Indian Journals of Science and Technology, *Vol 9(16), DOI: 10.17485/ijst/2016/v9i16/86536, April 2016*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Neural based Security Approach for Cloud Databases using Counter Propagation

S. Jegadeeswari[1]*, P. Dinadayalan[2] and N. Gnanambigai[3]

[1] Bharathiar University, Coimbatore - 641046, Tamil Nadu, India; jega_sathya@yahoo.co.in

[2] Department of Computer Science, Mahatma Gandhi Government Arts College, Mahe - 673311, Kerala, India; pdinadayalan@hotmail.com

[3] Department of Computer Science, Indira Gandhi College of Arts and Science, Puducherry - 605009, Tamil Nadu, almsyzdykov@gmail.com India; dgnanambigai@hotmail.com

Abstract

Neural Network is an efficient implementing technique for cryptographic algorithms to provide security in cloud environment. Cloud Computing is an outsourced on-demand computing service, where Privacy preserving is very difficult to provide. Secured Data Sharing is important in cloud storage aspect. The proposed Neural Data Security Model ensures high data confidentiality and security in cloud database environment. This Model is a combination of , Sensitive Data Component (SDC) and Counter Propagation Neural Data Security Component (CPNDSC). The Sensitive data Component is implemented for storing the fragmented sensitive data. In Neural Data Security Component the Neural cryptographic algorithm is used to encrypt the sensitive data to enhance the confidentiality level by using Counter Propagation Neural Network.This reasearch is carried on cloud databases and artificial Neural Network that achieves high data security in cloud environment.

Keywords: Cloud Data Security, Counter Propagation, Data Confidentiality, Grossberg Layer, Kohonen Layer, Neural Network

Introduction

Neural Network is composed of a highly interconnected processing element called Neuron. This has limited number of input and output, it also learns the recognized patterns. Learning is of two types, supervised or unsupervised. The learning activity is monitored by a master in supervised learning. However, in the unsupervised realizing there is no expert for observing the learning[1]. Counter-Propagation Neural system is the procedure to consolidate an unsupervised Kohonen layer with the output layer. The operation for the Counter-Propagation Neural Network is identified with Learning Vector Quantization Network. The middle Kohonen layer can be adjustable by using lookup table, finding the nearest fit to an input stimulus and the output is to equilvalent mapped. The Counter-Propagation Network consists of two layers, both of which are implemented by bi-directional mapping between the input and output layers. To generate a classification pattern on the output layer the data are presented to the input layer. Here, the output layer gets an acknowledge with the further input vector and create the output layer on the system. The network acquire its nature from this counter-posing flow of data through Counter Propagation structure.

Counter-propagation is implemented using a structure of Kohonen paradigm to categorize the input sets into classification zones. Counter-Propagation is on the whole made

*Author for correspondence

up of processing component which learns to produce an output by using a particular input. Since the Kohonen layer includes competition, only one output is formed from the given input vector.The dynamic output from the competitive Kohonen layer value is one. The other various output nodes are zero, the main weighted output node is the winning node. For every dynamic processing node in the competitive layer to reproduce the secure pattern in the output layer. The number of competitive elements of the class the output from that element create a weightage in response to those competitive processing element and zero for all the others. As compared to learning with limited data set, combined learning improves the learning correctness. This integrate more data sets into the learning method which contribute parties that carry out learning on their own and also in data sets. In the significant growth of Cloud Computing, it has been further suitable than ever for users across the internet.While running the neural network learning to secured the data, to overcome the users' lack mutual trust on not revealing their respective private data sets. This solution support a random number of participants for each possessing randomly partitioned data sets efficiently. One critical issue is pertaining in the Internet-wide, which has combined neural network learning to protect the data of each participant.

The study is done on the various security issues in cloud computing through the different limitation of the existing cloud data security models. The work first focused on data confidentiality in cloud computing, then the Neural Network techniques. Storing of data in cloud environment is an important issue towards data confidentiality[2-4]. Cloud user are dependent on external CSP to store confidential user's sensitive data. Cloud database stores both low and high sensitive data and some important business information. Data leakage is the serious risk in cloud computing[5-8]. The users' privacy is seriously affected due to data leakage[9,10]. The various types of data confidentiality threats include the incompatible use of encryption, operational crash, authentication and authorization failures, and the lack of security in data storage location. The sever impact on customer data loss due to illegal deletions and modification[11].

[Chen et al., 2009] had proposed a learning model for Privacy Preserving, Multilayer Neural Networks. Privacy-Preserving two-party distributed algorithm using Back Propagation. This method provides a efficient protection mechanism for data sets including transitional results. It supports vertically partitioned data[12].

