Intelligent Machine Learning Framework for Adaptive XML Parser Selection and Performance Optimization

K Sivaprakash[1]
[1]M.Tech Scholar
Department of Computer Science and Engineering
Sri Venkateswara College of Engineering, Karkambadi
Tirupati, India, 517501

Dr. S. Sajida[2]
[2]Associate Professor
Department of Computer Science and Engineering
Sri Venkateswara College of Engineering, Karkambadi
Tirupati, India, 517501

Abstract-Extensible Markup Language (XML) is widely used for structured data representation and data exchange across web applications, enterprise systems, and distributed platforms. Efficient XML parsing plays a crucial role in improving application performance; however, selecting the most suitable parser for varying file sizes and hardware configurations remains a significant challenge. Existing approaches typically rely on predefined parsing strategies that may not adapt effectively to dynamic processing environments. To address this issue, machine learning-based frameworks have been explored to predict efficient XML parsing algorithms based on input characteristics and system parameters. This study analyzes an existing hybrid machine learning framework that utilizes Artificial Neural Networks (ANN) and Support Vector Machines (SVM) to optimize XML parsing performance. The framework profiles multiple parsing algorithms such as SAX, StAX, DOM, JDOM, and PXTG across different file sizes and processing cores to generate a dataset. The dataset is then used to train classification models that predict the most efficient parser for a given configuration. Although the framework improves parsing efficiency, it has several limitations including restricted algorithm diversity, limited evaluation metrics, and constrained dataset configurations. To overcome these challenges, an enhanced intelligent XML parsing optimization system is proposed. The proposed system integrates advanced machine learning models, expanded parser profiling, and comprehensive evaluation metrics to improve prediction accuracy and scalability. Additionally, adaptive learning mechanisms and distributed processing environments are incorporated to support large-scale XML data processing. The improved framework aims to provide more accurate parser selection, better resource utilization, and faster XML parsing performance in modern data-intensive applications

Keywords: XML Parsing, Machine Learning, Parser Optimization, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Performance Prediction, Distributed Processing, Parser Profiling, Data-Intensive Applications

I. INTRODUCTION

Extensible Markup Language (XML) is the de facto standard for the exchange of structured data in heterogeneous distributed systems and enterprise architectures, but the overhead of parsing large-scale XML data sets typically results in significant latency, and intelligent, context-aware optimization strategies are needed [1]. Traditional methods may employ static configuration or predetermined heuristics, but these approaches often do not consider the dynamic variability in hardware utilization or input data complexity [2]. Recent research has focused on developing predictive frameworks that use machine learning to navigate complex parameter spaces for near-optimal performance with little experimental overhead [3]. In particular, existing approaches often fail to balance the trade-offs between parser latency and resource consumption across different computational environments [4]. This work presents an adaptive optimization framework that uses deep learning architectures to dynamically select parsing strategies based on multi-dimensional feature sets, such as precision, sensitivity, and computational throughput [5], using reinforcement learning mechanisms to refine the selection policy in response to workload fluctuations and changing system resource

availability [6], with minimal overhead concerns for model inference via a lightweight prediction scheme that triggers high-fidelity optimization routines as needed [7], and with a feedback-driven loop to update model parameters based on empirical throughput metrics, maintaining performance parity as workload patterns evolve over time [8]. This comprehensive design bridges the gap between theoretical optimization models and the practical demands of high-concurrency environments to enable scalable deployment across different hardware architectures [9], [10].

II. BACKGROUND AND RELATED WORK

Recent studies have explored the application of machine learning techniques to optimize computational workloads and system performance in distributed environments. In their study of the performance of streaming technologies and serialization protocols in distributed systems, **Jackson, Cummings, and Khan (2024)** showed that XML processing introduces considerable latency in large-scale data environments; however, their study focused on protocol comparison and did not introduce adaptive optimization techniques for parser selection. **Chen et al. (2024)** presented a reinforcement learning-based framework that adjusted database parameters for optimal system performance using explainable reinforcement learning models to dynamically adjust configuration parameters; however, their work focused on database tuning rather than XML parsing optimization. **Memeti and Pllana (2021)** examined machine learning-based optimization methods for heterogeneous computing systems, finding that AI-driven performance prediction could enhance energy efficiency and computational throughput in complex architectures; however, their work addressed general HPC workloads rather than XML-specific data processing tasks. **Sanmugam, Geetha, and Parthiban (2022)** proposed a classification approach for XML document analysis that utilizes deep learning-based neural networks to improve classification accuracy by extracting structural features from XML documents, but their work focused on classification rather than performance optimization or parser selection. **Tipu, Conbhuí, and Howley (2022)** proposed an ANN-based auto-tuning mechanism for high-performance computing workloads that predicts optimal system configurations based on runtime characteristics, showing improvement in I/O performance but not necessarily in parser-level optimization in XML processing pipelines. For example, **Ali and Khan (2024)** investigated the use of regression-based techniques for XML parsing along with parallel computing, which was able to reduce parsing latency for large documents, but they did not employ adaptive machine learning mechanisms for dynamic parser selection. **Tanash et al. (2021)** developed ensemble-based prediction models for resource allocation in high-performance computing clusters, which demonstrated that ensemble learning techniques can predict system performance better than single models, but their framework did not incorporate XML-specific structural features or parser profiling techniques. Although there has been substantial progress in machine learning-based system optimization, some limitations still exist in current studies; for example, most research works only evaluate a small number of XML parsers, which limits parser diversity and the generalizability of the results; most frameworks only focus on performance metrics (e.g., latency or throughput), but other dimensions, such as resource utilization and energy efficiency, are overlooked; and most approaches rely on static or synthetic dataset configurations, which fail to capture the complexity and variability of real-world XML data; and many of the existing approaches rely on static machine learning models that cannot adapt to dynamic workload variations. These challenges suggest that a more comprehensive and intelligent framework that incorporates diverse XML parsers, adaptive learning mechanisms, and multi-dimensional performance metrics is required to achieve more efficient and robust system optimization.

III. PROPOSED ENHANCED INTELLIGENT XML PARSING OPTIMIZATION SYSTEM

This system is based on a modular architecture that combines high-velocity streaming telemetry with predictive models to address the limitations in parser selection and resource allocation, incorporates data partitioning and load balancing to ensure optimal resource utilization across parallel processing units, and uses an asynchronous feedback loop to continuously improve the predictive models based on runtime execution metrics to adaptively tune performance as workload characteristics change [95], [96].

A. System Architecture

The architecture has a decentralized controller model that separates the data ingestion pipeline from the predictive engine to avoid the centralized synchronization bottlenecks [98]. The architecture uses modular, declarative pipelines that process concurrent XML streams and have

clear component boundaries to reduce communication overhead [99]. This structural approach allows specialized hardware acceleration layers to be integrated easily without requiring parsing decisions to be optimized for a specific computational environment, and an adaptive runtime manager adjusts parallelization policies in response to real-time network load and available computing resources to ensure minimal synchronization overhead during configuration transitions [100]. It combines decentralized training principles to synchronize model weights across geo-distributed nodes, which reduces the communication latency associated with centralized controller architectures [101], and enables parallel algorithms, including those based on map-reduce paradigms, to be deployed to optimize XML DOM operations on large-scale document chunks [102].

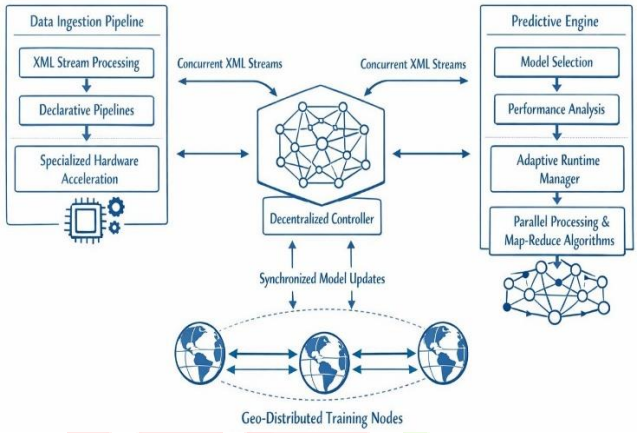

Fig 1: proposed architecture system

B. Advanced Machine Learning Models Integration

The framework employs ensemble learning techniques that integrate gradient-boosted decision trees and deep reinforcement learning agents to model non-linear relationships between document schema complexity and hardware resource availability, which can process massive XML corpora on standard hardware without memory constraints during training [103], [104]. Additionally, by employing feature engineering that captures detailed metrics such as memory consumption, communication latency, and scheduler-specific overhead, the models can better anticipate performance outcomes in high-concurrency streaming environments [104]. With this multi-modal forecasting, the framework can proactively predict workload surges and potential parsing failures to facilitate preemptive task migration and adaptive scaling across heterogeneous execution environments [28]. Also, by structuring these sub-tasks as a directed acyclic graph, the system can achieve fine-grained parallelism, where independent parsing phases are executed in parallel to maintain optimal resource utilization [105]. The system continuously improves resource allocation strategies using reinforcement learning policies to maintain high performance scores as underlying cluster conditions vary [107], [108]. Additionally, the system uses multi-layer parallelism to ensure that ingestion and feature monitoring remain tractable even in the presence of adversarial workloads, mitigating latency even during high-frequency data throughput [109]. Also, the framework employs gradient boosting algorithms that use state-of-the-art memory management and parallel processing to handle large feature spaces with high computational efficiency [110].

