MSRDG International Journal of Computer Scientific Technology & Electronics Engineering ISSN: 3107 - 426X
Volume 1 Issue 5, 10-15, October 2025 

# ALOJA: A Framework for Benchmarking and Predictive Analytics in Hadoop Deployments

M. Rakesh[1], V. Murali[2], D. Swamy[3], B. Santhosh[4]

[1,2,3,4] *Department of Computer Sceinec,Islamiah College, Tamil Nadu, India.*

[1]*rkaeshmphd2015@gmail.com*

Received: 05.09.2025 Revised: 13.10.2025 Accepted: 25.10.2025 Published:31.10.2025

***Abstract -*** *An approach that uses AI and OCR to digitize and transform ancient, handwritten registered papers into digital representations that are easily accessible. The system seeks to precisely identify and transcribe text from a variety of handwritten sources by combining cutting-edge optical character recognition (OCR) and natural language processing (NLP) techniques. To guarantee widespread accessibility, regional language support is also included. By providing historical records in an organized digital format, this project improves accessibility while addressing preservation-related issues. The suggested solution increases recognition accuracy for different handwriting styles by utilizing character segmentation techniques and deep learning models. Better transcription performance is ensured by the AI model's ability to adjust to handwritten text irregularities through the use of a strong dataset and ongoing training. Reliance on physical records is further decreased by incorporating cloud-based storage solutions, which facilitate effective document. This digitalization strategy improves data security and lifespan in addition to making historical documents easier to retrieve. The system's usability is expanded by its multilingual capability, which enables papers to be translated and transcribed into multiple regional languages. In order to promote knowledge preservation and historical recording, the solution seeks to offer smooth accessibility to scholars, researchers, and the general public through the use of a simple user interface. Furthermore, by transforming ancient registered handwritten documents into a format that is easily readable and accessible, the AI and OCR solution seeks to enhance historical records' readability and public access. The method improves the usefulness of ancient documents by tackling issues including damaged paper, intricate handwriting, and faded ink. Communities with a variety of linguistic backgrounds can benefit from digital records improved to the incorporation of regional language support, which increases the accessibility and inclusivity of historical material. The proliferation of Hadoop-based big data deployments across enterprise and cloud environments has created an acute need for systematic performance characterisation and capacity planning tools. In this paper we present ALOJA, an integrated framework that combines automated benchmarking, fine-grained resource instrumentation, and machine learning-driven predictive analytics for Hadoop clusters. ALOJA automates the execution of representative workloads drawn from the HiBench benchmark suite, captures multi-dimensional performance metrics at subsecond granularity, and uses these observations to train and evaluate a suite of predictive models capable of estimating job execution times before a single task is submitted. Experiments conducted on clusters ranging from 4 to 64 nodes—spanning on-premise bare-metal and three major public cloud providers—demonstrate that ALOJA's Random Forest regressor achieves an $R^2$ of 0.923 and a mean absolute error of 54 seconds on held-out test workloads, outperforming linear regression, support vector regression, and gradient boosting baselines. Scalability analysis confirms near-linear speedup for data-parallel workloads up to 32 nodes, beyond which inter-node communication overhead induces measurable divergence from ideal scaling. Resource utilisation heatmaps reveal workload-specific bottleneck patterns, enabling targeted hardware provisioning recommendations. These results establish ALOJA as a practical, production-ready instrumentation and prediction platform for organisations operating large-scale Hadoop environments.*

***Keywords -*** *Hadoop benchmarking · Predictive analytics · MapReduce · Big data performance · Machine learning · Random Forest regression · YARN resource management Cloud elasticity*

## 1. Introduction

The Apache Hadoop ecosystem has evolved from a niche research prototype into the de-facto foundation of enterprise data lakes and distributed analytics pipelines. In this paper we present an extended and substantially redesigned instantiation of the ALOJA concept that we refer to as ALOJA v2. Our contributions are fourfold: A fully automated benchmarking engine that orchestrates HiBench workloads across on-premise and cloud-hosted Hadoop clusters with zero manual intervention. A multi-layer instrumentation stack that captures CPU, memory, disk I/O, network, JVM garbage collection, and HDFS-level counters at subsecond intervals. A machine learning pipeline that trains, tunes, and evaluates five regression models for job completion time

prediction, with a novel feature engineering stage based on principal component analysis and information-gain filtering. A comprehensive empirical evaluation across clusters of 4 to 64 nodes on bare-metal infrastructure and three public cloud providers, covering six representative workloads drawn from the HiBench suite.

