Credit Cards Fraud Detection by Negative Selection Algorithm on Hadoop

(To Reduce the Training Time)

Hadi Hormozi
Computer Engineering and Information Technology Arak University,
Arak, IRAN
h.hormozi@qazd.ir

Elham Hormozi
Computer Engineering and Information Technology
Mazandaran University of Science and Technology,
Babol, IRAN
e.hormozi@ustmb.ac.ir

Mohammad Kazem Akbari
Computer Engineering and Information Technology Amirkabir University of Technology (Tehran Polytechnic) Tehran, IRAN
akbarif@aut.ac.ir

Morteza Sargolzaei Javan
Computer Engineering and Information Technology Amirkabir University of Technology (Tehran Polytechnic) Tehran, IRAN
msjavan@aut.ac.ir

Abstract—**This paper proposed a model for credit card fraud detection system, which is aimed to improve the current risk management by adding an Artificial Immune System's algorithm to fraud detection system. For achieving to this goal, we parallelize the negative selection algorithm on the cloud platform such as apache hadoop and mapreduce. The algorithm execute with three detectors set. The experiments show that by implement our fraud detection system on the cloud, the training time of algorithm in proportion to basic algorithm significantly decreases.**

Keywords-credit card; fraud detection; artificial immune system; apache hadoop; mapreduce.

I. INTRODUCTION

Cloud computing is an up-and-coming architecture with strengths and room for improvement. Cloud vendors use a large number of identically configured, low-end servers to scale the computing supply. Cloud systems can provide much of today's commerce depends on plastic. Also, People use their credit cards and debit cards to purchase products, get cash, and pay bills. Because of quick development in the electronic commerce technology, the use of credit cards has dramatically increased. Credit and debit cards are issued by banks. Fraud detection methods come into action when security approaches fail to stop it [1].

We have addressed credit card fraud detection in this paper. Fraud detection is the act of recognizing illegal activity and stopping it as soon as possible, before the transaction is accomplished. So, the real time fraud detection system is necessary for financial institutes [1, 2]. Therefore, the paper discusses the use of AIS on one aspect of security management, viz. the detection of credit card fraud and AIS has similarities to fraud detection system. This paper implement a fraud detection system based on AIS and Negative Selection Algorithm has selected. But due to the AIS algorithms have a long training time, the proposed model has been implemented using Apache Hadoop and MapReduce paradigm. With cloud computing, the organizations can access to a common data base of several types of frauds. Cloud Computing providing computing power and it can process high volume of transactions. The remainder of this paper is structured as follows: Section 2 describes fraud and fraud detection. Section 3 provides an overview of Artificial Immune System. Section 4 presents proposed model. Section 5 provides the results and the paper closes with a conclusion.

II. CREDIT CARD FRAUD DETECTION

Credit card fraud is a main issue in the financial industry. It is responsible for billions of dollars in losses per annum globally. It is well known that credit card as a method of payment increases the quantum of spending [2]. So, credit card payment systems must be supported by efficient fraud detection capability for minimizing unwanted activities by adversaries. Credit card fraud detection has drawn very large interest from the research community and a number of methods have been offered to counter fraud in this field [3]. Fraud detection is interesting for financial institutions. The appearance of new technologies as telephone, internet, automated teller machines (ATMs) and credit card systems have amplified the amount of fraud loss for many banks. Analyzing whether each transaction is legitimate or not is very expensive. Applying whether a transaction was done by a client or a fraudster by phoning all card holders is cost prohibitive if we check them in all transactions [4]. Before the transaction is accomplished, fraud detection should recognize fraud cases and stopping them as soon as possible.

III. ARTIFICIAL IMMUNE SYSTEM

Artificial Immune Systems (AIS) [5] are algorithms and systems that use the human immune system as inspiration. The algorithms typically operate the immune system's characteristics of learning and memory to solve a problem. The biological immune system is a highly parallel, distributed, and adaptive system. It uses learning, memory, and associative retrieval to solve recognition and classification tasks. In particular, it learns to recognize relevant patterns, remember patterns that have been seen previously, and use combinatorics to construct pattern detectors efficiently [6]. These remarkable information processing abilities of the immune system provide important aspects in the field of computation.

The immune system (IS) is a complex of cells, molecules and organs that represent an identification mechanism capable of perceiving and struggling dysfunction from our self cells and the action of exogenous nonself cells. The focal root of immunology was self-nonself discrimination through the principles of negative selection and clonal expansion. The first example of implemented AIS performing a useful computational task was an incarnation of a self-nonself discrimination system, used for the detection of computer virus executables [7]. The self-nonself discrimination system involved creating a behavior profile of sequences of system calls on a computer network during a period of normal function. To help in detecting malicious intruders, any subsequent sequences were matched against the normal profile, and any deviations reported as a possible intrusion [8]. Forrest et al in [7] proposed a NSA for several anomaly detection problems. This algorithm defines 'self' by building the normal behavior patterns of a monitored system. It generates a number of random patterns that are compared to each self pattern defined. If any randomly generated pattern matches a self pattern, this pattern fails to become a detector and thus it is removed (**Figure 1**). So, it becomes a 'detector' pattern and monitors subsequent profiled patterns of the monitored system. Consequently, if a 'detector' pattern matches any newly profiled pattern, it is then considered that new anomaly must have occurred in the monitored system (Figure **2**).

