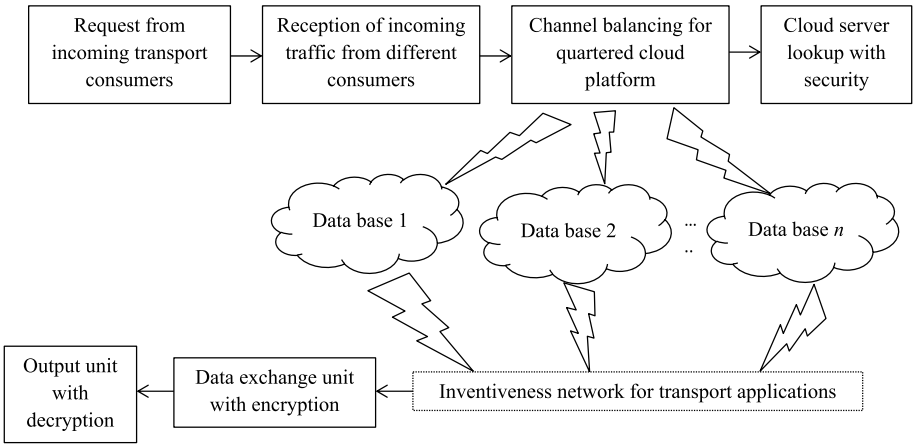

Fig. 1 Block diagram of multiple cloud system for transportation application

storage systems for use in transportation applications. New transport users submit the initial request, and the currently available free space is examined. In order to prevent user duplication when there are many clouds present, users will be admitted if there is any available space on the cloud platform. All data will be encrypted once users have access to the cloud platform, and the platform will be balanced in these circumstances. A lookup operation is carried out by transmitting the data over wireless media by sharing the channel since the data is balanced in the relevant cloud network. A database is built and designated as an inventive network for carrying out transport applications when the data has been shared with n users. The exchange unit will be activated by the aforementioned network and database, and at the very end, the consumer data is decrypted at the output units.

Main contributions

Many of the existing methods that uses IoT as major data application for transportation does not implement security factors where low emission are provided. Thus it is not possible to provide green IoT at all surrounding places where more amount of traffic is present in the system. In addition a generalized system model for green IoT applications is not integrated with other automated algorithms. Even if multiple cloud systems are present the downlink latency is much higher as synchronization of multiple data is not achieved with designed model. Moreover the energy factor remains as a major drawback in existing approach as the multiple cloud platforms are operated under green IoT it is essential to supply more amount of input power to all connected nodes that provides information about data vertices points.

To overcome the above mentioned drawbacks the proposed method is incorporated with blowfish algorithm and the major contributions are as follows,

- Design and examine the effect of green IoT arrangement for transportation applications using multiple cloud data base platform at minimized energy rate.
- To minimize the collision rate of multiple cloud data base system by allocating appropriate resources at low latency periods.
- To provide minimum data weight and high encryption ratio to all cloud system where idle data can be identified at high rapidity rate.

Paper organization

The rest of the article is organized as follows: Sect. 2 provides a basic indulgent on existing approaches that supports implementation of proposed method. Section 3 provides the system design model for both green IoT and multiple cloud platforms with secure data storage for transportation applications. Section 4 integrates the system model in loop based format with step-by-step implementations. Section 5 provides experimental outcomes of designed system model under five different scenarios and at last Sect. 6 concludes the paper with limitations and future scope.

Literature analysis

In order to combine fundamental formulations and techniques, this section gives information on existing models. Modernizing cloud computing systems is impossible without understanding the implementation cases' step-by-step processes. As a result, some of the fundamental implementation tests that have recently been made using

case studies and their behaviour in diverse applications are supplied. Typically, all current practises use a single cloud platform with a lot of storage, but the Internet of Things (IoT) process is completely different in every setting. Given that all issues were resolved in the aforementioned situation, it is referred to as a "green IoT" scenario. [1] makes an energy trade-off by introducing three main goals—learning, processing, and providing high security to entire cloud systems.The cloud platform is introduced with encryption code at all edge computing environments by using three objective scenarios. Although edge performance offers great detection accuracy, real-time applications employing hardware setup do not use these enabling technologies. Additionally, the energy trade-off offered by the application platform is significantly more challenging to perform in adjacent scenarios, so no direct connection is made to the smart detecting system. Fifth generation networks change the entire network design to boost the energy efficiency of IoT processes for various applications [2]. This update results in a singular communication channel being used by all IoT-enabled electric appliances, where all data is gathered and managed in cloud storage systems. However, since the data is continually sent through the same communication route, the cost of using a different channel type rises erratically.

High security methods are used to process a scalable framework network [3] with a high connection matrix where continuous connectivity is ensured. The aforementioned connectivity is processed using the same channel, but the placement of the nodes is arbitrary. No clear information on the output data is obtained because of the increased data transfer to the cloud caused by the nodes' random positions. Even cloud computing systems are pushed through their paces with different application scenarios, such as health care monitoring systems [4], where the performance of the entire network is improved with less computer power. The cost of implementation is lower because there are fewer resources available, but this method also uses random placement techniques. When examining the output units using an event simulator on a simulation platform, it is discovered that the proposed method only delivers the desired performance when two separate cloud parameters are used, while the other cloud parameters are left inactive. Since a dependable storage space must be created for the cloud network's communication tool to process all necessary data, a coded segment is created under various network categories. A time-dimensional matrix is used to create the coded format, and more data centre paths with low probability rates are offered. As the likelihood rates are lower, the network components are distributed more evenly throughout the entire system, boosting the dependability of the cloud storage device. Even if the system is more reliable, because the time allocation approach is based on polynomial distribution, a sequence of time changes can be seen using cloud systems' adaptive nature. In one specific application of IoT in real-time networks, vehicle applications are selected using a route-based technique [6] and demand parameters in cloud systems are examined. Due to the significantly lower demand, it has been noticed that time management tactics are used, leading to the random selection of various routes. The cost of the data collection unit would climb even more if there were many data collection plans existing in the cloud management system.

Additionally, a unique routing technique is used with a cloud hop counting mechanism [7], where many targets are chosen in the system, to monitor precarious cloud energy. Every goal is developed and processed using a different cloud platform. However, one disadvantage of the setup procedure is that cloud storage space is offered for specific goals that are not required if data is delivered at a much slower rate. For cloud support systems, a fog computing model is also taken into account [8]; it is based on the theoretical calculation of complete systems. The observed computational model is replicated in five distinct case studies, and it is discovered that only 25% of the data is passed in the cloud with high security measures, while 75% of the data can be retrieved using public encrypted keys. Additionally, a survey was conducted as a first step to identify the green IoT process using cloud computing techniques, and more changes are noted in this survey [9], as green IoT can only be represented for systems that have low energy for processing multiple applications and is in no way related to cloud management techniques. But only when data processing and administration are done correctly can cloud storage methods always represent the term "green IoT." Additionally, a system must work for processing data with different keys regardless of the application platform being used, allowing for the control of any external assault. With the help of a deep learning algorithm, the researchers [10] have made minor adjustments to the system, which now uses a cloud density method to detect any anomalies. A neural network with hidden cloud units is used in the direct realization of applications as the cloud computing approaches are handled with the best possible selection path.

Due to the existence of direct implementation scenarios, large applications that are determined for the smart development of diverse cities employing transportation indicators can be processed [11]. This intelligent development process needs encrypted keys that can be decoded using segments that are 128 bits long. In contrast, only 128 bits are required in encrypted scenarios, so making several choices for cloud storage security is still pointless.

