Performance Evaluation of a Fraud Detection System based Artificial Immune System on the Cloud

Elham Hormozi
Computer Engineering and Information Technology
Mazandaran University of Science and Technology
Babol, Mazandaran (North of Iran), 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

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

***Abstract*—Fraud is defined as the unlawful and intentional misrepresentation which can lead to actual or potential disadvantage to another individual or group. Fraud detection is a topic applicable to many industries including banking and financial sectors, insurance, law enforcement, and more. Credit card fraud is a major problem in the financial industry [1]. Therefore, real time fraud detection is a vital issue. Indeed, we implement the progress of Negative Selection Algorithm an anomaly detection approach in AIS on the cloud. As a result of our experiments, in serial NSA the time of training phase is around 23800s whereas parallel NSA training phase time is 78s. Also, in parallel algorithm the detection rate increased around %50 compared to the serial algorithm. But we concluded false positive rate a little raised that compared to increase the detection rate is almost negligible. Tests are done with real data sets, and all executions are run using mapreduce and apache hadoop.**

Keywords-fraud; fraud detection;credit card fraud; artificial immune system; cloud computing; negative selection algorithm.

I. INTRODUCTION

Fraud is the obtaining of financial advantage or causing of loss by implicit or explicit deception; it is the mechanism through which the fraudster gains an unlawful advantage or causes unlawful loss. Credit card fraud is a major problem for financial institutions globally. It is responsible for billions of dollars in losses per annum globally. Fraud can be defined as criminal deception intended to result in financial gain. Along with the developments in the Information Technology, fraud has been extending all over the world with results of huge financial losses [10]. With the increased use of credit cards, fraudsters are also finding more opportunities to fraudulent activities which effects bank as well as card holders to large financial losses [1]. Fraud detection is a vital business function for minimizing the effects of unauthorized transactions upon organizations customer service delivery, bottom line expenditure and business reputation through deployment of innovative fraud technology frameworks. Fraud detection is about identifying fraud as soon as possible and responding to it [16]. Institutions are now moving towards increasingly proactive methods of fraud detection for real time screening of financial data, and triggering of a preventive response prior to transaction completion in order to minimize the potential fraud deficit [2]. This work uses the negative selection algorithm for fraud detection on credit card transaction data with mapreduce interface. The Negative Selection Algorithm (NSA) is one of the artificial immune system algorithms (AIS).

The remainder of the paper is organized as follows. Section II introduces Artificial Immune System. Section III presents parallelization and its problems. Section IV provides implementation of NSA on cloud, Section V, explains performance critical and finally in Section VI we present experiments and results.

II. ARTIFICIAL IMUUNE SYSTEM

Artificial Immune system is inspired from biological immune system. 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. AIS is an umbrella term that covers all

the efforts to develop computational models inspired by biological immune systems. Among various mechanisms in the immune system that are explored for AIS, negative selection, immune network model and clonal selection [12] are still the most discussed models [2]. So AIS is suitable to be used in credit card fraud detection.

A. Negative Selection Algorithm

The Negative Selection is one of the mechanisms of the natural immune system that has inspired the developments of most of the existing AIS. The purpose of negative selection is to provide self-tolerance to T-cells. It detects unknown antigens, without reacting with the self cells. In the T-cell maturation process of the immune system, if a T-cell in thymus recognizes any self cell, it is eliminated before deploying for immune functionality. Similarly, the negative selection algorithm generates detector set by eliminating any detector candidate that match elements from a group of self samples [3]. The main idea of this algorithm is to generate a set of detectors by first randomly making candidates and then discarding those that recognize training self-data, and then these detectors can later be used to detect anomaly. The starting point of this algorithm is to produce a set of self strings, S, that define the normal state of the system. The task then is to generate a set of detectors, D, that only bind/recognize the complement of S. These detectors can then be applied to new data in order to classify them as being self or non-self, thus in the case of the original work by Forrest *et al* , highlighting the fact that data has been manipulated. The algorithm of Forrest *et al* produces the set of detectors via the process outlined in below [4].

