Middle-East Journal of Scientific Research 25 (6): 1303-1307, 2017
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2017.1303.1307

Experimental Analysis of Sequential Pattern Estimation with Huge Sequence of Data Processing

[1]K. Subramanian and [2]S. Surya

[1]H.H The Rajah's College, Pudukkottai,Tamil Nadu, India
[2]J.J. College of Arts and Science, Pudukkottai,Tamil Nadu, India

Abstract: The term Data Mining usually handles large set of data arranged in structured pattern. Numerous data mining tasks are performed by users in this modern era such as clustering the data, classify the items, mining the required patterns and performing association rule mining. In this kind of data mining tasks, the crucial or harmful thing may happen in some areas; those things are referred as Spams. These kinds of spams spread its identity to all over the data, in which it starts from the basic or initial cluster and spread the spam content to all over the cluster. Spam does not follow any pattern it just spread over the harmful content to the subsequent items one by one until the end of the data set. The main objective of this paper is to propose a new methodology to resolve the clustered spam content and provide effective prevention from the spams as well as control the spams to spread over the clusters further. The new algorithm Spam Control-Fuzzy C-Means (SC-FCM) algorithm is defined and solves the problems caused by spam. There are many types of spamming methods are available called data spam, link spam and more. The major cause is happened via data Spams, which mis-leads the users required work to an unwanted works or alerts or arising interruption and causes the work to fail in certain circumstances. Artificial Bee Colony (ABC) is also used in this approach to evaluate the k-NN dataset to prove the SC-FCM methodology is more better compare to all the other algorithms in past.

Key words: Spam Estimation · Spam Removal · SC-FCM · ABC · Spam Analysis · Filtering and Removal Process

INTRODUCTION

Successions are an imperative sort of information which happen much of the time in numerous fields, for example, restorative, business, budgetary, client conduct, instructions, security and different applications. In these applications, the examination of the information should be done in various approaches to fulfill diverse application necessities and it should be done in an effective way. Spam usually concentrates more on Cluster of data instead of single record ort item, because it is the easy way to spread the spam content to the usual portions without any identity. In this modern era all the user works are based on the system or PC. The solutions provided by the system are more better, time saving and accurate compare to the manual process. Because of this purpose all the users require the work to be done by smart way with the help of computers.

The major threat affected the system is caused by spams, which produces crucial affections to users to break their work with arising unwanted interruptions. The motto of the spam is to break the user's regular work and divert them to unwanted things. For instance, if the user named A working with Tally Software to manipulate the year ending process of a Company, Spam causes disturbances like close the Software without any knowledge of the respective user A or else open some unwanted tools installed into the machine. These kind of affections causes the user to think that the particular software or system having no much efficiency to operate with.

This paper approaches based on the dataset provided by KDD surveys and the dataset is named as k-NN dataset, which contains a cluster of data and several affected portions in it. Our experimental result will prove that the affection should be identified and removed with the help of two best classification algorithms. One is the classical Artificial Bee Colony Algorithm (ABC) and the other one is our proposed algorithm called Spam Control-Fuzzy C-Means (SC-FCM) Algorithm.

Corresponding Author: K. Subramanian, Assistant Professor, H.H The Rajah's College, Pudukkottai,Tamil Nadu, India.

Literature Review: In [1], the issue of finding what things is purchased together in an exchange" over bushel information was presented. While related, the issue of finding what things are purchased together is worried with finding intra-exchange designs, though the issue of finding successive examples is worried with between exchange designs. An example in the first issue comprises of an unordered arrangement of things though an example in the last case is a requested rundown of sets of things.

Finding designs in successions of occasions has been a range of dynamic exploration in AI (see, for instance, [6]). Be that as it may, the center in this group of work is on finding the tenet fundamental the era of a given arrangement keeping in mind the end goal to have the capacity to foresee a conceivable succession continuation (e.g. the tenet to foresee what number will come next, given a succession of numbers). We on the hand are keen on finding every single regular example implanted in a database of successions of sets of occasions (things).