The measurement of daily activities in powered framework models is also used to calculate individual cloud scores [12–16]. A customer satisfaction framework model using multiple cloud platform is examined [17] where a level agreement is provided for increasing the attention of all customers. During the above mentioned customer satisfaction framework the address of competing customers are provided by approving third party customers and it is provided as reliable framework method. But if third party users are present then external users can able to enter into the encrypted multiple cloud platform thus there is a high possibility of data breach. After keeping track of the cloud's health, it has been discovered that under low point scenarios, resources can be distributed more sparingly while maintaining high cloud operating efficiency. In the proposed method, which is detailed in Sect. 2, all the above indulgent are taken into account when framing the analytical model.

Analytical model

The analytical model that is put forth in this section is used to calculate all the values linked to the operation of transportation apps, where the output fields are automatically stored on a variety of cloud platforms with strict security controls. The earliest stages of formulation are carried out by utilising IoT processes with various wireless sensor installations. It is crucial to assess the uplink and downlink latency prominence of multiple cloud platforms during this installation, which is calculated using Eq. (1) as shown below.

$$l_i = \sum_{i=1}^{n} u_l(i) + d_l(i) + s_l(i) + r_l(i) \tag{1}$$

Where,

u_l, d_l represents uplink and downlink latency of multiple storage clouds

s_l denotes synchronization period of multiple cloud platform

r_l describes allocation of resources to all encrypted cloud storage systems

Equation (1) requires that the cloud system's synchronization values guarantee a high likelihood of controlling the parametric values, where all relevant demands in the transportation application must be met. If more than one demand is made and a collision occurs, the presence of more collisions can be reduced using Eq. (2) as shown below.

$$C_i = \min(1 - \sum_{i=1}^{n} e^{\frac{\rho_i}{z_i}}) \tag{2}$$

Where,

ρ_i indicates the approximation of resources in cloud

z_i represents available transportation cloud parametric values

The number of cloud nodes will be significantly reduced if the parametric values for both resources are lowered, preventing direct collisions with a high probability. The power of each individual node must be decreased in the manner described below in order to minimise the use of all cloud resources.

$$\rho_i = min \sum_{i=1}^{n} power_i * \gamma_i \tag{3}$$

Where,

$power_i$ indicates input power that is transmitted to cloud

γ_i denotes latency period of individual cloud platform

Since there are various clouds, a centralized data center platform is required to collect all the data. As a result, the analytical model uses a cloud graphing point, which is framed as a minimization point using Eq. (4) as shown below.

$$\tau_i = min \sum_{i=1}^{n} (v_i - \frac{w_i}{dc_i}) \tag{4}$$

Where,

v_i represents the data vertices in individual cloud

w_i indicates data weight in a particular unit

dc_i describes the amount of data centers that are available for multiple cloud platform

As data is collected utilizing various cloud encrypted sets, there may be a high chance of failure during the data transmission phase. Therefore, using Eq. (5), the failure rate during the transmission phase must be reduced as follows,

$$f_i = min \sum_{i=1}^{n} \left(\frac{vol_i}{1 - da_i} \right) * \beta_i \tag{5}$$

Where,

vol_i denotes total volume of data in all clouds

da_i indicates available data centers

β_i represents the decision variable of data transmission stages

The data transmission stage in Eq. (5) is subject to following constraint,

$$\beta_i = \begin{cases} 0 \; if \; da_i \; is \; not \; available \\ 1 \; if \; da_i \; is \; availabe \end{cases} \tag{6}$$

If $\beta_i = 1$then the number of available data transmission stage must be secured using tuple node formation method that is formulated in Eq. (7) as follows,

$$t_i = \sum_{i=1}^{n} ID_i + act_i + inact_i + \vartheta_i \tag{7}$$

Where,

ID_i indicates unique ID of a particular cloud storage node

act_i, $inact_i$ denotes active and inactive states of transmission process

ϑ_i represents type of transceiver performance

Using cloud computing, where data is processed with high security, the objective function of a defined transportation application is generated. As a result, the minimization issue described by Equation is a combination of all the Eq. (8).

$$obj_i = min \sum_{i=1}^{n} C_i, \rho_i, \tau_i, f_i \tag{8}$$

By tying together the required parts of the system, the objective function in Eq. (8) is applied in the real-time cloud computing toolkit. Since the proposed method is implemented using much lower data weights even with multiple cloud based systems low cost of operation is guaranteed. In addition the number of resources that are provided to each cloud platform is much lesser than threshold value thus using lower amount of resources proposed method operates without any collision. Therefore at low operational cost projected method can able to provide better efficiency with high encryption rate. Additionally proactive monitoring is provide in projected system model thus reducing unnecessary energy expenditure with low demand values. But an optimization approach has been adopted and is discussed in Sect. 3 in order to improve the effectiveness of the proposed method's functioning capability.

Optimization algorithm

In the entire system, the sensor nodes must be encrypted in order to increase the security of data in transportation applications. Thus, to meet high encryption requirements, the Improved BlowFish Algorithm (IBFA) cypher algorithm is selected. IBFA is also built using an S-shaped design, with one segment used for input case encryption and other segments used for input case decryption. The main benefit of selecting IBFA is that it offers encryption codes to all data units more quickly, allowing all imports in transportation applications to be reported to customers as soon as feasible [18–21]. IBFA is recommended over other data encryption standards because it requires fewer operational procedures than other encryption methods. The key management procedure is distributed among all the required replacement blocks using various array sizes when the optimization of IBFA begins. Subkeys are formed with 16 round periods in the following stage, and they are extended for 32 further periods until the final data is encrypted. As a result, a parallel structure with independent register units will be used to continuously cycle through and monitor all of the important periods. Additionally, the IBFA applies the pipeline idea in circumstances of direct realization, which leads to a reduction in the time it takes for uplink and downlink frequencies to synchronize. The IBFA mathematical model is as follows,

$$encryption_i = \sum_{i=1}^{n} (ed_i + ek_i) \tag{9}$$

Where,

ed_i, ek_i indicates encrypted data and corresponding key segment

Equation (9) represents the encrypted data and key in terms of bytes per second, and Eq. (10) represents the time required to encrypt a specific piece of data before sending it to the cloud as follows,

$$encryption_t(i) = \sum_{i=1}^{n} \frac{s_d(i)}{r_i} \tag{10}$$

Where,

s_d indicates corresponding size of data that is present in cloud

r_i represents rapidity rate of data to be transferred

In order to transfer the data as quickly as feasible, the speed of all the data across numerous cloud platforms must be enhanced.

At the conclusion of each cycle period, the aforementioned technique is also used to decode the appropriate codes. Equation (11) can therefore be used to compute the rapidity rate as follows,

$$r_i = \sum_{i=1}^{n} \frac{C_p(i)}{C_b(i)} \tag{11}$$

Where,

C_p, C_b denotes data that is transferred in cycles per second and bits respectively

Algorithm – Improved BlowFish Algorithm (IBFA)

Input

Initialize multiple cloud storage platform with latency periods for both uplink and downlink periods, and synchronization latencies by allocating corresponding resources for transportation applications ;

Output

Collision avoidance multiple cloud platform at minimized input power using key encryption code at reduced data volume;

Step 1: At first, the objective function is constructed with the collision factor using C_i;

Step 2: Initialize the amount of resources that needs to be supplied for transportation application that must be followed by certain improvements in power allocation factor $power_i$ with $0 \leq i \leq 1$, and its individual latency period determination γ_i with prevention of direct collision cases;

Step 3: While do.

