Secure Cloud Computing Platform Advantaged by Data Encryption and CS Optimized Ffbpnns

Kanav Sadawarti[a], Satish Saini,[b]
[a]Computer Science & Engineering, RIMT, Mandi, Gobindgarh, India
[b]Electronics Communication & Engineering, RIMT, Mandi, Gobindgarh, India
[a*]Sadawartikavan90@gmail.com, [b*]Satishsainiece@gmail.com

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract: Cloud computing is one of the latest techniques involved in the splendid field of information communication technology. It is bestowed with endless services to cloud users that offer access to vivid software applications, processing, and storage space irrespective of the time and place constraints. Security is the most dominating parameter deciding the quality of service of the cloud environment. In the present paper, author had addressed the security aspects adjoining cloud computing with the involvement of encryption algorithm followed by Cuckoo Search (CS) optimized Feed Forward and Back Propagation Neural Network (FFBPNN). To enhance the security of stored data technique namely, Rivest–Shamir–Adleman (RSA) is employed with Advanced Encryption Standard (AES) and Triple Data Encryption Standard (TDES). Experimental evaluation in terms of delay, energy consumption and Service Level Agreement (SLA) had shown that the proposed design proved to offer a secure cloud computing environment with 9-26% lower SLA violations, 6-9% reduced energy consumption and 21-42% decline in the data transmission delay as compared to existing work.

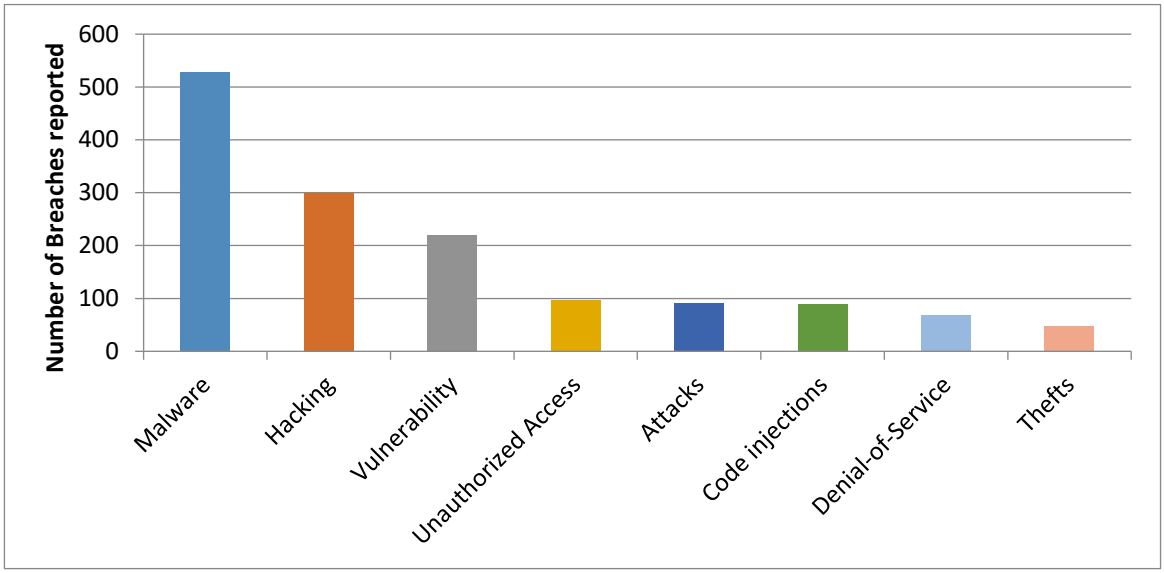
Keywords: Cloud Computing, Cloud Security, Cuckoo Search, Feed Forward and Back Propagation Neural Network, Rivest–Shamir–Adleman (RSA), Advanced Encryption Standard (AES), Triple Data Encryption Standard (TDES)

1. Introduction

Currently, cloud computing has been the latest drift and the most renowned service prevailing in the information technology sector. It offers access to vivid online softwares, applications and larger storage space to demonstrate high computing power that is regardless of time and place boundaries. National Institute of Standards and Technology (NIST) had defined the computing resources and services as key aspects of the cloud that reflect cloud computing as a required business model. It also defined software, infrastructure and platform as three service models and private, public, community and hybrid as four cloud service deployment models that preside over the distribution of cloud service **(Mell & Grance 2011)**. In cloud computing services, service providers and service users are the two key elements that hold the top position. Cloud Service Providers (CSPs) have adjoining Virtual Machines (VMs) that support multiuser sharing in various applications. The existence of VM has proved to be highly advantageous to offer a cost effective and flexible interface to a larger section of service users in addition to delivering an expandable and interoperable platform **(Ramachandra & Bhattacharya 2020)**. However, cloud computing is highly susceptible to various types of breaches and attack incidents.

