Graph Neural Networks for Anomaly Anticipation in HPC Systems

Martin Molan
DEI Department, University of Bologna
Italy
martin.molan2@unibo.it

Junaid Ahmed Khan
DEI Department, University of Bologna
Italy
junaid.ahmedkhan@studio.unibo.it

Andrea Borghesi
DISI Department, University of Bologna
Italy
andrea.borghesi3@unibo.it

Andrea Bartolini
DEI Department, University of Bologna
Italy
a.bartolini@unibo.it

ABSTRACT

In this paper, we explore the use of Graph Neural Networks (GNNs) for anomaly anticipation in high performance computing (HPC) systems. We propose a GNN-based approach that leverages the structure of the HPC system (particularly, the physical proximity of the compute nodes) to facilitate anomaly anticipation. We frame the task of forecasting the availability of the compute nodes as a supervised prediction problem; the GNN predicts the probability that a compute node will fail within a fixed-length future window.

We empirically demonstrate the viability of the GNN-based approach by conducting experiments on the production Tier-0 supercomputer hosted at CINECA datacenter facilities, the largest Italian provider of HPC. The results are extremely promising, showing both anomaly detection capabilities on par with other techniques from the literature (with a special focus on those tested on real, production data) and, more significantly, strong results in terms of anomaly prediction.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

High performance computing (HPC), Anomaly anticipation, Graph neural network (GNN)

ACM Reference Format:
Martin Molan, Junaid Ahmed Khan, Andrea Borghesi, and Andrea Bartolini. 2023. Graph Neural Networks for Anomaly Anticipation in HPC Systems. In *Companion of the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE '23 Companion), April 15–19, 2023, Coimbra, Portugal.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3578245.3585335

ICPE '23 Companion, April 15–19, 2023, Coimbra, Portugal

ACM ISBN 979-8-4007-0072-9/23/04…$15.00
https://doi.org/10.1145/3578245.3585335

1 INTRODUCTION

High Performance Computing (HPC) systems and the datacenters that host them are gaining importance in today's society and industry. The size and complexity of modern HPC systems necessitate the introduction of automated and more efficient methodologies to support the work of system administrators and facility managers[22]; this is compounded by the current trends in worldwide HPC installation, which forecasts growing demand for computational resources[3] and announcements of funding programs [23]. With almost an order of magnitude higher computational capabilities than today's most powerful supercomputers, the Exascale machine will bring supercomputer-assisted calculations into our daily life, revolutionising many aspects of our society (manufacturing, transportation, health and decision making, among others).

Overall, it is a daunting task for system administrators, and users to optimise supercomputer/jobs performance and power consumption, identify anomalous behaviours, faulty situations, and guarantee systems operating in optimal conditions. The scale of the problem motivates the development of an automated procedure for data-driven anomaly detection in current supercomputers. A very promising direction for such automated approaches is constituted by Artificial Intelligence (AI) techniques, which, when fed with large amounts of data and provided with sufficient computing resources, have demonstrated to be extremely well suited for this task[7]. So far, AI methodologies in the HPC sector have mostly focused on performance optimization, anomaly detection, resource utilization, scheduling and energy savings [9]. Anomalies are, in the context of HPC monitoring, any events that deviate from the system's normal operation and negatively impact the system's availability (ability to execute compute jobs).

Anomaly prediction in HPC is more valuable than anomaly detection because it allows for proactive system management. Instead of waiting for anomalies to occur and then detecting them, anomaly prediction uses Machine Learning (ML) algorithms to forecast future behaviour and identify potential anomalies before they happen. This allows system administrators to take proactive measures to prevent or mitigate the impact of anomalies, resulting in improved system reliability and performance [20]. At the same time, for several reasons, anomaly prediction is also considered more difficult than anomaly detection. Firstly, it requires high-quality training data to build accurate predictive models. The training data must represent normal behaviour, and any changes in the

system's behaviour must be reflected in the training data. This can challenge dynamic and complex systems like high performance computing, where behaviour can change rapidly and unpredictably [8]. Secondly, anomaly prediction requires more sophisticated ML algorithms to handle time-series data and predict future behaviour. This is a more challenging task than simply identifying deviations from normal behaviour, as the algorithms must consider both the past and future behaviour of the system.