[Boser et al.,1990] just considers the two-party scenario though it supports arbitrarily partitioned data set. The existing methods have solved some issues, but still lacks an some clarification that adds mutual BPN network learning with privacy preservation in the multiparty setting[13].

[Schlitter et al., 2008] establishes a privacy preserving BPN network learning system that permits multi parties to perform BPN without leaking their own private data sets. The Only solution is proposed for horizontal partitioning data. Moreover, this procedure cannot defend the intermediate results, which possibly will also contain sensitive data, during the learning process[14].

[Bansal et al. 2014] proposed a result for randomly partitioned data. This enhanced scheme directly extending them to the multiparty surroundings, which will set up a computation difficulty in many participants, such a difficulty represents an incredible expenditure on each party[15].

[Blessy, et al. 2014] the cipher text data are arbitrarily partitioned and uploaded in the cloud. The BPN in Neural Network learning algorithm without awareness of any private data, the drawback is to enable multiparty learning without the help of trusted auditor[16].

[S. S. Sayyad, et al., 2012] the problem of Privacy Preserving for Vertically Partitioned Dataset was solved using Back Propagation Algorithm, permits a Neural Network to be trained without require either party to reveal its data to the others. It encrypts the private data with the system public key and then storing the cipher texts to the cloud[17].

[Jiawei et al., 2014] here each party encrypts its sensitive data locally and loads the ciphertexts into the cloud. The cloud executes the operations in learning algorithms on ciphertexts without knowing the private data. To maintain the operations over ciphertexts, it offers BGN "doubly homomorphic" encryption algorithm. The experimental result analysis shows secure, efficient, and accurate. It does not support the multi party collaborative[18].

[Sumeet Bajaj, Radu Sion, 2014] presented an outsourced database model to permit clients to execute SQL queries. The Trusted DB, a trusted hardware based relational database are of data confidentiality and no

limitations on query cost, and finally to implement the detailed query optimization method is a trusted hardware-based on query processing[19].

[Singh et al., 2015] had described how efficiently distribute data and preserve it in the cloud. Back Propagation Neural Networks assist to prepare data and load data to improve the accuracy of the outcome. In this method each party encrypts its own private data locally & uploads ciphertexts into the cloud, and restricted in such a way of data partitioning only for two parties[20].

[Ciriani, S. D. C. di Vimercati, 2007] had proposed a model which improves the data security of relational databases by combining fragmentation, such that of violating the sensitive data relations and encryption on rendering the data[21]. In addition, they attach unencrypted attributes are provide confidentiality, in order to allow the efficient selection process without searchable encryption technique, that was explained by [Ciriani, S. D. C. di Vimercati, 2013] based on these queries, different attribute combinations are to be revealed, ensuing in data redundancy[22].

[Vali et al., 2015] In this scheme, each owner encrypts their sensitive dataset using the AES cryptography methods and stores cipher texts in cloud. Cloud perform arthimetic operation on the ciphertext via BGN homomorphic algorithm. and the collaborative learning obtain place without revealing the private data of the owners[23].

[Damiani et al. 2003] explains about indexing for outsourced encrypted data are arranged to sustain the balance between efficiency and confidentiality. This mapping shoud be secret and stored at user's side. Aggregate queries, such as SUMs etc are not supported in this bucket-based method[24].

[Aleksandar Hudic., Shareeful Islam 2013] had discussed on various data privacy preservation threats in cloud. To process the splited fragments into unlink able locations Data confidentiality is processed by using fragmentation by applying relational databases. Thus it can be process both vertical and horizontal fragmentation on relational databases on the independent fragments[25].

[Thansekhar and Balaji, 2014] Hashing is a method to find the direct address of the data record on the storage disk of fragmented data. Dynamic hashing present a method that allow data to be added and removed on-demand. The dynamic hash structure in cloud uses an capable method for data location verification[26].

The rest of the paper is planned as follows, Section II introduces the architecture of the proposed work and discusses the components. Section III discuss about the implementation and the results. Section IV summarises the conclusions.

2. Proposed Model

The main aim of the research is to find and recognize the security issues which affect the performance of Cloud Computing. Also, to understand the security method which are organised to mitigate these security issues. The main goals of this research are:

- To give data isolation and data security on the sensitive data.
- To provide a cryptographic algorithm using Neural Network concepts key generation.
- To achieve Pattern Recognition and Pattern Completion by using Neural Network.

The Neural Data Security Model is an efficient and effective process for all kinds of queries, and the level of confidential is high. The Sensitive Data Component (SDC) provides the data confidentiality with sensitive data sets and efficient data isolation. The Component is used to encrypt and decrypt the sensitive data by using the Counter Propagation Neural Network. This model provides less expensive, high performance and an expandable storage system to enhance security.