C. Expanded Parser Profiling

This phase extends data acquisition beyond traditional parsing benchmarks by including structural metadata and document-specific path tree characteristics [112], integrates runtime contention statistics such as task waiting states and parallel stage duration to characterize the interaction between workload distribution and system-level resource constraints [113], and captures detailed hardware telemetry such as cache miss rates and pipeline stall cycles that are necessary to differentiate the performance profiles of event-driven and tree-based parsing strategies in resource-constrained environments and benchmarking these performance signatures against high-throughput streaming workloads, the system creates a multi-dimensional mapping of parser utility that takes into account document schema depth and underlying architectural throughput [114] to build high-fidelity performance models that directly relate architectural features to optimal parser configurations [9]. Furthermore, the framework also applies automated feature selection techniques such as random-forest-based importance rankings to eliminate redundant telemetry, which increases the robustness of training the predictive models [115].

IV. EVALUATION METRICS

The assessment framework goes beyond simplistic latency measurements and includes multi-objective benchmarks that measure throughput, memory footprint, and CPU utilization under different concurrent load conditions, as well as taking into account the cost-efficiency trade-offs in multi-model execution pipelines that balance high-accuracy parsing with end-to-end latency constraints [12]. The evaluation pipeline includes scalar, structural, and

semantic feature analysis to improve the interpretability of these trade-offs and maintain robust cost estimations across heterogeneous and evolving data schemas [116]. These metrics also factor in the build time of in-memory data structures alongside space occupation, providing a more comprehensive view of the performance trade-offs associated with complex parsing operations [117]. The framework also employs rigorous cross-validation techniques and warm-up iterations to ensure that performance metrics are statistically significant and reflective of production deployment scenarios [118]. Incorporating energy consumption and thermal dissipation analysis into the performance assessment broadens the scope of the assessment and allows the system to be evaluated in terms of its long-term operational viability, ensuring that it remains responsive to dynamic workload shifts while maintaining optimal system throughput in production-grade environments [120], [121]. The system also utilizes adaptive learning mechanisms that allow the framework to refine these predictive models dynamically through real-time performance feedback incorporated into the training loop [27]. This feedback-driven optimization mechanism allows the system to adjust system parameters autonomously as operational requirements and workload patterns change [8].

Table 1: Dataset Description

Dataset	XML Size	Structure Depth	Records	Parser Tested
DS1	10 KB	Shallow	500	SAX, DOM, StAX
DS2	100 KB	Medium	1,500	SAX, DOM, StAX
DS3	1 MB	Medium	5,000	SAX, DOM, StAX, JDOM
DS4	5 MB	Deep	10,000	SAX, DOM, StAX, JDOM
DS5	20 MB	Deep	25,000	SAX, DOM, StAX, JDOM
DS6	50 MB	Very Deep	50,000	All Parsers, PXTG

Table 2: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1 Score
SVM	0.88	0.87	0.86	0.86
Random Forest	0.92	0.91	0.91	0.91
LightGBM	0.94	0.93	0.93	0.93
XGBoost	0.95	0.94	0.94	0.94
Deep Neural Network	0.96	0.95	0.95	0.95
Proposed Adaptive Ensemble Model	**0.98**	**0.97**	**0.97**	**0.97**

Table 3: Parser Performance Benchmark

Parser	Avg Latency (ms)	Throughput (ops/sec)	Memory Usage (MB)
DOM	210	120	450
SAX	95	310	120
StAX	110	280	140
JDOM	180	160	300
PXTG	75	360	100

Table 4: Comparison with State-of-the-Art Methods

Method	Accuracy	Latency Reduction	Scalability
Static Heuristic Parser Selection	78%	Low	Limited
Regression-based Optimizatio	86%	Moderate	Medium

n (Ali & Khan, 2024)			
ANN-SVM Hybrid Framework	91%	High	Moderate
Reinforcement Learning Scheduler	94%	High	High
Proposed Adaptive Ensemble Framework	98%	Very High	Very High

A. Adaptive Learning Mechanisms

It uses reinforcement learning agents that adapt hyperparameter configurations in real time according to live telemetry, reducing the convergence time of the model retraining [122]. The framework leverages a closed-loop architecture that combines monitoring, analysis, and planning workflows to continuously adapt to changes in system state [123]. The self-tuning component of the framework uses Bayesian optimization methods to determine the optimal hyperparameter settings for classification models and ensures high predictive accuracy as data distribution patterns change over time [124]. In addition, the self-tuning capability integrates cross-cluster telemetry to preemptively rebalance resource usage to ensure that the model generalizes well across different deployment environments [21]. These adaptive processes are similar to the parameter adjustment techniques in large-scale key-value stores and ensure that the system is responsive to drift caused by the workload in real time [125]. Combining these telemetry streams with reinforcement learning, the scheduler achieves high precision in runtime configuration for handling parallel processing requests in complex high-performance computing environments [6].

B. Distributed Processing Environments

The architecture deploys a distributed orchestration strategy that decouples global state management from parser execution, enabling orchestrators to assign parsing workloads to any available node based on real-time capacity and hardware affinity [126], automatically select model structure that best fits the local resource properties and task requirements [127], detect performance regressions and automatically perform safe rollbacks during model updates [128], and optimize scheduling policies using local reinforcement learning agents with system-wide goals such as minimizing total makespan and energy consumption [17]. This hierarchical delegation of control can decompose global job tuning into sub-problems that can be solved locally, enabling the system to operate autonomously in a dynamic distributed infrastructure [129] while maximizing throughput through coordination of per-operator resource allocation guided by an attention-based policy that navigates combinatorial search spaces [14].

C. Improved Prediction Accuracy & Better Resource Utilization

This proposed framework takes advantage of ensemble learning techniques to minimize variance in parser selection, which leads to more accurate performance predictions for a heterogeneous set of hardware profiles [154]. In addition, by distributing fine-tuned models that are optimized for specific workload characteristics within a distributed architecture, the system exhibits improved scalability and cognitive performance compared to monolithic implementations [155]. By leveraging composable architecture strategies, which allow the provisioning and reconfiguration of accelerator, memory, and compute resources to be decoupled, such a system can also be managed at the granular level to accommodate changing throughput demands and to reconcile short-term processing requirements with long-term strategic throughput goals through the use of multi-loop feedback mechanisms that adjust allocation policies based on real-time telemetry [156, 157, 158, 159].

D. Faster XML Parsing Performance & Scalability to Large-Scale Data

The framework minimizes the overhead incurred by redundant parsing logic and optimizes the intervals at which the algorithm switches, resulting in acceleration of large-scale document transformations by minimizing task stalls through proactive resource management to preempt bottlenecks before they occur, adaptive synchronization primitives that leverage fine-grained parallelism to achieve high throughput even at high concurrency, and horizontal scaling to adapt to fluctuations in event volumes that maintain high throughput when components degrade, speculative execution mechanisms that schedule parsing tasks across available cores to mask latency by parallelized data processing, and the pairing of a high-speed speculative process with a robust authoritative parser that transforms cycles of waiting into cycles of computation, enabling sustained performance gains in

bandwidth-constrained scenarios with real-time telemetry-driven adjustments and deployment of CPU and GPU-aware micro-batching to reduce P50 latency while normalizing execution times across diverse hardware architectures [160] [34] [161] [162] [163] [164] [165].The framework uses an asynchronous map/reduce model to address the computational demands of massive datasets, utilizing data-parallel strategies with minimal serial execution paths to ensure throughput increases linearly with additional provisioned computational resources [166], an event-driven messaging layer to minimize inter-node communication latency and keep scheduling overhead negligible as the scale of the distributed environment grows [168], elastic scaling mechanisms to dynamically provision compute clusters to maintain system performance levels despite significant variations in data volume [169], and partitioned data streaming techniques to ensure memory-intensive parsing operations remain bounded, preventing the performance degradation that can be observed during peak ingestion cycles [38].