In this work, we tackle the problem of anomaly prediction using an approach that, while not novel, has never been applied to the HPC domain. In particular we exploit Graph Neural Networks (GNNs). GNNs offer a promising solution to address the limitations of current methods for anomaly detection in high performance computing [26]. GNNs are a deep learning algorithm well-suited for processing graph-structured data, commonly found in high performance computing systems. Specifically, we are leveraging the spatial proximity of compute nodes in an HPC compute room. We then implement a GNN-based approach to forecast anomalies and we then conduct an experimental evaluation on a dataset collected from a real production supercomputer, namely we consider two racks from the M100 supercomputer, the Tier-0 machine hosted at the Italian supercomputing facilities of CINECA [1]. We thus demonstrate the effectiveness of our approach on real, production anomalies. The results are very promising, as we obtain an Area-Under-the-Curve (AUC) of 0.86 in the pure detection task (in par with other methods from the literature), while at the same time we can anticipate the anomalies by several hours; the prediction accuracy maintains very good results – around 0.84 AUC – with 8-hours look-ahead windows, slightly decreasing for longer time windows (e.g., 0.73 AUC for 24 hours). To the best of our knowledge, this is the first approach demonstrating this capability on real data from a production supercomputer.

In the rest of the paper, we start by providing an overview of other automated anomaly detection and prediction methods for HPC from the literature (Section 2); then, we briefly describe the supercomputer used to conduct the experimental evaluation (Section 3). In Section 4, the GNN-based approach is introduced, while (Section 5) examines the results of the empirical evaluation. Finally, (Section 6) concludes the paper.

2 RELATED WORKS

Anomaly detection is one of the earliest and most widely adopted applications of operational data analytics in high performance computing. The primary purpose of anomaly detection is to identify unusual or unexpected patterns or behaviours in the data generated by high performance computing systems [15]. Anomalies can include identifying deviations from normal resource utilization patterns, unusual behaviour in system logs, or unusual network traffic patterns. The goal is to detect anomalies as early as possible so that they can be investigated and addressed quickly and thus minimize the downtime of the HPC system.

Anomaly detection algorithms use statistical methods, ML techniques, or a combination of both to analyze data and identify deviations from normal behaviour. These deviations can then be flagged as potential anomalies, and the system administrator can be alerted for further investigation [19]. Anomaly detection in high performance computing is considered a mature topic with good results due to the significant advances made in the field over the years [18]. The extensive use of high performance computing systems in various industries has driven the development of sophisticated algorithms and tools for detecting anomalies in large amounts of data.

Different lines of work have explored the anomaly detection problem in HPC from multiple angles. Borghesi et al.[5, 6] presented a seminal work where they applied a particular ML model (autoencoder network) trained in a semi-supervised fashion to perform anomaly detection of failures in a supercomputer. The key limitation of that work was that it was tested only in anomalies injected in the HPC system by the researchers themselves, thus incurring in the risk of unwanted bias. Later approaches mixed semi-supervised methods with supervised ML models, taking advantage of more monitoring infrastructure which allowed the creation of annotated data set. These approaches have been tested on real anomalies found on production HPC systems [7, 8].

More recently, Molan et al. [16] introduced a fully unsupervised approach for failure detection on HPC compute nodes, again tested on real anomalies from a Tier-0 supercomputer. The fact that labels are not needed in this case makes this approach more versatile and more easily deployable in production HPC systems, as it can start detecting anomalies even without a preliminary training phase (or at least requiring only a very short one). The key limitation of current methods for anomaly detection is that they have shown a relatively poor ability to forecast the insurgency of anomalies, a feature extremely desired by system administrators to integrate this automated approach into their workflow.

