Exploring Data Caching Policy with Access Patterns from dCache Logs

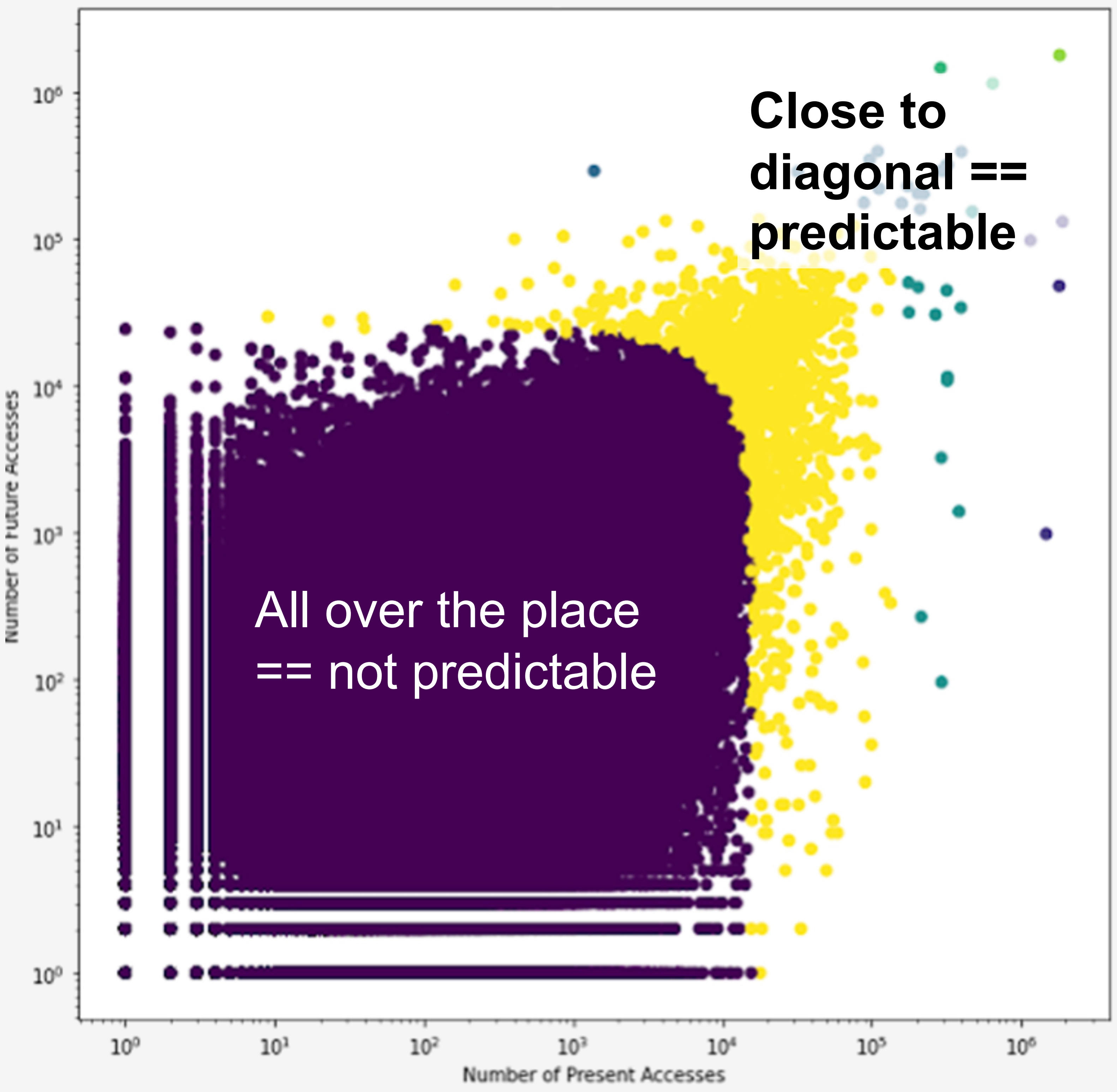
Jacob Aldrich[1,2], Vincent Garonne[3], Hironori Ito[3], Eric Lancon[3], Alex Sim[1], Kesheng Wu[1], Shinjae Yoo[3]
1. Lawrence Berkeley National Laboratory, 2. UC Berkeley, 3. Brookhaven National Laboratory.

Abstract

In this study, we develop machine learning models to predict dataset access frequencies in the dCache storage system at Brookhaven National Lab. Despite the inherent challenges in predicting data access counts, our neural network and gradient-boosted regression tree models show promising results.