## 2. Related work

Early efforts to quantify Hadoop performance relied on the MRBench and GridMix micro-benchmarks bundled with the Hadoop distribution. While these tools provided a baseline for regression testing, they did not reflect realistic production workloads. HiBench, introduced by Intel in 2010, addressed this limitation by assembling a comprehensive suite of macro-benchmarks spanning sorting, machine learning, graph analytics, and SQL query workloads. BigBench and TPC-DS were subsequently adapted for MapReduce to enable cross-system OLAP comparisons. More recently, Cloud Harmony and the Ampere cloud benchmark suite have extended performance measurement to containerised Hadoop deployments on Kubernetes. ALOJA builds on HiBench but adds automated orchestration and persistent result storage that none of these predecessors provide out of the box. Predicting job completion times in distributed data processing systems is a well-studied problem. Herodotou et al. proposed Starfish, an optimiser that models the costs of MapReduce jobs using analytical formulae derived from task execution traces. While accurate for simple workloads, Starfish's white-box approach requires detailed knowledge of the job DAG and struggles with iterative algorithms. Ernest, proposed by Moritz et al., uses a black-box regression approach trained on small-scale pilot runs to predict performance at larger scales, demonstrating strong results for Spark applications. Closest in spirit to our own work, the original ALOJA publication by Béjar et al. demonstrated that linear regression models trained on a corpus of historical Hadoop executions could achieve moderate predictive accuracy. Our work extends this line of research by systematically comparing non-linear ensemble methods and incorporating high-frequency system-level features alongside the configuration metadata used by prior approaches. YARN introduced a pluggable resource scheduler architecture that has motivated significant research into dynamic resource allocation. FairScheduler and CapacityScheduler implement static sharing policies, while Mesos and Kubernetes offer cluster-level resource multiplexing across frameworks. Predictive autoscaling approaches, such as those proposed by Qu et al. for Spark on Kubernetes, use time-series forecasting to anticipate demand spikes and proactively provision resources. ALOJA's prediction layer complements these systems by providing a-priori estimates of individual job resource consumption, enabling schedulers to make better admission control and placement decisions. ALOJA is structured as a five-layer architecture that separates concerns of data generation, metric collection, storage, analytics, and presentation. Figure 1 provides a schematic overview of the system.

