Anomaly Detection in High Performance Computing Systems: A Survey

Syed Zamil Hasan Shoumo
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
shoumohasan622@gmail.com

Ramkrishna Saha
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
mukutsaha123@gmail.com

Sabbir Hossain
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

Annajiat Alim Rasel
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
annajiat@gmail.com

***Abstract*— As the scale and complexity of high performance computing (HPC) systems keeps rising, it is very important to ensure the reliability of the systems. The performance of HPC systems can be affected by various faults or anomalies. Moreover, HPC systems are very complex and heterogeneous in nature. Therefore, it is crucial to detect and identify the anomalies properly. Currently, in many HPC systems, the system administrators and final users need to discover the faults manually. Certainly, this method is not suitable for large-scale HPC systems. As a result, to alleviate the problem, there has been some research into how to automatically detect anomalies in HPC systems. This paper presents a survey into the existing research for the problem and makes a comparison among them.**

Keywords*— Anomaly Detection, HPC, High Performance Computing, Autoencoders, Decision Trees, Random Forests, AdaBoost, K-means clustering

I. Introduction

Supercomputers are becoming increasingly complex and large with time, consisting of millions of units [1,2]. Thus, there is a plethora of sources of faults ranging from hardware malfunctions or misconfigurations to unwanted software bugs and errors. It is becoming near to impossible for system administrators to identify the source of the errors in a timely manner and keep the system operational.

Therefore, an automated system for error detection in HPC systems is very essential which can spot any anomalies in real-time. Currently, in most HPC systems, the system administrators rely on system log messages generated by software tools. [3,4] This method requires a lot of effort and time, as each system has different requirements and ways of working. Thus the approach based on system logs is highly generalized and inefficient.

The supercomputers today however have hardware components with sensors to monitor the physical parameters [5,6]. Therefore it is possible to read the metrics and collect them into a single gathering point to be analyzed later. Thus it seemed possible to collect data to look for possible anomalies. Later, these data can be analyzed using various techniques such as Machine Learning. For example, a supervised classifier can be trained using obtained data to distinguish between a healthy and an unhealthy system. In this paper, different techniques have been explained which were used to identify the anomalies in an HPC system- mainly autoencoders, supervised and unsupervised techniques.

The rest of the paper is organized as follows: Section II showcases some of the works that have been done in the past in this domain, Section III showcases the algorithms that were used for prediction, Section IV illustrates a comparison among the models and lastly, Section V concludes the paper with final remarks and future works.

II. Related works

In [7] Oliner and Strealy presents the system and methods of log collection, and how alerts are identified in an HPC System. They propose a filtering algorithm and define operational context to include information that were found missing from most logs. They considers logs from five systems to identify failures. Work on log analysis and large-systems reliability had been hindered by a lack of data about their behavior. Schroeder [8] studied failures in a set of cluster systems at Los Alamos National Lab (LANL) using the entries in a remedy database. This database was designed to account for all node downtime in these systems, and it was populated through a combination of automated procedures and the extensive effort of a full-time LANL employee, whose job it was to account for these failures and assign them a cause as soon as they occurred.

However, as mentioned in [5,6] many supercomputers have sensors to monitor various physical parameters. Thus it is now possible to collect huge amounts of data and analyze them. The following works represent the works done on anomaly detection in HPC systems based on Machine Learning.

In [13] Tuncar et. al. proposes a method in which they collect several measurements through monitoring, then a group of statistical features modeling the state of the supercomputer is obtainted from the features. The authors then train different machine learning algorithms to classify the behaviour of the supoercomputer using the statistical features. Baseman et al. [14] proposed a technique for anomaly detection in HPC systems. Then they applied a statistical technique called classifier-adjusted density estimation (CADE) in order to help the training of a supervised Random Forest classifier. The classifier decides the class (normal, anomaly, etc) of each data point (set of physical measurements).

Dani et al. in [15] described an unsupervised technique for anomaly detection in HPC. Their work was very different since they considered only the console logs generated by computing nodes. Their purpose was to distinguish logs relative to faulty conditions from logs created by healthy nodes; the proposed approach used the K-means clustering algorithm. Their faults that can be recognized by a node itself and stored in log messages; this bounds the number and the types of detectable anomalies.