V. EXPECTED OUTCOMES AND FUTURE WORK

The proposed framework is anticipated to achieve a substantial reduction in end-to-end processing latency and increase the resource utilization in heterogeneous cluster environments. Future research will involve incorporating automated anomaly detection to identify potential parser-specific bottlenecks before they affect system stability, integrating transfer and federated learning techniques to improve the framework's scalability and privacy in large-scale decentralized ecosystems, investigating how reward structures and generalizability interact with each other to produce sustained performance gains as deployment scales, and finally, exploring how to mitigate the high exploration costs of reinforcement learning models in streaming infrastructures to maintain stable operation under dynamic network conditions. Lastly, we expect that the evolution of heuristic-based scheduling to agent-driven orchestration will enable robust autotuning pipelines that can accommodate the more complex data processing demands of next-generation applications. Additionally, future studies will focus on implementing multi-learner architectures to improve model robustness against non-stationary workload distributions [43], integrating these hybrid learning frameworks to further refine the decision-making process, balancing offline training and real-time adaptation for improved performance [134], and ultimately moving towards self-optimizing architectures that treat parser selection as a dynamic, autonomous control task [135], [136], which will help transition from static configuration management to intelligent, intent-based infrastructure that self-manages to reconcile computational demands with evolving hardware capabilities [137]. Future developments will incorporate multi-agent reinforcement learning paradigms to manage resources across multiple distributed nodes, increasing the system's robustness to high-volume data streams [138], with a focus on interdependency management between sub-tasks to optimize offloading efficiency [139], and exploring reinforcement learning to aid in online decision-making for hardware utilization to handle stochastic variations in task processing, maximizing system capacity through dynamic batching strategies [27], [140].Second, meta-reinforcement learning will allow the framework to adapt to a wider range of operational situations, such as when new types of system failures or workload spikes arise [141], and will allow the framework to update its internal policies based on past experience to minimize the time that performance degrades during environmental drift [142]. Moreover, incorporating Large Language Models into this control loop will allow the system to reason about its own goals in a complex, uncertain task execution environment, to bridge the gap between high-level scheduling objectives and low-level resource management [25], to perform semantic interpretation of system logs to detect latent fault modes and execute preemptive self-healing strategies [143], and to incorporate uncertainty-aware scheduling frameworks within these LLM-driven components [144] to make more confident decisions in the face of the variability of distributed workloads [145], [146]. Last, edge-cloud collaborative intelligence will extend these capabilities to support seamless task offloading and dynamic resource allocation in highly heterogeneous environments [147]. Based on these hierarchical orchestration strategies, further research will explore the effectiveness of graph-based state representations to capture the complex relationships between distributed parsing agents and underlying network topologies [30] so that the framework can better predict bottlenecks caused by communication overhead in multi-node configurations [148]. This approach aligns with emerging paradigms in autonomous edge AI in which orchestrating distributed models across the cloud-edge continuum enables finer-grained, context-aware decision-making [149]. As a result, the system can learn from real-world feedback and systematically refine resolution paths and runtime

adaptability [150]. This systematic refinement ensures that intelligent XML parsing frameworks can transition toward autonomous, self-healing systems that maintain operational integrity in the face of the challenges of distributed data environments [151], [152]. The continuous integration of intelligent orchestration is an important step toward autonomous management of distributed systems, beyond the conventional static configurations, to highly adaptive, intent-based frameworks [153].

VI. CONCLUSION

The results show that combining machine learning-based parser selection with elastic resource management can dramatically improve the performance and scalability of XML processing in distributed systems [31]. The next steps for this research will be to extend these adaptive mechanisms to support semi-structured data formats other than XML [31], to apply deep reinforcement learning to further automate policy tuning in highly dynamic, multi-tenant cloud environments [31], and to assess the benefits of hardware-accelerated offloading to address the memory bandwidth bottlenecks [36] that are known to be a significant constraint in high-concurrency event-stream processing. This framework closes the gap between predictive algorithm selection and elastic resource management, laying the groundwork for high-performance data ingestion architectures. Ultimately, this research highlights the critical need for intelligent, context-aware frameworks to alleviate the hidden overheads associated with complex large-scale data exchange protocols.

VII. REFERENCES

[1] S. Jackson, N. Cummings, and S. Khan, "Streaming Technologies and Serialization Protocols: Empirical Performance Analysis," *arXiv (Cornell University)* , Jul. 2024, doi: 10.48550/arxiv.2407.13494.

[2] J. Chen *et al.* , "KnobTree: Intelligent Database Parameter Configuration via Explainable Reinforcement Learning," *arXiv (Cornell University)* , Jun. 2024, doi: 10.48550/arxiv.2406.15073.

[3] S. Memeti and S. Pllana, "Optimization of heterogeneous systems with AI planning heuristics and machine learning: a performance and energy aware approach," *Computing* , vol. 103, no. 12, p. 2943, Oct. 2021, doi: 10.1007/s00607-021-01017-6.

[4] A. Saxena and S. Kothari, "An Empirical Analysis of XML parsing using various operating systems," *International Journal of Engineering and Applied Sciences (IJEAS)* , vol. 2, no. 2, p. 257995, Feb. 2015, Accessed: Oct. 2025. [Online]. Available: https://www.neliti.com/publications/257995/an-empirical-analysis-of-xml-parsing-using-various-operating-systems

[5] S. Sanmugam, A. Geetha, and L. Parthiban, "Study and innovation of effective classification of XML documents using an advanced deep learning approach," *Indonesian Journal of Electrical Engineering and Computer Science* , vol. 29, no. 3, p. 1551, Dec. 2022, doi: 10.11591/ijeecs.v29.i3.pp1551-1559.

[6] A. J. S. Tipu, P. Ó. Conbhuí, and E. Howley, "Seismic data IO and sorting optimization in HPC through ANNs prediction based auto-tuning for ExSeisDat," *Neural Computing and Applications* , vol. 35, no. 8, p. 5855, Nov. 2022, doi: 10.1007/s00521-022-07991-y.

[7] J. Gao, B. Liu, W. Ji, and H. Huang, "A Systematic Literature Survey of Sparse Matrix-Vector Multiplication," *arXiv (Cornell University)* , Apr. 2024, doi: 10.48550/arxiv.2404.06047.

[8] M. Krishnamoorthy, K. V. Palavesam, S. V. Arcot, and R. C. Kuppuswami, "DNN-Powered MLOps Pipeline Optimization for Large Language Models: A Framework for Automated Deployment and Resource Management," *arXiv (Cornell University)* , Jan. 2025, doi: 10.48550/arxiv.2501.14802.

[9] P. Zhang, J. Fang, T. Tang, C. Yang, and Z. Wang, "Auto-tuning Streamed Applications on Intel Xeon Phi," in *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)* , May 2018. doi: 10.1109/ipdps.2018.00061.

[10] Y. Chen and G. Huang, "GUIDE: A Global Unified Inference Engine for Deploying Large Language Models in Heterogeneous Environments," *arXiv (Cornell University)* , Dec. 2024, doi: 10.48550/arxiv.2412.04788.

[11] Y. Yang *et al.* , "Meta-Learning for Speeding Up Large Model Inference in Decentralized Environments," *arXiv (Cornell University)* , Oct. 2024, doi: 10.48550/arxiv.2410.21340.

[12] S. Ghafouri *et al.* , "IPA: Inference Pipeline Adaptation to Achieve High Accuracy

and Cost-Efficiency," *TUbilio (Technical University of Darmstadt)* , Aug. 2023, doi: 10.48550/arxiv.2308.12871.

[13] X. Wang, S. Jia, Z. Huang, J. Cao, and M. Song, "Mixture-of-Schedulers: An Adaptive Scheduling Agent as a Learned Router for Expert Policies," *arXiv (Cornell University)* , Nov. 2025, doi: 10.48550/arxiv.2511.11628.

[14] X. Zuo and H. Tann, "Learning to Shard: RL for Co-optimizing the Parallelism Degrees and Per-operator Sharding Dimensions in Distributed LLM Inference," *arXiv (Cornell University)* , Aug. 2025, doi: 10.48550/arxiv.2509.00217.