Data confidentiality is implemented by fragmenting the sensitive data. These fragments are stored into different locations. The architecture of Neural cloud Data Security Model is illustrated in Figure 1. The architecture depict the flow of the Neural cloud Data Security applicaion. This application stores data in an efficient and confidential way in the cloud. The sensitive data are encrypted using the cryptographic algorithm with Neural Network and stored in encrypted format to achieve higher confidentiality level. The fragmented sensitive data are encrypted using the cryptographic algorithm which increases the level of confidentiality and protect data.

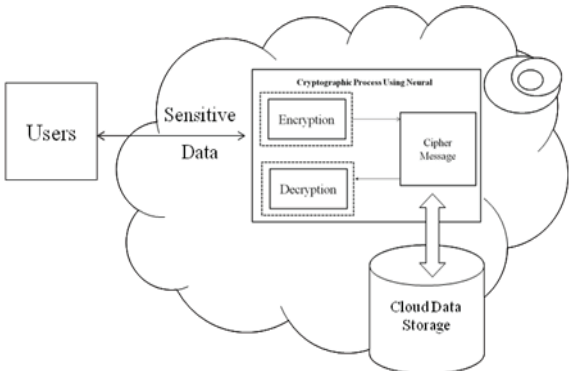

Figure 1. Architecture of the Neural Cloud Data Security Model.

The proposed model in this research work called, "A Neural Cloud Data Security", which assures High Confidentiality level and secured Storage Environment. Data users access data from a cloud service provider. The data owners ensure for authorized users to access the data from the service provider. the data owner encrypts the user's data and store it in the cloud storage provided by the CSP for its confidentiality.

The Model consist of Sensitive Data Component (SDC) and the Counter Propagation Neural Data Security Component (NDSC). In the Sensitive Data Component (SDC), the sensitive datasets are decomposed into sensitive data fragments. With the Neural Network the sensitive fragments are encrypted using cryptographic algorithm stored in different servers. The Counter Propogation Neural Network is implemented with two layered neural network, such as Kohonen and Grossberg layers. The Kohonen layer encrypts the sensitive data and the Grossberg layer decrypts the encrypted data to produce the original data.

2.1 The Sensitive Data Component (SDC)

The data sets from various data centers are stored and retrieved using cloud data bases. The Sensitive data Component is a dynamic form which reduce or enlarge their size based on the given fragmented data sets. The prime role of the SDC is to fragment only the sensitive content of the dataset for encryption. The SDC divides a single relation or a class of a database into different sensitive fragments.

Algorithm 1: ***The algorithm for Sensitive Data Component.***

Input - Cloud data Set from cloud database

Output - Fragmented Sensitive Encrypted data

Step 1. *Start.*

Step 2. *From the cloud data base the cloud data set are read which have sensitive data.*

Step 3. *Fragment the sensitive data sets.*

Step 4. *In dynamic hashing the fragmented data sets are stored .*

Step 5. *From the dynamic hashing the sensitive data sets are encrypted.*

5.1 *By using the cryptographic algorithm the encrypted mechanism is implemented with the neural network.*

5.2 *The public key is used to encrypt the sensitive data and the private key is used for decryption.*

Step 6. *By using the hashing structure the encrypted data is stored in the cloud.*

Step 7. *Stop.*

The SDC receives cloud data set as input and provides encrypted data as output that are stored in hash table.

2.2 Counter Propagation Neural Data Security Model

In the proposed model Neural Data Security Model, Neural Network concepts are applied with cryptographic technique to get well encrypted information. The Counter Propogation Neural Network concept is deployed in our cryptographic security technique, which is illustrated in figure 2. It has a two layered neural network concept such as Kohonen and Grossberg layers for implementing both the encryption and decryption. The Kohonen layer encrypts the sensitive data and the Grossberg layer decrypts the encrypted data to produce the original data. Counter Propagation Neural Network is to provide security so that, only the concerned user can access it. By securing the data, the unauthorized user is not allowed to access it. User sensitive data is stored after its encryption. When the data is needed, the user request for the data to CSP. The CSP checks the information generated with public and private keys. With the help of these keys the user is authenticated and delivers the data only if the user is valid user.

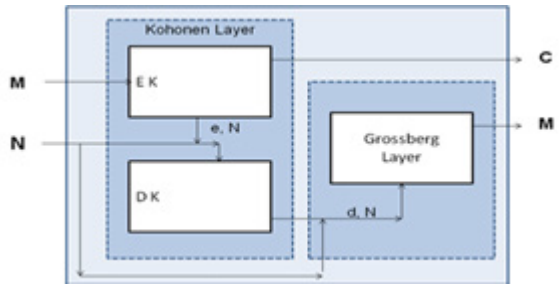

Figure 2. The Structure of the Counter Propagation Neural Data Security Model.