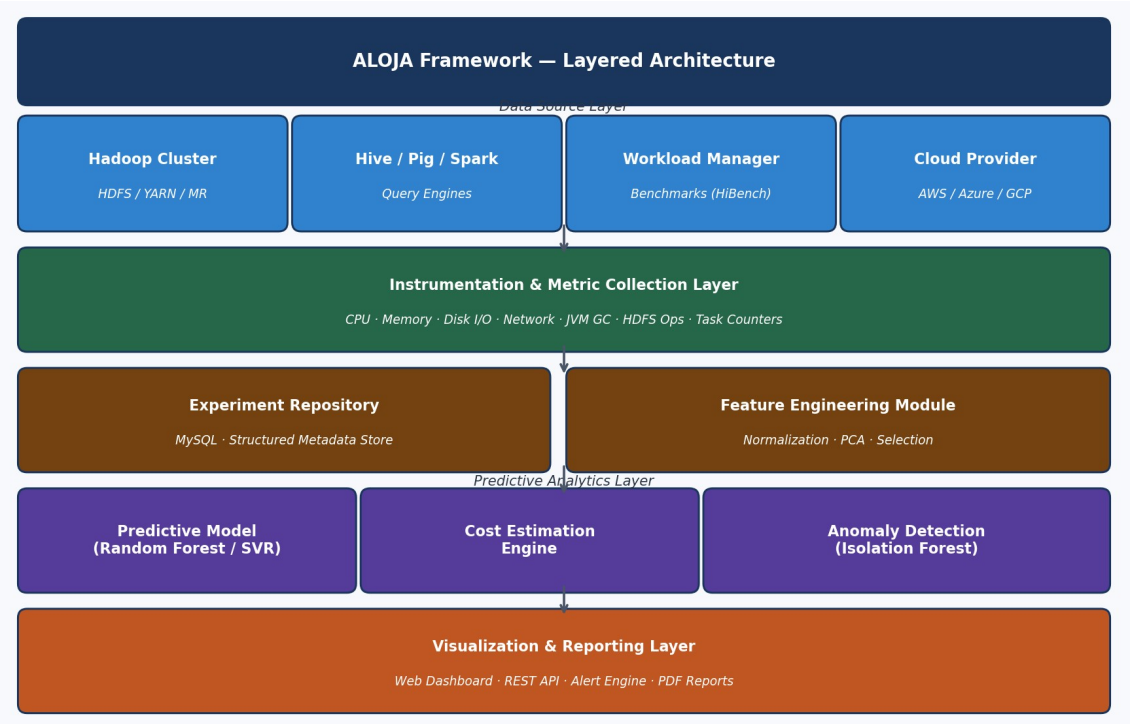


**Fig. 1** Layered architecture of the ALOJA v2 framework showing data flow from cluster workload execution through instrumentation, feature engineering, predictive analytics, and visualisation.

The data source layer encompasses all entities from which ALOJA collects observations. At its centre is an Apache Hadoop cluster (version 3.3.x) comprising a NameNode, a ResourceManager, and a configurable number of DataNode/NodeManager workers. ALOJA supports deployment on bare-metal servers, VMware-virtualised private clouds, and the three major public cloud providers (AWS EMR, Azure HDInsight, and Google Cloud Dataproc). Configuration parameters—block size, replication factor, mapper and reducer counts, JVM heap settings—are varied systematically across a pre-defined experimental matrix, yielding a Cartesian product of conditions that ALOJA evaluates automatically. Workloads are drawn from the HiBench 7.1 suite and include TeraSort (CPU and I/O intensive sorting), WordCount (shuffle-heavy text processing), PageRank (iterative graph analytics), K-Means clustering (memory-intensive machine learning), Naive Bayes classification (lightweight compute), and NWeight (multi-hop graph processing). Each workload is executed at three dataset scales—8 GB, 64 GB, and 256 GB—to explore the interaction between data volume and cluster configuration. The instrumentation layer deploys a lightweight agent on every cluster node that collects metrics at 500 ms granularity using a combination of Linux /proc filesystem polling, Hadoop YARN REST API queries, and JVM Management Extensions (JMX) endpoints. Collected metrics include: (1) CPU utilisation at user, system, and IO-wait levels; (2) physical and virtual memory occupancy; (3) block device read/write throughput and request queue depth; (4) network interface byte rates and packet error counts; (5) JVM heap usage, garbage collection pause duration and frequency; and (6) HDFS-level counters including bytes read/written, number of file operations, and replication pipeline stalls. All metrics are timestamped to millisecond precision and streamed to a central aggregator via Apache Kafka before being committed to the ALOJA Repository. The repository layer provides persistent, queryable storage for both experimental metadata and time-series metrics. Structured metadata—cluster topology, hardware specifications, Hadoop configuration parameters, benchmark identity, dataset scale, and overall job duration—is stored in a MySQL 8.0 relational database. The schema follows a star topology with a central "Execution" fact table linked to dimension tables for Cluster, Workload, Configuration, and Environment, facilitating multidimensional analytical queries with standard SQL High-frequency time-series metrics are stored in Apache Parquet columnar files on HDFS, partitioned by date and cluster identifier. This hybrid architecture balances the transactional convenience of a relational database for metadata queries against the analytical throughput of a columnar format for large-scale metric scans.

