Contents list available at JMCS
Journal of Mathematics and Computer Science
Journal Homepage: www.tjmcs.com

A New Method for Searching Keyword in Cloud Servers Using ANFIS

Fatemeh Goli
Mirdamad Institute for Higher Education, Gorgan, Iran
fateme.goli@hotmail.com
Hossein Momeni
Golestan University, Gorgan, Iran
h.momeni@gu.ac.ir
Ali Yavari
Sama technical and vocational training college, Islamic Azad University, Gorgan Branch, Gorgan, Iran
yavari@ustmb.ac.ir

Article history:
Received May 2014
Accepted June 2014
Available online July 2014

Abstract

With the popularity of computing cloud in recent decade, sensitive information is stored in cloud systems. To protect the sensitive data before saving, cryptographic operation must be done. In cryptography with the traditional method the user can search its data with high security ability, but the disadvantage of this method is to enter the same items in data for searching. And there is not any virtual error for false type. This major weakness causes not to harmonize the alone mentioned method in searching on cloud servers. In this paper we solve the search problem of cloud encrypted words in light and fuzzy system with maintaining the data security. Its aim is to search in cases which there is not the user's possible errors be changed into the rules using a light and fuzzy network, and discussed with the use of neural network to learning the pattern of the server data, and when searching the network provides the closest option to the user. Our experimental results show that this method of diagnosis is above 90 percent in the scope of the written rules for fuzzy system.

Keywords: cloud computing, fuzzy search, neural search, neuro-fuzzy, cryptography

1. Introduction

With the popularity of computing cloud in recent decade sensitive information is stored in cloud systems. This information class contains emails, medical records, governmental documents, etc. The responsibility for keeping the data owners from data physically is resolved with storing the data and they can enjoy keeping this service when being available. Yet, the data owners and cloud servers may not be too trusting. Owners may also be willing to share their information with certain users. The users also may want to search certain information. One of the methods for this is to select the search by keyword instead of searching all data in encrypted (such as Google). Unfortunately, encrypting data restrict searching by keywords in traditional method. Furthermore, keywords give brief information on the data that it is possible to endanger data security. Although encrypting the keywords can also make it more secure, the searching manner is like traditional methods. To enhance data security, encryption technique search has been in

past years [1] and [2]. In this method, a remarkable idea (i) with each keyword is considered that it is similar to extracting the features of the keyword, and is stored with the file of the keyword contents that this idea with keeping keywords information and main text can provide the search result. The above method must come the same keyword in the search field and there is not any virtual error for typing error like the words "office" that it was the users "office" or PO Box which it was P.O. Box. The presentation of a mechanism that it just checks the words from grammatical aspects cannot be an appropriate solution. For example, if a user type the word "house" it is correct from grammatical aspect, but in the sentence "go in my house" it is clear that the user has typed the word "house" false. Then grammar checking mechanism cannot be a good solution easily. A neurofuzzy can get a correct result with the correct combination of grammatical correction and the meaning understanding. In this paper we will use on searching the keywords in clause using neurofuzzy. In searching in neurofuzzy method in cases the user types the same item in Cloud, the result will be correct and when there is not the word in cloud, neurofuzzy network find the closest option with its seen trained models. The basis of the work is divided into several categories. Part 2 deals with the definitions, the basic concepts, the description of the system model and the description of the required background for implementation of this article. In part 3 we will collect and review the characteristics of the past works and we will express past problems. In part 4 we present the suggested approaches from the method of this article, and in part 5 we will evaluate our suggested approach. In part 6 the practical results are expressed and part 7 concludes the article.

2. Definitions and basic concepts

2.1. System model

In this paper we have a cloud data system including data owner and the users from data and the server. C is the stored data set in the server (C= (F1, F2,.., FN)). And the keywords are W(W= (W1, W2, ...,Wp)).

The cloud server provides the searching service on the set C. The user types a topic from data and the server is responsible for writing the topic search with the set C and presenting data from the set to him. The fuzzy keyword explorer returns the close topics using the following rules.

1) If the user's the search term was exactly like the keywords W, then the Cloud server return the text associated with it.

2) If from the terms and Or use in searching, then the server returns the closest answer.

2.2. Security model

Our assumption on an unreliable server in terms of data protection is information. It means that the server may reveal the other user's sensitive information in reply to searching a user. We will use from the previous security method [3] to protect the sensitive information.