input: S_{seen} = set of seen known self elements
output: D = set of generated detectors
begin
repeat
Randomly generate potential detectors and place them in a set *P*
Determine the affinity of each member of *P* with each member of the self set S_{seen}
If at least one element in *S* recognizes a detector in *P* according to a recognition threshold,
then the detector is rejected, otherwise it is added to the set of available detectors *D*
until Stopping criteria has been met
end

B. Challenges

- Long time of training phase for generating detectors (Low speed of training phase)
- The large amount of training records
- Low fraud detection rate

C. Proposed Solution

- Parallelization
- Implementation in cloud platform with *mapreduce*

III. PARALLELIZATION

To increase the speed of the process, we need to run parts of the program in parallel. In theory, this is straightforward: we could process different years in different processes, using all the available hardware threads on a machine. There are a few problems with this, however. First, dividing the work into equal-size pieces isn't always easy or obvious. In this case, the file size for different years varies widely, so some processes will finish much earlier than others. Even if they pick up further work, the whole run is dominated by the longest file. An alternative approach is to split the input into fixed-size chunks and assign each chunk to a process. Second, combining the results from independent processes can need further processing. In this case, the result for each year is independent of other years and may be combined by concatenating all the results, and sorting by year. If using the fixed-size chunk approach, the combination is more delicate. For this example, data for a particular year will typically be split into several chunks, each processed independently. We'll end up with the maximum temperature for each chunk, so the final step is to look for the highest of these maximums, for each year. Third, you are still limited by the processing capacity of a single machine. If the best time you can achieve is 20 minutes with the number of processors you have, then that's it. You can't make it go faster. Also, some datasets grow beyond the capacity of a single machine. When we start using multiple machines, a whole host of other factors come into play, mainly falling in the category of coordination and reliability. Who runs the overall job? How do we deal with failed processes? So, though it's feasible to parallelize the processing, in practice it's messy. Using a framework like Hadoop to take care of these issues is a great help [5].

IV. IMPLEMENTATION IN CLOUD PLATFORM

AIS has a long training time. That's why we have implemented the model in cloud computing systems to reduce this time [15]. To take advantages of the parallelization by using Apache Hadoop, we need to use our query as a MapReduce job. So, we will be able to run it on a cluster of machines [9].

A. Hadoop Mapreduce

MapReduce is a programming model and software framework introduced by Google to support distributed computing on large data sets on clusters of computers [14]. Hadoop MapReduce is an Apache open source project that develops parallel applications without any parallel programming techniques. The applications could be easy deployed and executed. Hadoop works with the HDFS[1] and processes huge datasets (e.g. more than 1TB) on distributed clusters (thousands of CPUs) in parallel [6]. It allows programmer writing the code easily and fast [7]. MapReduce works by breaking the processing into two

[1] **H**adoop **D**istributed **F**ile **S**ystem

phases: the map phase and the reduce phase (**Figure 1**). Each phase has key-value pairs as input and output, the types of which may be chosen by the programmer. The programmer also specifies two functions: the map function and the reduce function [6].