One thing shared by all anomaly detection methods for HPC is that they work at the *compute-node* level (each HPC has its own detection model), without taking into account the spatial structure of the HPC machines – e.g., the fact that the compute nodes are organized in racks with a well-defined structure. The neighboring compute nodes have higher probabilities to have correlated behavior w.r.t. distant nodes[12, 14]. It is possible to represent this spatial structure by representing the compute nodes as nodes in a graph. After this encoding, it is straightforward to apply graph-specific ML models to create automated anomaly detection and prediction models, capable of exploiting this form of locality.

In recent years GNNs have revealed to be extremely well-suited to deal with graph-like structures[25], especially to exploit node proximity[27]. One of the advantages of using GNNs for anomaly detection and prediction in high performance computing is that they allow us to train the models in a supervised manner. Unlike simple per-node supervised anomaly detection methods that overfit on the majority class. GNNs can use labelled training data more efficiently even when the classes are extremely unbalanced, as is the case in anomaly detection in HPC systems [17].

Graph modelling approaches have been successfully applied to understanding the operation of data centres. Specifically, the technique of landscape colouring has been applied to identify anomalous nodes (components) in the datacenter [11]. Alternatively, GNNs have been used to improve workload management on datacenters [10], identifying network issues [13], rack temperature

[1] CINECA is a public university consortium and the main supercomputing centre in Italy[24].

prediction[21], etc. To the best of our knowledge, however, this work is the first to explore the possibility of deploying graph neural networks (GNNs) to *predict* high-performance computing systems' failures. The use of GNNs in this context is a novel approach that has the potential to bring significant improvements to the field of operational data analytics.

3 BACKGROUND

The monitoring and data collection infrastructure in high performance computing (HPC) is crucial to ensuring the efficient and effective operation of HPC systems. The monitoring infrastructure is designed to collect data from various sources, including hardware sensors, software logs, and performance metrics, and store this data in a centralized repository. The specific monitoring infrastructure employed by CINECA on Marconi 100 (the target system of this work) is called Examon and has been developed by the University of Bologna and CINECA [4]. The architecture of the monitoring system Examon has been developed to exploit the MQTT protocol as a backbone to be lightweight while capable of ensuring differing quality-of-service levels. MQTT is a lightweight messaging protocol for the Internet of Things [1].

Examon collects multiple data sources, ranging from hardware sensors (for instance, CPU load of all the cores in the supercompute nodes, CPU clock, instructions per second, memory accesses, power consumption, fan speed, room and IT components temperature, etc.) to information pertaining the workload (e.g., jobs submitted and their characteristics) and the availability of the compute nodes, such as warning messages and alarms generated by the diagnostic software tools implemented by system administrators to carry out their tasks. In previous works we demonstrate that the various metrics data can be merged into holistic dashboards of the entire supercomputer [5]. Additionally, in the past we have also shown how to combine raw metrics with more elaborate information, such as the output produced by ML models, to obtain more informative views of the system; for instance, creates a digital twin of the M100 supercomputer as a 3D dashboard showing the status of all compute nodes (using different colours to highlight their state) [8].

The data experimented on in the dataset is part of a dataset collected from the Marconi 100 system that is in the process of being released as an open-source dataset. It contains 31 months of data and contains data from rack $n206$ and rack $n207$.

4 THE PROPOSED APPROACH: GNN FOR ANOMALY DETECTION

Graph Neural Networks (GNNs) are a type of deep learning algorithm that operates on graph-structured data. Unlike traditional deep learning algorithms designed to process grid-like structured data, such as images or text, GNNs handle data represented as a graph. A graph is a data structure that consists of a set of nodes and edges, where the nodes represent entities, and the edges represent relationships between the entities [26].

Convolutional Graph Neural Networks (GCNs) are a specific type of GNN designed to process graph-structured data. The main idea behind GCNs is to perform graph convolution operations on the graph data. These operations are designed to aggregate information from the local neighbourhood of each node in the graph and use this information to update the node representations [17]. The process of updating the node representations is performed iteratively, and after several iterations, the GCN algorithm can capture the global structure of the graph. This global structure can then be used for various tasks, such as node classification, link prediction, and graph classification.