2.3. Design aim

Introducing new mechanism for creating the fuzzy keywords set designing an appropriate fuzzy searching set and the designed system validation.

2.4. Distance model

There are several methods for grouping similar string. In the paper [4], the chahack distance [5-6] that we used from fuzzy optimization, the distance between w1 and w2 indicates with ed(w1, w2) and equals to the required operations to convert one of them instead of the other including three operations:

First) Substitution: Changing characters to the other in words

Second) Deleting: Deleting a character from the words

Third) Insertion: adding a character to the words for each search term the above three actions must continue to become d >ed(W, Wi). The amount of the number d is called simulation rate. We assume that there are the sets C, W and the d. The search string (K, W) is also accessible that K is the distance W from Wi.

If w ϵ wi then return (FIDwi)

Else if ed(W,Wi)< d then return (FIDwi)

Provided that ed (W,Wi)< min(k,d)

2.5. The adaptive neuro fuzzy inference system (ANFIS)

The structure of fuzzy neural network is shown in figure Y. IN this figure, the output of the nodes of the first layer is the degree of linguistic variables. Typically, bell membership functions are used in the layer. The structure of Bell membership function is shown in this formula [7]:

$$f(x) \quad = exp\left[\frac{-1}{2}\left(\frac{x - xi_1}{bi_1}\right)^2\right]$$

The second layer in fuzzy neural network is the law layer. The condition part of rules is calculated by the fuzzy operator of min, and the result is applied as the degree of law enforcement.

Learning activities in this layer are done by changing the degree of law enforcement regarding the training data which are injected into the network. In the third layer of the linear combination, results rate law is used to determine the membership degree in a specific category. In the fourth layer, the Sigmund membership function is used. Network training vector is injected into the network based on the following relationship [8]:

$$\{(\mathrm{x}^{\mathrm{k}}, \mathrm{y}^{\mathrm{k}})\ \mathrm{k} = 1,2, \dots, \mathrm{K}\}$$

Here xk refers to the kth pattern in input vector, then we will have:

$$y^k = \begin{cases} (1,0) & if x^k belongs to class\ 1 \\ (0,1) & if x^k belongs to class\ 2 \end{cases}$$

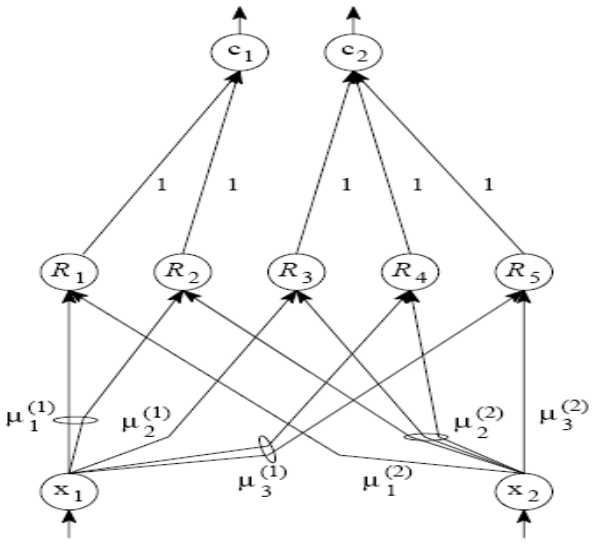

Fig. 1. Structure of fuzzy artificial neural network

Error function for pattern K is calculated from the following equation.

$$\mathrm{E}^{\mathrm{k}} = \frac{1}{2}\left[(\mathrm{o_1}^{\mathrm{k}} - \mathrm{y_1}^{\mathrm{k}})^2 + (\mathrm{o_2}^{\mathrm{k}} - \mathrm{y_2}^{\mathrm{k}})^2\right]$$

Equation should be written here. Where YK and OK are the desire output and the calculated output, respectively.

3. Related Works

The past works are divided into three categories:

The search of the fuzzy keywords: in recent decade, the importance of the fuzzy search has taken into consideration. [9] and [10]. This group allows the user to use from the technique try and see based on the final estimation by the fuzzy system for finding the required term.

