The International Journal Of Engineering And Science (IJES)
|| Volume || 4 || Issue || 5 || Pages || PP.92-98|| 2015 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

# Securing Privacy of User's Data on Cloud Using Back Propagation Neural Networks

[1]Manikamma Malipatil,[2]Susan Shieh,[3]Deepa Mathpati,[4]Nandita Madkunti
*[1,2,3,4]Department of Computer Science, Godutai Engineering College for Women*

## ABSTRACT

*To improve the accuracy of learning result, in practice multiple parties may collaborate through conducting joint Backpropagation neural network learning on the union of their respective data sets. During this process no party wants to disclose her/his data to others. Existing schemes supporting this kind of collaborative learning are either limited in the way of data partition or just consider two parties. There lacks a solution that allows two or more parties, each with an arbitrarily portioned data set, to collaboratively conduct the learning. this paper solves this open problem by utilizing the power of cloud computing. In our proposed scheme, each party encrypts his/her private data locally and uploads the ciphertexts into the cloud. The cloud then executes most of the operations pertaining to the learning algorithms over ciphertexts without knowing the original private data. To support flexible operations over ciphertexts, we adopt and tailor the BGN 'doubly homomorphic' encryption algorithm for the multi-party setting..*

***Keywords*** *– Backpropagation algorithm, Ciphertext, Cloud computing, Doubly homomorphic encryption, Neural networks.*

Date of Submission: 01-May-2015 Date of Accepted: 15-May-2015

## I. INTRODUCTION

Cloud computing is a computing term or a metaphor that evolved in the late 2000s, based on utility and consumption of computer resources. It can be defined as sharing of resources to achieve coherence and economics of scale. Virtually, we are going to use some other person's resources. Cloud computing involves different groups of remote servers and software networks that allow different kinds of data sources be uploaded for real time processing to generate computing results without the need to store processed data on the cloud. In cloud computing, the word cloud, is used as an image/metaphor for "the internet", so the phrase cloud computing means "a type of internet-based computing", where different services such as, servers, storage and applications are delivered to an organization's computers and devices through the internet. Clouds can be classified as public, private or hybrid [1][2].

Backpropagation, an abbreviation for "backward propagation of errors", is a common method of training artificial neural networks used in combination with an optimization method such as gradient descent. The method calculates the gradient of a loss function with respects to all the weights in the network. Backpropagation requires a known, desired output for each input value in order to calculate the loss function gradient. It is therefore usually considered to be a supervised learning method. The goal of any supervised learning algorithm is to find a function that best maps a set of inputs to its correct output. An example would be a simple classification task, where the input is an image of an animal, and the correct output would be the name of the animal. Artificial neural networks ANNs) are a family of statistical learning algorithms inspired by biological neural networks and are used to estimate approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons", which can compute values from inputs, and are capable of machine learning as well as pattern recognition thanks to their adaptive nature.

The BP learning process works in small iterative steps: one of the example cases is applied to the network, and the network produces some output based on the current state of its synaptic weights (initially, the output will be random). This output is compared to the known-good output, and a mean-squared error signal is calculated. The error value is then propagated backwards through the network, and small changes are made to the weights in each layer. The weight changes are calculated to reduce the error signal for the case in question. The whole process is repeated for each of the example cases, then back to the first case again, and so on. The cycle is repeated until the overall error value drops below some pre-determined threshold. At this point we say that the network has learned the problem "well enough".

## II. SYSTEM DESIGN

In the proposed system each party participant first encrypts her/his private data with the system public key and then uploads the ciphertexts to the cloud; cloud servers then execute most of the operations pertaining to the learning process over the ciphertexts and return the encrypted results to the participants; the participants jointly decrypt the results with which they update their respective weights for the BPN network. This process is carried out by three modules, namely, Trusted Authority, Participating Parties and Cloud Server.