According to report compiled by McAfee Labs, the incidents compromising the cloud security have increased as compared to the last quarter report of 2019. Top attacks in terms of number of breaches that are reported for the year 2018-2019 as shown in Figure 1. It shows that highest breach incidents have been observed for malware followed by account hacking, vulnerability, unauthorized access, attacks, code injection, dissemination of malicious scripts, Denial-of-Service and theft, etc. **(McAfee Labs Threats Report 2019)**. It is therefore highly recommended to design and deploy strategies to successfully defend the cloud computing environment. In the present work, author proposed the deployment of a secure cloud computing platform with the implementation of encryption techniques such as Rivest–Shamir–Adleman (RSA) with Advanced Encryption Standard (AES) and Triple Data Encryption Standard (TDES) approach. In the process, a database is generated to store the encrypted data that users could access on the basis of priority level obtained with Modified best fit decreasing (MBFD) technique. Further to enhance the security of the proposed cloud security model Cuckoo Search (CS) based optimization is performed followed by the incorporation of Feed Forward Back Propagation Neural Network (FFBPNN).

Figure.1 Top Attack Vectors of 2018-2019



The paper is organized into five sections with section 1 introducing the cloud computing environment and existing vulnerabilities. Section 2 is dedicated for the literature survey of the existing approaches dedicated for securing cloud environment, section 3 describes the proposed methodology, section 4 summarizes the model evaluation and results and section 5 concludes the paper.

