Anomaly Detection using Hadoop and MapReduce Technique in Cloud with Sensor Data

Ihab I.M. Alghussein
College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria, Egypt

Walid Mohamed Aly
College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria, Egypt

Mohamad Abou El-Nasr
College of Engineering, Arab Academy for Science, Technology and Maritime Transport, Alexandria, Egypt

ABSTRACT
This paper presents a model to observation the Cloud computing for any anomalous activity. Hadoop it is a largely used open source Cloud Computing framework to huge data. It uses the model Machine Learning technique to detect classify anomalies of sensory observation and help to in ensuring the stabilization of virtual sensor networks. The framework it's built on top of the Hadoop and MapReduce implementation which is use one of the Machines Learning techniques to detect these anomalies. Preliminary results show that our classification mechanism is promising and able to detect anomalous events that may cause a threat to the Cloud Computing.

General Terms
MapReduce, Hadoop, Anomaly Detection, Machine Learning, Cloud Computing, Weka.

Keywords
MapReduce, Hadoop, anomaly detection, Machine Learning, Cloud Computing.

1. INTRODUCTION
Cloud Computing is rapidly getting more and more common in distributed computing environment. Cloud environments are used for storage and processing of data. Cloud Computing supplies infrastructure, applications and programs by internet. Cloud computing is a model to enable an easy access on network demand to the shared pool of computing resources form like network ,services storage ,services and applications That can be quickly supplied and released with little management effort or services supplier interaction. The next models are showed by considering the deployment scenario as, Private Cloud, Public Cloud, Community Cloud and hybrid Cloud [1] [2].

MapReduce is a Cloud framework for processing match problems by mass dataset developed by Google as a popular open source execution of MapReduce, Hadoop has been largely used in big companies like, Yahoo and Ebay for data drastic careers [3].

However, successful execution of such jobs is not easy. On one hand, devices used in cloud are usually low cost ones which would make higher error probability in hardware, and on the other hand, some problems such as program bugs will also cause system performance degradation. MapReduce usually dividing the input data set into independent divisions that rely on the size of the dataset of the number of nodes used and have two main jobs Map and Reduce, The Map takes a series of (Key and Value) pairs, processes every one of them, and generates zero or more output (Key and Value) pairs. The input and output kind of the map can be "often are different from each one, and then the Reduce function aggregates and combines all intermediate values list" output coming from the Map function which have the same intermediate key [4].The Hadoop framework has distributed files system called (HDFS) Hadoop Distributed File System used to support the processing and management of big scale data sets. Furthermore, the MapReduce in Hadoop is designed to work efficiently with HDFS by moving the computation process for data and not the other way around to allow Hadoop to achieve high data locality [5].

Problems and errors are always reflected as system anomalies in which anomalies may cause longer job times and deterioration of data transfer speed. Moreover, if they're critical, the task might get interrupted. Therefore, it's essential to find out anomalies in time for reducing and avoiding losses. Certain characteristics of MapReduce make MapReduce differs and that makes various tasks with the same configuration environment that causes inconsistent execution times , although the same task executed at various times , Run time may differs too as a result of volatilization and doubt of the system . As a result, some of the ordinary ways depended on response time are not operative to find out anomalies in MapReduce area [6]. Those methods use common immovable time out threshold, where tasks are going to be known as anomalies if their implementation times go beyond the threshold. Besides MapReduce although has specifications of multi nodes and divided. i.e. like Ebay which owns 532 nodes clusters (8*532 cores , 5.3PB) in total for MapReduce [7]Tasks in Map Reduce are going to be implemented on a lot of nodes which are connected to each others . Thus, the methods intend to find out anomaly in one node case [8, 9] are not matched for Map Reduce environment.

The rest of this paper is organized as the next. In part II. We suggest related work focused on some previous researches using Hadoop and MapReduce in detection task either anomaly or non anomaly and presents their results. Experimental discussion and evaluation are described in part III. Part IV shows result and future work.

2. RELATED WORK
In these years, various algorithms have been employed to reduce the parameter dimensions of various problems. This section shows the related researches briefly.