In this paper we decided to adopt GCN, as preliminary experiments revealed that this topology would yield better performance. In this initial work we did not perform a systematic search in the hyperparameters space, but we rather relied on background knowledge and manual fine-tuning; we will exhaustively search for better architectures in future works. After the initial empirical exploration, the GNN architecture with the best performance resulted to be the following:

- Graph convolution layer of shape (416,300)
- Graph convolution layer of shape (300,100)
- Graph convolution layer of shape (100,16)
- Dense layer of shape (16,16)
- Dense layer of shape (16,1)

The graph neural network is trained on the node classification task. The label is constructed as follows: the node is labelled 0 if no anomalies occur in the future window T. Otherwise, the node is labelled as anomalous.

The graph representation of compute nodes takes advantage of the nodes' physical layout in racks. In this work, we have created an individual model for each compute rack. As explored in other approaches [16], the modular approach allows scalability with the future large exascale systems. Each rack is represented as a line graph (as depicted in figure 1), where nodes are vertices in a (line) graph. Each node is connected to the node above and below it in a compute rack. From a ML perspective, anomaly prediction is set up as a node classification problem - we are trying to predict the probability of the anomaly (a label) for each of the compute nodes in a rack (vertices in a line graph).

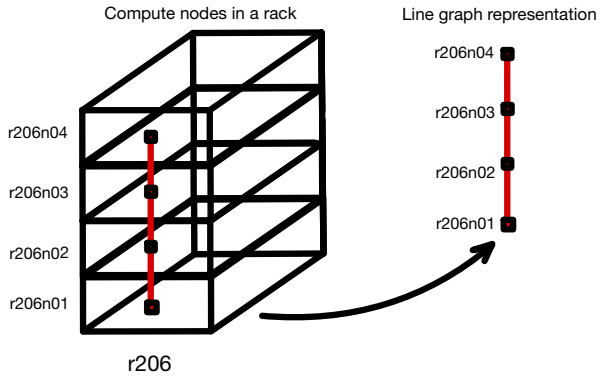

Figure 1: Graph representation of the physical proximity of the compute nodes in a rack. Each node is directly connected to the compute node above and below it.

5 EXPERIMENTAL EVALUATION

In this section, we describe the experimental evaluation conducted to evaluate the capability of the proposed approach, namely the

GNN model. The experimental evaluation is complicated by two factors: 1) there are not many works that can be directly compared to our approach, as the majority of the research works presented in the literature is tested only on synthetic anomalies and/or with data coming from non-production HPC systems; 2) there are no works explicitly targeting anomaly prediction ([8] is the one coming the closest as it reveals some anticipation capability as a by-product of the main goal, that is anomaly detection).

For these reasons we decided to compare the GNN approach to one of the most most robust and top-performing method for anomaly detection in *real anomalies* from production supercomputers, namely we chose RUAD as baseline. This simplifies the comparison for an additional reason: each model is configured to output the anomaly probability, so the area under the curve (AUC) metric can be used for a fair comparison. This baseline will serve to demonstrate the capability of the GNN approach to perform *anomaly detection*. In order to test its usefulness for performing *anomaly anticipation* as well we compare the GNN method with another, simpler approach (albeit typically used as an initial baseline in practice), namely a last-value predictor which simply produces the last seen value as prediction for the next value (i.e., the state of the HPC node in the following time-step).

5.1 Experimental setting

The experimental evaluation was performed on Marconi 100 (M100) HPC system located in CINECA. The training of graph neural networks was performed on a single compute node of M100 consisting of 32 IBM POWER9 cores, 256 GB of RAM and 4 NVIDIA V100 GPUs with 16 GB of RAM.

The dataset consists of 31 months of observation of two compute racks of the Marconi 100 system. This data has not been used in previous research works. First, 80 per cent of the data has been used as a train set, and the last (historically) 20 per cent as the test set. The models are implemented in the PyTorch framework. Hyperparameters for RUAD are adapted from the original paper [16], and hyperparameters for GNN are determined based on preliminary exploration and experimentation. The exact hyperparameters for the RUAD and GNN are described in sections 5.2 and 4, respectively.