2. Literature Review

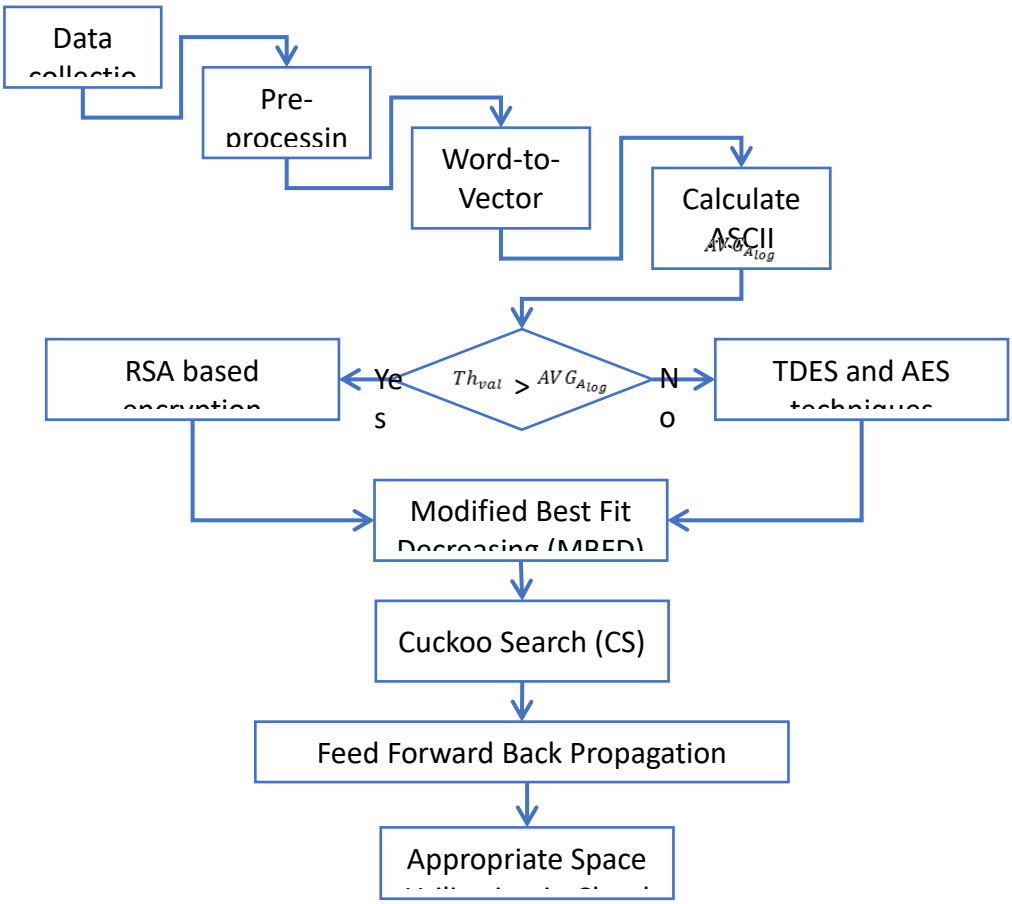
This section summarizes the some of the existing approaches proposed by various researchers to offer a secure cloud computing environment. To start with, in 2012 **Sammy et al.** had postulated a highly secure and energy efficient offerings of cloud services. They had implemented Dynamic Round Robin algorithm to decrease the energy consumption of data centers without compromising the data security. The technique had also shown lower SLA violations. Cloud computing is characterised by virtualization, multi-user and stability, etc. The security and energy challenges adjoining these features were addressed by **Kamboj and Rana (2017)**. They had taken into considerations the rising demand for cloud services that challenges the energy efficiency in terms of CPU usage, memory consumption, storage space and resource utilization at the part of data centres. They concluded that innovative approach that could go parallel with the eco-metrics is the highly required. **Longofono et al. (2019)** had implemented a MACE technique to address the challenges related to cloud storage space, energy efficiency and life time of PCM with the involvement of AESXTS based encryption strategy. Analysis had demonstrated the 15% mean decline in the energy consumption and improved the lifetime that significantly improved the memory utilization. **Mohiuddin et al. (2019)** had introduced an idea of multi-tenancy and designed a Secure Distributed Adaptive Bin Packing Algorithm for the allocation of secure bins in the cloud computing environment. The experimentation had shown that the proposed design requires less time as compared to Best Fit Allocation and First Fit Allocation techniques for storage space allocation. **Nasr et al. (2019)** had presented a priority based framework known as Highest Priority First Execute to offer a secure platform to multi-user and multi-task cloud. The experimental evaluation had shown the effectiveness of the proposed work in decreasing the makespan while enhancing the load balancing degree in comparison to first come first serve, simulated annealing, genetic algorithm, Min-Max and Max-Min algorithms. **Jouini and Rabai (2019)** had addressed the security issues prevailing in the cloud computing environment and applied a quantitative risk assessment technique known as Multidimensional Mean Failure Cost. This generic strategy proved to successfully resolve the most of the security issues of cloud. **Sathishkumar and Venkatachalam (2019)** had postulated a trust based encryption that is measured with kernel fuzzy c-means clustering approach. Here, consumers are verified with verification keys followed by file encryption performed by double encryption technique and blowfish algorithm. This encrypted data clusters are accumulated in cloud. The outcomes had demonstrated the enhanced encryption with the prevalence of legitimate data in cloud. **Yuan et al. (2019)** had proposed a highly secure data de-duplication method that implemented convergent all-or-nothing transform and bits that were randomly acquired from Bloom filter. The experimental analysis of security aspects had proved the outperformance the proposed design in terms of secure re encryption technique. **Njuki et al. (2019)** had proposed the implementation of best fit hybrid algorithm for VM and cloud data security. The hybrid involved AES 256 bit, RSA, SHA 256 bit and ECC that together proved to offer a highly secure and speedy computing environment to end users. The bulk and larger data size was addressed by the involvement of

Homomorphic Encryption. High speed decryption was achieved with the reduction of indices while paring hybrid of encryption algorithms and homographic encryption technique. **Bhise and Latif (2020)** had postulated cryptographic idea for data storage in cloud to meet the desired security level. To achieve this they had implemented RSA with AES algorithm for securing the cloud data using encryption and decryption strategy. In the process, the size of cipher text and decrypted key is also kept constant and demonstrated the enhanced security of the cloud environment with the involvement of experience based trust. **Ibtihal and Hassan (2020)** had mainly focussed the data outsourcing and sharing in cloud environment. They had proposed a highly secure framework with private cloud addressing the encryption and decryption followed by public cloud that stores the data. Encryption was performed with paillier's homomorphic cryptosystem designed specifically to address encryption of image data while homomorphic property was tests with Watermarking algorithm DWT. **Wang et al. (2020)** had postulated a SLA aware resource algorithm to offer a highly secure and efficient cloud storage that is based on I/O throughput along with backend node space exploitation. Experimental evaluation of the proposed work had shown that it outperformed the existing work in terms of lesser SLA violations with improvement in the involvement of number of hosts. **Lei et al. (2020)** had addressed the cloud security based on the combination of LDA and workd2vec models with an aim to establish balance between speed and accuracy adjoining various cloud services. Analysis had shown the effectiveness of the proposed work in warm and cold start environments when compared with existing approaches.