Shubhalaxmi Kher et al. proposed a model to watch the smart grid for any unexpected malicious attack or activity. The sample uses machine learning to find out and classify anomalies from the sensory notices. It's helpful for emphasizing the and stability security of the smart grid. The sample is supported by the real time data collected using

Wireless sensing Network as an overlay network on the power distribution grid. The overlay "WSN" devices uses a cluster topology at all towers to collect local information about the tower that is increased by the linear chain topology to join to the main station . Experiment Design Trained & Input data was acquired from the grid "WSN" (34% for testing and 66% for training) ,Analysis the sensor data using machine learning software ,the experiment goal is to decide when an anomaly being induced or not ,Experiment used Decision Tree J48 classifier Results showed that 98.14% correct anomalies detection .The results were contrasted with other machine learning techniques:, Decision Table, ADtree, ZeroR and Random Forest The comparison shows that categorization is best in the experiment of C4.5"J48" classifier. The main goal of this research is to detect anomalies using machine learning technique in sensor network smart grid, Sensor data was collected from multiple sensors, Data analyzed using decision tree J48 to detect anomalies. A result shows that decision tree J48 is the highest classification rate compared with other machine learning techniques [10].

Kai Wang's model depends on peer similarity, where he uses density depend on clustering to find out in MapReduce environment on OS level metric to achieve true time analysis. Both our anomaly detection way and the peer similarity are appraised through experiment. So, when we differentiate that with other methods suggest reflect easy, efficient and sensitive features. Moreover, this method can be spread in both online and offline area. It also does not need to treatment with complicated and large logs and the difficulty of algorithm it's just O(n*m), where n and m stand for the amount of slave nodes and the number of metrics. Moreover, the anomalous information could be reported in time even if there are many nodes. This method also provides the basic information related to anomaly like the super ordinate node, the occurring time, the type of the abnormal metric, and the deviation from the normal metric value. Later, there is a plan to store historical information to perform correlation analysis to examine if the anomaly is true or false, and to also analyze the main reason of anomaly [11].

Shilton et al. proposed a Support Vector machine approach to multi class classification and anomaly detection in WSNs. Their work requires data to be known classes to be classified into after that these data points which can't be classified are considered anomalous. One of the issue that the authors present is the difficulty in setting one of the algorithm's parameters. In particular, changing the value has a direct impact on the rate in which the algorithm produces false negatives, or in which the algorithm detects true positives to reduce the effect of the computational complexity of these algorithms [12].

Lee et al. proposed an approach to detect anomalies by leveraging Hadoop. Apache Hadoop is an open source software framework that strengthens a wide range of applications, stream processing, including machine learning, graph computation and ETL. Their work is preliminary in nature and mostly addresses concerns and discussion related to anomaly detection in Big Data [13].

Xie et al. proposed an online anomaly detection algorithm where their work uses a histogram based approach to detect anomalies within hierarchical WSNs. The disadvantage in their approach its lack of consideration for multivariate data. The work focused accurately in developing histograms for the contents of the data but not the context of the data. The authors focused on future work, indicating that inclusion of contextual data would improve the generality and detection performance of their algorithm [14].

3. METHODOLOGY

3.1 Data Sensing

The form of the data stream collect Dataset Collection form Sensors which may be large volumes of true time notices collected from the environment [15] ,some sensors collect data like temperature , damp light and chemicals etc., . This type of data is named univariate data. Wireless Sensors Networks are planned to collect multiple types of data form from the field at the same time that data is named multivariate. In these networks each node is supported by more than one sensor to collect various kinds of data in same time. In multivariate data, each kind of data is named a feature attribute. A sensing data measurement in univariate data with attribute can be readily find out if one or more than an attribute anomalous with respect to that attribute of other data compared with the attributes with the attributes of other data instances[16] [**Figure 1**]. By contrast, anomaly detection in multivariate WSNs is forced with the individual attribute which may not display irregular behavior, but when they are taken together they may show anomalous behavior [17]. Analysis of multivariate data cost is expensive, and anomalies exposure on multivariate data provides high accuracy if the connections between various attributes are accurately used [18] [19].