Researchable encryption: Traditional researchable encryption [1], [2] and [3] is widely used in encrypted texts. The first encrypted search was done by song and his colleagues. [11] GA and the colleagues, [12] have used from bloom filter for extracting the idea from the original text. Chang [13] and Kartmola [3] have used from a kind of the index that corrected the idea table from the original text. Bonch and the colleagues used from a global key such as Sang's work and the colleagues. A remarkable point in the above work is to search on the basis of the same term.

Other works: A private watch [15] widely uses from the solidarity factor between data for their searching. This method is used for watching two information sets with each other very much. Private information retrieval PIR [16] is another technique for reaching to information with high security.

The use of the fuzzy system for finding the closest option also did by Li and the colleagues [4] by using the corrected distance between the searchable term and Cloud data in 2010. It presented the best results until now. The main problem for their method is very high closing for the exit winning number to making the fuzzy with each other increasing the error percent highly.

At first we compare how the fuzzy keywords work on encrypted data. The encrypted function is like the following form:

$$E^k = \frac{1}{2}\left[(o_1{}^k - y_1{}^k)^2 + (o_2{}^k - y_2{}^k)^2\right]$$

Where sk is the secret key and setup is secret algorithm with the Landa parameter and enc is encoder and dec is the encoder detector. Twi is to map the way of no return from Wi. This mapping can do these operations with the function f such as [1], p[2] and p[3]. With giving SK and Wi to the function f: Twi=f(SK,Wi)

For every wi, we define a set of the fuzzy word d. this set is called Wi and contain all possible form that is d> (w,w')ed. For example, on the first characters from the set W=ray, all possible from for W equals to aay ,bay, cay, ... , zay. For making an index on Wi the data owner from a mapping with secret key sk use in Twi = f (sk,W'i) where sk share between the user and he. The data owner also encrypts FIDwi with the function ENC(sk, FIDwi). Then, the index Twi and ENC (sk, FIDwi) and the term W by the user, he computes the mapping W with the same Tw =f (sk,w) and send it to the server. Then, the server considers when sending the search term to the server mapping ed(W, W'i) for all W'i and selects the smallest and return ENC (sk,FIDwi). Then the user decodes and uses this file. But there is a basic problem in the above method. For example, in English data with the length L for d=1 is the cases number W'i the same with changing the word which equals to 1 multiplies 26. and for d=2 in the cases with the change is L*26 and in the cases with two changing equals to (L(L+1)/2)*26^2 which is for d=2 altogether:26L+(L(L+1)/2)*26^2And for the word 10000 with the length 10 bits, the index forms are (26*10+

(SS*26*26)*10000*10=3Gbit and meanwhile the main text includes: 10000*10+97 kbit. Therefore, it is necessary to need a kind of the fuzzy keywords with less size.

4. Proposed approach

To eliminate the size problem in fuzzy rules on one hand and the search system is better than to a neurofuzzy model. Hereinafter, our basic aim will be on two principles. First, it is the construction of the neurofuzzy system whenever the search term was different from the user's aim; it can give him the closest aim and second is to design a safe and useful search for the user.

For this purpose, first, we assume that the set W contains the English lowercase, then the included keywords in the cloud.Servers were encoded. Ski has an equivalent (sk + 97) to (sk + 122) with words using membership functions are fuzzy in figure 1. We use snorm for creating the fuzzy statement and the method snorm is an algebra sum.

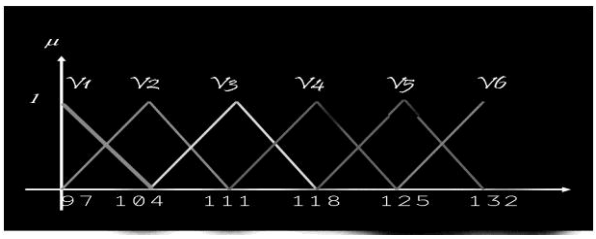

Fig. 1: Membership functions and language variables

Now we must deal with writing the fuzzy rules. It forms a rule for every combined keyword. From creating every letter between its letters. For example, W=ray, all possible cases for w′ equal to *ray, r*ay, ra*y and ray*.for d=2 is L+(L(L+I)/2) altogether. And for 10000 words with the length 10 bits. The index cases numbers are (10+(ss))*10000*10= 6. 1Mbit and this method decrease the data sizes of the fuzzy system 500 times than that of before method. We can consider the following rule for every term sbz which it is the keyword encryption ray with sk=1:

If s[0]=1 or s[1]=3 or s[2]=3 or … then y=1 where y=1 is the exit fuzzy. In brief this rule is written 1337777777

Where 7 is the lack of the membership function. It can also be written sbz in the following forms:*sbz, s*bz, sb*z,sbz*. Where the four rules are in the following rule:{1, 3, 3, 7, 7, 7, 7, 7, 7, 7}, {1, 7, 3, 3, 7, 7, 7, 7, 7, 7},{1, 3, 7, 3, 7, 7, 7, 7, 7, 7}, {7, 1, 3, 3, 7, 7, 7, 7, 7, 7}.

