Machine Learning Methodologies to Support HPC Systems Operations: Anomaly Detection

Published Version:
Molan M., Borghesi A., Benini L., Bartolini A. (2023). Machine Learning Methodologies to Support HPC Systems Operations: Anomaly Detection. Cham : Springer [10.1007/978-3-031-31209-0_24].

Availability:
This version is available at: https://hdl.handle.net/11585/952304 since: 2024-01-06

Published:
DOI: http://doi.org/10.1007/978-3-031-31209-0_24

Machine Learning methodologies to support HPC systems operations: Anomaly detection

Martin Molan[1[0000−0002−6805−2232]], Andrea Borghesi[1[0000−0002−2298−2944]], Luca Benini[1,2[0000−0001−8068−3806]], and Andrea Bartolini[1[0000−0002−1148−2450]]

[1] DISI and DEI Department, University of Bologna, Bologna, Italy
{martin.molan2, andrea.borghesi3, luca.benini, a.bartolini@unibo.it}@unibo.it
[2] Institut für Integrierte Systeme, ETH, Zürich, Switzerland

Abstract. The increasing complexity of modern and future pre-exascale high-performance computing (HPC) systems necessitate the introduction of machine learning methodologies that support systems administrators. The key element of these monitoring and support systems is anomaly detection. This presentation discusses my current work - as part of my Ph.D. research - in developing anomaly detection systems for the HPC systems. Specifically, I discuss my ongoing work in improving upon the previous SoA anomaly detection system. The proposed approach is evaluated on the Maroni 100 supercomputer located in CIENCA. Based on a large-scale evaluation (on all 980 nodes), we see that the proposed approach outperforms the previous SoA.

Keywords: Anomaly detection · High-performance computing · Machine learning

1 Introduction

In the move towards exascale, modern High-performance computing (HPC) systems are becoming increasingly larger and more complex [9]. A typical modern HPC system consists of hundreds of compute nodes with future pre-exascale systems expending this number into thousands [6]. This increased complexity necessitates the introduction of monitoring systems supported by AI/ML methodologies that support system administrators in managing the HPC system.

The most critical application of AI tools in support of system administrators is the introduction of anomaly detection systems. Anmol detection systems are crucial as they allow system administrators to react to the downtime (or unavailability event) faster and thus reduce the time between the anomaly (evet) and their response. This faster response time severely reduces the time the compute nodes are unavailable and increases the overall availability of the HPC system [8]. This anomaly detection signal is then included in the dashboard presented to the system administrators as a part of a digital twin of the datacentre.

The foundation for the creation of AI-augmented monitoring systems is the holistic monitoring infrastructure that combines out-of-band power monitoring,

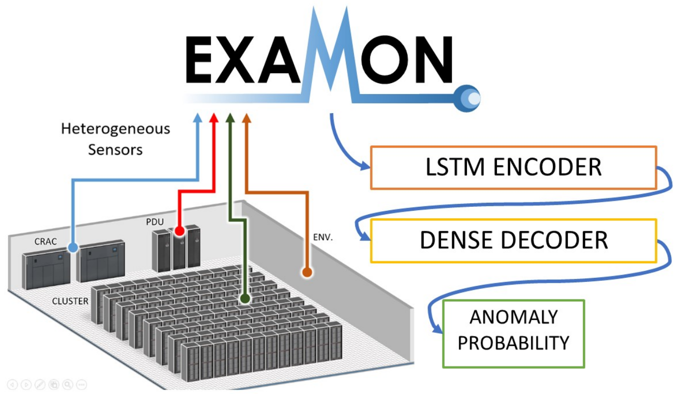

Fig. 1. Data collection schema of M100 HPC system. Data from various sensors is collected by ExaMon and then passed to the encoder/decoder network.

system monitoring, and historical availability data [3, 7]. My Ph.D. thesis focuses on data collected by the ExaMon monitoring system developed by the University of Bologna [5]. I study the data collected from the Marconi 100, which is a Tier-0 HPC system located in CINECA Italy [2] (ranked 9th in Jun. 2020 Top500 list[1]).