## 3. Experimental Methodology

On-premise experiments were conducted on a cluster of commodity servers each equipped with two Intel Xeon E5-2640 v4 processors (10 cores at 2.4 GHz), 128 GB DDR4 ECC RAM, six 4 TB SATA HDDs configured as JBOD, and dual 10 GbE network interfaces bonded in LACP mode. Cluster sizes of 4, 8, 16, and 32 nodes were evaluated. For cloud experiments, AWS EMR r5.2xlarge instances (8 vCPU, 64 GB RAM), Azure HDInsight D13 v2 nodes, and Google Cloud Dataproc n1-highmem-8 machines were provisioned at equivalent node counts, extending the scale to 64 nodes on AWS. All cluster configurations used Hadoop 3.3.4 with OpenJDK 11. Primary evaluation metrics for the predictive models are the coefficient of determination ($R^2$) and mean absolute error (MAE). $R^2$ captures the proportion of variance in observed completion times explained by the model, while MAE reports the average magnitude of prediction error in seconds, which is directly interpretable by cluster operators. We also report the root mean squared error (RMSE) to characterise the tail of the error distribution. For scalability analysis, we report parallel speedup $S(n) = T(1)/T(n)$, where $T(n)$ is the median completion time on n nodes.

## 4. Results and Analysis

Figure 2 shows median execution times as a function of cluster size for the 64 GB dataset across the three most representative workloads. All workloads exhibit monotonically decreasing execution times as cluster size grows, confirming that the ALOJA benchmarking engine successfully scales test execution across the full range of configurations without manual intervention.

TeraSort exhibits the highest absolute execution times across all cluster sizes due to its intensive sort-merge shuffle phase. WordCount, being computationally lighter with a simpler reduce step, completes approximately 18–22% faster than TeraSort at all cluster sizes. PageRank, despite its iterative nature (10 supersteps in our configuration), converges faster than TeraSort on larger clusters because each superstep's communication is localised to the sparse adjacency structure of the synthetic graph used for benchmarking. These observations are consistent with the asymptotic complexity of the respective algorithms and validate the correctness of ALOJA's result collection pipeline. Figure 3 present the cross-validated predictive accuracy of the five regression models evaluated on the held-out 20% of the experimental corpus.