Provide the vertices rate of individual cloud platform v_i in both presence and absence of data centers in a systematic way for computing the graphing points in transportation process by using Eq. (4);
Verify the data weight values in multiple cloud platform using corner vertices values τ_i for identifying the critical data changes;
If the critical data changes are higher τ_i is not at $(\tau_i < N)$ do
Modify the volume of data in multiple cloud platform in all available data centers that is allocated to a particular transport which is having different decision variables using Equations (4) and (5) f_i with $1 \leq i \leq N$ into N number of volume data states;
// Data transmission stage
Update the decision variables β_i with random tuple node function t_i by generating the unique ID ID_i using active and inactive states as shown in Eq. (7);
//Data encryption phase
Select the encrypted data and key matrix with changes in rapidity rate r_i as defined in Equations (9) and (10);
Update the rapidity rate using Eq. (11) with cycle per second and bit values of corresponding key vectors followed by the data segment values of cloud and compute the new secured data position;
The improvements in vertices segments in separate areas are updated by using Eq. (9);

$cloud_{new} = cloud_{old} + 1;$

End;

Step 4: If $(f_i < 0)$ then

$f \leftarrow 0$; *//Interchange the existing solution in the current loop with the new solution;*
End if;

Step 5: If $(\beta_{MAX}[0,1] < 1)$ then

Re-initialize the cloud values with new segments;
Obtain the overall best solution;
End if;

Step 6: If $(f_{max} < N)$ //Existing solution is replaced with the new solution

$cloud_i = cloud_{modified}$

;

$f_{min} = N$; //Attain the most feasible solutions for determining the overall best solution;

Increment the count $cloud_{new}$ by 1;

Return the best overall solution;

End;

The above mentioned step-by-step implementation is provided in flow chart for direct implementation as represented in Fig. 2.

Results and discussions

In this section, the experimental verification scenarios for the suggested method are explained, where all formulated variables and mathematical expressions are connected in a loop structure. The selection of the number of cloud storage systems in this type of cloud computing procedure is based on the data volume. It is not possible to add new cloud computing capabilities if a user selects both the storage system and the data based on the current scenario but if that condition changes in the future. The number of parameters connected to output determinations is not modified because the proposed method is also evaluated and simulated for transportation applications. The number of control centres for the cloud storage system is 52, and each of these unique units is connected to the central point of view in order to achieve the results suggested by the suggested method. Therefore, numerous cloud data sets are linked using time-demand output blocks if there are any unidentified actions in the transportation application. Additionally, IBFA is integrated with the suggested formulation for graph analysis, resulting in a constant 425 configuration setups. The following scenarios were chosen to validate the proposed multiple cloud computing system using high encryption standards,

Scenario 1: Evaluation of latency.
Scenario 2: Minimization of data collision.
Scenario 3: Resource allocation and vertices determination.
Scenario 4: Amount of failure rate.
Scenario 5: Encryption rapidity rate.

All five of the aforementioned scenarios—where hardware and software systems are combined with identical configuration setups—are evaluated, simulated, and compared with current models. The starting configuration is established for the simulation situation so that there is more data than there are generated packets. However, the suggested method has a significant limitation in that all packets are arranged linearly, necessitating multiple

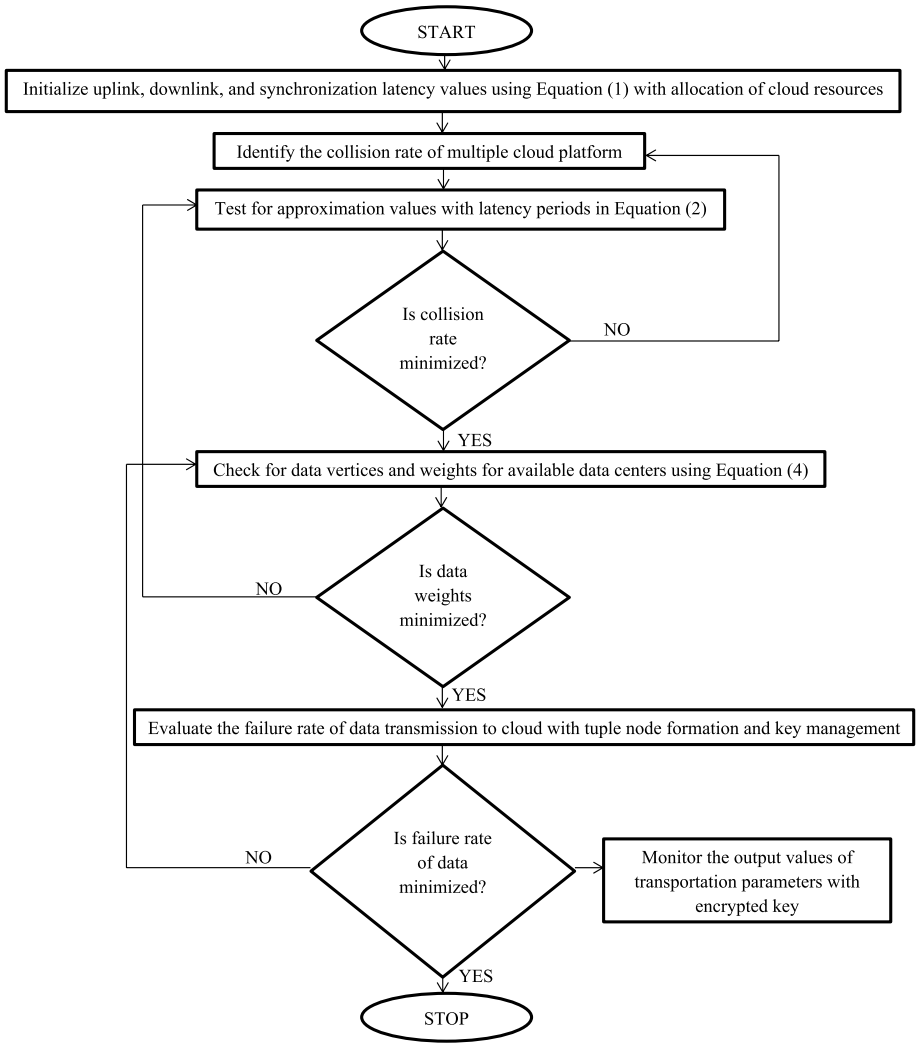

Fig. 2 IBFA for transportation applications

cloud segments to simulate a single situation. Additionally, the suggested solution addresses the drawback of simulation with existing models by offering active services to all required cloud stations. As a result, stations that are necessary are turned on for a while, and stations that are not necessary are turned off. The following is a full description of each scenario.

Scenario 1

Using uplink and downlink loop values, the latency period of data that is present during the transmission and reception stages is calculated. The values of crucial parameters will vary even every second in transportation applications, and all of these changes must be reorganized in the cloud storage system. If these types of changes are reported to a central data center, the system is said to be extremely secure with regard to any data changes. If the data is inactive for a predetermined amount of time, the relevant channel that transfers the data needs to be examined. The cloud storage system offers a new key for determination cases during this checking period and decodes the same at the receiver. Before moving on to the next level, more resources must be assigned to the same data if it was correctly received after decoding. The data may, however, occasionally become completely lost. In these cases, synchronization between uplink and downlink data is offered, retransmitting the transportation parametric data to the appropriate centers. The latency times for the data in the suggested technique are shown in Fig. 3.