3. Proposed Methodology

In the present work, author has addressed encryption, optimization and classification techniques to offer a secure cloud computing environment.

Figure. 2 Workflow of the proposed cloud security model



In the first stage, encryption algorithm is applied to store and secure the data followed by second stage that is dedicated to enhance the security aspects of the data with the implementation of Cuckoo Search (CS) and Feed Forward Back Propagation Neural Network (FFBPNN). Figure 2 shows the overall work flow of the proposed methodology.

3.1. Data collection and pre-processing

The very first step in this scenario is the collection of data from number of users. This data is processed with the implementation of stop word removal technique. It is one of the pre-processing techniques that transform the data into computer understandable format while filtering out useless data (words) commonly referred to as stop words. For instance, common stop words used in the process are 'a', 'an', 'the', 'in', 'are', 'of' and 'on', etc. The algorithm used for the processing is as follows:

Stop Word Removal (SWR) Algorithm

1. Input: $data_{user}$ // user data
2. Initialize: $Stop_{words}$ // represents the list of stop words
3. $foreach\ i\ in\ data_{user}$
4. $foreach\ j\ in\ Stop_{words}$
5. Check if $data_{user_i} == Stop_{words_j}$
6. Assign: $data_{SWR} = data_{user_i}$ // stop work free data elements
7. Else, $data_{SWR} = ''$
8. End_{if}
9. End_{for}
10. End_{for}
11. Output: $data_{SWR}$ // data obtained after stop word removal process

The above algorithm processes the user data according to the stop words present in the input user data and finally returns a stop word free data. This process considerably reduces the unnecessary utilization of the storage space and compromising processing time.

The outcomes of SWR algorithm is further processed with word to vector technique that represents the text document in numerical form. In this process, ASCII codes were used to represent the text present in the document followed by computation of the logarithmic values as follows:

$$Log_{val} = \int_0^N \log_{10}(\log_2(A_{val}))$$

Where, Log_{val} represents the logarithmic value corresponding to ASCII value denoted by A_{val} and N corresponding to the number of documents. Following this, $AVG_{A_{log}}$ average of the logarithmic value of representing each ASCII code is calculated as follows:

$$AVG_{A_{log}} \sum_{i=1}^{N} \frac{Log_{val}}{N}$$

Now, the average value $AVG_{A_{log}}$ is compared with the Th_{val} threshold value and if it is found to be lesser then RSA based encryption is performed otherwise TDES and AES techniques are applied for encryption of the user data.

Next, Modified Best Fit Decreasing (MBFD) Technique is implemented that represents a best fit decreasing heuristic approach responsible for the selection of active host that demonstrates least CPU consumption fitting the current VM. In case of multiple results, a next check is performed base on the RAM and the one with lower RAM is selected. It is incorporated in order to reduce energy consumption by either sending rest of the servers to sleep mode or turn them off. The encrypted data of the last step is fed to MBFD using following algorithmic steps.

Modified Best Fit Decreasing (MBFD) algorithm

1. Input: $data_{w-v}$ // word to vector data
   $data_{e-v}$ // encrypted vector data
2. $Foreach\ element\ in\ data_{w-v}$
3. Calculate: $min_{priority} = max\ (energy)$ //highest energy instance was assigned with least priority
4. Initiate variable:
   $storage_{space} = []$ // allocate storage space with an empty matrix

5. Foreach i in $\text{data}_{\text{e-v}}$
6. $\text{P}_{\text{predicted}} = \text{estimate}(\text{data}_{\text{w-v}}, \text{data}_{\text{e-v}_\text{i}})$ //predicted priority value
7. $\text{P}_{\text{mean}} = \text{mean}(\text{P}_{\text{predicted}})$ //estimate the mean value of the predicted priority
8. If ($\text{P}_{\text{predicted}} < \text{P}_{\text{mean}}$)
9. $\text{storage}_{\text{space}} = \text{data}_{\text{e-v}_\text{i}}$ // allocate value to storage space
1. $P_{min} = P_{predicted}$ // store the priority value as the min priority
2. If $storage_{space} \neq []$
3. Assign: $data_{SED} = sort(data_{e-v_i}, P_{predicted})$ // sort the encrypted data according to priority values
4. End_{if}
5. End_{for}
6. End_{for}
7. Output: $data_{SED}$ // sorted encrypted data

The above algorithm inputs two variables, $data_{w-v}$ word to vector converted data and $data_{e-v}$ encrypted data. In an iterative manner it assigns the minimum priority to data values represented by highest energy. Further, this information is utilized for returning a sorted encrypted data as $data_{SED}$.