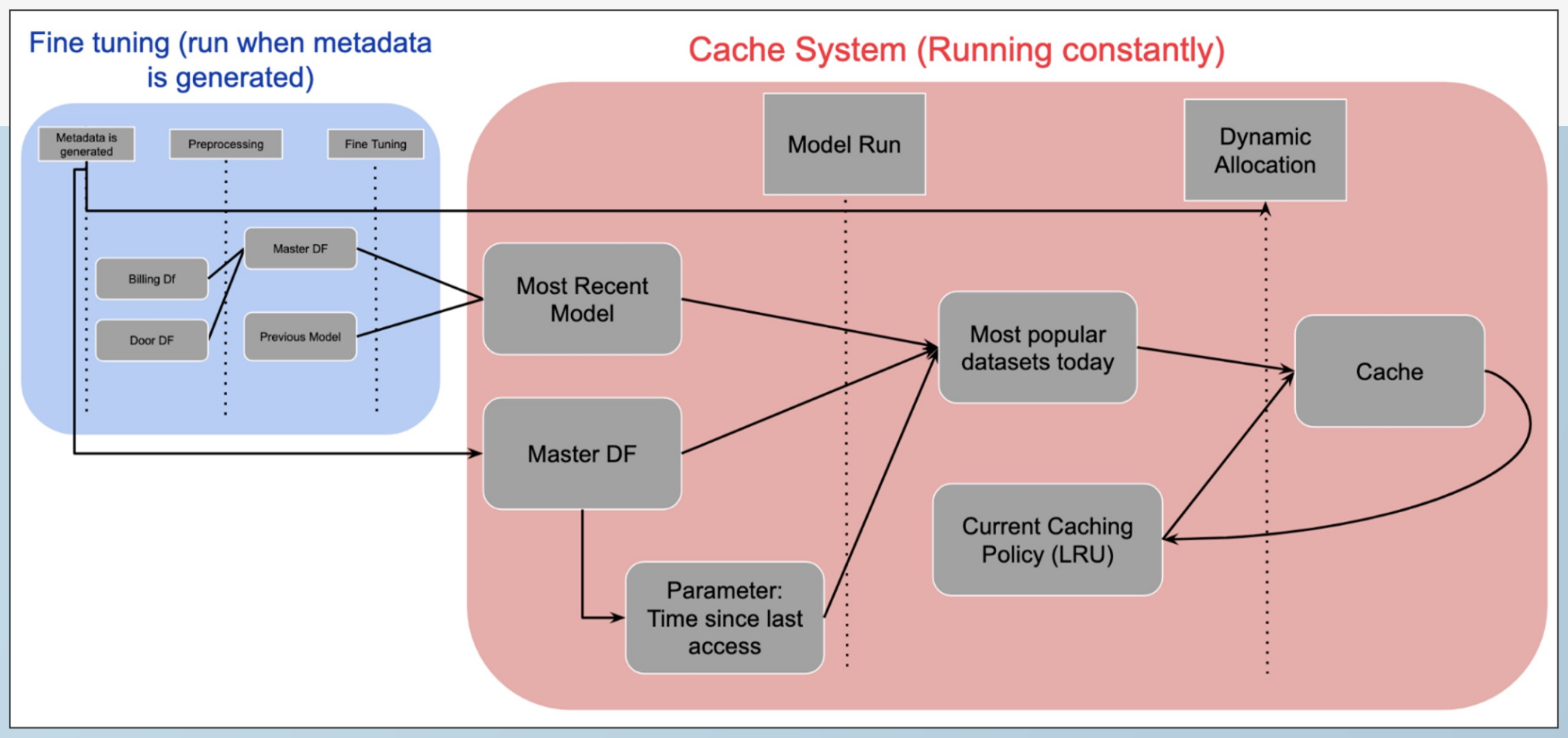
Exploratory Data Analysis

K-Means clustering (predicted access counts (x-axis) vs. actual access count) suggests that the less popular files are not predictable, while the more popular ones are likely more predictable (points closer to the diagonal).