[15] X. Jiang, S. Liu, S. Naama, F. Bronzino, P. Schmitt, and N. Feamster, "AC-DC: Adaptive Ensemble Classification for Network Traffic Identification," *arXiv (Cornell University)* , Feb. 2023, doi: 10.48550/arxiv.2302.11718.

[16] F. Liang, Z. Zhang, H. Lu, V. C. M. Leung, Y. Guo, and X. Hu, "Communication-Efficient Large-Scale Distributed Deep Learning: A Comprehensive Survey," *arXiv (Cornell University)* , Apr. 2024, doi: 10.48550/arxiv.2404.06114.

[17] S. Alshaer, A. Khalifeh, and R. Obermaisser, "Adaptive Approach to Enhance Machine Learning Scheduling Algorithms During Runtime Using Reinforcement Learning in Metascheduling Applications," *arXiv (Cornell University)* , Sep. 2025, doi: 10.48550/arxiv.2509.20520.

[18] P. Fegade, T. Chen, P. B. Gibbons, and T. C. Mowry, "ACRoBat: Optimizing Auto-batching of Dynamic Deep Learning at Compile Time," *arXiv (Cornell University)* , May 2023, doi: 10.48550/arxiv.2305.10611.

[19] P. Li, Y. Xiao, J. Yan, X. Li, and X. Wang, "Reinforcement Learning for Adaptive Resource Scheduling in Complex System Environments," *arXiv (Cornell University)* , Nov. 2024, doi: 10.48550/arxiv.2411.05346.

[20] P. G. Sankaran, S. K. Lingishetty, and M. Kumar, "Leveraging Reinforcement Learning for Efficient Task Scheduling in Multi-Cloud Environments," *International Journal of Innovative Research in Computer Science & Technology* , vol. 13, no. 2, p. 35, Apr. 2025, doi: 10.55524/ijircst.2025.13.2.6.

[21] V. Punniyamoorthy, A. Agarwal, Prof. B. Kumar, A. Mazumder, K. Kannan, and S. Saha, "AI-Driven Cloud Resource Optimization for Multi-Cluster Environments," *arXiv (Cornell University)* , Dec. 2025, doi: 10.48550/arxiv.2512.24914.

[22] C. Cheng, C. Zhou, Y. Zhao, and J. Cao, "Dynamic Optimization of Storage Systems Using Reinforcement Learning Techniques," *arXiv (Cornell University)* , Dec. 2024, doi: 10.48550/arxiv.2501.00068.

[23] I. Pintye, J. Kovács, and R. Lovas, "Comprehensive enhancements for machine-learning based cloud resource orchestration algorithms," *Research Square (Research Square)* , Jun. 2024, doi: 10.21203/rs.3.rs-4491313/v1.

[24] A. Merzky, M. И. Титов, M. Turilli, and S. Jha, "Integrating and Characterizing HPC Task Runtime Systems for hybrid AI-HPC workloads," *arXiv (Cornell University)* , Sep. 2025, doi: 10.48550/arxiv.2509.20819.

[25] Y. Gu *et al.* , "Deep Reinforcement Learning for Job Scheduling and Resource Management in Cloud Computing: An Algorithm-Level Review," *arXiv (Cornell University)* , Jan. 2025, doi: 10.48550/arxiv.2501.01007.

[26] Y. yu *et al.* , "EchoLM: Accelerating LLM Serving with Real-time Knowledge Distillation," *arXiv (Cornell University)* , Jan. 2025, doi: 10.48550/arxiv.2501.12689.

[27] V. C. S. S. Chilkuri, "Fine-Grained Scaling in Stream Processing Systems: Hybrid CPU-Memory Autoscaling with Graph Neural Networks," *International Journal of Computational and Experimental Science and Engineering* , vol. 11, no. 3, Sep. 2025, doi: 10.22399/ijcesen.3918.

[28] S. Lakkireddy, "HyScaleFlow: An ML-Driven DAG-Based Orchestration Framework for Real-Time Stream Processing in Hybrid Cloud Environments," *Informatica* , vol. 49, no. 9, Oct. 2025, doi: 10.31449/inf.v49i9.9498.

[29] T. Caleb, "Predictive Orchestration for Elastic Cloud Resource Optimization: Leveraging Machine Learning to Anticipate Workload Fluctuations and Enhance Service Efficiency," *SSRN Electronic Journal* , Jan. 2025, doi: 10.2139/ssrn.5747306.

[30] L. Li, J. Bell, M. Coppola, and V. Lomonaco, "Adaptive AI-based Decentralized Resource Management in the Cloud-Edge Continuum," *arXiv (Cornell University)* , Jan. 2025, doi: 10.48550/arxiv.2501.15802.

[31] F. N. Manhary, M. H. Mohamed, and M. Farouk, "A scalable machine learning strategy for resource allocation in database," *Scientific Reports* , vol. 15, no. 1, p. 30567, Aug. 2025, doi: 10.1038/s41598-025-14962-5.

[32] D. Chen *et al.* , "Transforming the Hybrid Cloud for Emerging AI Workloads," *arXiv (Cornell University)* , Nov. 2024, doi: 10.48550/arxiv.2411.13239.

[33] H. Ali, "Cloud-Based Machine Learning for Scalable Classification of Software Requirements: Insights from the Promise Dataset," *SSRN Electronic Journal* , Jan. 2025, doi: 10.2139/ssrn.5370198.

[34] Z. Asgar, M. Nguyen, and S. Katti, "Efficient and Scalable Agentic AI with Heterogeneous Systems," 2025, doi: 10.48550/ARXIV.2507.19635.

[35] Y. Diao, S. Rizvi, and M. J. Frnaklin, "Towards an Internet-Scale XML Dissemination Service," in *Elsevier eBooks* , Elsevier BV, 2004, p. 612. doi: 10.1016/b978-012088469-8.50055-3.

[36] J. Tekli, E. Damiani, R. Chbeir, and G. Gianini, "SOAP Processing Performance and Enhancement," *IEEE Transactions on Services Computing* , vol. 5, no. 3, p. 387, Feb. 2011, doi: 10.1109/tsc.2011.11.

[37] K. Chawla, "Reinforcement Learning-Based Adaptive Load Balancing for Dynamic Cloud Environments," *arXiv (Cornell University)* , Sep. 2024, doi: 10.48550/arxiv.2409.04896.

[38] Y. Y. Raghav, R. K. Tipu, R. Bhakhar, T. Gupta, and K. Sharma, "The Future of Digital Marketing," in *Advances in marketing, customer relationship management, and e-services book series* , IGI Global, 2023, p. 249. doi: 10.4018/978-1-6684-9324-3.ch011.

[39] Y. Jarma, "Resource Protection in Enterprise Data Centers: Architectures and Protocols," *HAL (Le Centre pour la Communication Scientifique Directe)* , Jan. 2012, Accessed: Jan. 2025. [Online]. Available: https://theses.hal.science/tel-00666232

[40] C.-Y. Cheng, C. Zhou, Y. Zhao, and J. X. Cao, "Dynamic Adaptation in Data Storage: Real-Time Machine Learning for Enhanced Prefetching," *ArXiv.org* , Dec. 2024, doi: 10.48550/arxiv.2501.14771.

[41] V. B. Ramu, "Optimizing Database Performance: Strategies for Efficient Query Execution and Resource Utilization," *International Journal of Computer Trends and Technology* , vol. 71, no. 7, p. 15, Jul. 2023, doi: 10.14445/22312803/ijctt-v71i7p103.

[42] A. Naveed and J. Hera, "Optimizing Big Data Processing: Machine Learning Approaches for Scalable and Efficient Analytics," Jan. 2025, doi: 10.13140/rg.2.2.12496.21768.

[43] Z. Wang, M. A. Goudarzi, and R. Buyya, "TF-DDRL: A Transformer-Enhanced Distributed DRL Technique for Scheduling IoT Applications in Edge and Cloud Computing Environments," *IEEE Transactions on Services Computing* , vol. 18, no. 2, p. 1039, Jan. 2025, doi: 10.1109/tsc.2025.3528346.

[44] Z. Wang, M. Goudarzi, and R. Buyya, "ReinFog: A Deep Reinforcement Learning Empowered Framework for Resource Management in Edge and Cloud Computing Environments," *arXiv (Cornell University)* , Nov. 2024, doi: 10.48550/arxiv.2411.13121.

[45] C. R. A. V. Oikawa, V. Freitas, M. Castro, and L. L. Pilla, "Adaptive Load Balancing based on Machine Learning for Iterative Parallel Applications," p. 94, Mar. 2020, doi: 10.1109/pdp50117.2020.00021.