3.2. Encrypted Data Storage

The next step in proposed work is the wiser management of the data that has been encrypted in the last steps. This is achieved with the implementation of Cuckoo Search which is a nature inspired based taking advantage of meta-heuristic approach. It was put forward by Yang and Deb and is based on the brood parasitism demonstrated by cuckoo's **(Wang et al., 2020)**. Further CS and FFBPNN hybrid is used to locate secure cloud space based on the list of sorted encrypted data. FFBPNN trains the sorted list of encrypted data to find the underutilized or free and secure cloud storage space. The steps involved in the hybrid comprising Cuckoo Search and FFBPNN is as follows:

Cuckoo Search (CS) optimized FFBPNN Algorithm

1. Input: $Data_{training}$ // property list of representing senor data as training data
   cat_{data} // category data
2. Initialize variables for CS:
   E_{size} // number of eggs representing sensor nodes property
   E_{OT} // other eggs
   $data_{opt_{train}}$ // optimized training data
3. $R = length(data_{opt_{train}})$ //length of optimized training data
4. Initialize variable:
5. $data_{opt_{train}} = []$ //initialize optimized training data variable
6. $Foreach\ i\ in\ R$
7. $E_{current} = data_{opt_{train_i}}$ // representing selected $node_{property}$ from current data sensor nodes
8. $E_{threshold} = average(data_{opt_{train}})$ // representing $threshold_{property}$
9. $if\ E_{current} < E_{threshold} = other\ Th_{properties}$ // threshold properties
10. $F_{fit} = fit(E_{current}, E_{threshold})$
11. $F_{fit} = \begin{cases} 1, & True \\ 0, & False \end{cases}$
12. $Best_{property} = CS(F_{fit}, Data_{training}, CS_{variables})$
13. End_{for}
14. Initialize parameters for FFBPNN
    E_{num} // number of epochs
    Itr_{num} // number of iterations
    N_{num} // number of neurons
    Performance parameters: MSE, Mutations, Gradient and Validation
    Techniques: Levenberg Marquardt
    Data Division: Random
15. $Foreach\ i\ in\ data_{opt_{train}}$
16. If ($data_{opt_{train}}\ belongs\ to\ over_{load}$)

17. Assign $cat_1 == data_{opt_{train_i}}$ // abnormal sensor node
18. If ($data_{opt_{train}}$ $belongs\ to\ under_{load}$)
19. Assign $cat_2 == data_{opt_{train_i}}$ // abnormal sensor node
20. Else
21. Assign $cat_3 == data_{opt_{train_i}}$ // representing normal storage space
22. End_{if}
23. End_{for}
24. $Net_{storage} = Newff(data_{opt_{train}}, cat, N_{num})$ //call neural networks initialization function
25. $Net_{train} = train(Net_{storage}, data_{opt_{train}}, cat)$ //call training of the network

Verification of the model

26. $E_{c_{prop}} = property(E_{current})$ // property of current sensor node
27. $R_{verify} = simulate(Net_{train}, E_{current})$ //verification results of current sensor node
28. If $R_{verify} == True$
29. $Network_{property} = genuine$ //consider for data transmission
30. Else
31. $Attack_{node} = attacker$
32. End_{if}
33. Output: $Attack_{node}$ // identifies attack nodes

The above algorithm first performs optimization which is followed by FFBPNN based classification to find out the attacked nodes. CS exhibits dual property of local and global coverage that is regulated by discovery probability. Fitness function helps CS to identifying suitable region in the database. The optimized output is used as optimized training data to feed input layer of FFBPNN.