### 2.1 Trusted Authority

A trusted authority that is trusted by all the parties, when an user gives the input data, to trusted authority it computes a secret key and creates a ciphertext and then this can be retrieved by the user. The trusted authority uses RSA algorithm for the creation of ciphertext. This ciphertext is also known as encoded information because it contains a form of the original plain text that is unreadable by human or computer without the cipher to decrypt it. The flow of trusted authority is shown in Fig. 2.1.

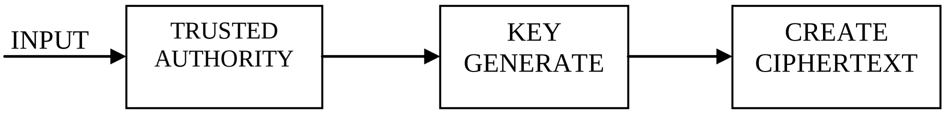


Fig 2.1: Trusted Authority

### 2.1.1 RSA algorithm

[4]Rivest-Shamir-Adleman, RSA algorithm is widely accepted scheme for public key cryptography. It utilizes modular arithmetic and factorization of large numbers. The public and private keys are generated based on the following rules.

a) Choose two large prime numbers p and q such that the product is equal to n.

b) Find a number e that is relatively prime to (p-1)(q-1). Two numbers are said to be relatively prime if they have no common factors except one. The public key consists of {e,n}.

c) Find a number d such that de=1mod((p-1)(q-1)). In other words d and e are multiplicative inverse of each other modulo ((p-1)(q-1)). Private key consists of {d.n}.

d) For any key integer p<n the following property holds $p^{de}$(mod n)=p(mod n).

e) Suppose p is an integer that corresponds to a block of text. RSA encrypts p as follows, $C=p^{e}$(mod n).

f) To decrypt the ciphertext C, RSA algorithm uses,
$P=C^{d}$(mod n)
$=(p^{e})^{d}$(mod n)
$=p^{de}$(mod n)
=p(mod n).

The flow chart is shown in the Fig.2.1.1

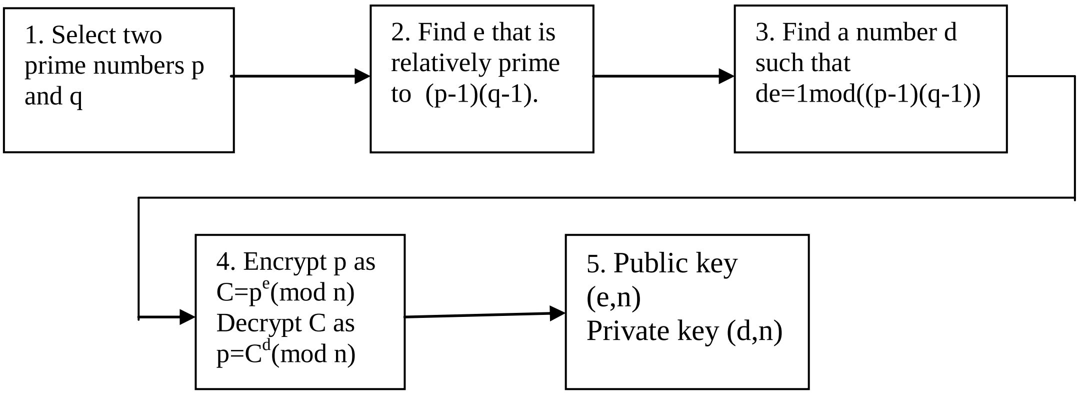

1. Select two prime numbers p and q
2. Find e that is relatively prime to (p-1)(q-1).
3. Find a number d such that de=1mod((p-1)(q-1))
4. Encrypt p as C=p^e(mod n) Decrypt C as p=C^d(mod n)
5. Public key (e,n) Private key (d,n)

Fig 2.1.1: RSA Algorithm Flowchart

### 2.2 Participating parties

The generated key is issued to the participating parties by trusted authority. Participating parties does not trust any other party than the trusted authority. The participating parties can retrieve the ciphertext from trusted authority whenever required. Then the user can upload the data(text, image, video etc) to the cloud server. The cloud server is not allowed to fully learn about the sensitive data, such as original data sets and intermediate data. The flow of participating parties is shown in Fig. 2.2.