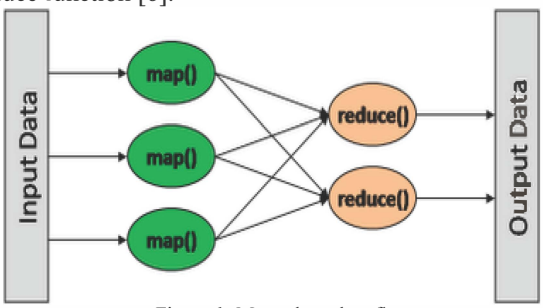


Figure 1. Mapreduce data flow

B. *Final Implemented Model*

As shown in **Figure 2** in training phase, at first we normalize input data and prepare the algorithm. It means that data should be between 0, 1 for simplicity. Then, each of mapper equally provides a random set of detectors. After that the Euclidean distance of detectors is calculated for each record (threshold). If this distance is less than the threshold (threshold is determined by programmer), it means that detector recognizes self cell, thus it must be eliminated. The self set is constructed using data patterns which represent the normal operation of the system. Patterns should be long enough to capture any important system behaviors. After that, the detector set is generated and negative selection is used to ensure that no detector matches any self pattern [8]. So, reducer gathers and sorts mapper output. Finally in testing phase, new data from system can be matched against these detectors to detect frauds. The testing phase is done serial.

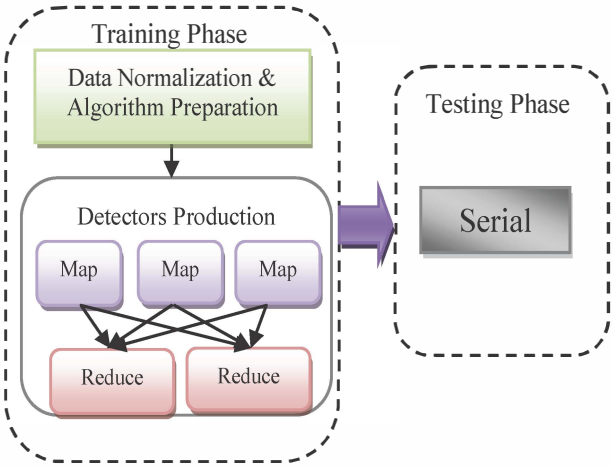


Figure 2. Final Implemented Model with Mapreduce

V. PERFORMANCE CRITERIA

The stream of accounts flagged by the fraud detection system includes the compromised accounts, true positives (TP), and incorrectly implicated cases, false positives (FP). Complementarily, accounts that pass through the fraud detection system with no alert created is a mixture of legitimate accounts, true negatives (TN) and missed fraudulent cases, false negatives (FN). Various performance criteria can be used in an application to the fraud detection problem. These include the measures such as misclassification rates, the area under receiver-operating curve (ROC) and more recently proposed criteria such as the area under the modified ROC curve called (the performance curve). It is typically considered that the error committed in assessing a fraudulent case as legitimate (FN) is more serious than the complementary type of error (FP) [2].

VI. EXPERIMENTS AND RESULTS

This section presents the experimental results of the implemented NSA over the cloud computing platform on credit card transactions provided by the collaborating bank. In general we consider %70 of dataset for train and %30 for test. 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.

A. *Time*

Experiments show that the long time of training phase decreases with implementation on cloud. As shown in **Figure 3**, in serial NSA the time of training phase is around 23800^{s} whereas parallel NSA training phase time is 78^{s}.

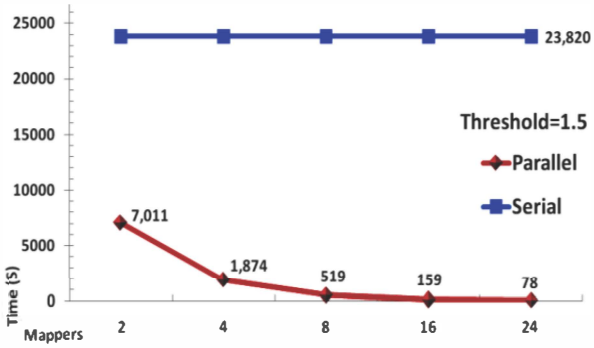


Figure 3. Training phase time in serial NSA and parallel NSA with different mapper (Threshold=1.5)

But as you see in **Figure 4**, threshold value increases by increasing the amount of time.