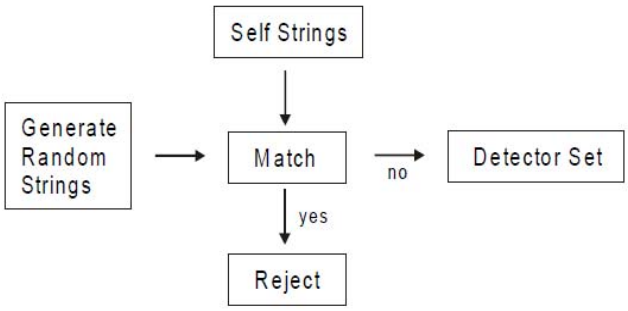

Figure 1. Detector Set Generation of NSA [9]

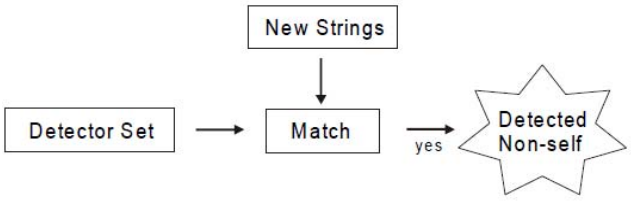

Figure 2. Detection of nonself patterns by Detector set [9]

IV. METODOLOGY

One of the big problems for running AIS algorithms is having long time of training phase for generating detectors (Low speed of training phase) [8]. So, for resolving this issue we implement our work on Hadoop Platform. The Apache Hadoop software library is a framework that allows for the distributed processing of big data sets across clusters of computers using MapReduce programming model. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The Apache Hadoop project [10] develops open-source software for scalable, reliable, distributed computing. Also, Hadoop Map/Reduce is a programming model for easily writing applications which process large amounts of data in-parallel on large clusters of commodity hardware in a reliable, fault-tolerant way [11]. A Map/Reduce job splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel method. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Usually, both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks [12, 13]. Typically the compute nodes and the storage nodes are the same, that is, the Map/Reduce framework and the Hadoop Distributed File System (HDFS) are running on the same set of nodes (Figure **3**).

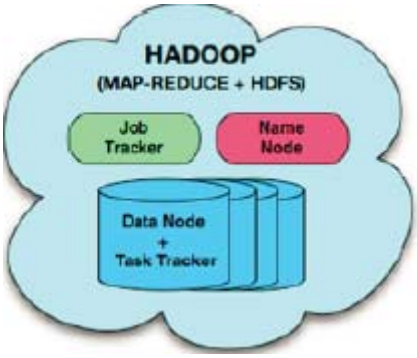

Figure 3. Apache Hadoop [11]

a. Training and Testing Phase

In training phase, we try to normalize input data and prepare the algorithm. Afterwards, generate normal detectors randomly for each mapper (by the Random

Detector Function). So, Affinity is calculated using Euclidean distance. For matching detectors with strings the NSA needs a threshold. In training phase, if the distance between detector and records was less than threshold, the detector has been removed because it detects a self record, otherwise, the detector detects a nonself record and the system keeps that for fraud detection.

After that, the detector set is generated and NSA is used to ensure that no detector matches any self pattern. So, reducer shuffle and sorts mapper output. At least, in testing phase, new data from system can be matched against these detectors to detect frauds. The testing phase is done serial. But in testing phase if the distance was less than threshold, the detector detects a nonself record.

V. EXPERIMENTS

a. *Data set*

We have obtained our database from a large Brazilian bank, with registers within time window between Jul/14/2004 through Sep/12/2004. The dataset consists of 300,000 records. Each register represents a credit card authorization, with only approved transactions excluding the denied transactions. All data fields are considered in numerical form. Totally, we consider %70 of dataset for train and %30 for test.

b. *Time (second)*

We executed the NSA on the cloud with three thresholds amounts. By the running NSA on the cloud platform with the several of Mapper, time significantly reduce. Also, when the threshold goes to higher value, the time of training phase increases too.

- Threshold=1

  As you see in Figure **6**, the time of training phase in based algorithm is about 10,620s, while in the parallelized NSA on the hadoop and running with several of Mapper reduced to 74s.

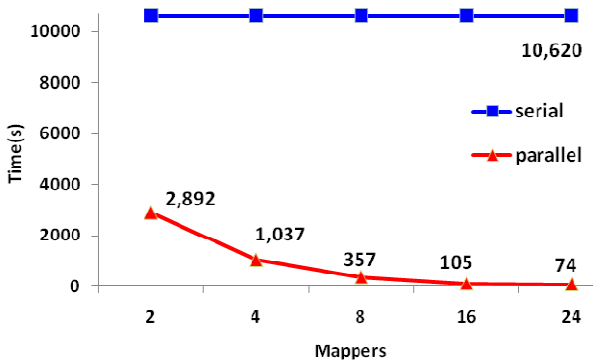

Figure 4. Time with Threshold= 1

- Threshold=1.5

  Figure **5** show that, the long time of running serial algorithm is about 23,820s, while in the parallelized NSA on the hadoop and running with several of Mapper that reduced to 78s.