Figure 3; Table 1 shows that five different data segment with uplink latency rates of 2.33, 3.47, 4.84.5.12,

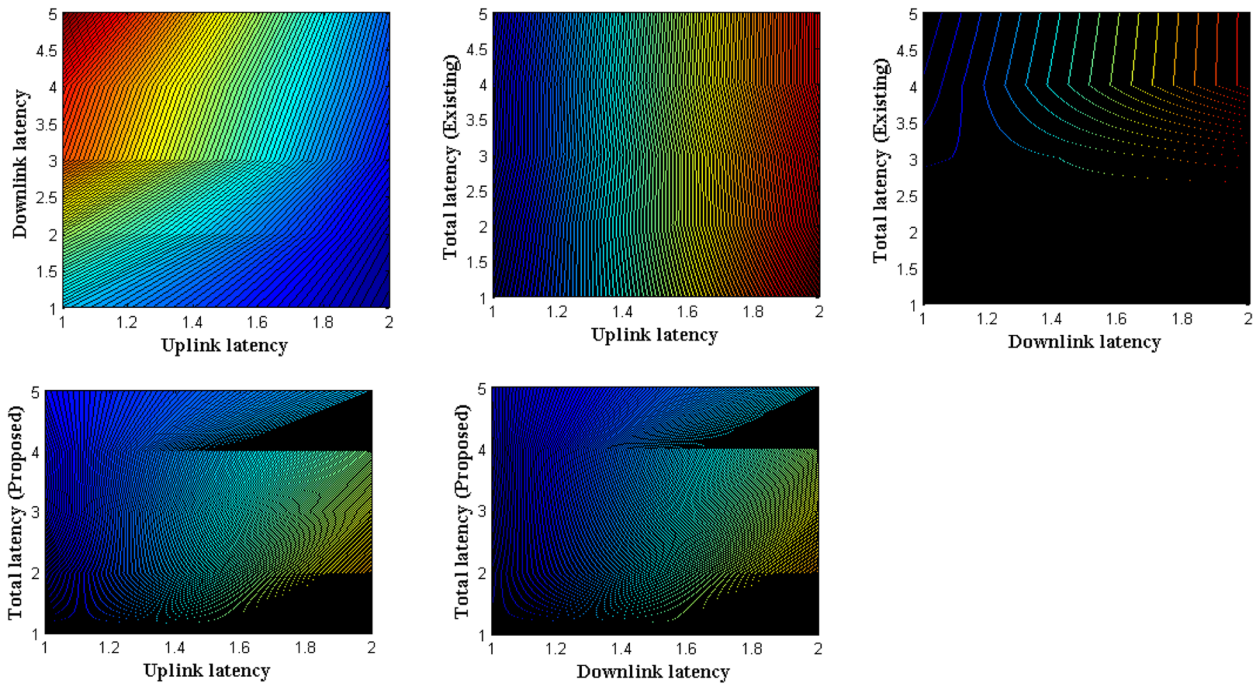

Fig. 3 Latency periods with synchronization

Table 1 Latency using synchronization

Uplink	Downlink	Total latency [4]	Total latency (Proposed)
2.33	0.7	56	34
3.47	0.9	53	25
4.84	1.3	52	21
5.12	1.6	50	18
5.9	1.8	50	12

and 5.9 s are taken into consideration (in the case if the data is transmitted). Accordingly, the downlink latency rates are given as 0.7, 0.9, 1.3, 1.6, and 1.8. The total latency of a projected multiple cloud storage system is compared to a single cloud platform using the rates mentioned above, where the exiting method has more inactive periods [4]. This can be demonstrated using an uplink time of 5.12 s and a downlink time of 1.6 s for the same data. In this case, the total latency provided by the proposed method is 18 s, including synchronization and other required resources. In contrast, the current technique allows for a 50-second latency interval while sending a single data packet to the cloud. Because there is only one cloud platform available, all of the data revolves around the same encrypted code that cannot be changed. This is the main cause of the prolonged periods of inactivity.

Scenario 2

Data duplication and collision are both possible as a result of the availability of several cloud storage systems. So, using experimental analysis, the quantity of resources allotted to a specific encrypted cloud is looked at in order to ascertain the collision rate of data transmission systems. Additionally, data synchronization raises the system's overall collision rate, leading to the need for approximations while handling transportation parametric data. The transportation characteristics will also be dynamic in another mode, necessitating separate access for various cloud systems. Even the dynamic nature mentioned above will result in collision effects, and the proposed method determines the exponential rate in order to avoid this kind of collision. Due to the collision process' exponential rate, Fig. 4's simulation shows virtually little change. Exact data collision rates across various cloud platforms are provided by the distinction between resources allotted and values stored.

It is shown in Fig. 4; Table 2 that the parametric values that are altered in random form are separated from approximation values that are varied from 10 to 50. The changes in parametric values are represented, respectively, by 2,8,14,23, and 26. Due to these modifications,

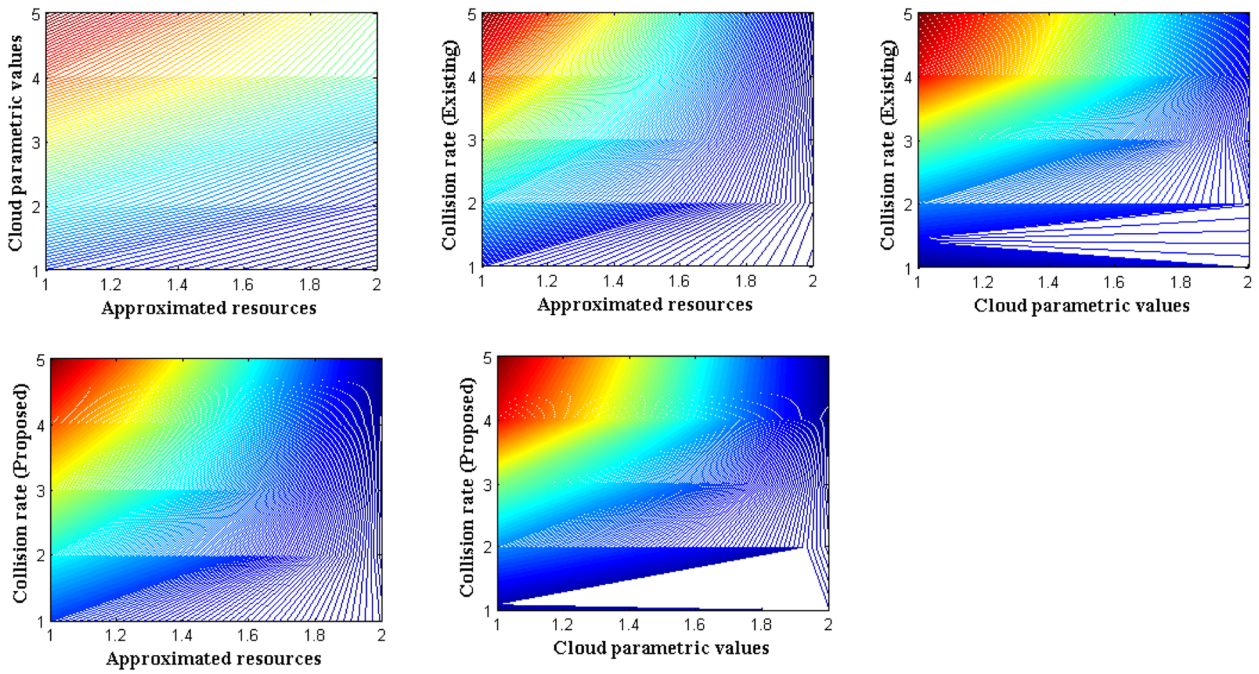

Fig. 4 Collision rates of multiple cloud metrics

Table 2 Collision rate of multiple cloud data

Approximated resources	Cloud parametric values	Collision rate [4]	Collision rate (Proposed)
10	2	4.4	2.6
20	8	4.8	2.2
30	14	4.3	2
40	23	4.5	1.9
50	26	4.9	1.7

the exponential term continues to produce negative data transmission values; hence, storage systems provide a difference of 1. The total collision rate is measured, simulated, and compared with the current model [4] after negative values have been avoided. It can be seen from the comparison case that the existing method has a high collision rate because there is only one cloud storage system. In contrast, the proposed method separates data into different storage systems within each cloud platform, which lowers the collision rate. This can be confirmed by comparing 30 different approximated data values with 14 different cloud parametric values. According to this specification, the collision rates of the current and proposed methods are 4.5 and 1.9 for all data transmitted in the cloud system, respectively.