Our issue is identified with the issue of finding content subsequences that match a given customary expression (c.f. the UNIX grep utility). There likewise has been work on finding content subsequences that around match a given string (e.g. [5] [12]). These strategies are situated toward finding matches for one example. In our issue, the difficulty is in figuring out what examples to attempt and after that efficiently finding out which ones are contained in a client grouping.

Procedures in light of different arrangement [11] have been proposed to find whole content groupings that are comparative. There likewise has been work to find locally comparative subsequences [4] [8] [9]. Be that as it may, as pointed out in [10], these procedures apply when the found examples comprise of sequential characters or numerous arrangements of continuous characters isolated by a fixed length of clamor characters.

Nearest to our issue is the issue plan in [10] with regards to finding likenesses in a database of hereditary groupings.

The examples they wish to find are subsequences comprised of continuous characters isolated by a variable number of commotion characters. An arrangement in our issue comprises of rundown of sets of characters (things), instead of being basically a rundown of characters. In this way, a component of the consecutive example we find can be an arrangement of characters (things), as opposed to being just a character. Our answer methodology is totally different.

The arrangement in [10] is not ensured to be finished, though we guarantee that we have found every successive example of interest that are available in a specified least number of groupings. The calculation in [10] is a primary memory calculation in light of summed up suffix tree [7] and was tried against a database of 150 groupings (in spite of the fact that the paper contains a few indications on how they may extend their way to deal with handle bigger databases). Our answer is focused at a huge number of client successions.

Dataset Formation: Spam/Spamming is a threat which causes unwanted time wastages and cost expensiveness problems, which leads several drawbacks over the respective user. A new approach called SC-FCM is proposed in this paper to detect the spam and act as a filter.

The solution we provide to the spam affections is done by offline mode and the k-NN based dataset is taken for the process, which contains spam affection over the cluster of data internally. Patter Mining with Sequential flow estimates the dataset with one by one sequence and resulting with high accuracy in finding out the spam content and eliminate it separately as well as our proposed approaches like ABC and SC-FCM rectifies the spam content efficiently.

Table 1: Dataset with Sample Rows and Columns

1	eco_i	private	S0	0	0	0
0	domain_	http	SF	295	811	0
2	eco_i	smtp	SF	792	330	0
0	Tcp	http	SF	321	2486	0
0	eco_i	ftp_data	SF	3468	0	0
0	Tcp	http	SF	227	4073	0
0	domain_	private	S0	0	0	0
0	eco_i	private	S0	0	0	0
0	Tcp	http	SF	203	1046	0
0	Tcp	http	SF	209	1558	0
0	eco_i	http	SF	242	392	0
0	Icmp	eco_i	SF	8	0	0
0	Tcp	private	S0	0	0	0
0	eco_i	private	S0	0	0	0
0	Icmp	ecr_i	SF	1032	0	0
0	Icmp	ecr_i	SF	1032	0	0
0	eco_i	http	SF	315	1102	0
0	domain_u	http	SF	201	7911	0
0	Tcp	http	SF	234	11485	0
0	domain_u	http	SF	54540	8314	0
	…	…	..	…	…	…

Maximum Limitation and Expansion of Dataset: The input Dataset for this approach is expanded up to 1 Lakhs number of rows and the main content specified into the respective dataset is defined with its affection range, the affection content is defined by the user program logic

only such as if the user try to find out the Neptune or Icmp content is spam means the program definitions should be like that and the algorithm approaches or proceeds based on the corresponding logic only. Once the identifications are done; the implemented algorithm proceeds for rectification process.

System Configuration: For this application we require the following system configurations, RAM 2 GB, Hard disk of 250 GB, Monitor and the system is implemented by means of NetBeans version 7.1.2 with the framework of Java in any version above Jdk 1.7.