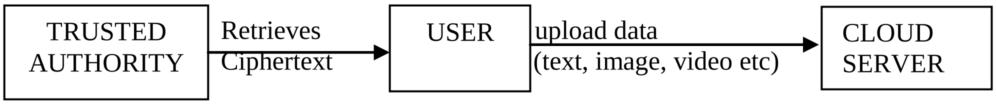


Fig 2.2: Participating Parties

### 2.3 Cloud Server

Cloud servers execute most of the operations pertaining to the learning process over the ciphertexts and return the encrypted results to the participants; the participants jointly decrypt the results with which they update their respective weights for the BPN network.

### 2.3.1 BPN Network

BPN network learning algorithm, our proposed scheme lets each party encrypt her/his private data set and upload the encrypted data to the cloud, allowing the cloud servers to perform most of the operations, i.e., additions and scalar products. Neural network is an interconnected group of nodes, same as that of the vast network of neurons in our brain. A neural network of hand writing recognition is defined by a set of input neurons which may be activated by the pixels of an input image. After being weighed and transformed the activities of these neurons are then passed on to other neurons [5][6]. This process is repeated until final output neuron is activated. The following Fig 2.3.1 shows the interconnected group of neural network, this is also known as the generalized network

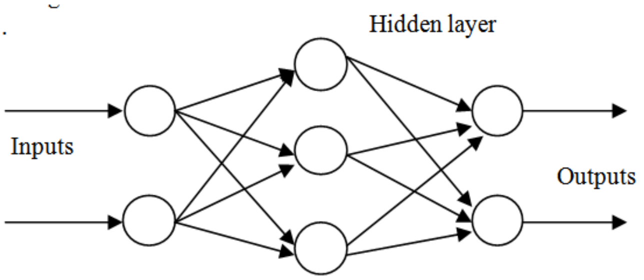


Fig 2.3.1: A Generalized Network

Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another. Each neuron receives a signal from the neurons in the previous layer, and each of those signals is multiplied by a separate weight value. The weighted inputs are summed, and passed through a limiting function which scales the output to a fixed range of values. The output of the limiter is then broadcast to all of the neurons in the next layer. So, to use the network to solve a problem, we apply the input values to the inputs of the first layer, allow the signals to propagate through the network, and read the output values.

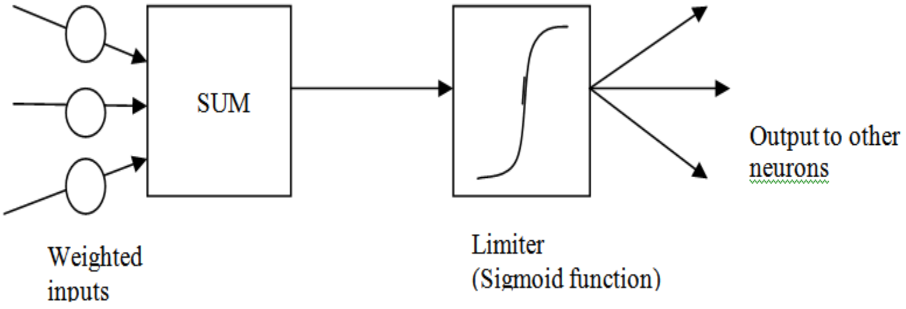


Fig 2.3.2:The structure of a neuron

The Fig 2.3.2 shows the structure of a neuron. Since the real uniqueness or 'intelligence' of the network exists in the values of the weights between neurons, we need a method of adjusting the weights to solve a particular problem. For this type of network, the most common learning algorithm is called Back Propagation (BP). A BP network learns by example, that is, we must provide a learning set that consists of some input examples and the known-correct output for each case. So, we use these input-output examples to show the network what type of behavior is expected, and the BP algorithm allows the network to adapt. The Bp learning process works in small iterative steps. Here, we must provide a learning set that consists of some input examples and the known correct output for each case.