The Counter Propagation Neural Data security component is designed to generate a complication of the keys generation in the cryptographic process. The Structure of Counter Propogation Consists of Encrypted Kohonen Layer, Decrypted Kohonen Layer and Grossberg Layer. Counter-Propagation Neural Networks is composed of two layers, the Kohonen layer and Grossberg layer. The first Kohonen layer implements encryption key generation and second Kohonen layer deals with decryption key generation. Both Kohonen layers use unsupervised learning mechanism. Kohonen layer is the input layer. The grossberg layer is the second layer which produces the output. Grossberg layer uses supervised learning having the target patterns.

The encrypted data from the Kohonen layer is used to generate a key. This training process is continued until the key value reaches one. Figure 3 shows the kohonen layer

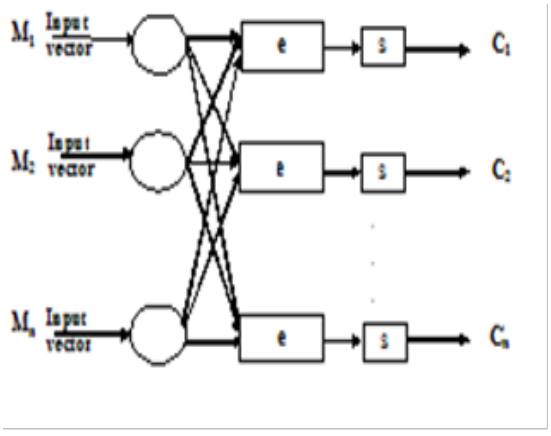

Figure 3. Structure of the Kohonen Data Security Model for Encryption.

for the encryption. Algorithm 1 gives the procedure for Encrypted Kohonen layer for Encryption.

Algorithm2: ***The algorithm for kohonen Neural Data Security Model for Encryption.***

Input - Cloud data Set Value (M,N,E={e1,e2,.....en})

Output - Encrypted Sensitive data ,C

Step 1. *The input vectors p and q is taken and N is obtained from the random set for network training. The function ø(N)is used determine the encryption key .evaluate ø (N), i.e. (p-1) (q-1)*

Step 2. *To get e, by choosing a random encryption key "e "where 1 < e < ø (N), such that the result of GCD must be equal to one. The training process continues... i.e., GCD (e, ø (N)) =1, this can be achieved parallel.*

Step 3. *The output of the network added with N i e. N = p * q to produce Public key KU= {e, N}.*

Step 4. *With the key 'ku' the encryption.*

$C=m^e \pmod n$ *where C-cipher text, e-encryption key, m-message.*

The structure of the Encrypted Kohonen layer consists of an input layer and an encrypted computational layer. In the input layer these are only fan out points which do not have any process. This input layer has original message M, N & E where M is the actual message, N is the product of p and q, E-set of prime numbers where e lies between 1< e < n. The Kohonen layer calculates the encryption key and generates the public key as output. This layer computes ø (N) and e (encryption) where e lies between 2 and ø(N). This layer continues to process until GCD (e, ø (N)) =1. The output of the Computational layer is encrypted from the given inputs. Finally the encrypted Kohonen layer generates the encrypted message by using KU = {e, N} where KU is public key, e is encryption key and N = p * q. By relating in the equation C =M^e (mod n), e generates the encoded form of message C. Where e is a random number generated between the input layer and the competitive layer. The random matrix d is generated between the competitve and output layer. The Euclidean distance between input vector and the weights of each competitive layer node is calculated by the Kohonen network using unsupervised learning. This layer finds the winner node with the shortest distance. Then the Grossberg network takes the winner node's as random number KU^e as output and adjusts the output random number as K^d corresponding to supervised learning.

$$EK_{new}^{ei} = EK_{old}^{ei} + \mu[EK_{new}^{ei}, \gamma(t)] \quad (1)$$

The training process of Kohenan layer is calculated by using equation (1). EK_{new}^{ei} and EK_{old}^{ei} are represented encrypted keys. The new encrypted EK_{new}^{ei} is replaced by $EK_{old}^{ei+i[\gamma(t),EK_{old}^{ei}]}$. The competition process is stopped when the winning neuron EK_{new}^{ei} achieves value one. So, the encryption key value is generated. The input vector sets are α, β where α, β are two distinct random numbers set. Here ρ is the product of α, β. μ used to calculate GCD of EK_{old}^{ei} and $\gamma(t)$ represents the product of α -1 and β -1 where t denotes the learning rate.