In [16] Borghesi et. al. propose a novel and automated technique for anomaly detection in HPC systems. They use a type of neural network called Autoencoder. The method they proposed had really good accuracy with real-time performance. Here autoencoders were used to recreate a given input during the test phase and check for reconstruction loss. In [17] Borghesi et. al. propose a semi-supervised anomaly detection model. In addition to [16], here the authors make a comparison between a distributed approach to where there is an autoencoder with every node and a more central approach where all the data was fed into one single autoencoder.

III. Algorithms

In this paper, we will first look into some of the machine learning algorithms that are used in detecting anomalous behavior in high performance computing systems. Supervised, unsupervised, and semi-supervised approaches have been followed in this regard. Some of the machine learning strategies which have been used in the detection of anomalous states are described as follows:

In supervised learning, we adopt a strategy where the machine can learn from labeled data. The output can be either discrete or continuous. But in all cases, the output is labeled so the machine has a better idea of the outcome of its inputs. Now for detecting anomalous behavior, different researchers have used different supervised learning models showing different levels of effectiveness. Tree-based models such as Random Forest (RF) and Decision Trees (DT) have been used for this purpose. In tree-based algorithms, the tree is formed based on the input data and the final outcome of the corresponding inputs. Each level of the tree comprises a decision node and its corresponding leaf or outcome. Data is split on the basis of certain features or decision nodes and its corresponding leaf will represent the outcome based on that decision node. Unlike DT models, an RF model will build up multiple decision trees and use certain techniques to combine or average up the results hence the term "forest".

Support Vector Machine (SVM) is another supervised machine learning algorithm that has been used to detect anomalies in HPC systems. SVM's are strong linear models which can be used for both classification and regression tasks. The main objective of an SVM model is to separate the given labeled data using a hyperplane. Even though it is a linear model, due to the presence of different kernels (kernel trick), it can also be used to detect nonlinear characteristics. Another algorithm that is commonly used in this scene is the K nearest neighbor (KNN). Similar data points are termed as "neighbors" and part of the same group. We can create as many "groups" we want by changing the value of "K" here. Nearest neighbors are created based on the distance between the data points. Neighbors tend to have the least distance between themselves. Generally, Euclidean distance is the metric that is used to calculate this minimum distance. KNN is a simple but effective tool for supervised learning.

In addition to supervised learning, there is another type of machine learning strategy known as unsupervised learning. Unlike supervised learning, unsupervised learning does not require the presence of labeled data. Algorithms of this nature learn to identify the correlations between similar data points and then on the basis of this information, they group the similar data points into clusters. Here each cluster consists of instances that are similar to each other in terms of their basic characteristics and can hypothetically belong to the same "label". Their ability to identify patterns in unlabelled data and group together similar instances is why they are often preferred over supervised learning when there is scarcity of labelled data. In this modern age, different kinds clustering algorithms have been introduced to us. K means clustering, Hierarchical clustering, Gaussian Mixture Models, Fuzzy C means clustering etc are just some of the many popular clustering algorithms used in machine learning. But for anomaly detecting, the most commonly used clustering technique is the K means clustering. The K means clustering is a centroid based clustering technique. It selects random centroids and creates clusters base don these centroids using the input data points. Then it tries to iteratively maximize the effectiveness of the created clusters by trying to minimize the sum off distances between the data points and the corresponding cluster the data point belongs to.

Besides the above mentioned methods, the usage of neural networks also present an interesting approach to tackle this issue. Multi layer perceptrons, recurrent neural network, auto encoders; all of them have been used to detect anomalies in high performance computation systems. The use of auto encoders is a common and noteworthy approach in this regard. Generally the autoencoders are used in a semi supervised environment. Here the labels are required during the pre processing phase, but they are not utlized when training or testing the model. Autoencoders have the ability to recreate the inputs that have been given to it with a certain amount of error. So the main aim of an auto encoder model is to recreate the input all the while minimizing this error (reconstruction error).