### 2.3.3 Secure scalar product and addition with cloud

This algorithm allows multiple parties to perform secure scalar product and homomorphic addition operations on ciphertexts using cloud computing, encrypts her/his data with the system public key and uploads the ciphertexts to the cloud. The cloud servers compute the sum of original messages based on their ciphertexts. If the original messages are vectors, the cloud computes the scalar product of the vectors. During this process the cloud does not need to decrypt nor learn about the original messages. The final result of the sum or scalar product is returned to all the parties in ciphertext [7].

### 2.3.4 Secure sharing of scalar product and sum

For privacy protection, the actual intermediate results shall also be protected and cannot be known to each party or the cloud server. While collaboratively running the BPN network learning algorithm, the parties need to execute consecutive operations such as addition and multiplication. However, the 'BGN' algorithm [7] only supports one step multiplication over ciphertext. To support consecutive multiplications, the parties need to decrypt the intermediate results first. However, straightforward decryption of the intermediate values will reveal these values to the parties, which may have privacy implications and shall be avoided [8]. To protect these intermediate results (scalar products or sum), we introduce a secure sharing algorithm which enables each participating parties to get a random share of the intermediate result without knowing its actual value.

### 2.3.5 BPN algorithm

```
[9] Algorithm for a 3-layer network (only one hidden layer):
Initialize network weights (often small random values)
do
   foreach training example ex
       prediction=neural-net-output(network. ex) // forward pass
       actual=teacher-output(ex)
       compute error(prediction-actual)at output units
       compute $\Delta w_h$ for all weights from hidden layer to output layer // backward pass
       compute $\Delta w_i$ form all weights from input layer to hidden layer // backward pass continued
       update network weights// input layer not modified by error estimate
       until all examples classified correctly or another stopping criterion satisfied
   return the network
```

As the algorithms name implies, the errors propagate backwards from the output nodes to the input nodes. Technically speaking, back propagation calculates the gradient of the error of the network regarding the networks modifiable weights[2]. This gradient is almost always used in simple stochastic gradient descent algorithm to find weights that minimize the error. Often the term "back propagation" is used in a more general sense, to refer to the entire procedure encompassing both the calculation of the gradient and its use in stochastic gradient descent. The back propagation algorithm for calculating the gradient has been rediscovered a number of times, and is a special case of a more general technique called automatic differentiation in the reverse accumulation mode[10]. The following Fig 2.3.5 the working of BPN algorithm in the form of a flow chart.

