

Transfer Learning Workflow for High-Quality I/O Bandwidth Prediction with Limited Data

Dmytro Povaliaiev
RWTH Aachen University
Aachen, Germany
dmytro.povaliaiev@rwth-aachen.de

Radita Liem
RWTH Aachen University
Aachen, Germany
liem@itc.rwth-aachen.de

Julian Kunkel
University of Göttingen/GWDG
Göttingen, Germany
julian.kunkel@gwdg.de

Jay Lofstead
Sandia National Lab
Albuquerque, New Mexico, USA
gflfst@sandia.gov

Philip Carns
Argonne National Laboratory
Lemont, Illinois, USA
carns@mcs.anl.gov

1 INTRODUCTION

The I/O performance of a scientific application is difficult to predict due to multiple intertwined variables coming from the hardware, middleware, and the application layer[1]. Within the application layer itself, multiple components are affecting the performance, such as different programming paradigms and I/O operation algorithms[2]. This makes predicting I/O performance a good candidate problem for machine learning due to the complex relationships of the variables involved. However, making a high-quality prediction requires a large amount of high-quality data, and collecting it is a big challenge for most data centers. Comprehensive I/O performance data from various types of applications can take years to gather and is rarely done in practice by small to medium data centers due to their limited resources.

In this project, we explore the use of transfer learning to predict the I/O performance by utilizing the publicly available I/O performance data in the form of Darshan[3] logs that were collected at the Blue Waters supercomputer operated by NCSA during 2012-2021¹. We devise a workflow to train a neural network model on the data extracted from these I/O logs and then use it as a base to predict the POSIX I/O bandwidth performance of other clusters (CLAIX18 [4] at the RWTH Aachen University and Theta [5] at the Argonne National Laboratory) as presented in Figure 1. Transfer learning enables us to retrain the model to predict the performance of a different unrelated cluster while requiring a significantly smaller amount of data than the usual approach of using only the data from one particular cluster. In our experiments, the bandwidth predictions for those clusters were obtained using the datasets of as few as <1% records (compared to around 680,000 Darshan logs to build the base neural network model).

The outcome of our project is a workflow that employs transfer learning to predict the I/O performance. Our workflow consists of training the base model on the initial data from Blue Waters, fine-tuning it on much smaller data from the target cluster, and verifying the model using cross-validation and explainable AI techniques. The workflow achieves a promising result, suggesting it can be used to predict the I/O bandwidth of another system with a mean absolute error better or equivalent to the state-of-the-art model built directly on the data from the target cluster [6].

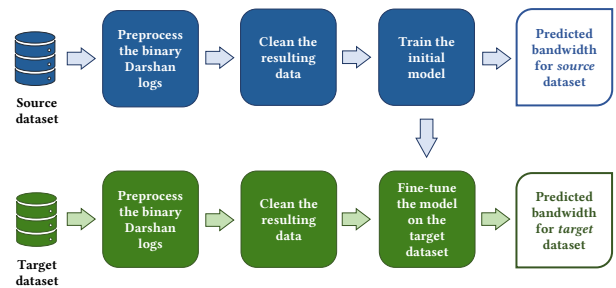


Figure 1: Overview of the Transfer Learning Workflow

2 RELATED WORKS

Multiple machine learning studies and experiments have been done regarding the performance prediction in HPC environments. In terms of I/O performance specifically, there are several existing works trying to cover various aspects of performance such as access patterns[7], data transfer[8][9], scheduling strategy[10], access time[11], memory usage[8], and bandwidth[6][12][13]. Using Darshan logs is one of the common ways to produce the training dataset[6][13][14]. Transfer learning was also used in various related papers to predict runtime[15] and conduct performance autotuning[16][17] in the HPC domain.

3 WORKFLOW

Our workflow can be divided into four parts: 1) preparing and cleaning the data by removing the erroneous records, all-zero features, and bandwidth outliers (the latter using the interquartile range (IQR) method[18]), 2) training the initial neural network model, 3) fine-tuning/transfer learning the initial model on a much smaller target dataset, and 4) verifying the results using cross-validation and explainable AI methods.