The inference engine used is the product motor with the minimum tnorm. We use from a creative method for non-fuzzy. The fuzzy statement prior to every group from the rules were computed which used from a range and divided by their numbers. So 5 groups of the number obtains from non-fuzzy of every range. We multiply these 5 groups by 50 to become the number which changes into binary 5 bits. Now, we turn into each number to the binary. Now, we bound outputs of each 5 bits into others bits to obtain a 25-bit number from each keyword string.

Therefore each term accessible in Cloud server changes into 25 bits. Table1 shows converting 5 keywords daivid, paul, ray, benjamin, anderson is 25 bits.

Table 1.Fuzzificationof 5 cloud server's data

Name	Grade	XX	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a	a
0	1	1	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0
1	1	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	0

1	0	0	0	0	1	1	1	0	0	1	1	1	0	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	1	1	0	0	1	0	1	0	0	1	1	1	0	0	1	1	0	0	0
0	1	0	0	0	1	1	1	0	0	1	1	0	1	0	1	1	1	0	0	0	0	1	1	0

Now we use from a neural network with back propagation architecture with 25 input layers and 100 secret layers and 5 output layers. Among them 25 fuzzy input and output layers which obtained in the previous phase. 5 output layers are 5 fuzzy keywords as 10000=anderson, 01000=benjamin, 00100=ray, 00010=paul, 00001=daivid are saved. We consider the outputs in the following form (Table 2).

Table2.Non-fuzzy of output

Name	Grade	XX	a	a
1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

The network architecture is like Figure2.

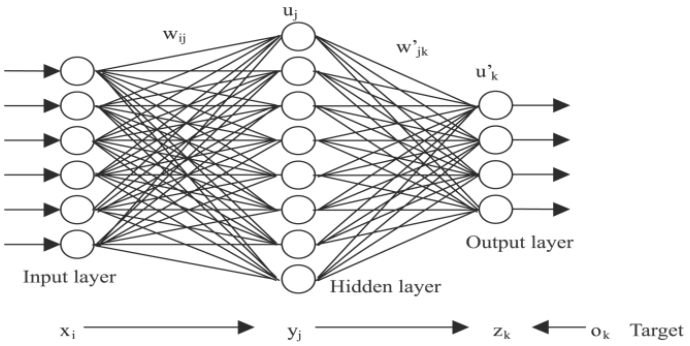

Fig. 2.Architecture of proposed neural network

The number of the secret layer was selected in a trial and error method to find recognition and work percent maximum with high speed and the output layers were selected on the basis of experimental keyword numbers. During the learning process was modulated the number of secret layer nerves and learning coefficient and momentum coefficient to obtain the least amount of error and all errors were computed from the following relation:

$$\mathrm{MSE} = \frac{1}{2} \sum_{\substack{1 \le i \le k \\ 1 \le j \le m}} (\mathrm{P}(i,j) - \mathrm{O}(i,j))$$

Where O is a real output and P is an aim. The obtained results are observed from training the network from 72 epochs in table3.

Table 3. Training results of network (72 epochs)

Name	Grade	XX	a	a
0.68	0.21	0.09	0.12	0.11
0.09	0.85	0.05	0.14	0.13
0.07	0.03	0.17	0.51	0.27

0.05	0.03	0.14	0.63	0.26
0.06	0.08	0.08	0.27	0.53

The network is observed after finishing the training in epoch 94 which has learned the output model and the error rate is acceptable (Table 4).

Table 4. The network after finishing the training in epoch 94

Name	Grade	XX	a	a
0.88	0.08	0.18	0.01	0.04
0.06	0.91	0.03	0.02	0.06
0.08	0.01	0.79	0.14	0.08
0.02	0.01	0.15	0.88	0.09
0.04	0.05	0.09	0.06	0.88

5. Evaluating the proposed approach

Now we must give the fuzzy data from the term into the network with updated weights to specify the outputs. For this purpose we have two data categories. First, the same exact data with the ones on cloud server to evaluate the made system.