Scenario 3

The primary resource designated for processing all data is known as power, since input data can only be delivered to different cloud platforms if there is enough power in the system. A boundary value is produced for this kind of input source that is measured in terms of vertices, and only particular regions inside those vertices will certain data be sent at high power. If the moving transportation area remains the same, the data will remain static on the same cloud platform, necessitating no change in power. Additionally, the weight of the data in each cloud must be calculated in order to distribute the data center in the proposed approach that is in charge of the precise weight. However, the current approach does not allocate data centers according to the weight of a given piece of data, so a single cloud signal is unable to explore the vertices rate. Additionally, the latency of each cloud is measured in relation to shifting weights, leading to the establishment of graphing points with minimization values. Simulated values for centralized data points are shown in Fig. 5.

Figure 5; Table 3 shows that the power is changed stepwise from 5 to 10 watts, with the corresponding boundary indices being 6.33, 7.21, 7.86, 8.24, 8.93, and 9.12. With regard to the allocated data weights in the

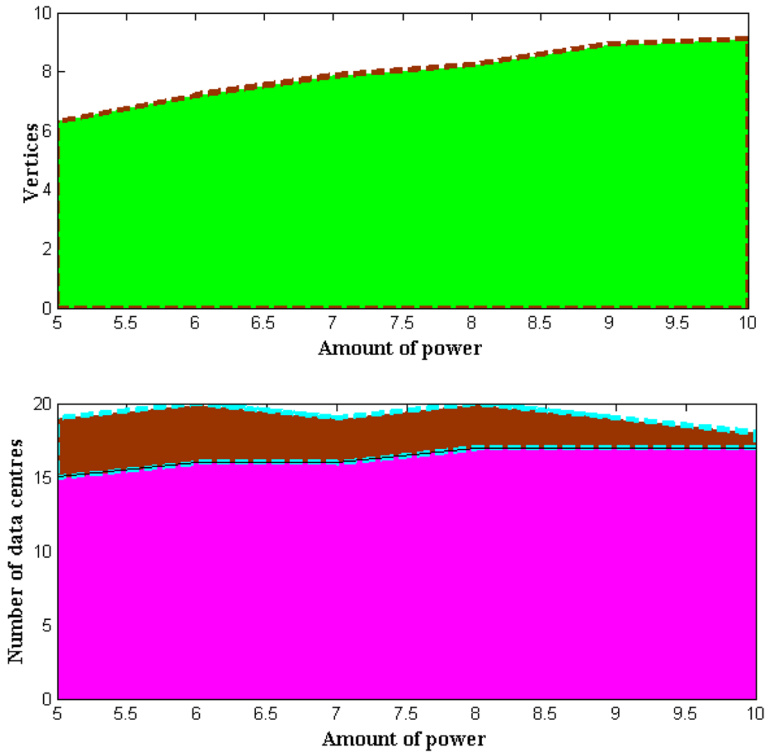

Fig. 5 Comparison of power with data centers

cloud system, the number of data centers is reduced throughout these adjustments. Additionally, the proposed method is able to assign transportation data to the appropriate data center according to a specific area due to the presence of vertices. However, the encrypted key management system is unaffected in the event of a sudden channel shift. As a result, there is still only one data center in the final phase of the process. This may be shown by utilizing a boundary that is 8.93 watts on all sides and an input power of 9 watts. The existing method uses 17 different data centers, whereas the proposed method only allocates 1 during this data specification, resulting in significant energy waste for a single cloud platform.

Table 3 Amount of power

Amount of power	Vertices	Number of data centers [4]	Number of data centers (Proposed)
5	6.33	15	4
6	7.21	16	4
7	7.86	16	3
8	8.24	17	3
9	8.93	17	2
10	9.12	17	1

Scenario 4

In applications that transport data from cloud storage, there is a significant chance that the data won't arrive at its destination on time because of the mobile environment. Even though the amount of data is much higher, this type of failure rate must be lower, and for peace of mind, ad hoc nodes can also be installed in a way that fully supports end devices. Determining the number of central and subordinate data centers in the cloud system can also help to lower the failure rate of transportation nodes. It is possible to avoid data redundancy by separating two different data centers and transmitting the data directly to the central station. The aforementioned procedure is only used when the total amount of data exceeds the threshold cloud data limit. A decision variable that uses the current threshold limit of data in the corresponding application is reproduced to determine the maximum limit of data. Thus, Fig. 6 shows the simulated values.

Figure 6; Table 4 shows that a total of 500 megabytes of data is transmitted in one location from various modes of transportation, and the system has allocated relevant data centers. For the total volume of data, there are 22,463,683, and 99 data centers, respectively. Practically speaking, the proposed method outperforms the current method [4] in the comparison case with a low failure

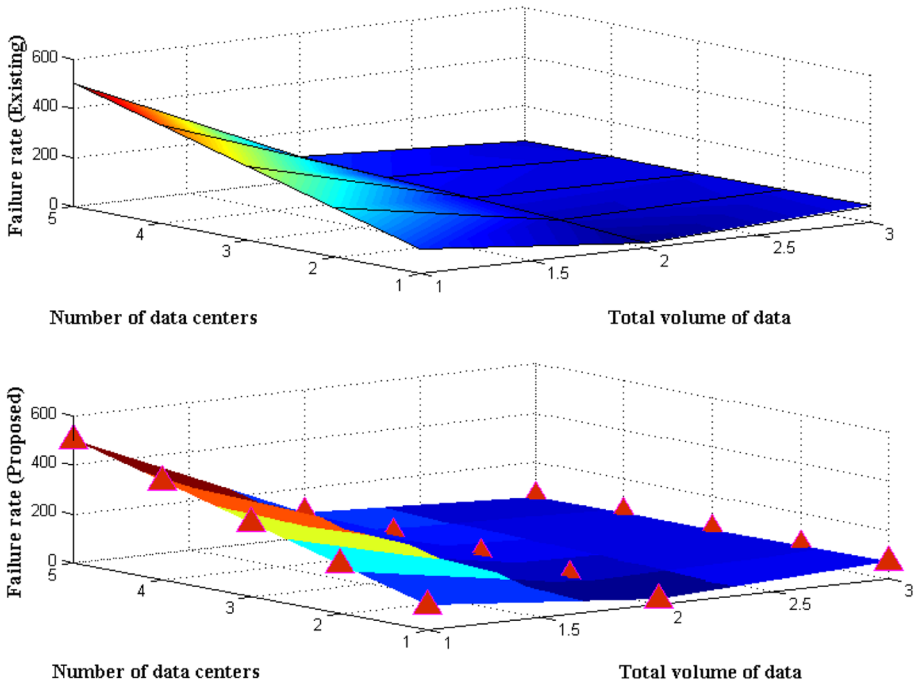

Fig. 6 Failure rate of multiple data clouds

Table 4 Failure rate of data in multiple cloud

Total volume of data	Number of data centers	Failure rate (Percentage) (Existing)	Failure rate (Percentage) (Proposed)
100	22	67	30
200	46	64	21
300	68	61	17
400	83	60	11
500	99	56	9

rate of data that is transmitted either through the same cloud link or through a different cloud link. Even though the volume of data is much higher in different places, the projected method's failure rate is still less than 10%. This may be demonstrated in a real-world experimental scenario with a total data volume of 500 and 99 data centers, respectively. One central data server will be present for the data management system instead of 99 data centers, as was previously mentioned. The proposed method offers a data failure rate of 9% with the aforementioned volume, while the current method offers a substantially higher failure rate of 56%.