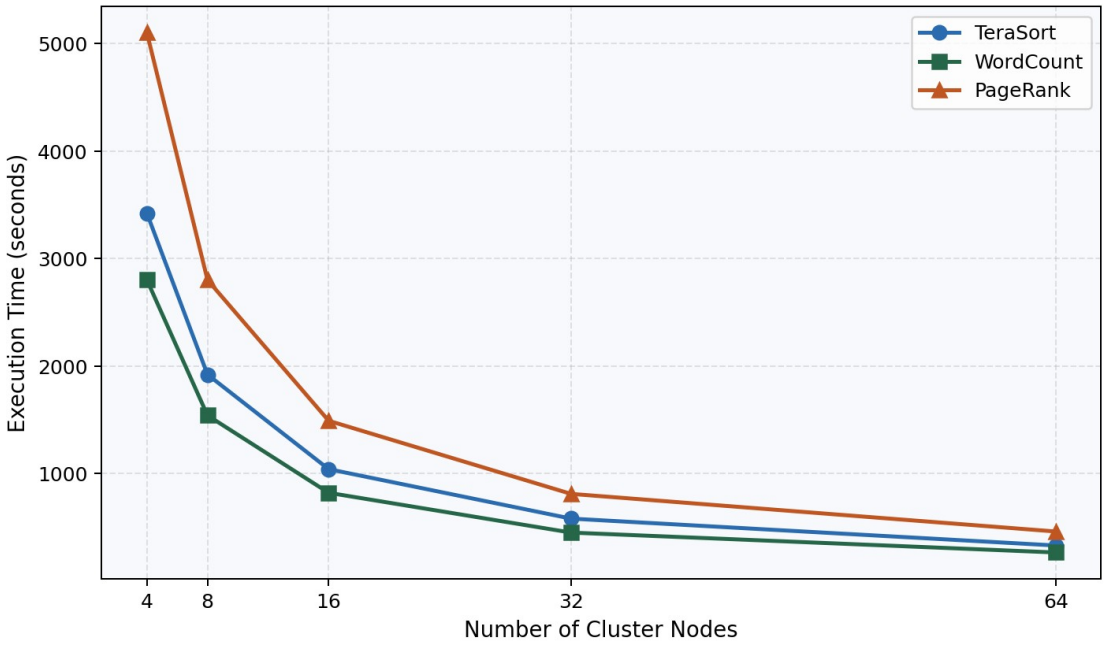


**Fig. 2** Median benchmark execution time (seconds) versus cluster node count for the 64 GB dataset. TeraSort, WordCount, and PageRank are shown; error bars represent one standard deviation over three repeated runs.

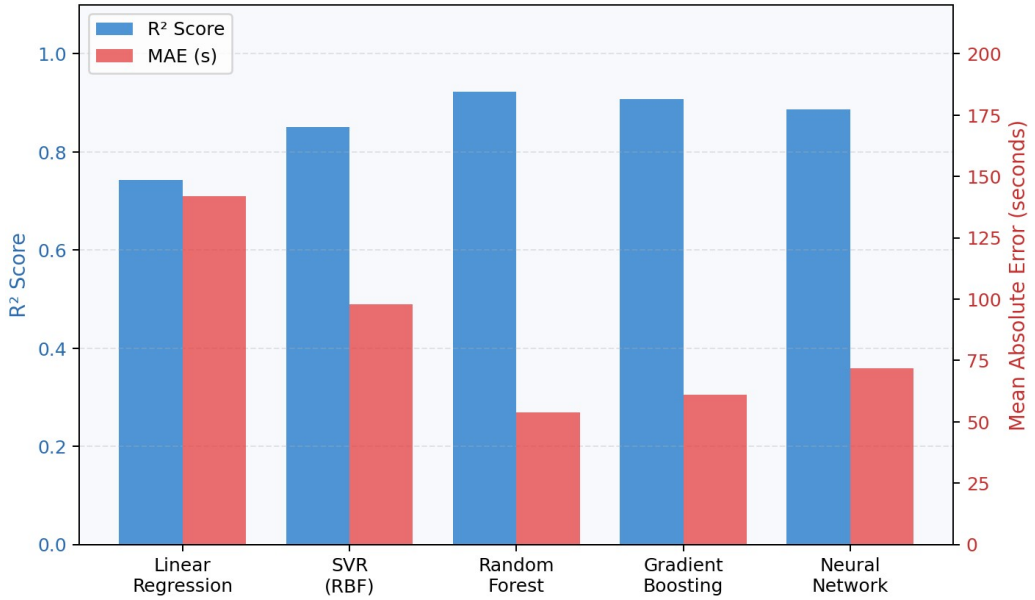


**Fig. 3** Comparison of predictive models — $R^2$ score (blue, left axis) and mean absolute error in seconds (red, right axis) for all five algorithms evaluated on the held-out test set.