4. Results

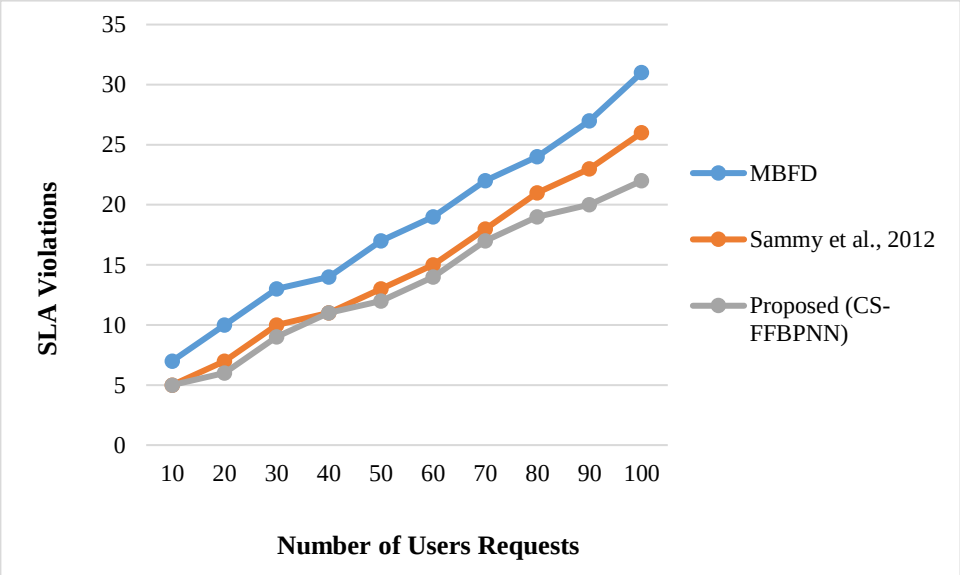
The current section evaluates the designed secure cloud computing architecture in terms of performance parameters, namely, transmission delay, PDR, energy consumption and SLA violations. The results of the proposed design are evaluated against the security offered by MBFD Algorithm and GA-PSO hybrid algorithm by **Sammy et al., 2012**. Table 1 shows the comparison of SLA violations of the proposed work with the existing algorithms.

Table 1 SLA Violations comparison

Number of Users Requests	SLA Violations		
	MBFD	Sammy et al., 2012	Proposed (CS-FFBPNN)
10	7	5	5
20	10	7	6
30	13	10	9
40	14	11	11
50	17	13	12
60	19	15	14
70	22	18	17
80	24	21	19
90	27	23	20
100	31	26	22

Service Level Agreement (SLA) comparison is performed to assure if the service offered by the cloud service provider agrees with the SLA requirements. It case of violations, cloud service provider has to pay the penalty to the service users. Figure 3 shows that proposed work exhibited least SLA violations as compared to MBFD and **Sammy et al., 2012** work. Average violations of 13.5, 14.9 and 18.4 were observed for proposed, **Sammy et al., 2012** and MBFD. Overall, it is observed that proposed work exhibited 9.39% lesser SLA violations as compared to Sharma and Reddy's work and 26.63% lesser as compared to MBFD.

Figure. 3 SLA Violations comparison



The next important parameter in cloud computing environment is energy consumption. It has to be minimal for a better performance. The observed energy consumption by proposed, MBFD and **Sammy et al., 2012** work is shown in Table 2.