Second, different and close data with the server terms. For the first state we use the term search paul. It returns the result in table 5.

Table 5.Searching the same term in the server (paul)

Name	Grade
Bmj	31
Npibnnbe	25
Iftbn	2
tbsb	23
ljbsbti	7

And for state2, we do the search benja6amin which is a changed input from benjamin observed in table 6. The written strong neurofuzzy engine could search the user's term with keeping the security among the servers.

Table 6.Search the changed term with the server

Name	Grade
Bmj	31
Npibnnbe	25
Iftbn	2
tbsb	23
ljbsbti	7

6. Practical results

Proposed methods in this paper implemented in c#.Net language in the software Visual studio 2008. For testing the software performance after training the neural network with 5 keywords. 3 data groups have given for testing. The first group contained the same term in the server like paul

and the second group consisted of the terms with one different preposition like benja6min and the third one is the data with two or more difference like benja6mfin who the recognition results are observed in Table7.

Table7. Recognition percent of data testy good

	10 data	30 data	100 data
The same term	%۱۰۰	%۱۰۰	۱۰۰%
A difference	%۹۰	%۹۳	%۹۷
Some difference	%۲۰	%۲۵	%۲۸

It is observed that our system has been very good for one difference and the system is very weak for several differences. Its reason is not to write the relevant rules for some differences. With considering the full fuzzy seta including all rules. This percent must get over 90%.

7. Conclusion

Searching in Cloud server has always been a basic challenge when the search term is different from the server data. In this paper a method presented to improve the done works in recent years using a neurofuzzy network until the system outputs of the fuzzy search is close together and the error percent was high using the neurofuzzy of back propagation deal with the learning model of the fuzzy output and suggest the closest option to the user. In written rule field for the fuzzy system, the correct recognition percent of this method is over 90%. As a future work, we can use the optimization algorithms such as particles-gathering algorithms to optimize the keyword search in Cloud servers.

Reference

[1] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in Proceedings of Crypto 2007, volume 4622 ofLNCS. Springer-Verlag, (2007).

[2] F. Bao, R. Deng, X. Ding, and Y. Yang, "Private query on encrypted data in multi-user settings," in Proc. of ISPEC'08, (2008).

[3] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in Proc. of ACM CCS'06, (2006).

[4] Jin Li, Qian Wang, Cong Wang, Ning Cao, KuiRen, and Wenjing Lou. "Fuzzy Keyword Search over Encrypted Data in Cloud Computing", (2010).

[5] V. Levenshtein, "Binary codes capable of correcting spurious insertions and deletions of ones," Problems of Information Transmission, vol. 1, no. 1, pp. 8–17, (1965).

[6] B. Waters, D. Balfanz, G. Durfee, and D. Smetters, "Building an encrypted and searchable audit log," in Proc. of 11th Annual Network and Distributed System, (2004).

[7] J. M. Zurada, Introduction to artificial neural systems vol. 8: West publishing company New York ;,(1992).

[8] S. Mitra and Y. Hayashi, "Neuro-fuzzy rule generation: survey in soft computing framework," Neural Networks, IEEE Transactions on, vol. 11, pp. 748-768, (2000).

[9] C. Li, J. Lu, and Y. Lu, "Efficient merging and filtering algorithms for approximate string searches," in Proc. of ICDE'08, (2008).

[10] S. Ji, G. Li, C. Li, and J. Feng, "Efficient interactive fuzzy keyword search," in Proc. of WWW'09, (2009).

[11] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. of IEEE Symposium on Security andPrivacy'00, 2000.

[12] E.-J. Goh, "Secure indexes," Cryptology ePrint Archive, Report2003/216, (2003),

http://eprint.iacr.org/.

[13] Y.-C. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data,” in Proc. of ACNS’05, (2005).

[14] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in Proc. of EUROCRYP’04, (2004).

[15] J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. Strauss, and R. N.Wright, “Secure multiparty computation of approximations,” in Proc. Of ICALP’01.

[16] R. Ostrovsky, “Software protection and simulations on oblivious rams,”Ph.D dissertation, Massachusetts Institute of Technology, (1992).