The neural network architecture is based on a Multi-Layer Perceptron[19] with fully-connected hidden layers. It receives 96 various POSIX counters and the number of processes from the Darshan log as input and produces bandwidth in MB/s as output (see Figure 2). To ensure the robustness of our model, we perform a stratified 5-fold cross-validation[20] during the initial model training

¹<https://bluewater.ncsa.illinois.edu/data-sets>

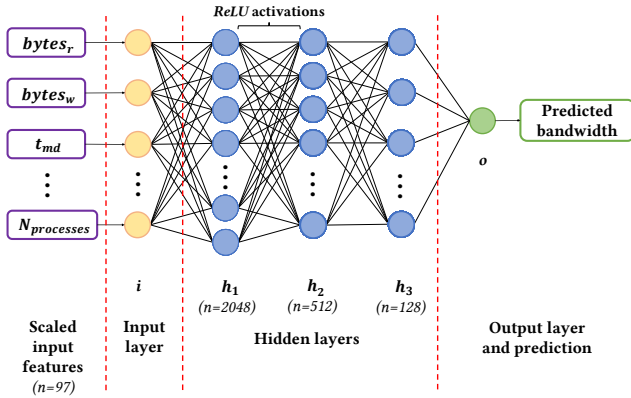


Figure 2: Neural network architecture used in our work. It consists of an input layer, three hidden layers connected with ReLU activations, and the output layer that produces the bandwidth value

stage using ten random seeds (resulting in 50 models) and then repeat the process for each base model during the fine-tuning/transfer learning stage, producing 250 models. To verify that the model is learning the correct features from the dataset, we apply multiple explainable machine learning techniques after the transfer learning stage. We employ nine approaches: Integrated Gradients[21], Integrated Gradients with Noise Tunnel[22], DeepLift[23], Feature Ablation[24], Shapley Value Sampling[25] [26], Guided Backpropagation[27], Feature Permutation[28], InputXGrad[29] and Saliency[30]. We average the results of these approaches to create a list of the most important features.

We evaluate two model variants to avoid unnecessary over-generalization caused by training the model to predict the data highly unlikely to appear at the target cluster. The first variant is trained on the whole Blue Waters dataset, and the training data for the second variant contains only the records with the number of processes that also appears in the target cluster data. The reasoning behind this is that the bandwidth from a large number of processes will never be achieved by the target cluster because of its limited hardware resources (nodes and cores). For the target dataset, we collected around 1,300 Darshan logs from scientific applications and benchmarks running on CLAI18 (< 1% compared to the initial training data) and randomly selected around 60,000 Darshan logs from those that were already collected on Theta (< 10% relative to the initial training data).

The Darshan binary logs were processed on one node of CLAI18 (2 Intel Skylake with 2.1 GHz and 48 cores in total and 192 GB of memory). For training the deep learning models, we used the CLAI16 GPU partition (NVIDIA P100-SXM2 16 GB GPU with 1 Intel Broadwell 2.2 GHz and 12 cores and 64 GB memory).

4 RESULTS

Figure 3 compares the results of the initial training stage and the fine-tuning/transfer learning stage side-by-side. In this Figure, we use the percentage error relative to the bandwidth mean as the comparison metric. Our base neural network performed better than

state of the art from Isakov et al.[6], and the transfer learning result on the limited number of processes is relatively comparable to state of the art. We also compare our approach against two baseline models: 1) random guessing within the IQR range and 2) the model trained directly on the target cluster data without using transfer learning. Table 1 shows the mean percentage error relative to the mean bandwidth across all our tested variants. These results show that transfer learning has an undeniable impact since it outperforms both baseline models.

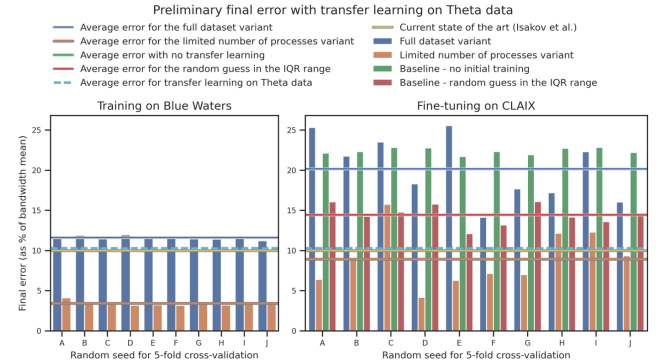


Figure 3: Comparison of the results from the base neural network model on the Blue Waters data (left chart) and the models on the CLAI18 and the Theta data after transfer learning (right chart)

Table 1: Final error across all stages and variants as a percentage of mean bandwidth

Variant	Initial Training	Fine-tuning
Full Dataset	11.6%	20.1%
Limited # of Processes	3.4%	8.92%
Random Guess in IQR	95.9%	14.4%
Without Initial Training	-	22.4%
State of the Art	10%	10%

The result from our explainable AI analysis shows that the most important features are related to time, data transfer sizes, and the number of processes. Overall, the model can learn the formula used to calculate the bandwidth on its own with no pre-existing notion of it. This result shows that our proposed Transfer Learning Workflow can produce a high-quality I/O bandwidth prediction for clusters while requiring limited I/O performance data. Our testing on the data from two systems with different architectures shows that the workflow is generalizable. We will extend this workflow in the future to create more fine-grained predictions based on specific components like the filesystem or application type.