$$C = \delta^{EK_{new}^{ei}ek}, \mathrm{mod}\,\rho \quad (2)$$

Where C is the cipher text, is the Message, EK is the encryption key and is product of random numbers. After unsupervised learning, the training begins to the second level ofsupervised learning. In the decrypted Kohonen layer, the decryption technique is used to generate the original message . The training process continues until the decryption reaches e * d = 1 mod ø(N) where 0 ≤ d ≤ N of the Kohonen layer contains the output of Kohonen layer and N. The encrypted Kohonen layer consist of E neurons where 2<e<ø(N) -1.These neurons work simultaneously. If GCD(e, ø(N))=1 arrives this is the winner neurons, other neurons set the value as 0. So, the winner neuron generates cipher text. The decrypted Kohonen layer consists of d number of neurons, each neurons calculates private key{d,N}, using the formula e * d = 1 mod(n) where d lies 0<d<N. The output of decrypted Kohonen layer (i,e) {d, n}.

Figure 4 shows the structure of Grossberg layer for decryption. Algorithm 3 explains the procedure of the Grossberg layer for decryption. The Grossberg layer uses supervised learning (i.e) it is having the target patterns. The input of Grossberg layer is {d,n} which is the output of the encrypted Kohonen layer. The computation of the Grossberg layer is using the method $M=C^d$ mod n where C-cipher text, d-decryption, M-original message, compares itself with the actual pattern of the target pattern. If target and actual messages match with the original message then it a is valid one, otherwise the message is invalid.

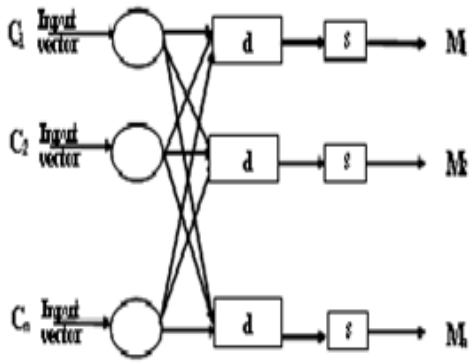

Figure 4. Structure of the Grossberg Neural Data Security Model for Decryption.

Algorithm 3: ***The algorithm for Grossberg Neural Data Security Model for decryption.***

Input -- Cloud data Sensitive Encrypted data ={e, N}

Output -- Original Message

Step 1. *The input for the Decrypted Kohonen layer is taken from the output of the encrypted kohonen layer. Decrypted Kohonen consists of d number of neurons for d=0 to N.*

Step 2. *To achieve private key, the computation of decrypted kohonen layer uses the formula for finding decryption key d: (i.e) e * d = 1 mod ø(N) and 0 ≤ d ≤N.*

Step 3. *The output the decrypted kohonen layer produces Private Key KR= {d, N}*

Step 4. *The output of DK is the input of Grossberg layer which is generating and recognising the message.*

Step 5. *The Grossberg layer computes the actual message by using the formula $m= C^d$ (mod n) where C-cipher text, d-decryption key, m-message.*
Once m is obtained, the user can get back the data.

Step 6. *If actual message and original message matches, the original message is valid*
Otherwise the given message is invalid.

The input for the decrypted Kohonen layer is the output of Kohonen layer of the encrypted message, in which the key is generated. The decrypted Kohonen layer calculates the decryption key with the private key d. This layer computes ø (N) and d (decryption key) where d lies between 0 ≤ d ≤N. This layer continuous until e * d = 1 mod ø (N) is satisfied. The output of{d, N} is the decrypted form of the given inputs. The output layer generates the decrypted message by using Kr= {d, N} where Kr is the private key, N=p*q and 'd' is a random matrix between the competitive layer and the output layer. In competitive layer, neurons are generated dynamically instead of being

located in advance. Then the Grossberg network takes the winner node as random number K^e as output and adjusts the output random number as K^d according to supervised learning. The supervised learning method is to learn a mapping from input objects to desired outputs, given training sets that consist of input and output pairs. If the chosen node in the competitive layer has already existed, then it will change according to following formula:

$$DK_{new}^{di} = DK_{old}^{di} + \pi[\gamma(t), DK_{old}^{di}] \quad (3)$$

In equation (3), for the training process of Grossberg layer we are generating decrypted key by using DK_{new}^{di} and DK_{old}^{di}. The new decrypted DK_{new} is generated by satisfing the equation 4,

$$EK_{old}^{di} * DK_{old}^{di} = 1 \bmod \gamma(t) \quad (4)$$

the competition process stopped when the equation satisfied and it is the neuron i.e new key value generated.

$$\delta = C^{DK_{new}^{ei}}, \bmod \rho \quad (5)$$

Where C is the cipher text, is the Message, DK is the decryption key. By applying in the equation 5, we generate the decrypted message . This message has to be compares with the actual target pattern. If target pattern and actual message matches, then the message is a valid one, otherwise the message is invalid. The NDSC model is used to identify the task behavior and predict the corresponding random number for the both the keys. Thus, the predicted random number can reduce the CPU computation time.