Pseudo code: SC-FCM Process

```
FuzzyCMAlgo scfcm=new FuzzyCMAlgo();
StandDeviation std=new StandDeviation();
try
{
String rl[] = {"phf.", "imap."};
//String ur[] = {"perl.", "loadmodule.", "xterm."};
//String dos[] = {"Smurf.", "Back.", "neptune."};
BigFile1 file = new BigFile1("Attack.txt");
for (String line : file) {
// jTextArea2.append(line + "\n");
int l = line.lastIndexOf(",");
String sat = line.substring(l + 1, line.length());
for (int i = 0; i < rl.length; i++) {
if (sat.equalsIgnoreCase("phf.") ||
sat.equalsIgnoreCase("imap.")) {
System.out.println(line);
jTextArea1.append(pi+line.substring(0, l) + "\t" + rl[i]
+ "\n");
c++;
} else {
System.out.println("---->" + line);
jTextArea3.append(pi+line + "\n"); }
}
 d++;
}
BigFile1 f1 = new BigFile1("normal.txt");
d=d-c;
for (String line1 : f1) {
jTextArea2.append(line1 + "\n");
e++;
Logger.getLogger(Algoriths.class.getName()).log(Lev
el.SEVERE, null, ex);
}
jLabel1.setText("Injured:" + c);
txtSCER.setText(""+scfcm.getErrorRate(c));
jLabel2.setText("Rectified:" + (d+e));
xpval=Long.parseLong(""+(d+e));
txtSCI.setText(""+scfcm.getIterationSCFCM(c,  (d+e)));
itr2=Integer.parseInt(txtSCI.getText());
txtSstd.setText(""+(""+std.stdDev(0, itr2)));
txtSmean.setText(""+(""+std.stdMean(0, itr2)));
```

Experimental Outcomes: The experimental results of the proposed approaches of SC-FCM and ABC algorithms are proved one by one in the following illustrations as snaps. In this below figures we show that our proposed Spam-Control FCM approach is more efficient than the classical Artificial Bee Colony approach.