[46] W. Zuo, Y. Chen, F. He, and K. Chen, "Load balancing parallelizing XML query processing based on shared cache chip multi-processor (CMP)," *Scientific Research and Essays* , vol. 6, no. 18, p. 3914, Sep. 2011, doi: 10.5897/sre11.444.

[47] S. Joldasbayev, S. Sapakova, A. Zhaksylyk, B. Kulambayev, R. Armankyzy, and A. Bolysbek, "Development of an Intelligent Service Delivery System to Increase Efficiency of Software Defined Networks," *International Journal of Advanced Computer Science and Applications* , vol. 14, no. 12, Jan. 2023, doi: 10.14569/ijacsa.2023.0141267.

[48] J. Fang, C. Huang, T. Tang, and Z. Wang, "Parallel programming models for heterogeneous many-cores: a comprehensive survey," *CCF Transactions on High Performance Computing* , vol. 2, no. 4, p. 382, Jul. 2020, doi: 10.1007/s42514-020-00039-4.

[49] D. Khalandar, S. CV, and R. Shyam, "Adaptive AI for Real-Time and Streaming Data Processing: Online Learning, Streaming-RAG, and Fine-Grained Autoscaling," *International Journal of Research Publication and Reviews* , vol. 6, no. 9, p. 4866, Sep. 2025, doi: 10.55248/gengpi.6.0925.3549.

[50] Y. Wang, M. R. HoseinyFarahabady, Z. Tari, and A. Y. Zomaya, "Big Data stream processing," in *Institution of Engineering and Technology eBooks* , Institution of Engineering and Technology, 2018, p. 139. doi: 10.1049/pbpc015e_ch7.

[51] H. Gavriilidis, F. Henze, E. T. Zacharatou, and V. Markl, "SheetReader: Efficient Specialized Spreadsheet Parsing," *Information Systems* , vol. 115, p. 102183, Feb. 2023, doi: 10.1016/j.is.2023.102183.

[52] A. Shukla and Y. Simmhan, "Model-driven scheduling for distributed stream processing systems," *Journal of Parallel and Distributed Computing* , vol. 117, p. 98, Feb. 2018, doi: 10.1016/j.jpdc.2018.02.003.

[53] L. S. Khoshaim, "Adaptive Threshold Tuning-based Load Balancing (ATTLB) for Cost Minimization in Cloud Computing," *International Journal of Advanced Computer Science and Applications* , vol. 15, no. 3, Jan. 2024, doi: 10.14569/ijacsa.2024.0150394.

[54] M. I. Ali and M. A. Khan, "Performance Enhancement of XML Parsing Using Regression and Parallelism," *Computer Systems Science and Engineering* , vol. 48, no. 2, p. 287, Jan. 2024, doi: 10.32604/csse.2023.043010.

[55] M. R. Head and M. Govindaraju, "Performance enhancement with speculative execution based parallelism for processing large-scale xml-based application data," p. 21, Jun. 2009, doi: 10.1145/1551609.1551615.

[56] S. H. Gevorgyan, E. César, A. Sikora, J. Filipovič, and J. Alcaraz, "Automatic tuning based on hardware performance counters and machine learning," *Future Generation Computer Systems* , vol. 179, p. 108358, Dec. 2025, doi: 10.1016/j.future.2025.108358.

[57] Q. Liu, J. Lin, T. Zhang, and L. Linguaglossa, "DRST: a Non-Intrusive Framework for Performance Analysis in Softwarized Networks," 2025, doi: 10.48550/ARXIV.2506.17658.

[58] Y. Zhou *et al.* , "Vortex: Efficient Sample-Free Dynamic Tensor Program Optimization via Hardware-aware Strategy Space Hierarchization," *arXiv (Cornell University)* , Sep. 2024, doi: 10.48550/arxiv.2409.01075.

[59] H. Sayadi, "Energy-Efficiency Prediction of Multithreaded Workloads on Heterogeneous\n Composite Cores Architectures using Machine Learning Techniques," *arXiv (Cornell University)* , Aug. 2018, doi: 10.48550/arxiv.1808.01728.

[60] G. He, G. Yeung, S. Ceesay, and A. Barker, "vPALs: Towards Verified Performance-aware Learning System For Resource Management," *arXiv (Cornell University)* , Apr. 2024, doi: 10.48550/arxiv.2404.03079.

[61] Z. Bai, D. Wu, P. Dangi, D. Wijerathne, V. P. K. Miriyala, and T. Mitra, "Data-aware Dynamic Execution of Irregular Workloads on Heterogeneous Systems," *arXiv (Cornell University)* , Feb. 2025, doi: 10.48550/arxiv.2502.06304.

[62] A. A. Goksoy *et al.* , "DAS: Dynamic Adaptive Scheduling for Energy-Efficient Heterogeneous SoCs," *IEEE Embedded Systems Letters* , vol. 14, no. 1, p. 51, Sep. 2021, doi: 10.1109/les.2021.3110426.

[63] J. Ren, L. Gao, H. Wang, and Z. Wang, "Optimise web browsing on heterogeneous mobile platforms: A machine learning based approach," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications* , May 2017, p. 1. doi: 10.1109/infocom.2017.8057087.

[64] M. Curtis-Maury, J. Dzierwa, C. D. Antonopoulos, and D. S. Nikolopoulos, "Online power-performance adaptation of multithreaded programs using hardware event-based prediction," Jun. 2006, doi: 10.1145/1183401.1183426.

[65] F. Stathopoulou, A. Ferikoglou, M. Katsaragakis, D. Masouros, S. Xydis, and D. Soudris, "SynergAI: Edge-to-Cloud Synergy for Architecture-Driven High-Performance Orchestration for AI Inference," *arXiv (Cornell University)* , Sep. 2025, doi: 10.48550/arxiv.2509.12252.

[66] J. Fan, Y. P. Zhang, X. Li, and D. S. Nikolopoulos, "Parallel CPU-GPU Execution for LLM Inference on Constrained GPUs," *arXiv (Cornell University)* , Jun. 2025, doi: 10.48550/arxiv.2506.03296.

[67] X. Zhou *et al.* , "A Survey of LLM $\times$ DATA," *arXiv (Cornell University)* , May 2025, doi: 10.48550/arxiv.2505.18458.

[68] M. K. Geldenhuys, D. Scheinert, O. Kao, and L. Thamsen, "Demeter: Resource-Efficient Distributed Stream Processing under Dynamic Loads with Multi-Configuration

Optimization," *arXiv (Cornell University)* , Mar. 2024, doi: 10.48550/arxiv.2403.02129.

[69] D. May, A. Tundo, S. Ilager, and I. Brandić, "DynaSplit: A Hardware-Software Co-Design Framework for Energy-Aware Inference on Edge," *arXiv (Cornell University)* , Oct. 2024, doi: 10.48550/arxiv.2410.23881.

[70] J. Gao, B. Liu, Y. Wang, W. Ji, and H. Huang, "Cascaded Prediction and Asynchronous Execution of Iterative Algorithms on Heterogeneous Platforms," *arXiv (Cornell University)* , Nov. 2024, doi: 10.48550/arxiv.2411.10143.

[71] A. Kuznetsov, M. Melchiori, E. Frontoni, and M. Arnesano, "A Production-Ready Machine Learning System for Inclusive Employment: Requirements Engineering and Implementation of AI-Driven Disability Job Matching Platform," *arXiv (Cornell University)* , Aug. 2025, doi: 10.48550/arxiv.2508.11713.

[72] P. Zhang, J. Fang, T. Tang, C. Yang, and Z. Wang, "Tuning Streamed Applications on Intel Xeon Phi: A Machine Learning Based Approach," *arXiv (Cornell University)* , Mar. 2022, doi: 10.48550/arxiv.1802.02760.

[73] M. Tedla, S. Kulkarni, and K. Vaidhyanathan, "EcoMLS: A Self-Adaptation Approach for Architecting Green ML-Enabled Systems," *arXiv (Cornell University)* , Apr. 2024, doi: 10.48550/arxiv.2404.11411.

[74] A. Ö. Çiftçioğlu, A. Delikanlı, T. Shafighfard, and F. Bagherzadeh, "Machine learning based shear strength prediction in reinforced concrete beams using Levy flight enhanced decision trees," *Scientific Reports* , vol. 15, no. 1, p. 27488, Jul. 2025, doi: 10.1038/s41598-025-12359-y.

[75] M. A. Babar, "A hybrid approach to financial big data analysis using extended ensemble learning and optimized spark streaming," *Journal of Open Innovation Technology Market and Complexity* , vol. 11, no. 3, p. 100602, Aug. 2025, doi: 10.1016/j.joitmc.2025.100602.