3. Result and Discussion

To evaluate the impact of our improvements from existing schemes we have simulated the Cloud Environment using CloudSim. The test was done for different kinds of cost analysis such as retrieval, insertion, deletion and updation. Each test contains sensitive instances of sample data sets which consists of user's confidentiality data. In the retrieval process analysing, we made use of select queries, such as Scalar, Range and Nested queries. The execution time for each operation have been successfully analysed and executed. From the simulation using CloudSim test, 42% of data sets are sensitive confidential data are done on data sets. In this section, we numerically evaluate the performance of our proposed scheme in terms of execution of queries and compare it with the existing techniques.This is implemented in this approach with the following steps:

- The first algorithm is designed to fragment the user data into sensitive data only using the sensitive data component mechanism and the sensitive data are arranged in different locations.
- The second stage of the process is of two algorithms implemented using Counter Propagation Neural Network for encrypting and decrypting the data in the cloud environment.

3.1 Computation of Sensitive Data Component

Consider the relation R, which contains Employee details. The Employee details of a company are described with different attributes. The table describes about the Employee details of a company. In the company the Employee details have to be maintained confidential as they are sensitive data. The company needs to maintain all the details of the employee, but there are some details that need not to be maintained confidentially. The attributes such as Employee ID, Employee Name, Employee No, Department, Employee Type, Address, Pincode and Salary about the employee are maintained. In this, some particular attributes need to be maintained as sensitive data such as Employee ID, Reference No and salary in the Table 1.

3.2 Computation of Encryption using EK Layer

Let M =A101 be the key attribute which can be encrypted using Counter Propagation neural network, where M is the user requested data. In this model, the data has to process in two layers for encryption and decryption. The Counter Propagation Neural Network consists of Kohonen layer and Grossberg layer. The input of CPN is the attribute of M which is passing to Kohonen layer. The training of Kohonen layer is self-organizing map. The input layer is fully connected to Kohonen layer which is the encryption process. To analyse the data process in the encryption, we process it with example values. The output

layer is producing the public encryption key={e, N}. This output layer obtains public key of recipient ={e, N} using the rule C = M^e mod N. where C-cipher text, e-encryption key, m-message. The computation of Kohonen layer follows: Let p = 227 and q = 229 be the input vectors with N = p * q = 227 * 229 = 51983 taken from the random set for network training. We calculate the encryption key using the function ø(N). To achieve e, by Selecting at random encryption key 'e' where 1 < e < ø (N), such that the result of GCD must be equal to one. Calculate ø(N), i.e. (p-1) (q-1) = (227 - 1) * (229 - 1) = 35964. The Kohonen layer has ø (N) neurons. The Kohonen element with the gcd (e, ø (N)) =1 is the winner and outputs a one to the output layer. This is a competitive win, so all other processing elements are forced to zero for that input vector. The output of the network added with N i e. N = p * q to produce Public key KU= {e, N} = 45329, 51983. With the key 'ku' the encryption = me (mod n), where Cipher text C generated. C = A101 45329 mod 51938 = 34529 15059 19918 15059. Thus C is the encrypted message, which is in the coded format. These coded sensitive data are stored in the cloud data center.

3.3 Computation of Decryption DK and Grossberg Layer

In the second layer, the decryption is implemented using DK layer Grossberg layer. A DK layer consists of a ø(N) number of neurons, and each neurons contains a computation units. Each neurons receives its inputs directly from the EK layer. The output of DK layer is the private key During the training of a Grossberg layer the decryption key d,e * d = 1 mod ø(N) and 0 ≤ d ≤N is processed many times .i.e d = 18977 * 29693 = 563484061 such that the mod value is 1. The output of the DK layer is the secret private decryption key R = (D,N) =18977,51983. The input of grossberg layer is secret private decryption key, R = (D,N) = {18977,51983}. The training of Grossberg layer is supervised learning which incorporates an target output (actual attribute). The Grossberg layer processes the output for Kohonen layer and compares its resulting outputs against the desired outputs. Also the Grossberg layer calculates the decrypted Ciphertext M = C^d mod N = (34529 15059 19918 15059 189779693 mod 51983. Thus the original Message is retrieved by decrypting as output M = A101. Therefore, the message is processed securely an from the user and cloud provider.

3.4 Performance Analysis

To evaluate performance we have taken retrieval queries like Scalar, Range and Nested. In retrieval process analysis, we got the results for unfragmented and fragmented types of table data by using Back Propagation and Counter Propagation algorithms on the cloud data as described in Table 2. From the experiment done for scalar query, the execution time for unfragmented and fragmented, results are lower when compared with unfragmented data. The confidentiality comparison is done between Back Propagation algorithms and Counter Propagation algorithm, in which Counter Propagation is efficient than the existing Back Propagation algorithm.