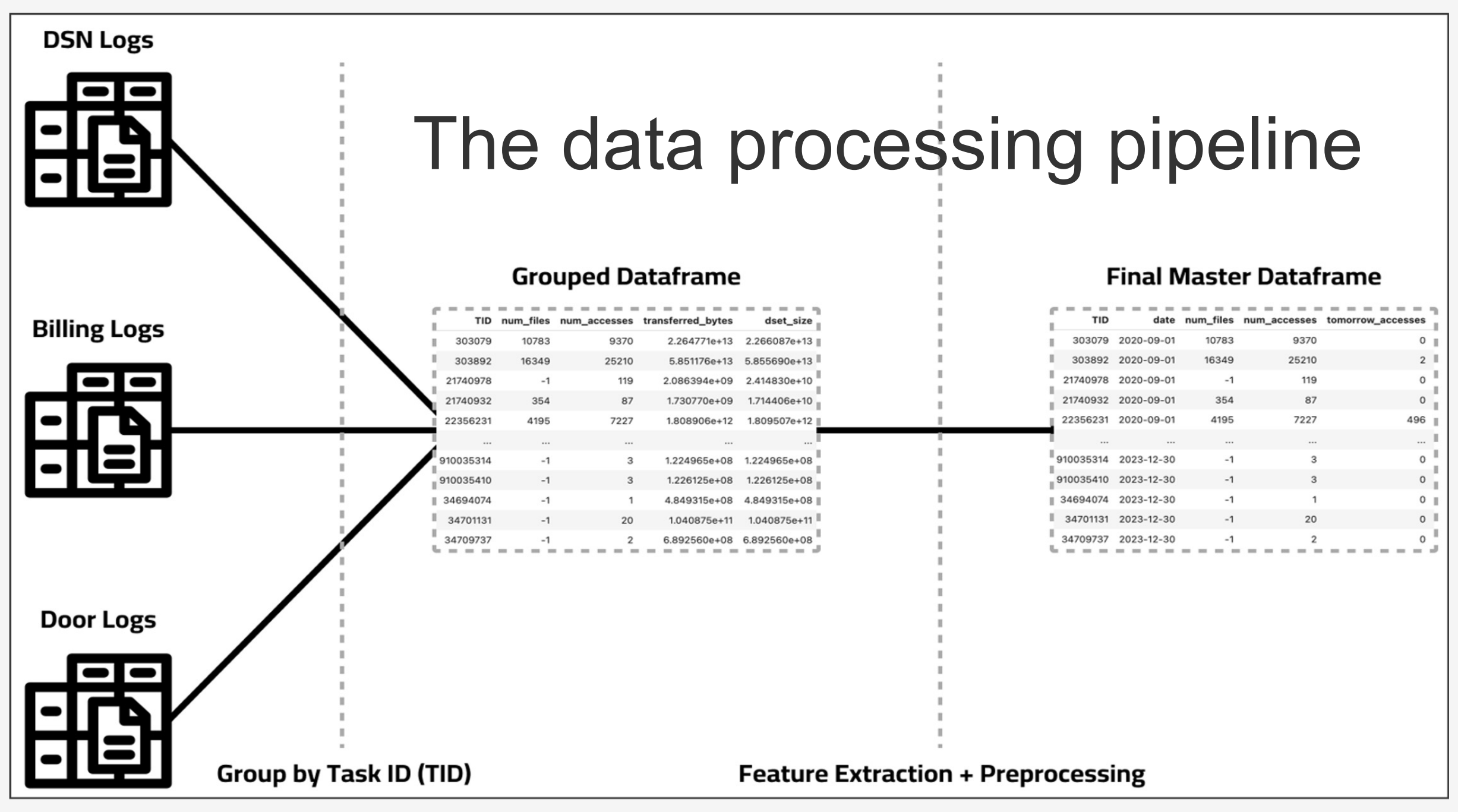
Caching System

The intelligent caching system for dCache stores the most popular datasets through a dynamic workflow involving four key steps: (1) Metadata Processing, (2) Model Fine-tuning, (3) Popularity Prediction, and (4) Dynamic Cache Allocation. The system automatically adjusts to changes in dataset access frequencies as well as metadata updates (or a lack thereof). This caching system integrates with the existing dCache infrastructure, optimizing storage and improving efficiency.


Fine tuning (run when metadata is generated)
Cache System (Running constantly)
Metadata is generated
Preprocessing
Fine Tuning
Billing Df
Door DF
Master DF
Previous Model
Model Run
Dynamic Allocation
Most Recent Model
Master DF
Parameter: Time since last access
Most popular datasets today
Current Caching Policy (LRU)
Cache