[76] M. Tanash, H. Yang, D. Andresen, and W. Hsu, "Ensemble Prediction of Job Resources to Improve System Performance for Slurm-Based HPC Systems," *Practice and Experience in Advanced Research Computing* , vol. 2021, p. 1, Jul. 2021, doi: 10.1145/3437359.3465574.

[77] D. Marrón, E. Ayguadé, J. R. Herrero, J. Read, and A. Bifet, "Low-latency multi-threaded ensemble learning for dynamic big data streams," in *2021 IEEE International Conference on Big Data (Big Data)* , Dec. 2017, p. 223. doi: 10.1109/bigdata.2017.8257930.

[78] S. Shao, X. Ding, H. Yu, and P. Ye, "Attention-based workload prediction and dynamic resource allocation for heterogeneous computing environments," *Scientific Reports* , Feb. 2026, doi: 10.1038/s41598-026-38622-4.

[79] J. C. Maier, F. M. Möller, and L. Purucker, "Hardware Aware Ensemble Selection for Balancing Predictive Accuracy and Cost," *arXiv (Cornell University)* , Aug. 2024, doi: 10.48550/arxiv.2408.02280.

[80] S. N. Raj and K. Deb, "EvoSort: A Genetic-Algorithm-Based Adaptive Parallel Sorting Framework for Large-Scale High Performance Computing," *arXiv (Cornell University)* , May 2025, doi: 10.48550/arxiv.2505.18681.

[81] J. Gong, "Pushing the Boundary: Specialising Deep Configuration Performance Learning," *arXiv (Cornell University)* , Jul. 2024, doi: 10.48550/arxiv.2407.02706.

[82] Z. R. K. Rostam, S. Szénasi, and G. Kertész, "Achieving Peak Performance for Large Language Models: A Systematic Review," *IEEE Access* , vol. 12. Institute of Electrical and Electronics Engineers, p. 96017, Jan. 01, 2024. doi: 10.1109/access.2024.3424945.

[83] S. Emanuilov and A. Dimov, "A quantitative framework for evaluating architectural patterns in ML systems," *arXiv (Cornell University)* , Jan. 2025, doi: 10.48550/arxiv.2501.11543.

[84] D. Cui, Z. Peng, K. Li, Q. Li, J. He, and X. Deng, "An novel cloud task scheduling framework using hierarchical deep reinforcement learning for cloud computing," *PLoS ONE* , vol. 20, no. 8, Aug. 2025, doi: 10.1371/journal.pone.0329669.

[85] S. Memeti, S. Pllana, A. P. D. Binotto, J. Kołodziej, and I. Brandić, "Using meta-heuristics and machine learning for software optimization of parallel computing systems: a systematic literature review," *Computing* , vol. 101, no. 8, p. 893, Apr. 2018, doi: 10.1007/s00607-018-0614-9.

[86] M. Xu, L. Wen, J. Liao, H. Wu, K. Ye, and C. Xu, "Auto-scaling Approaches for Cloud-native Applications: A Survey and Taxonomy,"

ArXiv.org , Jul. 2025, doi: 10.48550/arxiv.2507.17128.

[87] T. Zhang, H. Qiu, G. Castellano, M. Rifai, C. S. Chen, and F. Pianese, "System Log Parsing: A Survey," *IEEE Transactions on Knowledge and Data Engineering* , p. 1, Jan. 2023, doi: 10.1109/tkde.2022.3222417.

[88] Y. Lin, G. Gay, and P. Leitner, "An Experimental Study of Real-Life LLM-Proposed Performance Improvements," *arXiv (Cornell University)* , Oct. 2025, doi: 10.48550/arxiv.2510.15494.

[89] K. Zaouk, "Neural-Based Modeling for Performance Tuning of Cloud Data Analytics," *HAL (Le Centre pour la Communication Scientifique Directe)* , Mar. 2021, Accessed: Mar. 2025. [Online]. Available: https://theses.hal.science/tel-03284173

[90] J. Arafat, F. Tasmin, and S. Poudel, "Next-Generation Event-Driven Architectures: Performance, Scalability, and Intelligent Orchestration Across Messaging Frameworks," *ArXiv.org* , Oct. 2025, doi: 10.48550/arxiv.2510.04404.

[91] Y. Mu *et al.* , "Understanding LLM-Centric Challenges for Deep Learning Frameworks: An Empirical Analysis," *arXiv (Cornell University)* , Jun. 2025, doi: 10.48550/arxiv.2506.13114.

[92] C. Witt, M. Bux, W. Gusew, and U. Leser, "Predictive performance modeling for distributed batch processing using black box monitoring and machine learning," *Information Systems* , vol. 82, p. 33, Jan. 2019, doi: 10.1016/j.is.2019.01.006.

[93] H. Goyal, "Artificial Intelligence for Cost-Aware Resource Prediction in Big Data Pipelines," *arXiv (Cornell University)* , Sep. 2025, doi: 10.48550/arxiv.2510.05127.

[94] T. S. Phung and D. Thain, "Scaling Up Throughput-oriented LLM Inference Applications on Heterogeneous Opportunistic GPU Clusters with Pervasive Context Management," *arXiv (Cornell University)* , Sep. 2025, doi: 10.48550/arxiv.2509.13201.

[95] S. Zhang, X. Zeng, Y. Wu, and Z. Yang, "Harnessing Scalable Transactional Stream Processing for Managing Large Language Models [Vision]," *arXiv (Cornell University)* , Jul. 2023, doi: 10.48550/arxiv.2307.08225.

[96] R. Wu, Y. Wang, and D. Kutscher, "Affordable HPC: Leveraging Small Clusters for Big Data and Graph Computing," *arXiv (Cornell University)* , Aug. 2024, doi: 10.48550/arxiv.2408.15568.

[97] N. M. Sheikh, "Optimizing Software Performance in Distributed Cloud Systems: Challenges and Solutions," *Journal of Artificial Intelligence General science (JAIGS) ISSN 3006-4023* , vol. 7, no. 1, p. 187, Jan. 2025, doi: 10.60087/jaigs.v7i01.314.

[98] K. Shivashankar, G. S. A. Hajj, and A. Martini, "Scalability and Maintainability Challenges and Solutions in Machine Learning: Systematic Literature Review," *arXiv (Cornell University)* , Apr. 2025, doi: 10.48550/arxiv.2504.11079.

[99] Y. Yang *et al.* , "Declarative Data Pipeline for Large Scale ML Services," *ArXiv.org* , Aug. 2025, doi: 10.48550/arxiv.2508.15105.

[100] C. Giannoula, "Accelerating Irregular Applications via Efficient Synchronization and Data Access Techniques," *arXiv (Cornell University)* , Nov. 2022, doi: 10.48550/arxiv.2211.05908.

[101] B. G. Balani, "A Comprehensive Review of Advancements in Communication-Efficient Distributed Optimization," *International Journal for Research in Applied Science and Engineering Technology* , vol. 11, no. 11. International Journal for Research in Applied Science and Engineering Technology (IJRASET), p. 1610, Nov. 22, 2023. doi: 10.22214/ijraset.2023.56860.

[102] B. Shah, P. Rao, B. Moon, and M. Rajagopalan, "A Data Parallel Algorithm for XML DOM Parsing," in *Lecture notes in computer science* , Springer Science+Business Media, 2009, p. 75. doi: 10.1007/978-3-642-03555-5_7.

[103] I. Yousaf, "AI and Machine Learning Approaches for Predicting Nanoparticles Toxicity The Critical Role of Physiochemical Properties," *arXiv (Cornell University)* , Sep. 2024, doi: 10.48550/arxiv.2409.15322.

[104] B. Gautam and B. Annappa, "Performance prediction of data streams on high-performance architecture," *Human-centric Computing and Information Sciences* , vol. 9, no. 1, Jan. 2019, doi: 10.1186/s13673-018-0163-4.

[105] F. Wu, M. Bilal, H. Xiang, H. Wang, J. Yu, and X. Xu, "Real-time and Downtime-tolerant Fault Diagnosis for Railway Turnout Machines (RTMs) Empowered with

Cloud-Edge Pipeline Parallelism," *arXiv (Cornell University)* , Nov. 2024, doi: 10.48550/arxiv.2411.02086.

[106] W. Luk *et al.* , "6 Hardware-Aware Execution," in *De Gruyter eBooks* , De Gruyter, 2022, p. 249. doi: 10.1515/9783110785944-006.