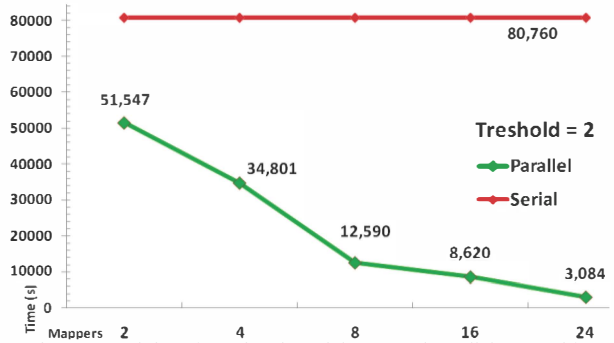
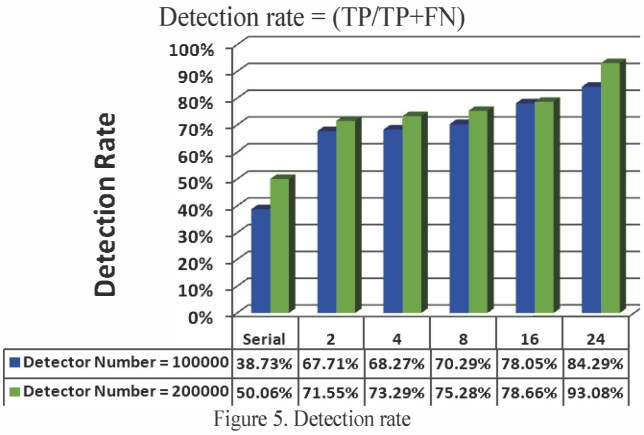


Figure 4. Training phase time in serial NSA and parallel NSA with different mapper (Threshold=2)

B. *Detection Rate*

As shown in **Figure 5**, the detection rate increases with implementation on cloud environment. In parallel algorithm the detection rate increased around %50 compared to the serial algorithm. We used two sets of detectors to experiment and we conclude that increasing the number of detectors increases the detection rate.

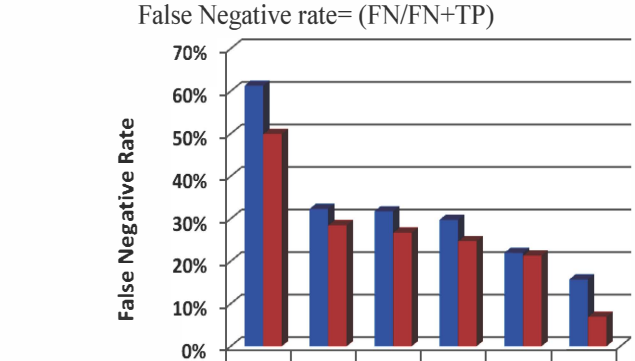


	Serial	2	4	8	16	24
Detector Number = 100000	38.73%	67.71%	68.27%	70.29%	78.05%	84.29%
Detector Number = 200000	50.06%	71.55%	73.29%	75.28%	78.66%	93.08%

Figure 5. Detection rate

C. *False negative rate*

As shown in **Figure 6**, false negative rate decreased too. We used two sets of detectors to experiment and we conclude that increasing the number of detectors increases the FN rate too.

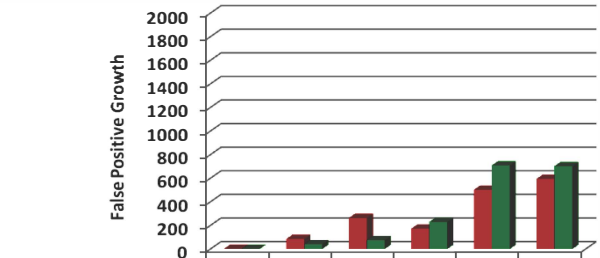


	Serial	2	4	8	16	24
Detector Number = 100000	61.27%	32.29%	31.73%	29.71%	21.95%	15.71%
Detector Number = 200000	49.94%	28.45%	26.71%	24.72%	21.34%	6.92%

Figure 6. False Negative rate

D. *False positive rate*

The experiments show that false positive rate a little raised that compared to increase the detection rate is almost negligible. As shown in **Figure 7**, the FP raised. However, the FN is more serious than the FP. So false positive rate a little raised that compared to increase the detection rate is almost negligible. Also we conclude that increasing the number of detectors increases the FP rate.