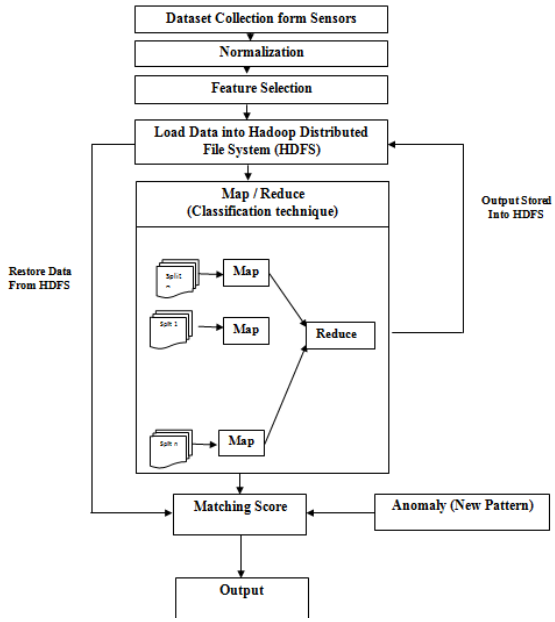

Fig 1: System block diagram

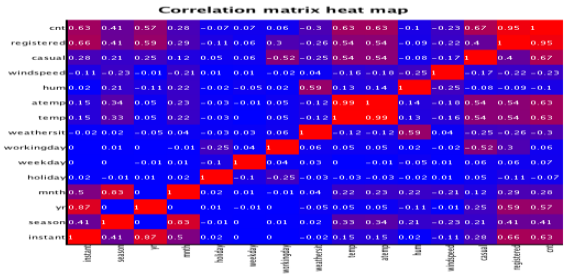

Fig 2: Correlated to some events in the town which easily are traceable

Sensor Data correlation has temporal and spatial connection between sensor readings. When sensor readings are tested at closer time stamps, they tend to be similar and this is what temporal correlation means. Thus, the readings monitored at one time immediate are joined to the readings watched at the last time instants. While spatial correlation observes that the readings from sensor nodes geographically near to each one should be not different or correlated [20].The emergent and Spatial correlation between sensing data attributes are helpful to limit the source of anomaly [**Figure 2**].

3.2 Hadoop Distributed File System (HDFS)

HDFS is a scaling portable and distributed file system written in Java for the Hadoop framework that is the file system component of Hadoop, It stores file system metadata and application data alone. As in the case in other distributed file systems, like Lustre, GFS and PVFS, HDFS stores metadata on dedicated servers called the NameNode. Applications data it is stored on other the servers called the DataNode. All servers are fully communicated and connected with eachother used TCP based protocols. Default HDFS stores three separate copies of the all the data block to ensure reliability, availability, and performance. In large clusters through three these replicas are spread across different physical racks, so HDFS is flexible towards two common failure scenarios: individual datanode breakdown and failures the networking equipment that combines an entire rack offline. Replicating blocks across physical machines it also increases opportunities to share locating data processing in the time table of MapReduce jobs, since many of the copies yield more opportunities of exploitation locality [21].

3.3 MapReduce

MapReduce, Records it is dealt with isolation by tasks called the MappersThe output of the Mappers is then got together within into the second set of tasks names the Reducers, where outputs form various Mappers can be joined together. Problem fitting for treating with MapReduce have to usually be readily separated into independent subtasks that can be treated in parallel. The Map and Reduce function are both specialized in the terms of date structured in key and value pairs. Each reduce take just treats and receives information for one specific key at a time and outputs the data it treats as a (Key, Value) pairs. The Hadoop MapReduce engine has JobTracker and one of many TaskTrackers.A MapReduce work has to be managed by job trackers which then separate the jobs into tasks processed by the task trackers. JobTrackers sends jobs and assigns splits (splits) to mappers or reduces as each step finishes. TaksTrackers implements task send by the JobTracker and reports rank to JobTrackers.(Maps) step: The master node gets the input and splitting into smaller sub problems then dispenses then to worker nodes that may do this again in turn which leads to a multi level structure of tree. The worker node treats the passes answer block to its master node and smaller problem reduce step. The master node the collects the answers to all the sub problems and joins the in some way to from the output of answer to find solution to the problem [22].