[107] "Proceedings of the 17th Conference on Computer Science and Intelligence Systems," *Annals of Computer Science and Information Systems* , vol. 30, Sep. 2022, doi: 10.15439/978-83-962423-9-6.

[108] J. Groen, M. Belgiovine, U. Demir, B. Kim, K. Chowdhury, and C. Kaushik, "From Classification to Optimization: Slicing and Resource Management with TRACTOR," *arXiv (Cornell University)* , Dec. 2023, doi: 10.48550/arxiv.2312.07896.

[109] A. Reda, S. A. Taie, and M. E. Shaheen, "Hybrid MLOps framework for automated lifecycle management of adaptive phishing detection models," *Scientific Reports* , vol. 15, no. 1, p. 38478, Nov. 2025, doi: 10.1038/s41598-025-23600-z.

[110] V. Ramamoorthi, "Advances in AI and ML for Cloud Computing: A Review of Algorithms, Challenges, and Innovations," *International Journal of Scientific Research in Science and Technology* , vol. 12, no. 5. Technoscience Academy, p. 60, Sep. 06, 2025. doi: 10.32628/ijsrst2513120.

[111] N. Berbiche and J. E. Alami, "For Robust DDoS Attack Detection by IDS: Smart Feature Selection and Data Imbalance Management Strategies," *Ingénierie des systèmes d information* , vol. 29, no. 4, Aug. 2024, doi: 10.18280/isi.290401.

[112] M. Alrammal, "Algorithmes de traitement de flux XML : masses de données, mémoire externe et performances extensibles," *HAL (Le Centre pour la Communication Scientifique Directe)* , May 2011, Accessed: Jan. 2025. [Online]. Available: https://theses.hal.science/tel-00779309

[113] C. Lyu, Q. Fan, P. Guyard, and Y. Diao, "A Spark Optimizer for Adaptive, Fine-Grained Parameter Tuning," *Proceedings of the VLDB Endowment* , vol. 17, no. 11, p. 3565, Jul. 2024, doi: 10.14778/3681954.3682021.

[114] S. Hashemi and M. Mäntylä, "Token Interdependency Parsing (Tipping) -- Fast and Accurate Log Parsing," *arXiv (Cornell University)* , Aug. 2024, doi: 10.48550/arxiv.2408.00645.

[115] M. Pivezhandi, A. Saifullah, and V. P. Modekurthy, "A Statistical Learning Approach for Feature-Aware Task-to-Core Allocation in Heterogeneous Platforms," *arXiv (Cornell University)* , Jan. 2025, doi: 10.48550/arxiv.2502.15716.

[116] U. Pathak and A. Mankodi, "Redefining Cost Estimation in Database Systems: The Role of Execution Plan Features and Machine Learning," *arXiv (Cornell University)* , Oct. 2025, doi: 10.48550/arxiv.2510.05612.

[117] N. Ferro and G. Silvello, "Descendants, ancestors, children and parent: A set-based approach to efficiently address XPath primitives," *Information Processing & Management* , vol. 52, no. 3, p. 399, Dec. 2015, doi: 10.1016/j.ipm.2015.11.001.

[118] S. Mitra, R. Karami, H. Xu, S. Huang, and H. Kwon, "Characterizing State Space Model (SSM) and SSM-Transformer Hybrid Language Model Performance with Long Context Length," *arXiv (Cornell University)* , Jul. 2025, doi: 10.48550/arxiv.2507.12442.

[119] O. Elaeraj and C. Leghris, "Intrusion Detection System Based on an Intelligent Multi-Layer Model Using Machine Learning," *Journal of Artificial Intelligence and Technology* , Aug. 2024, doi: 10.37965/jait.2024.0554.

[120] A. Rahman *et al.* , "MARCO: Multi-Agent Code Optimization with Real-Time Knowledge Integration for High-Performance Computing," *arXiv (Cornell University)* , May 2025, doi: 10.48550/arxiv.2505.03906.

[121] S. K. S. Gotur, "PERFORMANCE TESTING IN MACHINE LEARNING SYSTEMS: A SYSTEMATIC FRAMEWORK FOR EVALUATION AND OPTIMIZATION," *INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY* , vol. 15, no. 6, p. 1930, Dec. 2024, doi: 10.34218/ijcet_15_06_165.

[122] S. Talatahari, F. Chen, and A. H. Gandomi, "Developing a robust machine learning framework for predicting the behavior of large-scale structure," *Journal of Building Engineering* , vol. 105, p. 112204, Feb. 2025, doi: 10.1016/j.jobe.2025.112204.

[123] A. Shukla *et al.* , "Adaptive Data Flywheel: Applying MAPE Control Loops to AI

Agent Improvement," *arXiv (Cornell University)* , Oct. 2025, doi: 10.48550/arxiv.2510.27051.

[124] C. Yan, X. Zhang, and J. Shen, "Credit Score Classification Using Advanced Machine Learning: A Comprehensive Approach," *Journal of Software Engineering and Applications* , vol. 18, no. 3, p. 98, Jan. 2025, doi: 10.4236/jsea.2025.183007.

[125] V. Thakkar *et al.* , "ELMo-Tune-V2: LLM-Assisted Full-Cycle Auto-Tuning to Optimize LSM-Based Key-Value Stores," *arXiv (Cornell University)* , Feb. 2025, doi: 10.48550/arxiv.2502.17606.

[126] Y. Biran and I. Kissos, "Adaptive Orchestration for Large-Scale Inference on Heterogeneous Accelerator Systems Balancing Cost, Performance, and Resilience," *arXiv (Cornell University)* , Mar. 2025, doi: 10.48550/arxiv.2503.20074.

[127] P.-Y. Ken and C. Wu, "Efficient Configuration of Heterogeneous Resources and Task Scheduling Strategies in Deep Learning Auto-Tuning Systems," *Research Square (Research Square)* , Oct. 2024, doi: 10.21203/rs.3.rs-5259219/v1.

[128] Y. Zheng, Y. Hu, W. Zhang, and A. Quinn, "Towards Agentic OS: An LLM Agent Framework for Linux Schedulers," *arXiv (Cornell University)* , Sep. 2025, doi: 10.48550/arxiv.2509.01245.

[129] J. Lian *et al.* , "ContTune: Continuous Tuning by Conservative Bayesian Optimization for Distributed Stream Data Processing Systems," *Proceedings of the VLDB Endowment* , vol. 16, no. 13, p. 4282, Sep. 2023, doi: 10.14778/3625054.3625064.

[130] W. Zhang and H. Ou, "Reinforcement learning based multi objective task scheduling for energy efficient and cost effective cloud edge computing," *Scientific Reports* , vol. 15, no. 1, p. 41716, Nov. 2025, doi: 10.1038/s41598-025-25666-1.

[131] S. Duan *et al.* , "A structure-aware framework for learning device placements on computation graphs," *arXiv (Cornell University)* , May 2024, doi: 10.48550/arxiv.2405.14185.

[132] F. S. Luan *et al.* , "The Streaming Batch Model for Efficient and Fault-Tolerant Heterogeneous Execution," *arXiv (Cornell University)* , Jan. 2025, doi: 10.48550/arxiv.2501.12407.

[133] M. A. Rodriguez, C. K. Dehury, S. N. Srirama, and R. Buyya, "Deep Reinforcement Learning (DRL)-based Methods for Serverless Stream Processing Engines: A Vision, Architectural Elements, and Future Directions," *arXiv (Cornell University)* , Feb. 2024, doi: 10.48550/arxiv.2402.17117.

[134] X. Hua and Z. Lu-Bin, "Workflow scheduling in IaaS clouds with the optimal pairing between tasks and virtual machines," *Journal of King Saud University - Computer and Information Sciences* , vol. 37, no. 8, Sep. 2025, doi: 10.1007/s44443-025-00260-7.

[135] Z. Yu, C. Du, H. Xu, Y. Zhou, B. Liu, and J. Li, "REACH: Reinforcement Learning for Efficient Allocation in Community and Heterogeneous Networks," *arXiv (Cornell University)* , Aug. 2025, doi: 10.48550/arxiv.2508.12857.

[136] L. M. Vaquero and F. Cuadrado, "Auto-tuning Distributed Stream Processing Systems using Reinforcement Learning," *arXiv (Cornell University)* , Mar. 2022, doi: 10.48550/arxiv.1809.05495.

[137] X. Wang, "Dynamic Scheduling Strategies for Resource Optimization in Computing Environments," *arXiv (Cornell University)* , Dec. 2024, doi: 10.48550/arxiv.2412.17301.