Scenario 5

The IBFA's provision of data encryption for end-user data transmission is put to the test using this scenario. The loss rate and delay are decreased when IBFA is integrated with analytical models because no delinquent data is transmitted, even on mobile platforms. However, if data is not encrypted, there will be a rise in data duplication across multiple clouds, which will lead to a failure in data transmission. Because there are many clouds, it is possible for real-time data from one cloud to be stored in another cloud, where users will not be able to access it even after decrypting it because system specifications differ. As a result, necessary storage is offered on the same cloud platform with encryption, which also reduces data duplication. The key management step is where the encryption process typically begins, and if there is more data, the encryption process' speed must be increased. Figure 7 uses a simulation to show the encryption process' speed.

According to Fig. 7; Table 5, the encryption rate of the proposed IBFA is significantly higher than the existing method [4], and the following data specifications, which have data sizes that are exactly the same as those in the previous scenario cases of 100 to 500, respectively, can attest to this. The encryption speeds for individual data that is spread over several cloud platforms are 160, 360, 540, 620, and 890 for total data sizes. With the proposed method using IBFA, the encryption time of a specific data is therefore less than 50 s by separating the data size and rapidity rate. After testing with the same configuration, the existing method [4] offers a higher encryption period that is significantly higher than 50%, even for

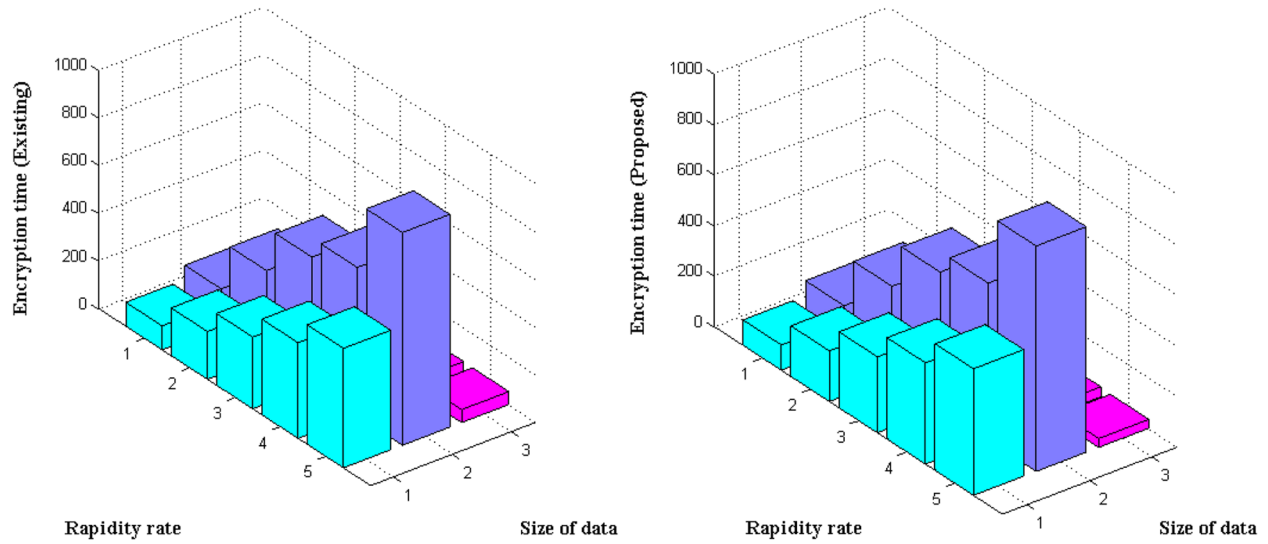

Fig. 7 Encryption time period of data

Table 5 Time period of encryption

Size of data	Rapidity rate	Encryption time [4]	Encryption time (Proposed)
100	160	68	52
200	360	63	50
300	540	60	43
400	620	56	40
500	890	54	35

small amounts of data. This can be demonstrated using data with a size of 500 and a common rapidity rate of 890 s, where one data is encrypted with key management techniques in the case of IBFA in 35 s. In contrast, other optimization techniques require at least 54 s to encrypt a piece of data, and after that, key management procedures are applied.

Performance metrics of IBFA

By taking into account various scenario instances, the parametric cloud computing problems that are established using analytical models are simulated and compared. The inclusion of IBFA in the suggested method must be addressed using performance evaluation measures. Hence, two case studies are taken into consideration in this section for resolving complexity issues in numerous cloud infrastructures. The proposed solution introduces multiple cloud metrics with growing data size since at a certain point in time, traffic conditions will be more intense, but since the size of data is much higher, more complexity will rise in a single cloud platform. The case studies below are taken into consideration as a result.

Case study 1: Time complexity.
Case study 2: Space complexity.

Case study 1

Data must arrive at the receiver at the proper time since the proposed method is used for transportation applications where information about all consumer items must be obtained. If the data does not reach numerous receivers, the cloud platform's access points are reviewed, and data faults are fixed. However, there are rare instances where data hasn't reached just a few receivers, in which case the receiver's channel is examined and necessary action is taken for any delayed data from a few receivers. Even if nodes in multiple clouds are unable to connect, there is a good chance that an ad hoc cloud security system can be set up at the same location by specifying certain ranges. The optimal epoch periods for Fig. 8's time complexity illustration is given in Table 6.

The best epoch periods, which are randomly selected between 10 and 100, are used to simulate Fig. 8. The following epochs: 20, 40, 60, 80, and 100 are taken into consideration for simple configuration, and each time complexity is compared with the current approach [4]. If the time complexity is less than one second, the used multiple cloud system transports the data without any complications. This time frame is attained by utilizing the suggested technique as opposed to the current way, although in the beginning, the complexity of the