IV. COMPARATIVE ANALYSIS

The anomaly detection process is a cumbersome prcedure which has a history of being done manually. But due to the recent advancements in machine learning, the entire process can now be automated and be done with mch more efficiency. In this section we will discuss some of the approaches that have been used to detect anomalous states high performance computing systems using machine learning

In [9] the authors have used SVM, RF, KNN, Adaboost, and DT to detect anomaly. They have compared their results with two other baseline models which were first showcased in [10, 11]. They have used F measure as the evaluation metric for each model Since this is done in a supervised environment, their dataset required both “healthy” and “anomalous” instances. So they manually created synthetic anomalous instances and injected them into their data. They data was prepared by the usage of two very reliable high performance computing infrastructure. Furthermore, for a more robust model, they use a 5 fold cross validation technique and repeated it 10 times. Now from their results it was clearly evident that their Random forest model had clearly outperformed all the other above mentioned models and presented itself as an accurate framework with low overhead for detecting anomalies in such systems.

In [12] the authors have taken up an unsupervised learning approach where they have used text mining techniques combined with K-means clustering to detect anomalous behaviour from log messages generated by the nodes of their hpc system. Different log messages were utlized of different formats. Each format was then clustered into healthy and abnormal / anomalous classes. The experiment was conducted by comparing 3 different clustering techniques; K means, DBSCAN and hierarchical clustering. And among these methods K means had shown the most promise in clustering the log messages in appropriate healthy and anomalous clusters.

In [18] we see that the authors have presented an approach based on Markov Chain Models. The data set here has also been prepared from system logs. The anomalous instances have been created manually and inserted into the data set. Additionally two scoring methods have also been proposed in the paper to evaluate the anomalousness of the messages. Both the proposed model and the scoring methods have shown quite promise in detecting anomalous states in high performance computation systems.

In [17] the authors have presented a semi supervised solution to this problem. They have proposed two models based on autoencoders and showcased a comparative analysis between the proposed model and other supervised learning models. Here the environment is semi supervised because no “anomalous” instances were used to train the model. The autoencoder models were trained only on “healthy” or “normal” state behaviours. Among the two proposed models, one infrastructure using an autoenconder connected and being fed data from every node of their hpc machine; and the other infrastructure is based on one central autoencoder being fed data from all connected nodes. As the models learn only healthy behaviour, they will not be able to reproduce any input which resembles anomalous behaviour with minimal reconstruction error. Any recreated point that will contain a reconstruction error above a certain threshold will be deemed as anomalous. All models were evaluated based on F measure. Both the proposed model have attained very good scores in comparison to the other supervised learning models. But among the two proposed models, the former one has attained a better score in comparison to the latter. But it has to be added that the latter is also easier to implement.

From the above discussion, we chave seen that much work has been done in this regard and this topic is of much significance to the world of computing. Many different approaches have been adopted under different environments and the authors have showcased many promising results. The use of supervised learning approaches is a viable solution to this problem if we have the privilege of having labelled data. Now in the absence of unlabelled data, we can take the help of unsupervised learning approaches. In this case we have to pay additional heed to the optimum number of clusters that our model will propose. Additionally for a robust model, the number of healthy and anomalous instances should be atleast close so that the model can learn both types of behaviour. But as we can see, in many cases the authors had to manually insert anomalous instances into the dataset. In this scenario we can use a semi supervised approach and utlize autoencoders for anomaly detection. In this case, we have to take a look into the reconstruction error and at what threshold can we deem an instance health or anomalous.

V. Conclusion

In this paper a comparative study was presented on the research related to the anomaly detection in HPC systems. Various models such as Autoencoders, Random Forest, Decision Tree, Support Vector Machine, KNN, K-Means clustering etc were described in conjunction with the context of HPC Systems. Based on the discuss models, we can see that this problem can be looked at from different perspectives and in all cases we do see interesting and effective results. As we know, labelling the data which is normally done by system administrators, is overall a cumbersome process. So we can look into more unsupervised learning strategies for this problem. Many authors have proposed to create online and real time versions of the proposed model which will certainly be much more effective in this regard.