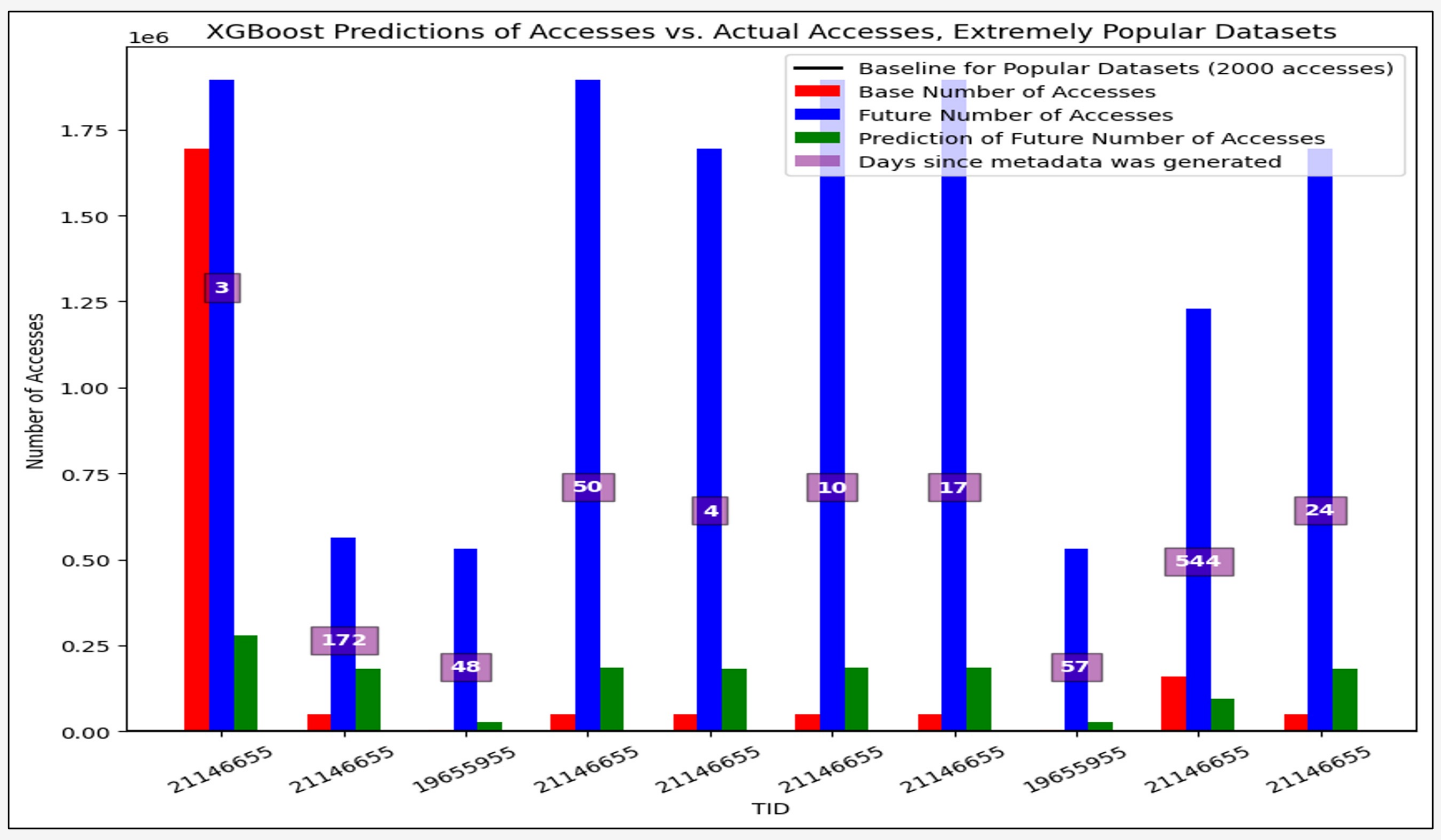
Background

The hierarchical dCache system dedicated to ATLAS [1, 2], combines disk and tape to offer long-term archival and fast performance [3]. To increase predictability, we group files into datasets using Task IDs (TIDs), as files with the same TID are typically accessed together [4].



Model Development

We evaluated two models to predict future dataset accesses: Gradient-boosted regression trees (XGBoost) [6] and neural networks [5]. Both models used 17 input features to forecast dataset accesses, trained on 127 million data points. XGBoost achieved a testing RMSE to standard deviation ratio of 0.28, outperforming the neural network's ratio of 0.84



XGBoost works quite well, but still struggles with extreme cases.

Conclusion

This study demonstrates that data accesses in the dCache system can be predicted using machine learning models. Our models show promising results in forecasting access frequencies, despite the inherent challenge in prediction. Additionally, the proposed dynamic caching system stands to improve dCache effectiveness significantly. While further testing is needed for implementation, our findings highlight the potential of integrating machine learning into storage management systems like dCache.

References

[1] M. Ernst, P. Fuhrmann, M. Gasthuber, T. Mkrtchyan, C. Waldman, dcache, a distributed storage data caching system, Journal of Physics: Conference Series (2001)
[2] G. Behrmann, P. Fuhrmann, M. Grønager, J. Kleist, A distributed storage system with dCache, Journal of Physics: Conference Series 119 (2008)
[3] Y. Wang, K. Wu, A. Sim, S. Yoo, S. Misawa, Access Patterns to Disk Cache for Large Scientific Archive, in ACM International Workshop on Systems and Network Telemetry and Analytics (2020), pp. 37–40
[4] R.W. Watson, R.A. Coyne, The Parallel I/O Architecture of the High-Performance Storage System (HPSS), in Proceedings of the 14th IEEE Symposium on Mass Storage Systems (1995), ISBN 0818670649
[5] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., Pytorch: An imperative style, high-performance deep learning library, CoRR abs/1912.01703 (2019), 1912.01703
[6] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, CoRR abs/1603.02754 (2016), 1603.02754
[7] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv (2017), 1412.698