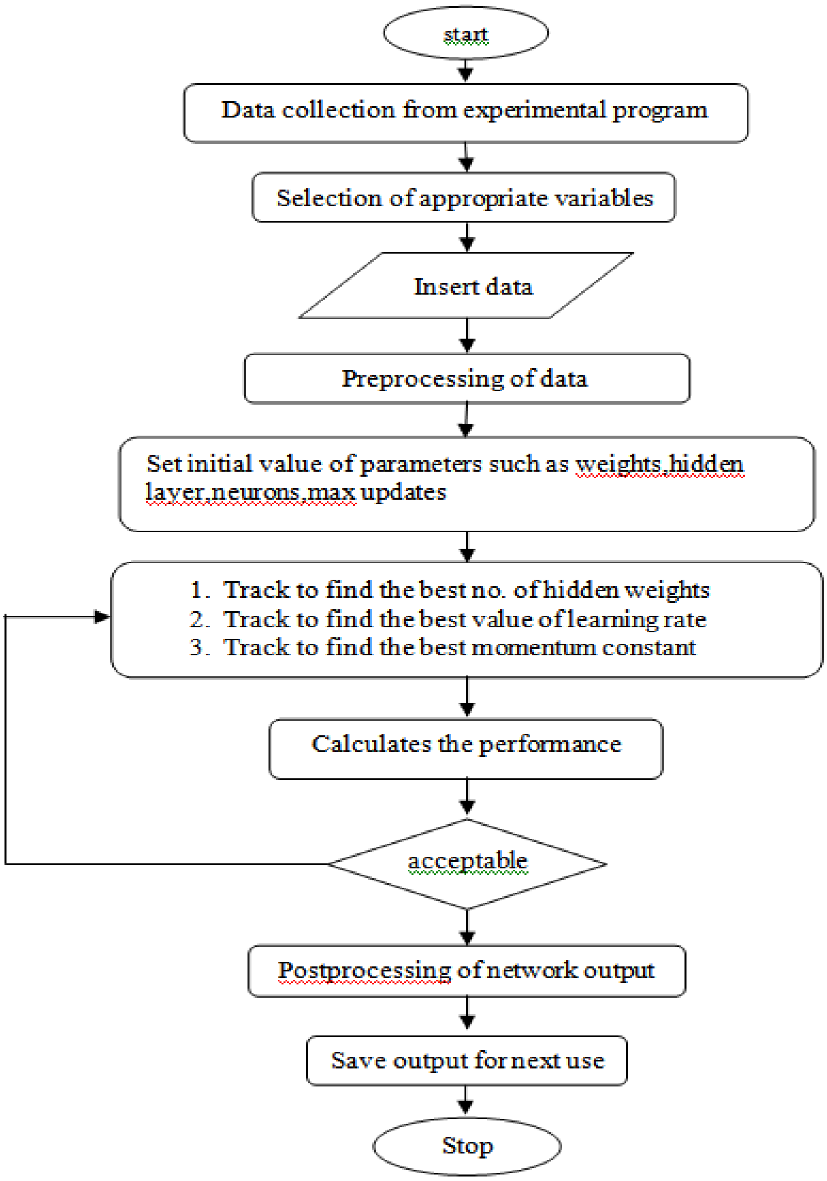

start
Data collection from experimental program
Selection of appropriate variables
Insert data
Preprocessing of data
Set initial value of parameters such as weights,hidden layer,neurons,max updates
1. Track to find the best no. of hidden weights
2. Track to find the best value of learning rate
3. Track to find the best momentum constant
Calculates the performance
acceptable
Postprocessing of network output
Save output for next use
Stop

Fig 2.3.5: BPN Algorithm Flowchart

The overall system design is shown in Fig 2.3.6

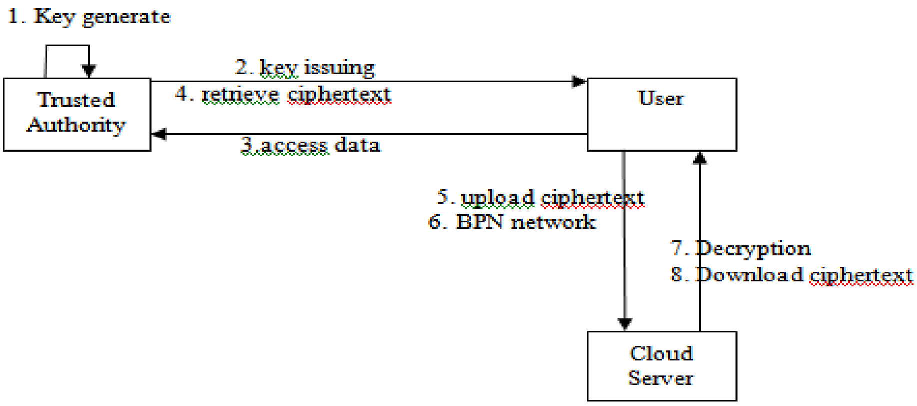


Fig 2.3.6 : System Design

## III. RESULT AND ANALYSIS

To login, an user needs to register himself/herself where, they are asked to fill the basic information such as name, e-mail, password etc. once the form is filled, the user must click on register button. The user is now registered and the registration successful page is displayed onto the screen. Fig 3(a) shows the login page, Fig 3(b) shows the registering form and Fig 3(c) shows registration successful page.