The state-of-the-art (SoA) approach to anomaly detection is to deploy a semi-supervised approach [8, 4]. This stems from the fact that the anomalies are rare events, and it would be impossible to collect a significant enough dataset for classical supervised classification methods [4]. The current SoA approach to anomaly detection proposed by Borghesi et al. [4] is a semi-supervised anomaly detection approach that takes minimal advantage of the temporal dependencies in the anomaly signal. Our current work - discussed in this paper - is thus how to extend this approach to include temporal dynamics. We propose to do this by deploying an encoder network consisting of Long Short-Term Memory (LSTM) cells.

1.1 Contributions

The key research question that this work discussed is how to extend our previous work [4] (which is the current SoA) with temporal dependency data. Specifically, we propose an LSTM-based approach that we evaluate in a very large scale experiment: we train two different models for each of the 980 nodes of Marconi 100; an extensive scale experiment thus supports the results (and the claim of the new SoA).

1.2 Anomalies and dataset

The proposed methodology is evaluated on the complete first ten-month operation history of all 980 nodes of the Marconi 100 HPC system. The collected dataset contains out-of-bound hardware monitoring data and system data. The complete list of used features is presented in table 1. The anomalies are determined as events where the nodes are unavailable to accept (or continue to execute) compute jobs. The dataset is prepared in collaboration between the University of Bologna and CINECA [2].

Source	Features
Hardware monitoring	ambient temp., dimm[0-15] temp., fan[0-7] speed, fan disk power, GPU[0-3] core temp. , GPU[0-3] mem temp. , gv100card[0-3], core[0-3] temp. , p[0-1] io power, p[0-1] mem power, p[0-1] power, p[0-1] vdd temp. , part max used, ps[0-1] input power, ps[0-1] input voltage, ps[0-1] output current, ps[0-1] output voltage, total power
System monitoring	CPU system, bytes out, CPU idle, proc. run, mem. total, pkts. out, bytes in, boot time, CPU steal, mem. cached, stamp, CPU speed, mem. free, CPU num., swap total, CPU user, proc. total, pkts. in, mem. buffers, CPU idle, CPU nice, mem. shared, PCIe, CPU wio, swap free

Table 1. An anomaly detection model is created only on hardware and application monitoring features. More granular information regarding individual jobs is not collected to ensure the privacy of the HPC system users.

2 The LSTM Autoencoder Network

To improve the current SoA, we propose an LSTM encoder-dense decoder model. The key innovation, compared to the current SoA [4] is that we are encoding a sequence leading up to (and including) the last timestamp. This improves upon the dense autoencoder as it better captures the temporal dependencies inherent in the dataset. The critical insight in this innovation is that while the data describing supercomputing nodes is composed of multi-variate time series, the state-of-the-art does not explicitly consider the temporal dimension – the dense

Anomaly detection method:	AUC:
Exponential smoothing	0.4276
Dense autoencoder (current SoA)	0.7470
LSTM autoencoder (proposed approach)	0.7582

Table 2. AUC performance of different AD models. Proposed approach outperforms the current SoA for AD.

autoencoder has no notion of time nor of *sequence* of data points. To overcome this limitation, our approach works by encoding the sequence of values *leading up to the anomaly.* The encoder network is composed of LSTM layers, which have often been proved to be well suited to the context where the temporal dimension is relevant. An LSTM layer consists of recurrent cells with input from the previous timestamp and the long-term memory. The latent layer (vector) output is passed into a dense decoder trained by reproducing the final vector in an input sequence. The decoder network is thus composed of fully connected dense layers.

The proposed LSTM encoder/dense decoder model takes as input a sequence of vectors of features $\boldsymbol{x}$ leading up to the current time t_0 and then tries to reconstruct only the last vector in the sequence $\hat{\boldsymbol{x}}_{t_0}$:

$$M : \boldsymbol{x}_{t_0-W}, \cdots, \boldsymbol{x}_{t_0} \to \hat{\boldsymbol{x}}_{t_0}. \tag{1}$$

The length of the input sequence W leading up to the current time t_0 is a tunable parameter. For experimental results, the length of the input sequence is set to 10. The proposed model M outputs the probability (estimated from the reconstruction error between $\boldsymbol{x}_{t_0}$ and $\hat{\boldsymbol{x}}_{t_0}$) that the node is in an anomalous state at time t_0.