	Serial	2	4	8	16	24
Detector Number = 100000	0	84	262	172	501	593
Detector Number = 200000	0	40	75	229	708	701

Figure 7. False Positive rate

CONCLUSION

In this paper, we present a performance evaluation of fraud detection based AIS on the cloud systems. For this work, we added two functions including map and reduce function to NSA. Finally the experimental results show that implementation of NSA on cloud and parallelization by mapreduce, significantly reduce the training time and produce higher detection rate and lower FN rate.

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] CA. W. Paasch, "Credit card fraud detection using artificial neural networks tuned by genetic algorithms," Thesis (Ph.D.), The Hong Kong University of Science and Technology, February 2008.

[2] M. Krivko, "A Hybrid Model For Plastic Card Fraud Detection Systems", Expert Systems with Applications, Vol. 37, No. 8, pp. 6070-6076, 2010.

[3] J.R. Al-Enezi, M.F. Abbod & S. Alsharhan, "Artificial Imuune Systems–Models, Algorithms and Applications," Electronic and Computer Engineering Department, School of Engineering and Design, Brunel University, UK, IJRRAS May 2010.

[4] AISWeb, The Online Home of Artificial Immune Systems, http://www.artificial-immune-systems.org/algorithms.shtml #neg-alg, (Last Modified: 25th Novemeber 2012).

[5] T. White, Hadoop: The Definitive Guide, 2nd Edition, O'Reilly, 2011, pp.

[6] D. Borthakur, “The Hadoop Distributed File System: Architecture and Design,” 2007 The Apache Software Foundation.

[7] MR. Lyu, Chu Yan Shing, Wu Bing Chuan, “Department of Computer Science and Engineering The Chinese University of Hong Kong,” Department of Computer Science and Engineering, CUHK 2010-2011.

[8] D. Dasgupta, K. KrishnaKumar, D. Wong, M. Berry, “Negative Selection Algorithm for Aircraft Fault Detection, Division of Computer Science, University of Memphis, TN.

[9] A. T. Velte, T. J. Velte, and R. Elsenpeter, “Cloud Computing: A Practical Approach”, McGraw-Hill Publishing, United States, pp. 3-22, 2010.

[10] S. Panigrahi, A. Kundu, and et al, "Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning", Information Fusion, Vol.10, No. 4, pp. 354-363, 2009.

[11] W. Albrecht, C. Albrecht, and et al, "Current Trends in Fraud and its Detection", Information Security Journal: A Global Perspective, Vol. 17, No. 1, pp. 2-12, 2008.

[12] Ada, G. L. & Nossal, G. J. V. (1987), “The Clonal Selection Theory”, Scientific, American, 257(2), pp. 50-57.

[13] Dr. Dipankar Dasgupta, “What Are Artificial Immune Systems ,” Department of Computer Science,University of Memphis, IEEE, 2009.

[14] M. Miller, “Cloud computing: Web-based applications that change the way you work and collaborate online”, Que Publication, 2008.

[15] N. Soltani, Mohammad Kazem Akbari, Mortaza Sargolzaei Javan, “A New User-Based Model for Credit Card Fraud Detection Based on Artificial Immune System,” The 16th CSI International Symposium on Computer Architecture & Digital Systems, CADS 2012.

[16] R. Huang, H. Tawfik, and A.K. Nagar, “A Novel Hybrid Artificial Immune Inspired Approach for Online Break-in Fraud Detection,” Faculty of Business and Computer Sciences, Liverpool Hope University, Liverpool, United KingdomInternational Conference on Computational Science, ICCS 2010.