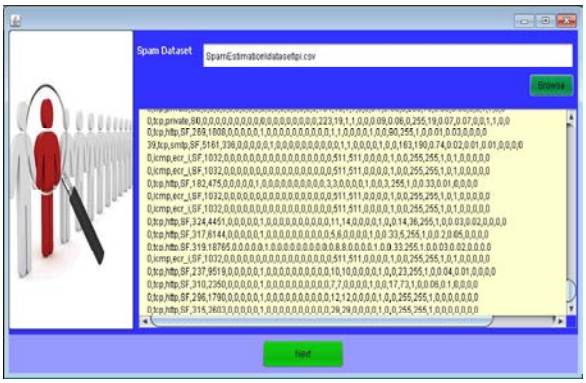

Fig. 1: Load Dataset

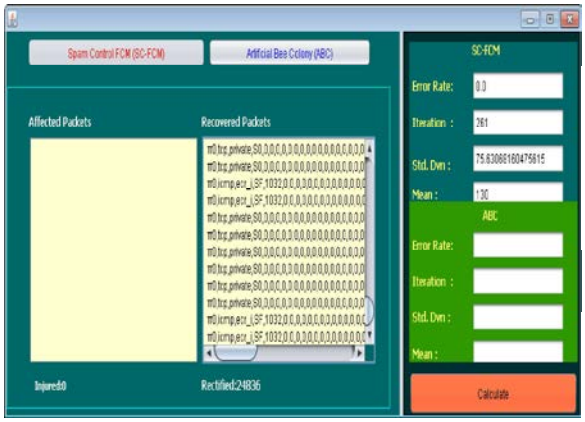

Fig. 2: SC-FCM Processing

Fig. 3: ABC Processing

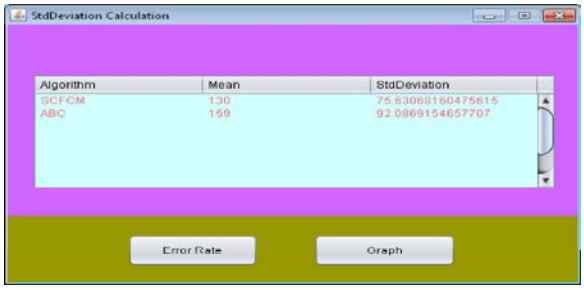

Fig. 4: Mean and Standard Deviation Estimation

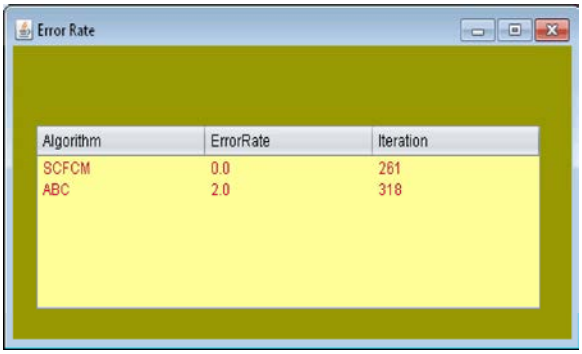

Fig. 5: Error Rate and Iteration Estimation

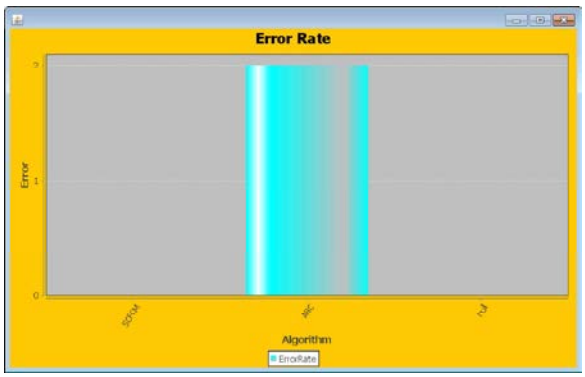

Fig. 6: Error Rate Graphical Representation

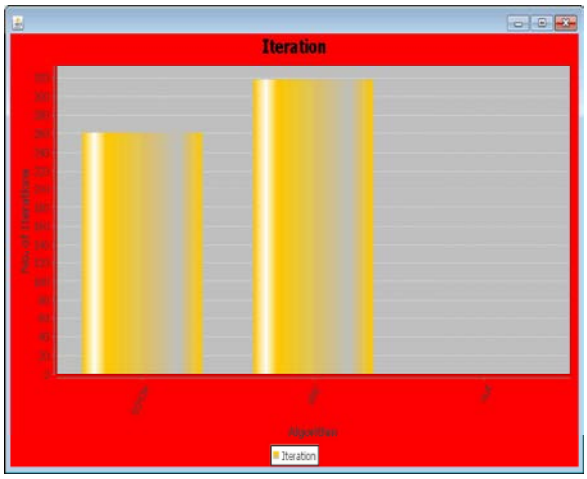

Fig. 7: Iteration Level Graphical Representation

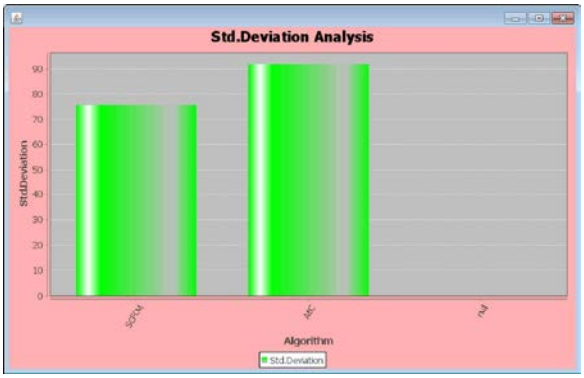

Fig. 8: Standard Deviation Graphical Representation

CONCLUSION

In this experimental analysis work, we completely illustrate the pattern estimations of spam content over data clusters with the help of two different algorithms such as Artificial Bee Colony (ABC) and Spam Control Fuzzy-C Means (SC-FCM). These modeling techniques efficiently find out the mean, standard deviation, error rate and iteration level of the input dataset. The experimental results prove that the SC-FCM is more efficient than ABC by means of identifying the content spams presented into the given dataset and rectify the spams with zero error rate. For all the complete analsyis clearly illustrates the performance and accuracy levels of two algorithms and the result are finer.