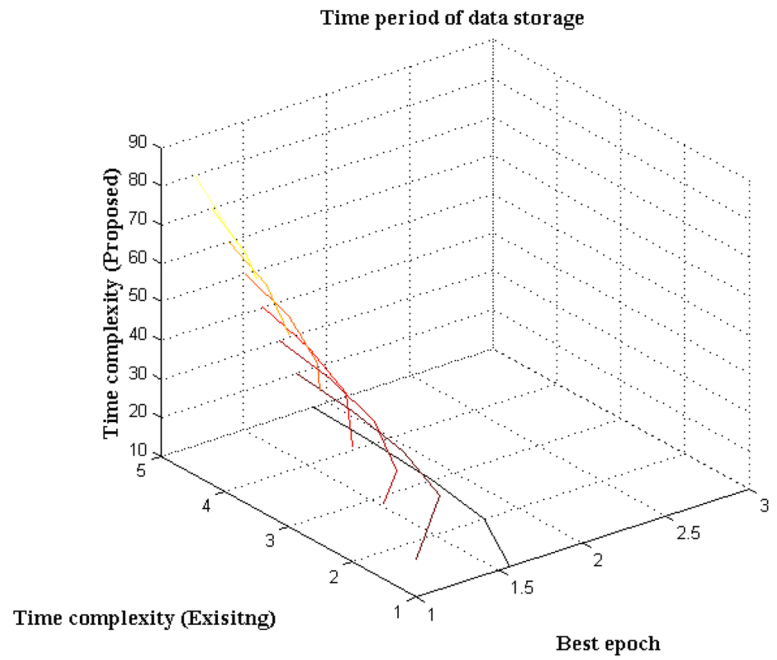

Fig. 8 Complexities in time periods

Table 6 Time complexities of IBFA

Best epoch	Time complexity [4]	Time complexity (Proposed)
20	2.34	1.26
40	2.25	1.15
60	2.2	0.9
80	2.13	0.7
100	2.11	0.5

Table 7 Space complexity representations

Best epoch	Space complexity [4]	Space complexity (Proposed)
20	0.4	0.13
40	0.36	0.1
60	0.35	0.07
80	0.33	0.04
100	0.32	0.03

proposed method employing IBFA in calculating the amount of cloud data is an issue. With an epoch of 80, this can be arbitrated; the projected method's time complexity is 0.7 s, compared to the existing method's 2.13 s.

Case study 2

The quantity of space in a given infrastructure is significantly influenced by the data storage capability of various cloud platforms. This case study compares the planned system's spatial complexity to the state of the traffic in the area. The suggested solution also assumes that the traffic system may change in the future, allocating additional capacity for specific data. Additionally, the proposed IBFA system is utilized to calculate the amount of space needed to encrypt a certain piece of data before storing it in space. Since the investigation indicated above is different from auxiliary space, stack segments connected to environmental changes are not taken into account. Figure 9; Table 7 shows the simulation results for the projected system's spatial complexity.

The best epoch periods are taken into consideration as inputs since Fig. 9 shows that all changes in space complexity are defined by utilizing input characteristics. Similar to the preceding scenario, the suggested system's space complexity is compared to that of the current system [4]. The suggested solution also offers reduced space difficulties with storage, encryption time, and unexpected increases in traffic situations in this comparison. This may be demonstrated using an epoch period of 100, where the space complexity of IBFA is 0.03 and that of the present approach is 0.32.

From Fig. 9 it is observed that all changes in space complexities are determined using input characteristics therefore the best epoch periods are considered as inputs. Similar to previous case five epoch is considered and space complexity of proposed system is compared

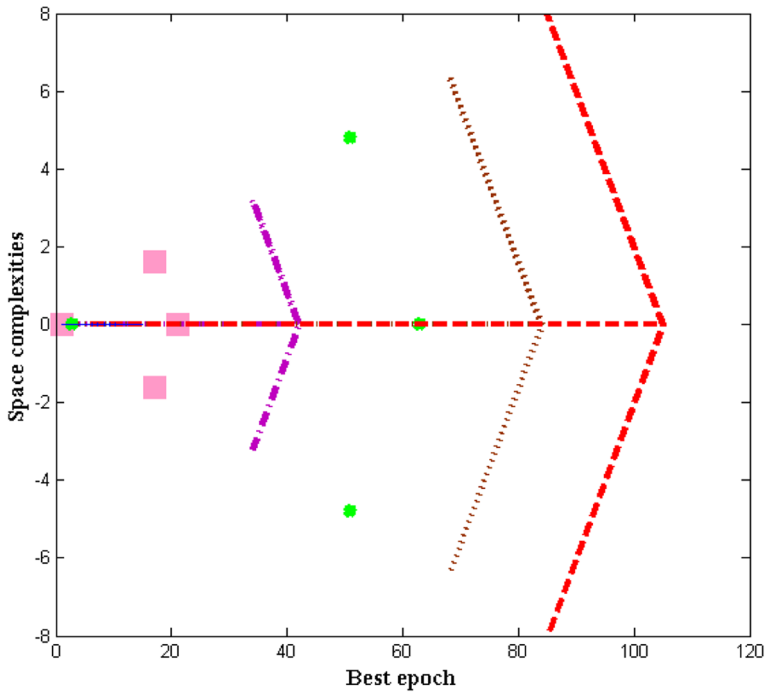

Fig. 9 Comparison of space complexities

with existing system [4]. In this comparison also the proposed method provides less space complexities with encryption time, storage and sudden increase in traffic conditions. This can be proved with epoch period of 100 where space complexity of IBFA is 0.03 whereas for existing method it is increased to 0.32.

Conclusion

A new analytical model is offered for a multiple-cloud platform with strong encryption for consumer application transmission, and it also analyses all issues linked to various data errors. The provided analytical model offers data information in order to synchronize both uplink and downlink data and decrease latency in each cloud data base. Additionally, all risk considerations connected to a single cloud platform for huge data operations are investigated, and solutions are rationalized in conditions that are energy efficient. Additionally, a control data system is created that connects data into various job setups that a cloud system must carry out. Multiple cloud systems work better on schedules as a result of individual time allocation, so the suggested solution is put into practice to reduce the data delay that occurs during transmission times. Additionally, because the suggested system incorporates several clouds, there is a significant risk of data collision if a channel is used by multiple users. To avoid this, the projected system uses IBFA, which allots sufficient resources for the entire cloud database. As a result, the system's collision rate is reduced, resulting in a more adaptable wireless transportation operation. When consumer data is set up on hardware in real time and communicated to the cloud, it is seen that all of the data has been encrypted using key management techniques. The aforementioned key is utilized at the receiver's side to decrypt all information that has been stored while using the least amount of power possible. The following scenarios, such as determining the vertices for multiple cloud systems, examining the failure rate of data packets, and analyzing the rapidity rate of encryption processes, are examined to verify the projected analytical model. The results of all of the aforementioned cases are compared with existing models, and the comparative results show that the proposed method on multiple cloud storage is superior for 43% at the start.

However once the scenarios are examined the real time outcomes for all seven different cases provides much better experimental results for about 32%. But in case of total representation at ending state projected model provides an average state outcome of about 27%. Therefore as compared to starting cases at ending period the percentage of outcomes is lesser but optimal results are achieved. If the proposed model is implemented in real time then infrastructure of data communication will be increased thus the data can be recovered even in case of

many disaster conditions. Since the data transfer is made as an intra-communication platform where all individuals throughout the universe can able to communicate with each other it is necessary to avoid vendor stock-in which is provided by multiple clouds. In addition a user can able to transfer the data based on different workloads in the system thus entire data loss is reduced. Therefore throughout the universe the multiple cloud applications are based on managing necessary components, providing centralized management facilities and balancing the data loads.

Limitations

Even though the projected method provides above mentioned advantages in real time applications by forming a green IoT environment the exploration characteristics of multiple cloud has some limitations. The major limitation of designed system model is that complex parameters can be established if the range of cloud systems is extended to wide range. Therefore for long distance data collection points proposed method will be implemented at high complexities thus in turn all industry alertness will be reduced thus increasing the cost of implementation. Even if multiple cloud systems are designed at wide range then reliability of the system will be reduced.

Acknowledgements
Not Applicable.