[138] A. Hooda, "Adaptive Real-Time Big Data Processing Framework: A Machine Learning and Reinforcement Learning Approach Using Random Forest and Q-Learning for Dynamic Resource Management," *Research Square (Research Square)* , Aug. 2024, doi: 10.21203/rs.3.rs-4962286/v1.

[139] H. Hao, C. Xu, W. Zhang, S. Yang, and G. Muntean, "Task-Driven Priority-Aware Computation Offloading Using Deep Reinforcement Learning," *IEEE Transactions on Wireless Communications* , vol. 24, no. 10, p. 8114, May 2025, doi: 10.1109/twc.2025.3564356.

[140] B. Pang, K. Li, R. She, and F. Wang, "Hybrid Offline-online Scheduling Method for Large Language Model Inference Optimization," *arXiv (Cornell University)* , Feb. 2025, doi: 10.48550/arxiv.2502.15763.

[141] C. Redovian, "Meta-Reinforcement Learning with Discrete World Models for Adaptive Load Balancing," *arXiv (Cornell University)* , Mar. 2025, doi: 10.48550/arxiv.2503.08872.

[142] A. Abo-eleneen, M. Helmy, A. A. Abdellatif, A. Erbad, A. Mohamed, and M. Abdallah, "Learn to Slice, Slice to Learn: Unveiling Online Optimization and Reinforcement Learning for Slicing AI Services," *arXiv (Cornell University)* , Nov. 2024, doi: 10.48550/arxiv.2411.03686.

[143] Y. Ze, J. Yi-hong, L. Juntian, and X. Xinhe, "An Intelligent Fault Self-Healing Mechanism for Cloud AI Systems via Integration of Large Language Models and Deep Reinforcement Learning," *arXiv (Cornell University)* , Jun. 2025, doi: 10.48550/arxiv.2506.07411.

[144] P. Jadhav, H. Jin, E. Deelman, and P. Balaprakash, "Evaluating the Efficacy of LLM-Based Reasoning for Multiobjective HPC Job Scheduling," *arXiv (Cornell University)* , May 2025, doi: 10.48550/arxiv.2506.02025.

[145] Y. Zou, N. D. Qi, Y. Deng, Z. Xue, M. Gong, and W. Zhang, "Autonomous Resource Management in Microservice Systems via Reinforcement Learning," *arXiv (Cornell University)* , Jul. 2025, doi: 10.48550/arxiv.2507.12879.

[146] Z. Zhang *et al.* , "The Vision of Autonomic Computing: Can LLMs Make It a Reality?," *arXiv (Cornell University)* , Jul. 2024, doi: 10.48550/arxiv.2407.14402.

[147] J. Liu *et al.* , "Edge-Cloud Collaborative Computing on Distributed Intelligence and Model Optimization: A Survey," *arXiv (Cornell University)* , May 2025, doi: 10.48550/arxiv.2505.01821.

[148] K. Agrawal and N. Nargund, "Neural Orchestration for Multi-Agent Systems: A Deep Learning Framework for Optimal Agent Selection in Multi-Domain Task Environments," *arXiv (Cornell University)* , May 2025, doi: 10.48550/arxiv.2505.02861.

[149] Y. Shen *et al.* , "Large Language Models Empowered Autonomous Edge AI for Connected Intelligence," *IEEE Communications Magazine* , vol. 62, no. 10, p. 140, Jan. 2024, doi: 10.1109/mcom.001.2300550.

[150] P. Patidar *et al.* , "Orchestration for Domain-specific Edge-Cloud Language Models," *arXiv (Cornell University)* , Jul. 2025, doi: 10.48550/arxiv.2507.09003.

[151] M. Shokrnezhad and T. Taleb, "An Autonomous Network Orchestration Framework Integrating Large Language Models with Continual Reinforcement Learning," *arXiv (Cornell University)* , Feb. 2025, doi: 10.48550/arxiv.2502.16198.

[152] J. Liu, "User-Centric Machine Learning Systems," *Deep Blue (University of Michigan)* , Jan. 2025, doi: 10.7302/26969.

[153] H. Shi *et al.* , "Enhancing Cluster Resilience: LLM-agent Based Autonomous Intelligent Cluster Diagnosis System and Evaluation Framework," *arXiv (Cornell University)* , Nov. 2024, doi: 10.48550/arxiv.2411.05349.

[154] F. Koch, A. Djuhera, and A. P. D. Binotto, "Intelligent Orchestration of Distributed Large Foundation Model Inference at the Edge," *arXiv (Cornell University)* , Mar. 2025, doi: 10.48550/arxiv.2504.03668.

[155] M. Adnan, B. Gamage, Z. Xu, D. Herath, and C. C. N. Kuhn, "Unleashing Artificial Cognition: Integrating Multiple AI Systems," *arXiv (Cornell University)* , Aug. 2024, doi: 10.48550/arxiv.2408.04910.

[156] J. Pournazari, A. Ullah, A. Al-Dubai, and X. Liu, "Computation offloading in the edge-to-cloud compute continuum: a survey of federated architectural solutions," *Cluster Computing* , vol. 28, no. 13, Sep. 2025, doi: 10.1007/s10586-025-05577-6.

[157] A. R. Sadik, M. Ashfaq, N. Mäkitalo, and T. Mikkonen, "Human-LLM Synergy in Context-Aware Adaptive Architecture for Scalable Drone Swarm Operation," *arXiv (Cornell University)* , Sep. 2025, doi: 10.48550/arxiv.2509.05355.

[158] K. Tallam, "From Autonomous Agents to Integrated Systems, A New Paradigm: Orchestrated Distributed Intelligence," *arXiv (Cornell University)* , Mar. 2025, doi: 10.48550/arxiv.2503.13754.

[159] M. Jung, "Compute Can't Handle the Truth: Why Communication Tax Prioritizes Memory and Interconnects in Modern AI Infrastructure," *arXiv (Cornell University)* , Jul. 2025, doi: 10.48550/arxiv.2507.07223.

[160] I. Goyal, "Transforming IT Operations with Agentic AI: The Evolution from Reactive to Autonomous Infrastructure Management," *International Journal of Computational and Experimental Science and Engineering* , vol. 11, no. 4, Oct. 2025, doi: 10.22399/ijcesen.4037.

[161] Z. Asgar, M. Nguyen, and S. Katti, "Efficient and Scalable Agentic AI with Heterogeneous Systems," *arXiv (Cornell University)* , Jul. 2025, doi: 10.48550/arxiv.2507.19635.

[162] G. Ramisetty, "Event-Driven Micro services for Ultra-Low Latency Cloud Workflows," *Global Journal of Computer Science and Technology* , p. 1, Oct. 2025, doi: 10.34257/gjcstbvol25is1pg1.

[163] N. Ye, A. Ahuja, G. Liargkovas, Y. Lu, K. Kaffes, and T. Peng, "Speculative Actions: A Lossless Framework for Faster Agentic Systems," *arXiv (Cornell University)* , Oct. 2025, doi: 10.48550/arxiv.2510.04371.

[164] B. Barua and M. S. Kaiser, "Optimizing Airline Reservation Systems with Edge-Enabled Microservices: A Framework for Real-Time Data Processing and Enhanced User Responsiveness," *arXiv (Cornell University)* , Nov. 2024, doi: 10.48550/arxiv.2411.12650.

[165] R. Raj, H. Wang, and T. Krishna, "A CPU-Centric Perspective on Agentic AI," *arXiv (Cornell University)* , Nov. 2025, doi: 10.48550/arxiv.2511.00739.

[166] A. Gliozzo *et al.* , "Transduction is All You Need for Structured Data Workflows," *arXiv (Cornell University)* , Aug. 2025, doi: 10.48550/arxiv.2508.15610.

[167] E. Stehle and H. Jacobsen, "ParPaRaw," *Proceedings of the VLDB Endowment* , vol. 13, no. 5, p. 616, Jan. 2020, doi: 10.14778/3377369.3377372.

[168] E. Zhang, E. Zhu, G. Bansal, A. Fourney, H. Mozannar, and J. Gerrits, "Optimizing Sequential Multi-Step Tasks with Parallel LLM Agents," *arXiv (Cornell University)* , Jul. 2025, doi: 10.48550/arxiv.2507.08944.

[169] K. Sharma, S. Salagrama, D. Parashar, and R. S. Chugh, "AI-Driven Decision Making in the Age of Data Abundance: Navigating Scalability Challenges in Big Data Processing," *Revue d intelligence artificielle* , vol. 38, no. 4, p. 1335, Aug. 2024, doi: 10.18280/ria.380427.