4. LEARNING ALGORITHMS

4.1 Random Forest

The Random Forests is an active prediction aid in data mining. It uses the Bagging way to give a rashly sampled set of training data for every of the trees. This Random Forests way also semi randomly chooses division properties; a rash part of a given size is given from the space of enabled division characteristic. The best division is property deterministically chosen from that part [23].

Pseudo code:

To generate c classifiers:
For i = 1 to c **do**
Rashly model the training data D with echange to give Di
Form a root node, Ni containing Di
Name BuildTree(i N)
Name BuildTree(i N)
End for
BuildTree(N):
If N has instances of just one class **then**
Return
Else
Rashly choose x% of the enabled division property in N
Choose the property F with the best information gain to split on
Form f kid nodes of N , 1 N ,..., Nf , where F has f enabled values (1 F , … ,Ff)
For i = 1 to f **do**
Set the contents of N i to D i, where D i is all instances in N that parallel F i
Name BuildTree(N i)
End for
End if

4.2 OneR(One Rule)

OneR is a naïve and very effective sort classification algorithm frequently use in machine learning applications. OneR creates a one level decision tree. OneR can infer typically naïve, yet accurate classification rule from a set of examples. OneR can also treat missing values and numeric attributes offering adaptation in spite simplicity. The OneR Algorithm makes one rule for each attribute in the training data, then chooses the rule with the least error rate as it (One Rule) to crate the rule for an attribute , the most frequent rank or each attribute value have to be limited . The most frequent rank is easily the rank that appears most often for that attribute value [24].

Pseudo code:

1. For each attribute A,
   i. For each value VA of the attribute, and make a rule as follows Counting how many time each class appears
      Find the most frequent class Cf
      Create a rule when A=VA
      Class attribute value = Cf
   ii. Calculate the error rate of all rules

4.3 Bagging

Bagging "bootstrap aggregating" is a different way to combine decision trees or other base classifiers. Both for boosting, the main learning algorithm is being used frequent in a series of rounds. from other hand, the way in which the base learner is called it is different than in boosting. Specially, on every round, the bootstrap replicate train the base learner of the real training slips. Suggest the training slip includes m sample, next a bootstrap replicate is a new training set that also includes of m Sample in which it is formed by repeatedly selecting uniformly at rash and with replacement m samples from the real training slip. This shows that the exact

sample may appear varied times in the bootstrapping replicate, or it may disappeared.

However, on every of T rounds of bagging, a bootstrapping replicate is formed from the real training slip. A main classifier is then prepared on this replicate, and the process goes on. After T rounds, a final joined classifier is created that easily predicts with the total vote of each main classifiers [25].

Pseudo code:

1. Given training data *(x1,y1),...,(xm,ym)*
2. For *t=1,..., T*:
   i. form bootstrap replicate dataset St by selecting m random samples from the training set with replacement
   ii. let *ht* be the output of training main learning algorithm on *St*
3. output combined classifier:
   H(x)=majority(h1(x),...,hT(x))

In the part two of the assignment it is to write an R function called "bag.trees" implements this bagging procedure. The input parameters and that are returned of this function should be of such as those for your boosting a function. such as the boosting function, your bagging function should used the decision trees as the base learner.

4.4 J48 (C4.5)

C4.5 creates decision tree from a set of training data using the conception of date entropy. The training data is a *S = s1, s2…*Of proposed classified models. Each sample will be *Si = x1, x2…*Is a vector where *X1, X2…* representing attributes of this model. The training information is augmented with vector *C = c1, c2…*a where *c1, c2…* represent the class not which sample follows [26].