Table 2. Time executions Comparison of Neural concepts

Queries	Un-fragmented table		Fragmented Table	
	BP	CP	BP	CP
Scalar	74.4	69.6	50.3	38.9
Range	81.9	72.4	52.7	40.8
Nested	88.2	78.2	55.6	43.4

The retrieval of data by using the counter propagation is faster and the time taken for the execution of the range, scalar and nested query is very minimum on the fragmented data. Such as scalar query takes time about processing 10 query at 88.8ms for Back Propagation on un-fragmented data, while the same queries are processed for Counter Propagation in un-fragmented data the time is 78.2ms. Where the same queries executed in fragmented data the time taken is 55.6ms and 43.5ms for Back Propagation and Counter Propagation respectively. Therefore for scalar query the time execution is very minimum to process on fragmented data.

While in the range query, the time taken to process 10 query at 81.9ms and 72.4ms using Back Propagation and Counter Propagation respectively in un-fragmented. Where the same queries executed in fragmented data the time taken is 52.7ms and 40.8ms for back propagation and counter propagation respectively. Therefore for range query the time execute is also very minimum to process on fragmented data when compared to un-fragmented. Similarly for nested query also the time execute is also very minimum to process on fragmented data are projected in chart-1 clearly. Thus, from the results obtained from the simulation, we come to a conclusion that the

time taken for the queries is more in unfragmented data when compared to fragmented data.

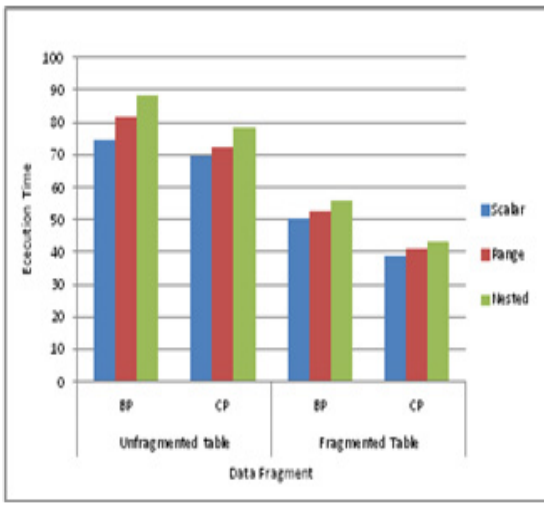

Chart1. Time executions Comparison of Neural concepts.

4. Conclusion

The major aspect of cloud is to provide confidentiality while accessing set of information from a cloud database with a high security level. This research proposed a new cloud data security model, A Neural Data Security Model ensures high confidentiality and security in cloud data storage environment for achieving data confidentiality. The Sensitive data Component deals for storing the fragmented sensitive data. that increase the confidentiality level. The fragmented sensitive data are stored efficiently using dynamic hashing concept. The Counter Propagation Neural Data Security Component is implemented for encrypting and decrypting the sensitive data by using Counter Propagation Neural Network. The Structure of Counter Propogation Consists of Encrypted Kohonen Layer, Decrypted Kohonen Layer and Grossberg Layer. This methodology is efficient and effective for all kinds of queries requested by the user data with different test cases. The performance of this work is better when compared with the existing work of Back Propagation. The new model is better than the conventional cloud data security models as it achieves a high data confidentiality level.