5.2 RUAD

The current SOTA for anomaly detection in production supercomputers is Recurrent Unsupervised Anomaly Detection (RUAD). The structure of the model is adopted from [16] and consists of the following:

- LSTM layer of shape (352, 16)
- LSTM layer of shape (16,8)
- Linear layer of shape (8, 16)
- Linear layer of shape (16, 352)

The main innovation of RUAD is the use of time-dependent encoder layers and dense (time-agnostic) decoder layers. The training task of RUAD is to predict the last element in the input sequence:

$$RUAD : \vec{x}_{t_0-W+1}, \cdots, \vec{x}_{t_0} \rightarrow \vec{x}_{t_0}. \quad (1)$$

The reconstruction error is then calculated on the last value in the input sequence; from the normalized reconstruction error, the anomaly probability is estimated. The time window for RUAD is adopted directly from the original paper [16], where the time window 20 achieved the best results.

In line with other works from anomaly detection, the primary evaluation metric for the proposed approach is the receiver-operator characteristic (ROC) curve [16]. The ROC curve enables the comparison of the classifiers without setting a specific decision threshold. This enables the comparison of approaches without selecting a specific trade-off between specificity and accuracy [2].

5.3 Anomaly anticipation

We start by conducting the same experiments described in the original paper presenting the RUAD approach, that is [16]. The authors originally claimed a AUC of 0.77, but in our experiments conducted for this paper we obtained an AUC of 0.86. This non-marginal increase in performance is justified by the fact that we are currently operating with a larger training set than the original one. In any case, this does not invalidate our comparison as we use the same more recent dataset to test the GNN approach as well.

The GNN approach, on the other hand, achieves stronger results in anomaly anticipation than RUAD does in anomaly detection. This is a remarkable result, as anomaly prediction is a more complex (albeit more useful) task. In anticipation window of up to 6 hours ahead, GNN outperforms RUAD in anomaly detection. It can be seen in the table 1 that the performance of the network declines as we increase the future window. Still, GNN achieves results comparable to previous approaches [8, 16] up to 24 hours ahead. This very promising result shows that the GNN truly unlocks the possibility of *long term* anomaly prediction in HPC systems.

In terms of prediction strength, the GNN anomaly detection approach is compared against the last value predictor, as currently there are no methods in the HPC literature that directly tackle the problem of anomaly anticipation (to the best of our knowledge). As evident in the table 1, the last value predictor achieves comparable results to GNN for short prediction intervals. This can be explained by the fact that the anomalies in the HPC system typically last for at least a few hours - if the nodes are unavailable at $t0$, it is thus very likely it will also be unavailable at $t + 6$ (or 1.5 hours in advance).

For longer time intervals, however, the difference between the last value predictor and the neural network becomes more significant. In other words, anomaly prediction becomes more difficult, and the last value predictor can simply explain anomalies. It is in longer periods where the strength of the approach shows - for predictions of up to 8 and 16 hours ahead, the difference of AUC of the last value and GNN is around 0.1. The increasing performance gap between GNNs and the last value predictor, as the future observation window increases, is also apparent from figure 2.

6 CONCLUSIONS AND FUTURE WORK

This paper introduces the first anomaly anticipation method for high performance computing systems based on graph neural networks and demonstrates its superiority over current anomaly detection baselines. Our method outperforms traditional methods by providing usable results in a long future observation window. This is a significant breakthrough in high-performance computing systems, as it shows that graph neural networks can effectively be used for long-term anomaly anticipation.