5 ACKNOWLEDGEMENTS

Training and evaluation of the machine learning model were performed with computing resources granted by RWTH Aachen University under project thes1067.

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) the State of Illinois, and as of December, 2019, the National Geospatial-Intelligence Agency. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

This work was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

REFERENCES

- [1] Jay Lofstead, Milo Polte, Garth Gibson, Scott Klasky, Karsten Schwan, Ron Oldfield, Matthew Wolf, and Qing Liu. 2011. Six degrees of scientific data: reading patterns for extreme scale science io. In *Proceedings of the 20th International Symposium on High Performance Distributed Computing (HPDC '11)*. Association for Computing Machinery, San Jose, California, USA, 49–60. ISBN: 9781450305525. doi: 10.1145/1996130.1996139.
- [2] Jean Luca Bez, Suren Byna, and Shadi Ibrahim. 2023. I/o access patterns in hpc applications: a 360-degree survey. *ACM Comput. Surv.*, (July 2023). Just Accepted. doi: 10.1145/3611007.
- [3] Philip Carns, Kevin Harms, William Allcock, Charles Bacon, Samuel Lang, Robert Latham, and Robert Ross. 2011. Understanding and Improving Computational Science Storage Access through Continuous Characterization. *ACM Transactions on Storage*, 7, 3, (Oct. 2011), 8:1–8:26. doi: 10.1145/2027066.2027068.
- [4] [n. d.] Claix (2018) - intel-hns2600bpb, xeon platinum 8160 24c 2.1ghz, intel omni-path | top500. <https://www.top500.org/system/179682/>. (Accessed on 08/05/2023). ().
- [5] [n. d.] Theta - cray xc40, intel xeon phi 7230 64c 1.3ghz, aries interconnect | top500. <https://www.top500.org/system/178926/>. (Accessed on 08/05/2023). ().
- [6] Mihailo Isakov, Eliakin del Rosario, Sandeep Madireddy, Prasanna Balaprakash, Philip Carns, Robert B. Ross, and Michel A. Kinsy. 2020. HPC I/O Throughput Bottleneck Analysis with Explainable Local Models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. (Nov. 2020), 1–13. doi: 10.1109/SC41405.2020.00037.
- [7] Jean Luca Bez, Francieli Zanon Boito, Ramon Nou, Alberto Miranda, Toni Cortes, and Philippe O. A. Navaux. 2019. Detecting I/O Access Patterns of HPC Workloads at Runtime. In *2019 31st International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. (Oct. 2019), 80–87. doi: 10.1109/SBAC-PAD.2019.00025.
- [8] Benjamin Shealy. 2021. *Intelligent Resource Prediction for HPC and Scientific Workflows*. Ph.D. Dissertation. Clemson University, (Dec. 2021). https://tigerprints.clemson.edu/all_dissertations/2956.
- [9] Benjamin T. Shealy, F. Alex Feltus, and Melissa C. Smith. 2021. Intelligent Resource Provisioning for Scientific Workflows and HPC. In *2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS)*. (Nov. 2021), 9–16. doi: 10.1109/WORKS54523.2021.00007.
- [10] Lu Pang and Krishna Kant. 2022. Server-Side Workload Identification for HPC I/O Requests. In *Proceedings of the 2nd Workshop on Performance Engineering, Modelling, Analysis, and Visualization Strategy (PERMAVOST '22)*. Association for Computing Machinery, New York, NY, USA, (June 2022), 15–22. ISBN: 978-1-4503-9314-0. doi: 10.1145/3526063.3535350.
- [11] Jan F. Schmidt and Julian M. Kunkel. 2016. Predicting I/O performance in HPC using artificial neural networks. *Supercomputing Frontiers and Innovations*, 3, 3, (Sept. 2016), 19–33. Retrieved Aug. 17, 2021 from <http://centaur.reading.ac.uk/77675/>.
- [12] Abdul Jabbar Saeed Tipu, Pádraig Ó Conbhú, and Enda Howley. 2022. Artificial neural networks based predictions towards the auto-tuning and optimization of parallel IO bandwidth in HPC system. *Cluster Computing*, (Dec. 2022). doi: 10.1007/s10586-022-03814-w.
- [13] David James Gabriel. 2022. *I/O Throughput Prediction for HPC Applications Using Darshan Logs*. Master's thesis. University of Nevada, Reno, (May 2022). Retrieved Oct. 29, 2022 from <https://scholarworks.unr.edu/handle/11714/8129>.