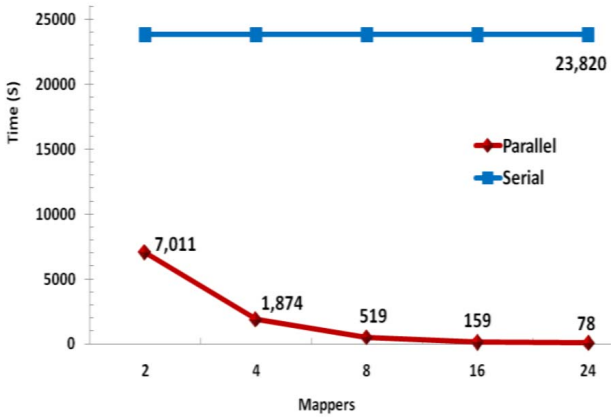

Figure 5. Time with Threshold= 1.5

- Threshold=2

  In the Figure **6** as you can see, time reduce to 3,084 in comparison to the time of based algorithm that is 80,760. So whatever

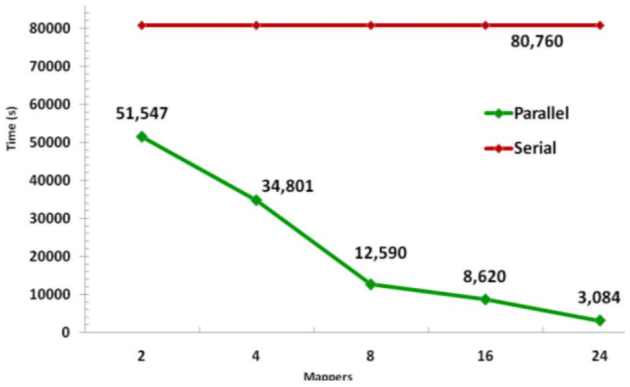

Figure 6. Time with Threshold= 2

CONCLUSION

The purpose of this paper is decrease the time of fraud detection system for credit card fraud. For achieving to this aim, we implement one of the AIS algorithms on the Hadoop by MapReduce programming model. Because of the AIS algorithms have the long time training time, we parallelized NSA on the cloud with several of Mapper. Two classes that called Map function and Reduce function added to the NSA. The result shown that the time of training time dramatically decreases. That means, our credit card fraud detection system can detect frauds quickly. As you seen in the results, with increasing of threshold amount, the time of algorithm rising too.

ACKNOWLEDGMENT

The authors would like to thank all those who contributed to this paper. Further to this, we gratefully acknowledge those in the cloud computing team at the Department of Computer engineering and Information Technology, Amirkabir University, IRAN and Mazandaran University of Science and Technology, Babol, IRAN.

REFERENCES

[1] A. Srivastava, A. Kundu, and et al, "Credit Card Fraud Detection Using Hidden Markov Model", IEEE Transactions on dependable and secure computing, Vol. 5, No. 1, pp. 37-48, 2008.

[2] A. Richard, "Credit Cards as Spending Facilitating Stimuli: A Conditioning. Interpretation," J. Consumer Research, vol. 13, no. 3, pp. 348-356, 1986.

[3] S. Ghosh and D.L. Reilly, "Credit Card Fraud Detection with a Neural-Network," Proc. Int'l Conf. System Science, pp. 621-630, 1994.

[4] M. Gadi, X. Wang, A. Lago, "Credit Card Fraud Detection with Artificial Immune System", Springer, 2008 K. Elissa.

[5] L. de Castro and J. Timmis. Artificial Immune Systems: A New Computational Approach. Springer-Verlag, London. UK., September 2002.

[6] L. N. de Castro and F. J. Von Zuben. (1999) Artificial Immune Systems: Part I—Basic Theory and Applications. FEEC/Univ. Campinas Brazil. [Online]. Available: http://www.dca.fee.unicamp.br/~lnunes/immune.html.

[7] S. Forrest, A. Perelson, L. Allen, and R. Cherukuri. Self-nonself discrimination in a computer. In Proc. of the IEEE Symposium on Security and Privacy, pages 202–209, IEEE Computer Society, 1994.

[8] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff. A sense of self for unix processes. In Proc. of the IEEE Symposium on Research in Security and Privacy, pages 120–128. IEEE Computer Society Press, 1996.

[9] Kim J, Bentley P (2001), Evaluating Negative Selection in an AIS for Network Intrusion Detection, Genetic and Evolutionary Computation Conference 2001, 1330-1337.

[10] Apache Hadoop project: http://hadoop.apache.org.

[11] Colin White, "MapReduce and the Data Scientist," BI Research January 2012.

[12] Dean, J. and Ghemawat, S.: MapReduce: Simplified data processing on large clusters. In Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation, USENIX Association, Volume 6, pp. 10-10, December 2004.

[13] M. Miller, Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online: Que, Aug. 2008.