5. References

1. Bonde SA, Shubhangi P, Surbhi A, Satish S. Review techniques of data privacy in cloud using back propagation neural network. International Journal of Emerging Technology and Advanced Engineering. 2014 Feb; 4(2):668–72.
2. Jothi Neela T, Saravanan N. Privacy Preserving Approaches in Cloud: A Survey. Indian Journal of Science and Technology. 2013 May; 6(5):4531–5. Doi: 10.17485/ijst/2013/v6i5/33258.
3. Lee J-Y. A Study on the use of secure data in cloud storage for collaboration. Indian Journal of Science and Technology. 2015 Mar; 8(S5):33–6. Doi: 10.17485/ijst/2015/v8iS5/61462
4. Kalaichelvi R, Arockiam L. EnBloAES: A unified framework to preserve confidentiality of data in public cloud storage. Indian Journal of Science and Technology. 2015 Aug; 8(18). Doi: 10.17485/ijst/2015/v8i19/72272.
5. Subashini, Kavitha V. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications. 2011; 1–11.
6. Zissis D, Lekkas D. Addressing cloud computing security issues. Future Generation Computer Systems. Elsevier Journal. 2012; 583–92.
7. Kirubakaramoorthi R, Arivazhagan D, Helen D. Analysis of Cloud Computing Technology. Indian Journal of Science and Technology. 2015 Sep; 8(21). Doi: 10.17485/ijst/2015/v8i21/79144.
8. Sugumar R, Sheik Imam SB. Symmetric Encryption Algorithm to Secure Outsourced Data in Public Cloud Storage. Indian Journal of Science and Technology. 2015 Sep; 8(23). Doi:10.17485/ ijst/2015/v8i23/79210.
9. Durairaj M, Manimaran A. A Study on Security Issues in Cloud Based E-Learning. Indian Journal of Science and Technology. 2015 Apr; 8(8):757–65. Doi: 10.17485/ijst/2015/v8i8/69307.
10. Kalpana V, Meena V. Study on data storage correctness methods in mobile cloud computing. Indian Journal of Science and Technology. 2015 Mar; 8(6):495–500. Doi: 10.17485 /ijst/2015/v8i6/70094.
11. Rajathi A, Saravanan N. A Survey on Secure Storage in Cloud Computing. Indian Journal of Science and Technology. 2013 Apr; 6(4):4396–401. Doi: 10.17485/ijst/2013/v6i4/31871.
12. Chen T, Zhong S. Privacy-Preserving Backpropagation Neural Network Learning. IEEE Trans Neural Network. 2009 Oct, 20(10):1554–64.
13. Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. Handwritten digit recognition with a back-propagation network. In Advances in Neural Information Processing Systems, Morgan Kaufmann. 1990; 396–404.

14. Schlitter N. A protocol for privacy preserving neural network learning on horizontal partitioned data. Proceedings of the Privacy Statistics in Databases (PSD). 2008 Sep; 1–9.
15. Chen BT, Zhong S. Privacy Preserving Back-Propagation Neural Network Learning over Arbitrarily Partitioned Data. Neural Computing Applications. 2011 Feb; 20(1):143–50.
16. Blessy S et al. Privacy Preserving Back Propagation Neural Network learning in signature scheme. International Journal of Engineering Research and Applications, IJERA. 2014 Mar; 1(4). ISSN: 2248-9622,
17. Sayyad SS, Kulkarni PJ. Privacy Preserving Back Propogation Algorithm for Distributed Neural Network learning. International Journal for scientific and Research Publication. 2012; 2(3):133–6. ISSN: 2250-3153.
18. Yuan J, Yu S. Privacy preserving Back-Propagation Neural Network Learning Made Practical with Cloud Computing. IEEE Transactions on Parallel and Distributed Systems. 2014; 25(1):212–21.
19. Bajaj S, Sion R. Trusted DB: A trusted hardware-based database with privacy and data confidentiality. IEEE Transactions on Knowledge and Data Engineering. 2014; 752–65.
20. Singh D, Satav S, Mulla M. Privacy Preserving Made Practical with Cloud Computing Back Propagation and Neural Network Learning. International Journal of Innovations and Advancement in Computer Science. 2015; 4(1):153–8.
21. Ciriani V, di Vimercati SDC, Foresti S, Jajodia S, Paraboschi S, Samarati P. Fragmentation and encryption to enforce privacy in data storage. Proceedings of the 12th European Symposium on Research in Computer Security (ESORICS'07), Dresden, Germany, LNCS, Springer-Verlag. 2007; 4734:171–86.
22. Ciriani V, di Vimercati SDC, Foresti S, Jajodia S, Paraboschi S, Samarati P. Combining fragmentation and encryption to protect privacy in data storage. ACM Trans Inf Syst Secur. 2013; 22(1):1–30.
23. Valli N, Anwar Basha H. Back-Propagation Neural Network Learning with Preserved Privacy using Cloud Computing. IOSR Journal of Computer Engineering (IOSR-JCE). 2015; 17(2):75–9.
24. Damiani E, di Vimercati SDC, Jajodia S, Paraboschi S, Samarati P. Balancing Confidentiality and Efficiency in Untrusted Relational DBMSs. ACM Transaction. 2003; 93–102.
25. Hudic V, Islam S, Kieseberg P, Weippl ER. Data Condentiality using Fragmentation in Cloud Computing. International Journal of Communication Networks and Distributed Systems. 2012; 1(3):325–9.
26. Thansekhar, Balaji. An Efficient Dynamic Indexing and Metadata Model for Storage in Cloud Environment. IEEE International Conference on Innovations in Engineering and Technology ICIET'14. 2014; 124–9.

Table 1. Sensitive Table Attributes for Sensitive Confidentiality Level

Emp ID	Ref No	Salary
A101	3102032	36024
B102	3102232	40025
F232	3102534	40025
A234	3106444	46025
A634	3102533	40420