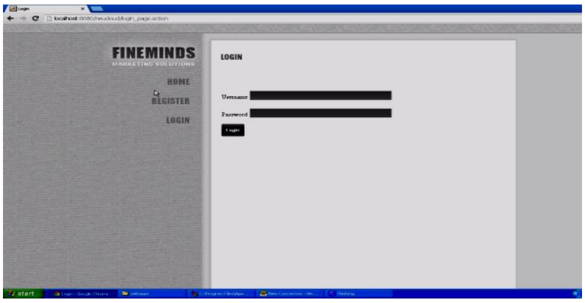
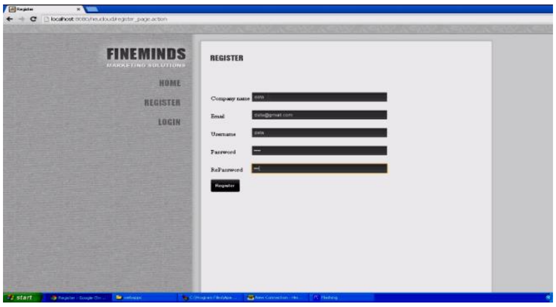


fig 3(a) : login page



Company name data
Email data@gmail.com
Username data
Password ----
RePassword ---

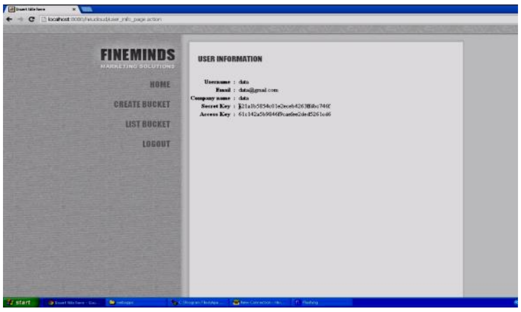
fig 3(b): registering form

Registered Successfully

fig 3(c): registered successful

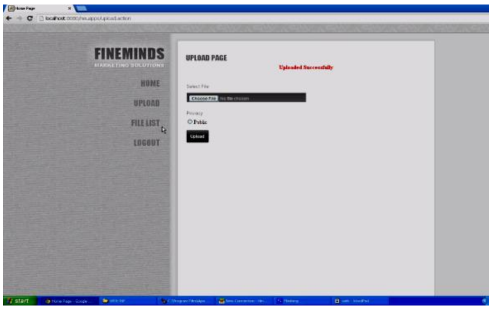
The trusted authority generates and issues the secret key to an user. This secret key and access key can be viewed in user information along with the user name, e-mail and company name. Fig 3(d) shows the user information. For a user to login cloud, it first needs to register the company in the cloud. To create an user in the cloud, the user needs to create an account, specify the access specifier as public or private or shared, along with the basic information such as name, e-mail address etc. the similar images can be seen in Fig 3(a), 3(b) and 3(c) for login, registering page and registration successful page respectively.

Once the account of an user is created in cloud server, the user can upload the data (text, audio, image etc) onto the cloud server. The access specifier is previously mentioned during the creation of the user account. The user needs to choose a file form its respective system and then upload. An upload successful page appears on the screen once the file is uploaded in the cloud server. The Fig 3(e) shows the process uploading a file on cloud server by the user.



Username : data
Email : data@gmail.com
Company name : data
Secret Key : 121a1b5854c01e2eceb42638f6bc746
Access Key : 61c142a5b9846f9caefee2ded5261cd6