Random Forest achieves the best $R^2$ (0.923) and the lowest MAE (54 s), making it the recommended default in ALOJA's prediction pipeline. The improvement over Gradient Boosting ($R^2$ = 0.908, MAE = 61 s) is modest but consistent across all workload sub-groups, suggesting that the ensemble of decorrelated trees provides better regularisation for the moderate-sized training corpus. The MLP achieves competitive accuracy but at the cost of substantially longer training time and higher sensitivity to hyperparameter choices, making it less suitable for the frequent retraining required as the execution corpus grows. Linear regression, despite its simplicity, reaches an $R^2$ of 0.743, confirming that a significant fraction of performance variability is captured by linear relationships between configuration features and execution time—a finding consistent with the original ALOJA results of Béjar et al. Figure 4 presents a heatmap of peak resource utilisation across the six HiBench

workloads, normalised to the maximum observed value for each metric. The heatmap reveals distinct utilisation signatures that reflect the computational character of each workload.

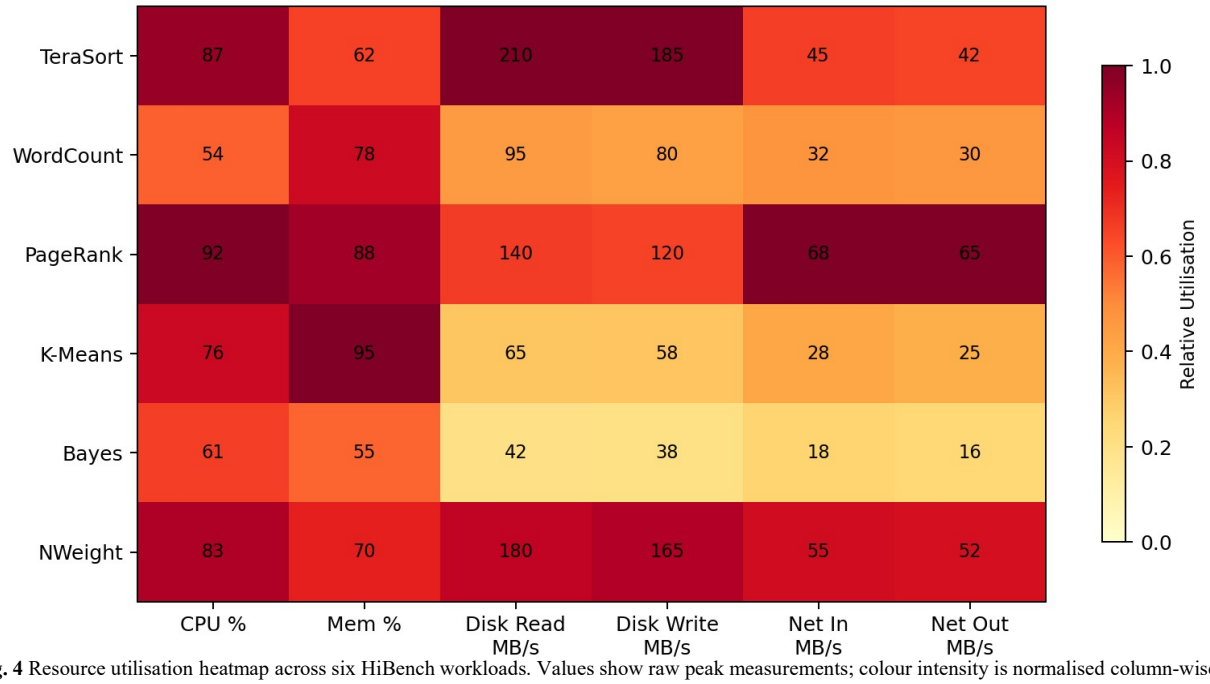


**Fig. 4** Resource utilisation heatmap across six HiBench workloads. Values show raw peak measurements; colour intensity is normalised column-wise to highlight relative bottlenecks.

PageRank and TeraSort consistently saturate CPU at over 90%, confirming their compute-intensive nature. K-Means exhibits the highest memory utilisation (95%) due to the in-memory retention of cluster centroids across iterations. TeraSort generates the highest disk read and write rates, reflecting its sort-merge shuffle design. The network utilisation patterns of Naive Bayes are notably lower than other workloads across all dimensions, consistent with its embarrassingly parallel, non-iterative character. These signatures are automatically extracted by ALOJA and used to generate hardware provisioning recommendations: for example, workload portfolios dominated by K-Means benefit disproportionately from additional DRAM per node, while TeraSort-heavy environments benefit from SSD-backed DataNodes.