- [14] Bing Xie, Zilong Tan, Philip Carns, Jeff Chase, Kevin Harms, Jay Lofstead, Sarp Oral, Sudharshan S. Vazhkudai, and Feiyi Wang. 2021. Interpreting Write Performance of Supercomputer I/O Systems with Regression Models. In *2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. (May 2021), 557–566. doi: 10.1109/IPDPS49936.2021.00064.
- [15] Preeti Malakar, Prasanna Balaprakash, Venkatram Vishwanath, Vitali Morozov, and Kalyan Kumar. 2018. Benchmarking Machine Learning Methods for Performance Modeling of Scientific Applications. In *2018 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*. (Nov. 2018), 33–44. doi: 10.1109/PMBS.2018.8641686.
- [16] Harshitha Menon, Abhinav Bhatel, and Todd Gamblin. 2020. Auto-tuning Parameter Choices in HPC Applications using Bayesian Optimization. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. (May 2020), 831–840. doi: 10.1109/IPDPS47924.2020.00090.
- [17] Matthieu Dorier, Romain Egele, Prasanna Balaprakash, Jaehoon Koo, Sandeep Madireddy, Srinivasan Ramesh, Allen D. Malony, and Rob Ross. 2022. HPC Storage Service Autotuning Using Variational -Autoencoder -Guided Asynchronous Bayesian Optimization. In *2022 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, Heidelberg, Germany, (Sept. 2022), 381–393. ISBN: 978-1-66549-856-2. doi: 10.1109/CLUSTER51413.2022.00049.
- [18] Katharina Benkert, Edgar Gabriel, and Michael M. Resch. 2008. Outlier detection in performance data of parallel applications. In *2008 IEEE International Symposium on Parallel and Distributed Processing*. (Apr. 2008), 1–8. doi: 10.1109/IPDPS.2008.4536463.
- [19] Jean-Gabriel Attali and Gilles Pagès. 1997. Approximations of Functions by a Multilayer Perceptron: A New Approach. *Neural Networks*, 10, 6, (Aug. 1997), 1069–1081. doi: 10.1016/S0893-6080(97)00010-5.
- [20] Sylvain Arlot and Alain Celisse. 2010. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4, none, (Jan. 2010), 40–79. doi: 10.1214/09-SS054.
- [21] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML '17)*. JMLR.org, Sydney, NSW, Australia, (Aug. 2017), 3319–3328. Retrieved Sept. 1, 2022 from.
- [22] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. SmoothGrad: removing noise by adding noise. (June 2017). arXiv: 1706.03825 [cs, stat]. doi: 10.48550/arXiv.1706.03825.
- [23] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML '17)*. JMLR.org, Sydney, NSW, Australia, (Aug. 2017), 3145–3153. Retrieved Sept. 8, 2022 from.
- [24] Luke Merrick. 2019. Randomized Ablation Feature Importance. (Oct. 2019). arXiv: 1910.00174 [cs, stat]. doi: 10.48550/arXiv.1910.00174.
- [25] Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*. Selected Papers Presented at the Tenth International Symposium on Locational Decisions (ISOLDE X) 36, 5, (May 2009), 1726–1730. doi: 10.1016/j.cor.2008.04.004.
- [26] Erik Strumbelj and Igor Kononenko. 2010. An Efficient Explanation of Individual Classifications using Game Theory. *The Journal of Machine Learning Research*, 11, (Mar. 2010), 1–18.
- [27] J. Springenberg, Alexey Dosovitskiy, Thomas Brox, and M. Riedmiller. 2015. Striving for Simplicity: The All Convolutional Net. In *ICLR (workshop track)*. Retrieved Sept. 9, 2022 from <https://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a/>.
- [28] Giuseppe Casalicchio, Christoph Molnar, and Bernd Bischl. 2019. Visualizing the Feature Importance for Black Box Models. In *Machine Learning and Knowledge Discovery in Databases (Lecture Notes in Computer Science)*. Michele Berlingerio, Francesco Bonchi, Thomas Gärtner, Neil Hurley, and Georgiana Ifrim, (Eds.) Springer International Publishing, Cham, 655–670. ISBN: 978-3-030-10925-7. doi: 10.1007/978-3-030-10925-7_40.
- [29] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. 2017. Not Just a Black Box: Learning Important Features Through Propagating Activation Differences. (Apr. 2017). arXiv: 1605.01713 [cs]. doi: 10.48550/arXiv.1605.01713.
- [30] K. Simonyan, A. Vedaldi, and A. Zisserman. 2019. Deep inside convolutional networks: visualising image classification models and saliency maps. *Proceedings of the International Conference on Learning Representations (ICLR)*. Retrieved Sept. 9, 2022 from <https://ora.ox.ac.uk/objects/uuid:ea777387-538b-4a01-843b-41b2f05dd287>.