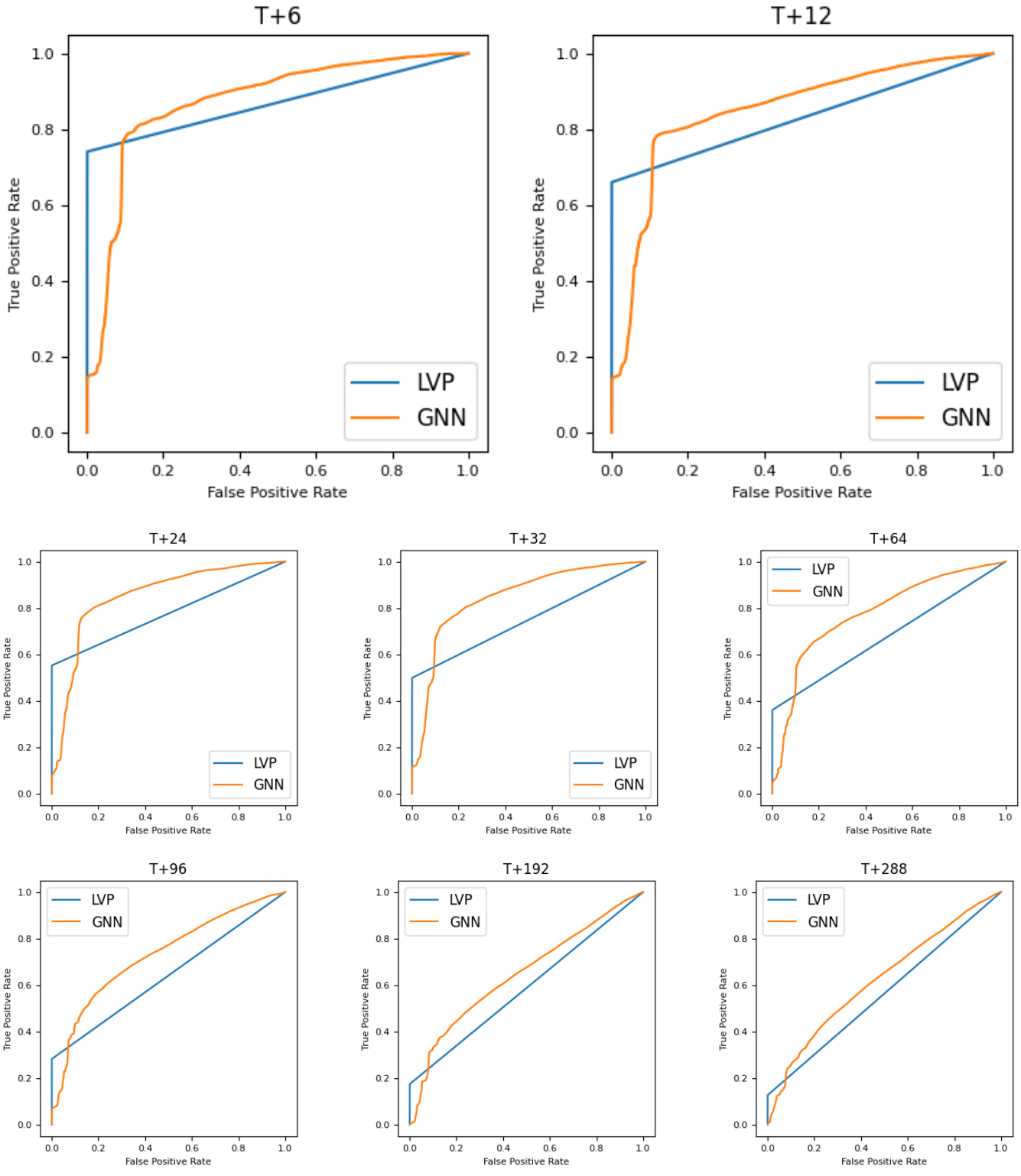

Figure 2: ROC curve of anomaly predictors at different future intervals. The GNN predictor is compared for each interval against the last value predictor (LVP).