REFERENCES

1. Dong, G. and J. Pei, 2007. Sequence Data Mining, Springer, 2007.
2. Agrawal, R. and R. Srikant, 1994. Fast Algorithms for Mining Association Rules, 20th Int. Conf. Very Large Data Bases, VLDB, Morgan Kaufmann, 1994, 1994, pp: 487-499.
3. Agrawal, R. and R. Srikant, 1995. Mining Sequential Patterns, 11th Int. Conf. on Data Engineering, IEEE Computer Society Press, Taiwan, pp: 3-14.
4. Wang, W. and J. Yang, 2005. Mining Sequential Patterns from Large Data Sets, Springer, 2005.
5. Zhao, Q. and B.S. Bhowmick, 2003. Sequential Pattern Mining: A Survey, Technical Report, CAIS, Nanyang Technological University, No. 2003118, Singapore, 2003.
6. Yang J., W. Wang, P.S. Yu and J. Han, 2003. Mining Asynchronous Periodic Patterns in Time series Data, IEEE Transaction on Knowledge and Data Engineering, 15(3): 613-628.

7. Yang, J., W. Wang, P.S. Yu and J. Han, 2001. Infominet: Mining Surprising periodic Patterns, In Proceeding of the 7th ACM Int. Conf. on Knowledge Discover and Data Mining (KDD), pp: 395-400.
8. Han, J. and M. Kamber, 2003. Data Mining Concepts and Techniques, Morgan Kanufmann, 2003.
9. Yang, J., W. Wang, P.S. Yu and J. Han, 2002. Mining Long Sequential Patterns in a noisy environment, In Proceeding of the 2002 ACM SIGMOD Int. Conf. on Management of data, ACM Press, 2002, pp: 406-417.
10. Wang, J.T.L., G.W. Chirn, T. G. Marr, B. Shapiro, D. Shasha and K. Zhang, 1994. Combinatorial pattern discovery for scientific data: Some preliminary results. In Proc. of the ACM SIGMOD Conference on Management of Data, Minneapolis, May 1994.
11. Waterman, M., 1989. editor. Mathematical Methods for DNA Sequence Analysis. CRC Press.
12. Wu, S. and U. Manber, 1992. Fast text searching allowing errors. Communications of the ACM, 35(10): 83.
13. Agrawal, R., T. Imielinski and A. Swami, 1993. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, pages 207{216, Washington, D.C., May 1993.
14. Agrawal, R. and R. Srikant, 1994. Fast algorithms for mining association rules. In Proc. of the VLDB Conference, Santiago, Chile, September 1994. Expanded version available as IBM Research Report RJ9839, June 1994.
15. Agrawal, R. and R. Srikant, 1994. Mining sequential patterns. Research Report RJ 9910, IBM Almaden Research Center, San Jose, California, October 1994.
16. Altschul, S., W. Gish, W. Miller, E. Myers and D. Lipman, 1990. A basic local alignment search tool. Journal of Molecular Biology.
17. Califano, A. and I. Rigoutsos, 1993. Flash: A fast look-up algorithm for string homology. In Proc. of the 1st International Converence on Intelligent Systems for Molecular Biology, Bethesda, MD, July 1993.
18. Dietterich, T.G. and R.S. Michalski, 1985. Discovering patterns in sequences of events. Artificial Intelligence, 25: 187(232.
19. Hui, L., 1992. Color set size problem with applications to string matching. In A. Apostolico, M. Crochemere, Z. Galil and U. Manber, editors, Combinatorial Pattern Matching, LNCS 644, pages 230{243. Springer-Verlag, 1992.
20. Roytberg, M., 1992. A search for common patterns in many sequences. Computer Applications in the Biosciences, 8(1): 57{64.