

Background & Motivation

With their immense computing power reaching exaflops per second, HPC systems play a vital role in addressing the world's most complex computational challenges.

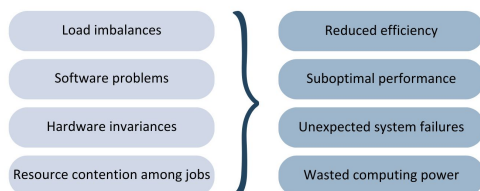


Figure 1: Reasons and effects of performance variations in HPC systems

It's crucial to mitigate performance variations within these systems as they directly impact computational efficiency and reliability, influencing diverse domains, thereby shaping critical outcomes in research and decision-making processes.

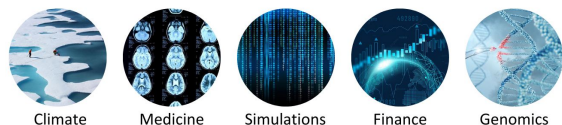


Figure 2: Diverse domains influenced by the performance variations in HPC systems

Objectives

The overarching goal of our project is to design scalable AI-based frameworks to automatically diagnose and mitigate performance anomalies in HPC systems.

One of the latest work of our lab is to design an unsupervised anomaly detection framework for HPC systems. The focus is to decrease the need for large number of labeled data in the anomaly detection process.

Methodology

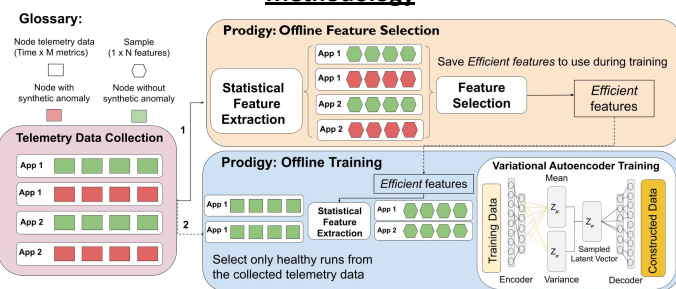


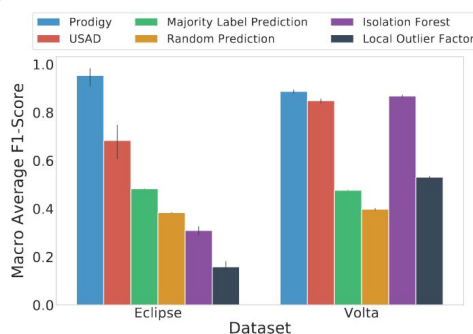
Figure 3: High level workflow of Prodigy framework [1]

- For feature extraction step, we use the open source TSFRESH [2] tool to compute a total of 794 features from time series telemetry data. The extracted features consist of both descriptive and an advanced set of metrics.
- We use the chi-square feature selection method to determine the most important metrics in the training stage.
- In the unsupervised, variational autoencoder based training step, the network learns the characteristics of healthy samples. The aim of this network is to reduce the reconstruction error.
- Anomaly detection bases on the reconstruction error threshold.

Experimental Design

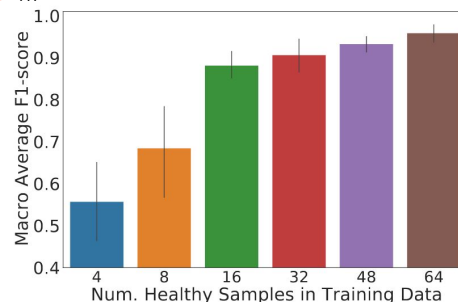
- We collect time series telemetry data from a production HPC system Eclipse and a testbed supercomputer Volta.
- We select some common real and proxy HPC applications (such as LAMMPS, SW4LITE ..) for the experimental procedure.
- We use synthetic anomalies in HPAS [3] to inject into tested applications:
 - memleak*: Simulates memory leakage
 - membw*: Simulates memory bandwidth contention
 - cpuoccupy*: Simulates excessive CPU utilization
 - cacheoccupy*: Simulates cache contention
- We compare five deep-learning based and traditional anomaly detection baseline methods - namely USAD, Local Outlier Factor (LOF), Isolation Forest (IF), Random and Majority Label Prediction - with the Prodigy framework.
- Train and test split ratio is 20-80% and remains the same across all experiments.

Results



Prodigy outperforms the baseline methods by achieving 0.95 and 0.88 F1 scores for Eclipse and Volta datasets, respectively.

Figure 4: Macro average F1 score comparison between Prodigy and other baseline methods f1



Prodigy can work with small number of labeled data and can achieve 0.9 F1 score with only 16 samples.

Figure 5: Prodigy's effectiveness with limited amount of labeled data [1]

Future Directions

We are currently working on reproducing and simulating GPU-based performance bottlenecks in heterogeneous HPC systems.

We will investigate and contribute to intelligent resource allocation methodologies for performance improvements in HPC systems.

References

- [1] Aksar, B., Sencan, E., Schwaller, B., Aaziz, O., Leung, V. J., Brandt, J., ... & Coskun, A. K. (2023, November). Prodigy: Towards unsupervised anomaly detection in production hpc systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1-14).
- [2] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr. 2018. Time series feature extraction on basis of scalable hypothesis tests (tsfresh—a python package). *Neurocomputing* 307 (2018), 72–77.
- [3] Ates, E., Zhang, Y., Aksar, B., Brandt, J., Leung, V. J., Egele, M., & Coskun, A. K. (2019, August). HPAS: An HPC performance anomaly suite for reproducing performance variations. In *Proceedings of the 48th International Conference on Parallel Processing* (pp. 1-10).