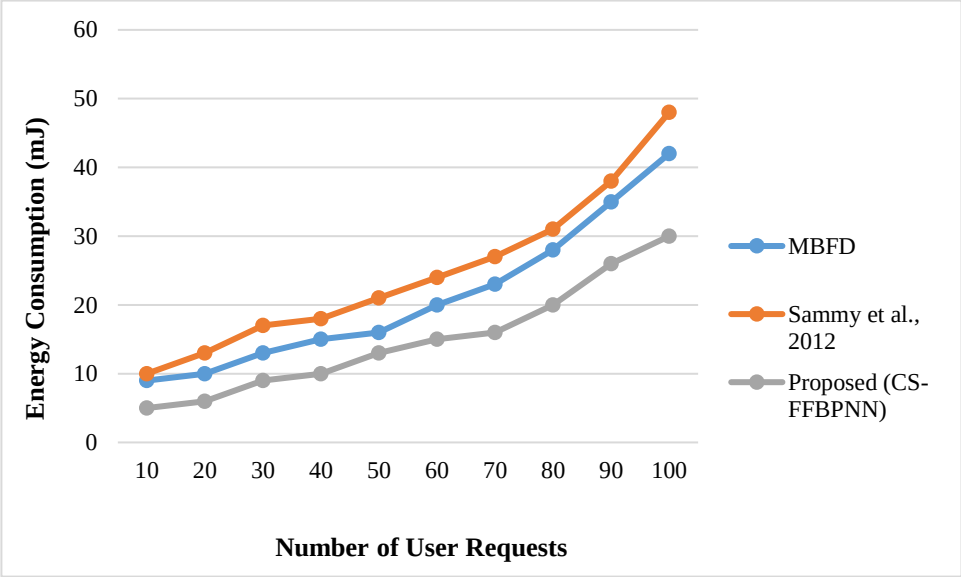
Table 2 Energy Consumption comparison

Number of Users Requests	Energy Consumption (mJ)		
	MBFD	Sammy et al., 2012	Proposed (CS-FFBPNN)
10	9	10	5
20	10	13	6
30	13	17	9
40	15	18	10
50	16	21	13
60	20	24	15
70	23	27	16
80	28	31	20
90	35	38	26
100	42	48	30

Figure 4 compares the energy consumption with varied number of user requests. It is observed that average energy consumption over 100 user requests for the proposed work is 15 mJ, however for MBFD it is 21.1 mJ and

for Sammy et al. it is 24.7mJ. It means that average energy consumption of the proposed work is 9.7 mJ and 6.1 mJ lower as compared to **Sammy et al., 2012** and MBFD, respectively. Thus, it demonstrates that there is reduction in energy consumption while utilizing CS-FFBPNN as compared to GA-PSO by **Sammy et al., 2012** and MBFD.

Figure. 4 Energy Consumption comparisons



Practically, data traffic in cloud environment is considerably increasing with the rising applications and merits of cloud computing. However, challenged speed of data transmission is reflected due to rising number of users, uploaded data size, rising traffic in communication channel and speed of the employed network. The proposed work postulated a CS-FFBPNN to reduce this observed delay and compared against existing work MBFD and GA-PSO based work as shown in Table 3.

Table 3 Delay comparison

Number of Users Requests	Delay (s)		
	MBFD	Sammy et al., 2012	Proposed (CS-FFBPNN)
10	0.914	0.757	0.485
20	1.265	0.864	0.674
30	1.451	1.012	0.754
40	1.958	1.157	0.854
50	2.153	1.322	0.984
60	2.542	1.543	1.354
70	2.958	2.024	1.547
80	3.145	2.517	1.954
90	3.845	3.024	2.584
100	4.512	3.957	3.14

Figure. 5 Delay comparison

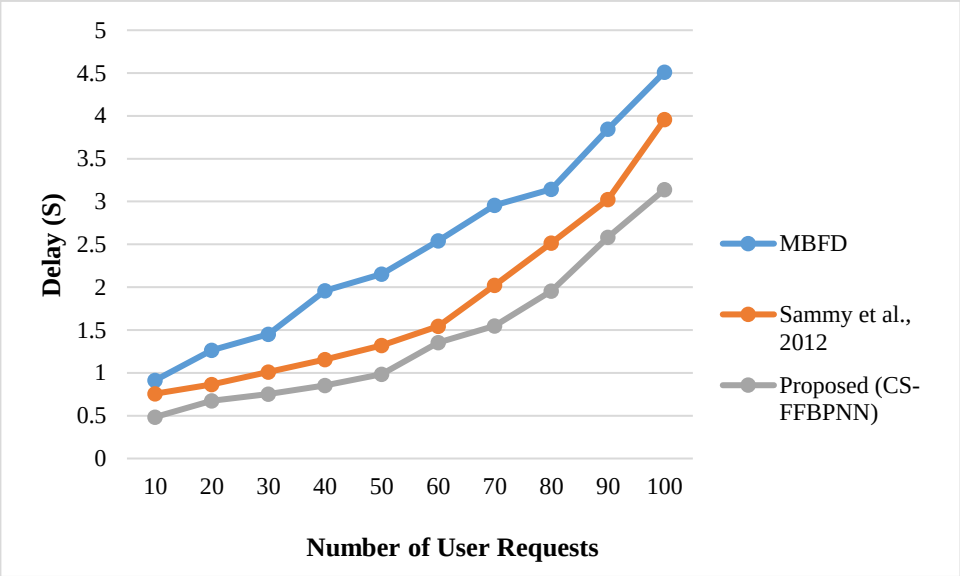


Figure 5 shows that the implementation of CS-FFBPNN in the proposed work has resulted in an average delay of only 1.433s as compared to 2.474s by MBFD and 1.81s by **Sammy et al., 2012** who had implemented GA-PSO hybrid in their work. Overall, the comparative analysis shows that the proposed work had outperformed the two existing works in offering a 21.16% and 42% higher speed data access and storage as compared to Sammy et al. and MBFD.