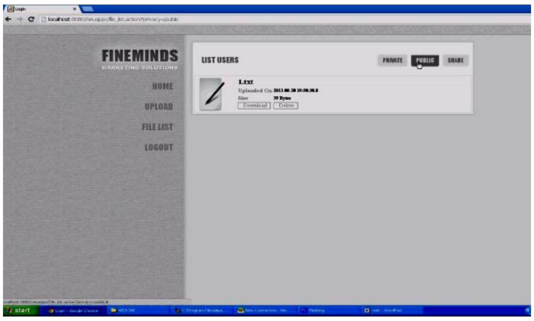
fig 3(d): user information



Uploaded Successfully
Select File
No file chosen
Privacy
○ Public

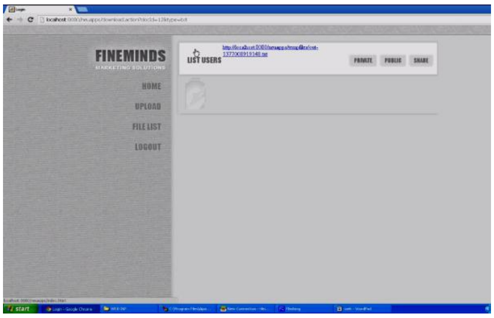
fig 3(e): upload page in cloud

if the other user's want to view or download the file uploaded by the other user's , they can check the list users and click on download. The cloud provides a download link to the user, using which the user can download and give the content of the file. Fig 3(f) shows the list user's, Fig 3(g) shows the download link provided by the cloud for the user and Fig 3(h) shows the content of the file.



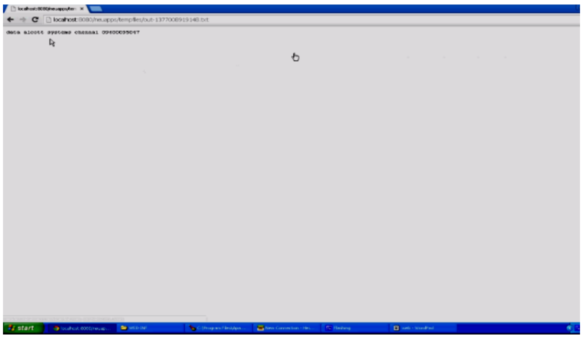
1.txt
Uploaded On 2013.08.20 19:50:30.0
Size 39 Bytes

Fig 3(f): list user's



http://localhost:8080/neuapps/tempfiles/out-1377008919148.txt

fig 3(g): download link



```
data alcott systems chennal 09600095047
```

Fig 3(h): content of the file

## IV. Conclusion

In this work, we proposed the first secure and practical multi party BPN network learning scheme over arbitrarily partitioned data. In our proposed approach, the parties encrypt their arbitrarily partitioned data and upload the ciphertext to the cloud. The cloud can execute most operations pertaining to the BPN network learning algorithm without knowing any private information. The cost of each party in our scheme is independent to the number of parties. This work tailors the BGN homomorphic encryption algorithm to support the multi party scenario, which can be used as an independent solution for other related applications. Complexity and security analysis shows that our proposed scheme is scalable, efficient and secure.

## REFERENCES

[1] Hassan, Qusay (2011). "Demystifying Cloud Computing". *The Journal of Defense Software Engineering* (CrossTalk) 2011 (Jan/Feb):16–21. Retrieved 11 December 2014.

[2] "The NIST Definition of Cloud Computing". National Institute of Standards and Technology. Retrieved 24 July 2011.

[3] Rumelhart, David E.; Hinton, Geoffrey E.; Williams, Ronald J. (8 October 1986). "Learning representations by back-propagating errors". *Nature* 323 (6088): 533–536.doi:10.1038/323533a0.

[4] Computer & Communication Networks, Nadir F Mir, Pearson Education, India

[5] McCulloch, Warren; Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics* 5 (4): 115–133.doi:10.1007/BF02478259. http://en.wikipedia.org/wiki/Artificial_neural_network

[7] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Proceedings of the Second international conference on Theory of Cryptography*, TCC'05, pages 325–341, Berlin, Heidelberg, 2005.

[8] T. Chen and S. Zhong. Privacy-preserving backpropagation neural network learning. *Trans. Neur. Netw.*, 20(10):1554–1564, Oct. 2009. http://en.wikipedia.org/wiki/Backpropagation#Algorithm

[9] Paul J. Werbos (1994). The Roots of Backpropagation. From Ordered Derivatives to Neural Networks and Political Forecasting. New York, NY: John Wiley & Sons, Inc.