Pseudo code:

1. Examine for main cases
2. For every attribute a
3. Find the normalized information obtain from division on a
4. Let a_best be the attribute with the most normalized information gain
5. Create a decision node that divides on a_best Recourse on the sub lists acquired by divide sing on a best, and add those nodes as children of node

5. EXPERIMENTAL RESULTS

To limit occurrences and areas of an event such as, anomaly detection on the Cloud Computing are produced next result and training information was got from the Cloud Computing Wireless Sensor Data experiment .We used WEKA machine learning software for analysis .The taken data owns 17380 examples is represented in ranks that represent true examples and different anomaly examples. Observed sensory information is analyzed using some learning techniques aids. The goal is to limit the case of the monitored environment through multi sensor Observation. Analysis is applied using WEKA [27], an open source machine learning data mining software largely used for analysis in various applications.

To measure and test the performance on the chosen algorithm that are OneR, J48 (c4.5), Bagging and Random Forest (RF) .We use the typical experimental implementation as proposed by WEKA, Just 75% of the total used for examining and training then get the model out . The model outputs are divided into a lot of sub parts for easier evaluation and analysis. The first part is truly and untruly classified examples will be divided in percentage value and the mean absolute error and root relative squared are error too in the percentage for references and evaluation also, TP Rate and FP Rate are going to be in numeric value

The outputs are found in Table 1 and 2 below. Table 1 majorly summarizes the outputs built on accuracy and the time taken for each test and Table 2 shows the result based on error, FP Rate and TP Rate.

Table 1. Cross validation and classification rates

Algorithm	oneR	J48	Random forest	Bagging
Correctly classified instances %	94.3989	94.5355	99.8634	97.1311
Incorrectly classified instances %	5.6011 %	5.4645%	0.1366 %	2.8689
Time	00:02:21	00:02:21	00:02:24	00:02:52
Precision	0.920	0.921	0.998	0.958
Recall	0.944	0.945	0.999	0.971

Table 2. Errors

Algorithm	oneR	J48	Random forest	Bagging
TP Rate	0.944	0.945	0.999	0.971
FP Rate	0.056	0.055	0.001	0.029
Mean absolute error	0.0002	0.0002	0.001	0.0011
Root mean squared error	0.013	0.0089	0.015	0.0167
Relative absolute error %	5.6093	5.4725	36.6338	39.2467
Root relative squared error %	33.495	23.394	40.234	44.067

[**Figure 3**] The graphical representations of results.

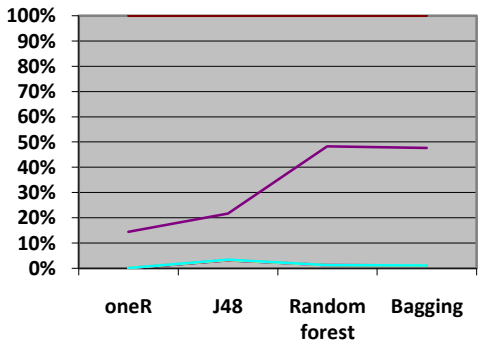

Fig 3. A graphical representation of Table 2

6. CONCLUSION

In this work, Cloud computing and Mapreduce monitoring framework are presented using machine learning for detecting anomalies via multiple sensors. The long Term goal is to predict events in the distribution network such as, intrusion

and suspicious activities in Cloud Computing. Preliminary study exhibits the usage of sensors and collection of sensory observations on a continuous basis if properly integrated for analysis where a better prediction model can be built. Data analyzed using Random Forest to detect anomalies the random forest is the highest classification rate compared to other machine learning techniques.

7. ACKNOWLEDGMENTS

- Many thanks to all members of the Waikato machine learning group and the external contributors for all the hard work they have put into WEKA.

- This dataset was provided by Fanaee T, Hadi, and Gama, Joao using data from Capital Bikeshare.

8. REFERENCES

[1] Mell and T. Grance, ―The NIST definition of cloud computing ,‖ NIST special publication, 800(145), 7, 2011.

[2] An architecture for overlaying private clouds on public providers Shtern, M. ; Simmons, B. ; Smit, M. ; Litoiu, M. Publication Year: 2012 , Page(s): 371 – 377.

[3] Hadoop website. http://hadoop.apache.org/. Last vist 14 augest 2014.

[4] "MapReduce: Simplified Data Processing on Large Clusters", by JeffreyDean and Sanjay Ghemawat; from http://research.google.com/archive/mapreduce.html Last vist 12 augest 2014.