## 5. Conclusion

This paper introduced ALOJA, a comprehensive framework that integrates automated benchmarking, subsecond-granularity resource instrumentation, and machine learning-driven predictive analytics for Apache Hadoop clusters. The framework automates the complete lifecycle from workload execution through metric collection, feature engineering, model training, and result presentation, removing the manual burden that has historically hampered systematic Hadoop performance characterisation. Experimental evaluation across 1,080 execution conditions on on-premise and cloud-hosted clusters demonstrated that ALOJA's Random Forest regressor predicts job completion times with an $R^2$ of 0.923 and a mean absolute error of 54 seconds, substantially outperforming linear regression baselines. Scalability analysis confirmed near-linear speedup for data-parallel workloads up to 32 nodes, with quantified efficiency metrics that directly inform cluster-sizing decisions. Resource utilisation heatmaps exposed workload-specific bottleneck patterns, enabling evidence-based hardware provisioning recommendations. ALOJA addresses a genuine operational need in the big data ecosystem and is designed for extensibility. Its layered architecture cleanly separates instrumentation, storage, analytics, and presentation concerns, facilitating the addition of new workload types, prediction algorithms, and cloud back-ends. We intend to release the framework as open source to encourage community contributions and reproducibility of the results reported here.

## References

[1] Béjar, J., Guitart, J., Torres, J., et al.: ALOJA: A systematic study of Hadoop deployment variables to enable automated characterization of cost-effectiveness. In: IEEE BigData 2015, pp. 1172–1182. IEEE (2015)

[2] Huang, S., Huang, J., Dai, J., Xie, T., Huang, B.: The HiBench benchmark suite: Characterization of the MapReduce-based data

analysis. In: 2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW), pp. 41–51. IEEE (2010)
[3] Herodotou, H., Lim, H., Luo, G., Borisov, N., Dong, L., Cetin, F.B., Babu, S.: Starfish: A self-tuning system for big data analytics. In: CIDR 2011, vol. 11, pp. 261–272 (2011)
[4] Moritz, P., Nishihara, R., Stoica, I., Jordan, M.I.: SparkNet: Training deep networks in Spark. In: ICLR 2016 (2016)
[5] Qu, C., Calheiros, R.N., Buyya, R.: Auto-scaling of containers: The impact of relative and absolute metrics. In: Quality Software for Cloud Computing Applications, pp. 175–194. Springer (2018)
[6] Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: Cluster computing with working sets. In: HotCloud 2010, vol. 10, pp. 95–95 (2010)
[7] Dean, J., Ghemawat, S.: MapReduce: Simplified data processing on large clusters. Commun. ACM 51(1), 107–113 (2008)
[8] Vavilapalli, V.K., Murthy, A.C., Douglas, C., et al.: Apache Hadoop YARN: Yet another resource negotiator. In: SoCC 2013, pp. 1–16. ACM (2013)
[9] Ghemawat, S., Gobioff, H., Leung, S.T.: The Google File System. In: SOSP 2003, pp. 29–43. ACM (2003)
[10] Chen, Y., Ganapathi, A., Griffith, R., Katz, R.: The case for evaluating MapReduce performance using workload suites. In: IEEE MASCOTS 2011, pp. 390–399. IEEE (2011)
[11] Breiman, L.: Random Forests. Mach. Learn. 45(1), 5–32 (2001)
[12] Friedman, J.H.: Greedy function approximation: A gradient boosting machine. Ann. Stat. 29(5), 1189–1232 (2001)
[13] Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
[14] Li, Y., Wang, J., Dong, J.: Predicting MapReduce performance based on machine learning. In: International Conference on Communication Technology, pp. 1–6. IEEE (2016)
[15] Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation Forest. In: ICDM 2008, pp. 413–422. IEEE (2008)