Authors' contributions
Data curation: Khaled H. Alyoubi, Abdulrhman M. Alshareef; Writing original draft: Hariprasath Manoharan, Shitharth S; Supervision: Shitharth, Fahad S. Alotaibi, Adil O. Khadidos. Project administration: Shitharth, Fahad S. Alotaibi, Adil O. Khadidos. Conceptualization: Hariprasath Manoharan, Shitharth S; Methodology: Shitharth S and Hariprasath Manoharan; Validation: Khaled H. Alyoubi, Abdulrhman M. Alshareef; Visualization: Khaled H. Alyoubi, Abdulrhman M. Alshareef; Resources: Shitharth S and Hariprasath; Overall Review & Editing: Shitharth, Fahad S. Alotaibi, Adil O. Khadidos. All authors reviewed the final manuscript. The author(s) read and approved the final manuscript.

Funding
No funding was received for this project.

Availability of data and materials
The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Declarations

Ethics approval and consent to participate
Not Applicable.

Consent for publication
Not Applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Department of Computer Science, Kebri Dehar University, Kebri Dehar, Ethiopia. [2]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. [3]Department of Electronics and Communication Engineering, Panimalar Engineering College, Poonamallee, Chennai, India. [4]Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia.

Received: 2 September 2022 Accepted: 11 October 2022
Published: 3 November 2022

References

1. Upadhyay D, Zaman M, Joshi R, and Srinivas Sampalli (2022) An Efficient Key Management and Multi-Layered Security Framework for SCADA Systems. IEEE Trans Netw Serv Manage 19(1):642–660. https://doi.org/10.1109/TNSM.2021.3104531
2. Suvorov AA, Ahmed AZaki, Diab AS, Gusev MV, Andreev NY, Ruban AB, Askarov RA, Ufa et al (2020) Comprehensive Validation of Transient Stability Calculations in Electric Power Systems and Hardware-Software Tool for Its Implementation. IEEE Access 8:136071–136091. https://doi.org/10.1109/ACCESS.2020.3011207
3. Shitharth S, Satheesh N, Kumar BPraveen, ' IDS Detection Based on Optimization Based on WI-CS and GNN, Algorithm (2021) in SCADA Network', Architectural Wireless Networks Solutions and Security Issues, Lecture notes in network and systems, Springer, vol. 196, Issue 1, pp. 247–266. https://doi.org/10.1007/978-981-16-0386-0_14
4. Kermani M, Adelmanesh B, Shirdare E, Sima CA, Carnì DL, and Luigi Martirano (2021) Intelligent Energy Management Based on SCADA System in a Real Microgrid for Smart Building Applications. Renewable Energy 171:1115–1127. https://doi.org/10.1016/j.renene.2021.03.008
5. Đorđević A, and Željko Đurišić (2019) Mathematical Model for the Optimal Determination of Voltage Level and PCC for Large Wind Farms Connection to Transmission Network. IET Renew Power Gener 13(12):2240–2250. https://doi.org/10.1049/iet-rpg.2018.5913
6. Samimi A, and Mehdi Nikzad (2017) Complete Active-Reactive Power Resource Scheduling of Smart Distribution System with High Penetration of Distributed Energy Resources. J Mod Power Syst Clean Energy 5(6):863–875. https://doi.org/10.1007/s40565-017-0330-z
7. Marpaung N, Lysbetti E, Ervianto R, Amri, and Hayatul Illahi (2020) Analysis of SCADA Application on Distribution System Reliability. Int J Electr Energy Power Syst Eng 3(2):46–52. https://doi.org/10.31258/ijeepse.3.2.46-52
8. Astolfi, Davide. 2021. "Perspectives on Scada Data Analysis Methods for Multivariate Wind Turbine Power Curve Modeling." Machines 9 (5). https://doi.org/10.3390/machines9050100.
9. Kong X, Chen Y, Xu T, Wang C, Yong C, Li P, and Li Yu (2018) A Hybrid State Estimator Based on SCADA and PMU Measurements for Medium Voltage Distribution System. Appl Sci (Switzerland) 8(9). https://doi.org/10.3390/app8091527
10. Kong H, Lu M, Que L, Xu F, Zhao J, and Ancheng Xue (2022) A New Four-Step Method to Identify the Parameters of Transmission Line Based on SCADA Data. IET Gener Transm Distrib 16(9):1822–1835. https://doi.org/10.1049/gtd2.12420
11. Pandit, Ravi, Athanasios Kolios. 2020 "SCADA Data-Based Support Vector Machine Wind Turbine Power Curve Uncertainty Estimation and Its Comparative Studies." Applied Sciences (Switzerland) 10 (23): 1–18. https://doi.org/10.3390/app10238685.
12. Alaa O, Khadidos AO, Khadidos H, Manoharan KH, Alyoubi, Abdulrhman M, Alshareef, Shitharth S (2022) " Integrating Industrial Appliances for Security Enhancement in Data Point Using SCADA Networks with Learning Algorithm," International transactions on Electrical Energy Systems, vol. Article ID 8457116, 17 pages, 2022. https://doi.org/10.1155/2022/8685235.
13. Temido J, Sousa J, Malheiro R (2014) SCADA and Smart Metering Systems in Water Companies. A Perspective Based on the Value Creation Analysis. Procedia Eng 70:1629–1638. https://doi.org/10.1016/j.proeng.2014.02.180
14. Mollah MB, Sikder SI (2012) "Towards IEEE 802.22 Based SCADA System for Future Distributed System." 2012 International Conference on Informatics, Electronics and Vision, ICIEV 2012, no. May 2012: 1075–80. https://doi.org/10.1109/ICIEV.2012.6317474
15. Kovaliuk O, Dmytro KM, Huza, Kovaliuk OO (2018) Development of SCADA System Based on Web Technologies. Int J Inform Eng Electron Bus 10(2):25–32. https://doi.org/10.5815/ijieeb.2018.02.04

16. Osman FA, Mohamed YM, Hashem, Mostafa AR, Eltokhy (2022) Secured Cloud SCADA System Implementation for Industrial Applications. Multimedia Tools and Applications 81(7):9989–10005. https://doi.org/10.1007/s11042-022-12130-9
17. Badshah A, Jalal A, Farooq U, Rehman G-U, Band SS, Iwendi C "Service Level Agreement Monitoring as a Service: An Independent Monitoring Service for Service Level Agreements in Clouds.Big Data." *ahead of print*. https://doi.org/10.1089/big.2021.0274
18. Alaa O, Khadidos H, Manoharan S, Selvarajan, Adil O, Khadidos et al (2022) In: Energies (ed) A Classy Multifacet Clustering and Fused Optimization Based Classification Methodologies for SCADA Security. MDPI. https://doi.org/10.3390/en15103624
19. Aghenta LO, Tariq Iqbal M (2019) Design and Implementation of a Low-Cost, Open Source IoT-Based SCADA System Using ESP32 with OLED, ThingsBoard and MQTT Protocol. AIMS Electron Electr Eng 4(1):57–86. https://doi.org/10.3934/ElectrEng.2020.1.57
20. Shitharth S, Prasad KM, Sangeetha K, Kshirsagar PR, Babu TS, Alhelou HH (2021) "An Enriched RPCO-BCNN Mechanisms for Attack Detection and Classification in SCADA Systems," in IEEE Access. 9:156297–156312. https://doi.org/10.1109/ACCESS.2021.3129053
21. Wang Q, Zhongli Z (2011) "Reinforcement Learning Model, Algorithms and Its Application." Proceedings 2011 International Conference on Mechatronic Science, Electric Engineering and Computer, MEC 2011, no. 1: 1143–46. https://doi.org/10.1109/MEC.2011.6025669

Publisher's Note