(T+N)	Time-ahead	AUC of GNN	AUC of LVP
T+6	1.5hr ahead	0.8826	0.8699
T+12	3hr ahead	0.8676	0.8297
T+24	6hr ahead	0.8661	0.7759
T+32	8hr ahead	0.8454	0.7491
T+64	16hr ahead	0.7801	0.6797
T+96	24hr ahead	0.7317	0.6408
T+192	48hr ahead	0.6455	0.5868
T+288	72hr ahead	0.6219	0.5630

Table 1: Comparision between LVP and GNN. The difference between GNN and the last value predictor increases for larger future windows.

The experimental results demonstrate the potential of graph neural networks to unlock the possibility of long-term anomaly anticipation in high performance computing systems. Using graph neural networks allows us to consider both the spatial and temporal relationships between the components in the system, leading to a more accurate and effective prediction of anomalies.

Future work in this field should focus on increasing the size of the graph neural network to include the entire compute room and exploring different types of connections in the graph. This will further improve the accuracy and effectiveness of the anomaly anticipation method, leading to a more robust and efficient high performance computing system. This increased complexity and size of the neural networks will be supported by integration with the Graph-Massivizer toolchain. In particular, the GNN applications on large graphs will integrate with massive graph ingestion and processing pipelines developed by the project. Overall, this research represents a significant step forward in developing effective anomaly anticipation methods for high performance computing systems and highlights the potential of graph neural networks in this area.

ACKNOWLEDGMENT

This research was partly supported by the EuroHPC EU Regale project (g.a. 956560), the HE EU Graph-Massivizer project (g.a. 101093202), the HE EU DECICE project (g.a. 101092582), and the EuroHPC EU Pilot for exascale EUPEX (g.a. 101033975). We also thank CINECA for the collaboration and access to their machines.