3 Experimental results

To remove the potential for bias by setting up the decision threshold, we compare the proposed approach against the current SoA ([4]) by evaluating the area under the receiver-operator characteristic curve (AUC ROC). An exponential smoothing baseline is implemented as a sanity check - if the anomalies were simple jumps in value, the exponential smoothing would be able to catch them. As it is clear from the results, this is not the case - exponential smoothing performs even worse than the trivial classifier (AUC smaller than 0.5). As seen in the table 2), the proposed model (combined results from 980 nodes) outperforms the current SoA. This confirms our hypothesis about the usefulness of considering temporal dependencies when modeling anomalies.

4 Conclusions

This work presents the genesis of developing anomaly detection systems on a Tier-0 supercomputer. It reevaluates our previous work [4] against a new proposed approach. This approach, based on LSTM cells, outperforms the old SoA approach.

Both deep learning-based approaches are evaluated on a very large-scale experiment consisting of the whole dataset collected from Marconi 100. Results from this large-scale experiment strongly support our claim that the proposed SoA approach sets a new SoA benchmark for anomaly detection in HPC systems.

5 Acknowledgments

All work discussed in the Ph.D. symposium is done in collaboration with and under the supervision of prof. Andrea Bartolini, prof. Luca Benini and prof. Andrea Borghesi.

This research was partly supported by the EuroHPC EU PILOT project (g.a. 101034126), the EuroHPC EU Regale project (g.a. 956560), EU H2020-ICT-11-2018-2019 IoTwins project (g.a. 857191), and EU Pilot for exascale EuroHPC EUPEX (g. a. 101033975). We also thank CINECA for the collaboration and access to their machines and Francesco Beneventi for maintaining Examon.

References

1. Top500list (2020), https://www.top500.org/lists/top500/2020/06/
2. Beske, N.: Ug3.2: Marconi100 userguide, https://wiki.u-gov.it/confluence/pages/viewpage.action?pageId=336727645, accessed: 2020-08-17
3. Borghesi, A., Bartolini, A., et al.: Anomaly detection using autoencoders in hpc systems. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)
4. Borghesi, A., Molan, M., Milano, M., Bartolini, A.: Anomaly detection and anticipation in high performance computing systems. IEEE Transactions on Parallel and Distributed Systems **33**(4), 739–750 (2022). https://doi.org/10.1109/TPDS.2021.3082802
5. Iuhasz, G., Petcu, D.: Monitoring of exascale data processing. In: 2019 IEEE International Conference on Advanced Scientific Computing (ICASC). pp. 1–5 (2019). https://doi.org/10.1109/ICASC48083.2019.8946279
6. Milojicic, D., Faraboschi, P., Dube, N., Roweth, D.: Future of hpc: Diversifying heterogeneity. In: 2021 Design, Automation Test in Europe Conference Exhibition (DATE). pp. 276–281 (2021). https://doi.org/10.23919/DATE51398.2021.9474063
7. Netti, A., Kiziltan, Z., Babaoglu, O., Sîrbu, A., Bartolini, A., Borghesi, A.: A machine learning approach to online fault classification in hpc systems. Future Generation Computer Systems (2019)
8. Netti, A., Shin, W., Ott, M., Wilde, T., Bates, N.: A conceptual framework for hpc operational data analytics. In: 2021 IEEE International Conference on Cluster Computing (CLUSTER). pp. 596–603 (2021). https://doi.org/10.1109/Cluster48925.2021.00086
9. Shin, W., Oles, V., Karimi, A.M., Ellis, J.A., Wang, F.: Revealing power, energy and thermal dynamics of a 200pf pre-exascale supercomputer. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. SC '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3458817.3476188, https://doi.org/10.1145/3458817.3476188