References

[1] H. Fu, J. Liao, J. Yang, and et Al., “The sunway taihulight supercomputer: system and applications,” Science China Information Sciences, vol. 59, no. 7, 2016.

[2] J. J. Dongarra, H. W. Meuer, and E. Strohmaier, “29th top500 Supercomputer Sites,” Top500.org, Tech. Rep., Nov. 1994.

[3] W. Barth, Nagios: System and network monitoring. No Starch Press, 2008.

[4] W. Xu, L. Huang, and M. I. Jordan, “Experience mining google’s production console logs.” in SLAML, 2010

[5] F. Beneventi, A. Bartolini, C. Cavazzoni, and L. Benini, “Continuous learning of HPC infrastructure models using big data analytics and in memory processing tools,” in Proceedings of the Conference on Design, Automation & Test in Europe. European Design and Automation Association, 2017, pp. 1038–1043.

[6] A. Bartolini, A. Borghesi, A. Libri, F. Beneventi, D. Gregori, S. Tinti, C. Gianfreda, and P. Altoe, “The D.A.V.I.D.E. big-data-powered fine-grain power and performance monitoring support,” in Proceedings of the 15th ACM International Conference on Computing Frontiers, CF 2018, Ischia, Italy, May 08-10, 2018, 2018,

pp. 303–308. [Online]. Available: http://doi.acm.org/10.1145/3203217.3205863

[7] A. Oliner and J. Stearley, “What supercomputers say: A study of five system logs,” in Dependable Systems and Networks, 2007. DSN’07. 37th Annual IEEE/IFIP International Conference on. IEEE, 2007, pp. 575– 584.

[8] B. Schroeder and G. Gibson. A large-scale study of failures in high-performance-computing systems. In Proceedings of the Intl. Conf. on Dependable Systems and Networks (DSN), Philadelphia, PA, June 2006.

[9] Tuncer, O., Ates, E., Zhang, Y., Turk, A., Brandt, J., Leung, V. J., ... & Coskun, A. K. (2017, June). Diagnosing performance variations in HPC applications using machine learning. In *International Supercomputing Conference* (pp. 355-373). Springer, Cham.

[10] Bodik, P., Goldszmidt, M., Fox, A., Woodard, D.B., Andersen, H.: Fingerprinting the datacenter: automated classification of performance crises. In: Proceedings of the 5th European Conference on Computer Systems, pp. 111–124 (2010)

[11] Lan, Z., Zheng, Z., Li, Y.: Toward automated anomaly identification in large-scale systems. IEEE Trans. Parallel Distrib. Syst. 21(2), 174–187 (2010)

[12] Dani, M. C., Doreau, H., & Alt, S. (2017, June). K-means application for anomaly detection and log classification in hpc. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 201-210). Springer, Cham.

[13] O. Tuncer, E. Ates, Y. Zhang, A. Turk, J. Brandt, V. J. Leung, M. Egele, and A. K. Coskun, “Diagnosing performance variations in hpc applications using machine learning,” in International Supercomputing Conference. Springer, 2017, pp. 355–373.

[14] E. Baseman, S. Blanchard, N. DeBardeleben, A. Bonnie, and A. Morrow, “Interpretable anomaly detection for monitoring of high performance computing systems,” in Outlier Definition, Detection, and Description on Demand Workshop at ACM SIGKDD. San Francisco (Aug 2016), 2016.

[15] M. C. Dani, H. Doreau, and S. Alt, “K-means application for anomaly detection and log classification in hpc,” in International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, 2017, pp. 201–210.

[16] A. Borghesi, A. Libri, L. Benini and A. Bartolini, "Online Anomaly Detection in HPC Systems," 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), 2019, pp. 229-233, doi: 10.1109/AICAS.2019.8771527.

[17] Borghesi, A., Bartolini, A., Lombardi, M., Milano, M., & Benini, L. (2019). A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems. Engineering Applications of Artificial Intelligence, 85, 634-644.

[18] Haque, A., DeLucia, A., & Baseman, E. (2017, November). Markov chain modeling for anomaly detection in high performance computing systemlogs. In *Proceedings of the Fourth International Workshop on HPC User Support Tools* (pp. 1-8).