REFERENCES

[1] [n. d.]. The standard for IOT messaging. https://mqtt.org/
[2] 2021. Receiver operating characteristic. https://en.wikipedia.org/wiki/Receiver_operating_characteristic
[3] Aleksandr Sergeevich Antonov, Il'ya Viktorovich Afanasyev, and Vl V Voevodin. 2021. High-performance computing platforms: current status and development trends. *Numerical Methods and Programming (Vychislitel'nye Metody i Programmirovanie)* 22 (2021), 135–177.
[4] Andrea Bartolini, Francesco Beneventi, Andrea Borghesi, Daniele Cesarini, Antonio Libri, Luca Benini, and Carlo Cavazzoni. 2019. Paving the Way Toward Energy-Aware and Automated Datacentre. In *Proceedings of the 48th International Conference on Parallel Processing: Workshops* (Kyoto, Japan) *(ICPP 2019)*. Association for Computing Machinery, New York, NY, USA, Article 8, 8 pages. https://doi.org/10.1145/3339186.3339215
[5] A. Borghesi, A. Bartolini, and et al. 2019. Anomaly detection using autoencoders in HPC systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 24–32.
[6] Andrea Borghesi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. 2019. A semisupervised autoencoder-based approach for anomaly detection in high performance computing systems. *Engineering Applications of Artificial Intelligence* 85 (2019), 634–644.
[7] Andrea Borghesi, Alessio Burrello, and Andrea Bartolini. 2021. ExaMon-X: a Predictive Maintenance Framework for Automatic Monitoring in Industrial IoT Systems. *IEEE Internet of Things Journal* (2021).
[8] Andrea Borghesi, Martin Molan, Michela Milano, and Andrea Bartolini. 2022. Anomaly Detection and Anticipation in High Performance Computing Systems. *IEEE Transactions on Parallel and Distributed Systems* 33, 4 (2022), 739–750. https://doi.org/10.1109/TPDS.2021.3082802
[9] B Evgeniy. 2019. Supercomputer beg with artificial intelligence of optimal resource use and management by continuous processing of large programs. *Glob Acad J Econ Buss* 1 (2019), 21–26.
[10] Zehua Guo. 2022. Graph Neural Network-Based Coflow Scheduling in Data Center Networks. In *Bringing Machine Learning to Software-Defined Networks*. Springer, 53–65.
[11] Olumuyiwa Ibidunmoye, Thijs Metsch, Victor Bayon-Molino, and Erik Elmroth. 2017. Performance Anomaly Detection Using Datacenter Landscape Graphs. In *2017 5th Intl Conf on Applied Computing and Information Technology/4th Intl Conf on Computational Science/Intelligence and Applied Informatics/2nd Intl Conf on Big Data, Cloud Computing, Data Science (ACIT-CSII-BCD)*. 295–302. https://doi.org/10.1109/ACIT-CSII-BCD.2017.40
[12] Arun S Jois and SR Jayasimha. [n. d.]. Telemetry Based Anomaly Detection and Correlation in Data Center. ([n. d.]).
[13] Michael Kenning, Jingjing Deng, Michael Edwards, and Xianghua Xie. 2022. A directed graph convolutional neural network for edge-structured signals in link-fault detection. *Pattern Recognition Letters* 153 (2022), 100–106.
[14] Wania Khan, Davide De Chiara, Ah-Lian Kor, and Marta Chinnici. 2022. Exploratory data analysis for data center energy management. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*. 571–580.
[15] Dejan Milojicic, Paolo Faraboschi, Nicolas Dube, and Duncan Roweth. 2021. Future of HPC: Diversifying Heterogeneity. In *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*. 276–281. https://doi.org/10.23919/DATE51398.2021.9474063
[16] Martin Molan, Andrea Borghesi, Daniele Cesarini, Luca Benini, and Andrea Bartolini. 2023. RUAD: Unsupervised anomaly detection in HPC systems. *Future Generation Computer Systems* 141 (2023), 542–554. https://doi.org/10.1016/j.future.2022.12.001
[17] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. 2016. Geometric deep learning on graphs and manifolds using mixture model CNNs. *CoRR* abs/1611.08402 (2016). arXiv:1611.08402 http://arxiv.org/abs/1611.08402
[18] Alessio Netti, Woong Shin, Michael Ott, Torsten Wilde, and Natalie Bates. 2021. A Conceptual Framework for HPC Operational Data Analytics. In *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. 596–603. https://doi.org/10.1109/Cluster48925.2021.00086
[19] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep Learning for Anomaly Detection: A Review. *ACM Comput. Surv.* 54, 2, Article 38 (March 2021), 38 pages. https://doi.org/10.1145/3439950
[20] Lynn A. Parnell, Dustin W. Demetriou, Vinod Kamath, and Eric Y. Zhang. 2019. Trends in High Performance Computing: Exascale Systems and Facilities Beyond the First Wave. In *2019 18th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*. 167–176. https://doi.org/10.1109/ITHERM.2019.8757229
[21] Fang Shen, Zhan Li, Bing Pan, Ziwei Zhang, Jialong Wang, Wendy Zhao, Xin Wang, and Wenwu Zhu. 2022. Inter-and-Intra Domain Attention Relational Inference for Rack Temperature Prediction in Data Center. In *Database Systems for Advanced Applications: 27th International Conference, DASFAA 2022, Virtual Event, April 11–14, 2022, Proceedings, Part III*. Springer, 481–492.
[22] Woong Shin, Vladyslav Oles, Ahmad Maroof Karimi, J. Austin Ellis, and Feiyi Wang. 2021. 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* (St. Louis, Missouri) *(SC '21)*. Association for Computing Machinery, New York, NY, USA, Article 12, 14 pages. https://doi.org/10.1145/3458817.3476188
[23] Thomas Skordas. 2019. Toward a European exascale ecosystem: the EuroHPC joint undertaking. *Commun. ACM* 62, 4 (2019), 70–70.
[24] Wikipedia. 2021. CINECA — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=CINECA&oldid=954269846. [Online; accessed 04-December-2021].
[25] Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Le Song. 2022. *Graph neural networks*. Springer.
[26] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. Graph Neural Networks: A Review of Methods and Applications. https://doi.org/10.48550/ARXIV.1812.08434
[27] Jing Zhu, Xingyu Lu, Mark Heimann, and Danai Koutra. 2021. Node proximity is all you need: Unified structural and positional node and graph embedding. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*. SIAM, 163–171.