[5] T. White, Hadoop: The Definitive Guide, original ed.O'Reilly Media, Jun. 2009.

[6] Kai Wang, Ying Wang, Bo Yin, "A Density-Based Anomaly Detection Method for MapReduce," nca, pp.159-162, 2012 IEEE 11th International Symposium on Network Computing and Applications, 2012.

[7] https://wiki.apache.org/hadoop/PoweredBy last vist 12 augest 2014. (Ebay).

[8] Magorzata Steinder, Adarshpal S. Sethi, "A survey of fault localization techniques in computer networks", in Sci Comput Program, Vol. 53, pp.165-194.

[9] Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey". In ACM Computing Surveys, 2009.

[10] Kher, S. ; Arkansas State Univ., AR, USA ; Nutt, V. Dasgupta, D. ; Ali, H.more authors"A detection model for anomalies in smart grid with sensor network" Future of Instrumentation International Workshop (FIIW), 2012.

[11] K. Wang, Y. Wang, and B. Yin, "A Density-Based Anomaly Detection Method for MapReduce", ;in Proc. NCA, 2012, pp.159-162.

[12] A. Shilton, S. Rajasegarar, and M. Palaniswami, "Combined multiclass classification and anomaly detection for large-scale wireless sensor networks," in Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on, 2013, pp.491–496.

[13] J. R. Lee, S.-K. Ye, and H.-D. J. Jeong, "Detecting anomaly teletraffic using stochastic self-similarity based on Hadoop," in Network-Based Information Systems (NBiS), 2013 16th International Conference on, 2013, pp. 282–287.

[14] M. Xie, J. Hu, and B. Tian, "Histogram-based online anomaly detection in hierarchical wireless sensor networks," in Trust, Security and Pri- vacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, 2012, pp. 751–759.

[15] Gaber, M. Data Stream Processing in Sensor Networks. In Learning from Data Streams Processing Techniques in Sensor Networks; Springer: Berlin/Heidelberg, Germany, 2007; pp. 41–48.

[16] Tan, P.-N.; Steinbach, M.; Kumar, V. Introduction to Data Mining; Addison Wesley: Boston, MA, USA, 2005.

[17] Aggarwal, C.; Yu, P. Outlier Detection for High Dimensional Data. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA, 21–24 May 2001.

[18] Janakiram, D.; Adi Mallikarjuna Reddy, V.; Phani Kumar, A.V.U. Outlier Detection in Wireless Sensor Networks Using Bayesian Belief Networks, In Proceedings of the First International Conference on Communication System Software and Middleware (COMSWARE 2006), Delhi, India, 8–12 January 2006; pp. 1–6.

[19] Li, Y. Anomaly Detection in Unknown Environments Using Wireless Sensor Networks; The University of Tennessee: Knoxville, TN, USA, 2010.

[20] Jeffery, S.; Alonso, G.; Franklin, M.; Hong, W.; Widom, J. Declarative Support for Sensor Data Cleaning. In Pervasive Computing, Springer: Berlin/Heidelberg, Germany, 2006; pp. 83–100.

[21] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler. Yahoo! Sunnyvale, California USA "The Hadoop Distributed File System", IEEE, 2010.

[22]] Module 1: Tutorial Introduction Mapreduce Yahoo https://developer.yahoo.com/hadoop/tutorial/module1.html. 12 augest 2014.

[23] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and Regression Trees. Wadswort, Belmont, 1984.

[24] OneR one Rule . http://www.saedsayad.com/oner.htm. Last vist 12 augest 2014.

[25] Shinde, Amit, Anshuman Sahu, Daniel Apley, and George Runger. "Preimages for Variation Patterns from Kernel PCA and Bagging." IIE Transactions, Vol. 46, Iss. 5, 2014.

[26] C4.5algorithm. Http://en.wikipedia.org/wiki/C4.5_algorithm. Last vist 12 augest 2014.

[27] WEKA: A machine learning tool set. Software downloadable from http://www.cs.waikato.ac.nz/ml/weka/index_downloading.html. last vist 12 augest 2014.