5. Conclusion

The present work has implemented strategies to offer secure cloud storage space for cloud service users. The security aspect has been addressed with the involvement of RSA with AES and TDES encryption methods followed by the implementation of MBFD for the storage of the encrypted data. Additionally, author had involved CS-FFBPNN to address the issues adjoining the allocation of storage space while selecting the most genuine VMs. The proposed work is evaluated in terms of SLA violations, energy consumption and delay observed in offering cloud services to end users. The comparative analysis against MBFD and GA-PSO had shown that the proposed work outperformed with lower SLA violations by 26.63% and 9.39%, lower energy consumption by 6.1% and 9.7% and delay by 42.08% and 21.16% over 100 analysed over user requests as compared to MBFD and GA-PSO hybrid, respectively. Overall, the proposed model exhibited a highly secure cloud computing environment as compared to existing approaches.

References

1. Bhise, A. S., & Latif, P. M. T. (2020). Secure Cloud Storage System by Integrating Trust with Role Based Access Control and Cryptographic Algorithm. In Techno-Societal 2018 (pp. 87-97). Springer, Cham.
2. Ibtihal, M., & Hassan, N. (2020). Homomorphic encryption as a service for outsourced images in mobile cloud computing environment. In Cryptography: Breakthroughs in Research and Practice (pp. 316-330). IGI Global.
3. Jouini, M., & Rabai, L. B. A. (2019). A security framework for secure cloud computing environments. In Cloud security: Concepts, methodologies, tools, and applications (pp. 249-263). IGI Global.
4. Kamboj, M., & Rana, S. (2017). Cloud security and energy efficiency. Advances in Computational Sciences and Technology, 10(5), 1245-1255.
5. Lei, C., Dai, H., Yu, Z., & Li, R. (2020). A service recommendation algorithm with the transfer learning based matrix factorization to improve cloud security. Information Sciences, 513, 98-111.
6. Longofono, S., Kline, D., Melhem, R., & Jones, A. K. (2019, October). Toward Secure, Reliable, and Energy Efficient Phase-change Main Memory with MACE.

In 2019 Tenth International Green and Sustainable Computing Conference (IGSC) (pp. 1-8). IEEE.

7. McAfee Labs Threats Report 2019, Accessed online at https://www.mcafee.com/
8. Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. Information Technology Laboratory: National Institute of Standards and Technology
9. Mohiuddin, I., Almogren, A., Al Qurishi, M., Hassan, M. M., Al Rassan, I., & Fortino, G. (2019). Secure distributed adaptive bin packing algorithm for cloud storage. Future Generation Computer Systems, 90, 307-316.
10. Nasr, A. A., Dubey, K., El-Bahnasawy, N. A., Sharma, S. C., Attiya, G., & El-Sayed, A. (2019). HPFE: a new secure framework for serving multi-users with multi-tasks in public cloud without violating SLA. Neural Computing and Applications, 1-21.
11. Njuki, S., Zhang, J., Too, E., & Dadye, H. B. (2019). Enhancing user data and VM security using the efficient hybrid of encrypting techniques. Journal of Theoretical and Applied Information Technology, 97(15).
12. Ramachandra, A. C., & Bhattacharya, S. (2020). Literature Survey on Log-Based Anomaly Detection Framework in Cloud. In Computational Intelligence in Pattern Recognition (pp. 143-153). Springer, Singapore.
13. Sammy, K., Shengbing, R., & Wilson, C. (2012). Energy efficient security preserving vm live migration in data centers for cloud computing. IJCSI International Journal of Computer Science Issues, 9(2), 1694-0814.
14. Sathishkumar, P., & Venkatachalam, V. (2019). Trust management scheme for authentication in secure cloud computing using double encryption method. International Journal of Business Intelligence and Data Mining, 15(1), 49-70.
15. Wang, Y., Tao, X., Zhao, F., Tian, B., & Sai, A. M. V. V. (2020). SLA-aware resource scheduling algorithm for cloud storage. EURASIP Journal on Wireless Communications and Networking, 2020(1), 1-10.
16. Yang, X. S., & Deb, S. (2010). Engineering optimisation by cuckoo search. arXiv preprint arXiv:1005.2908.
17. Yuan, H., Chen, X., Li, J., Jiang, T., Wang, J., & Deng, R. (2019). Secure Cloud Data Deduplication with Efficient Re-encryption. IEEE Transactions on Services Computing.