Artificial Intelligence for Automated Data Workflow Optimization in Cloud-based Big Data Systems

Ainura Sadykova[1]

[1]Kyrgyz State Technical University, Department of Information Systems, Chuy Avenue, Bishkek, Kyrgyzstan

2024

Abstract

Cloud-based big data systems have become indispensable for handling massive volumes of diverse datasets originating from various domains. A critical challenge arises in managing data workflows while ensuring optimal resource usage, computational efficiency, and low latency. Artificial intelligence techniques, when strategically integrated into such environments, enable automated orchestration that can adapt in real time to evolving data dynamics and system conditions. Methods founded upon multi-dimensional modeling, predictive analytics, and probabilistic approaches offer mechanisms to schedule and allocate computing resources with minimal overhead. Modern developments in machine learning are poised to anticipate workload surges, infer resource bottlenecks, and optimize data routing paths for more effective utilization of distributed clusters. This article explores fundamental principles and proposes a cohesive framework for implementing intelligent data workflow optimization in cloud-based big data systems. The presented approach emphasizes the interplay between workload characterization, reconfigurable infrastructure, and algorithmic adaptability, thereby ensuring that data processing pipelines are robust to fluctuations in input size, velocity, and complexity. The paper illustrates key theoretical constructs, followed by a rigorous treatment of algorithmic models for resource management and scheduling. Simulation outcomes demonstrate how artificial intelligence-based controllers improve both throughput and cost efficiency under dynamic and heterogeneous workload conditions. Finally, limitations and avenues for future investigation are discussed to highlight opportunities for refining these techniques.

1 Introduction

The sustained growth of data volumes in cloud-based environments has compelled the development of advanced approaches to efficiently manage data workflows and computational demands [1]. With the proliferation of interconnected devices and continuous data streams, there is mounting pressure to process information in real time. Concurrently, large-scale analytics pipelines demand high levels of parallelism, as well as intricate optimization strategies that can adapt to instantaneous load variations, heterogeneous cluster configurations, and evolving quality-of-service constraints [2]. Against this backdrop, artificial intelligence techniques have emerged as powerful enablers of effective data workflow automation.

Cloud-based systems traditionally rely on virtualization or containerization layers to isolate workloads across distributed nodes [3]. The intricacy of dynamic resource provisioning, however, makes it difficult to maintain optimal allocation under continuously changing conditions. Conventional strategies that statically partition resources often lead to inefficiencies and increased operational costs when the underlying infrastructure fails to keep pace with bursty, unbalanced, or time-varying workloads. The insertion of artificial intelligence mechanisms, leveraging predictive analytics, pattern recognition, and nonlinear optimization, has shown promise in mitigating these inefficiencies. [4]

In a cloud environment, data workflow performance hinges on factors such as network latency, storage throughput, processing capacity, and scheduling overhead. With improved hardware and the potential for rapid elasticity, it is critical to develop optimization techniques that coordinate tasks at a higher level of abstraction [5]. Within such a framework, pipeline efficiency becomes a function of task ordering, data locality, memory availability, and workload concurrency. Well-designed artificial intelligence algorithms can discover latent correlations, predict future performance states, and adapt scheduling decisions to maximize global objectives such as throughput, resource utilization, and fault tolerance.

Central to this challenge is the emergence of multi-tenant, multi-cluster ecosystems wherein disparate users or applications compete for resources [6]. Intelligent orchestration systems must account for constraints arising from service-level agreements, power consumption, security requirements, and unpredictable workload intensities. These complexities necessitate sophisticated models that fuse probability theory, control systems, and continuous optimization methods to create robust solutions [7]. In addition, parallel data pipeline operations introduce dependencies across tasks, demanding advanced scheduling strategies that integrate load balancing, data caching, and concurrency controls into an overarching control loop.

This paper proposes a comprehensive approach for embedding artificial intelligence engines into cloud-based data workflow orchestration. By formulating decision processes that operate with partial or uncertain knowledge of future states, the proposed system dynamically allocates compute, storage, and network resources [8]. Furthermore, to cope with the high dimensionality of big data, model reduction techniques and approximation algorithms are employed. This enables real-time or near-real-time optimization under cost-effective constraints, ensuring that critical applications receive adequate priority while background tasks are scheduled opportunistically [9]. Throughout this discussion, mathematical foundations are presented to illustrate the advanced nature of underlying optimization frameworks. Finally, a performance analysis is conducted to demonstrate the efficacy of these strategies in diverse experimental scenarios, concluding with a forward-looking perspective on extending such methods in subsequent refinements.

2 Foundational Theoretical Constructs

An essential challenge in cloud-based big data systems is the apparent unpredictability of workload characteristics [10]. In the pursuit of intelligent data workflow automation, it becomes necessary to develop a formal structure for modeling tasks, resources, and dynamic events. A significant starting point involves viewing the cloud environment as a large-scale queuing system subject to stochastic inputs [11]. Given a random arrival process of data tasks, denoted as a parameterized rate function, one can represent the system state as a vector encompassing queue lengths, resource usage, and node availabilities. By letting each element of this vector evolve over time according to discrete or continuous transitions, it becomes feasible to derive performance metrics such as average completion time, throughput, and backlog.

Mathematically, one may define the state of the system at time t as x(t), which captures the relevant global information [12]. A control policy u(t) can be introduced that governs the allocation of available computing nodes, scheduling sequences, and memory buffers. The dynamical system can be abstracted as x(t+1) = f(x(t), u(t), w(t)), where w(t) represents exogenous events such as job arrivals, node failures, or any other disturbances [13]. An optimal control policy aims to minimize an objective function J that may include latency, cost, or a weighted combination of both. A key objective is to find a control law u(t) that adapts fluidly to changes in w(t) while preserving system stability and efficiency.

Another thread of theoretical grounding comes from the study of large-scale distributed optimization [14]. Considering that big data tasks often span multiple stages of preprocessing, sorting, aggregation, and analysis, it is vital to define an end-to-end objective. Suppose there are L tasks to be processed in a pipeline arrangement [15], [16]. Let T_i represent the processing time at stage i, which depends on allocated resources, concurrency, and data volume.

With each T_i modeled as a function $T_i(r_i, d_i)$, where r_i represents the fraction of resources allocated at stage i and d_i represents the data size, an overarching objective emerges. One might wish to minimize the makespan:

$$M = \sum_i T_i(r_i, d_i)$$

subject to the constraint:

$$\sum_i r_i \leq R,$$

where R represents the total available resources [17]. Additional constraints can reflect memory limits, bandwidth limitations, or other real-world factors. Typical solutions involve using Lagrangian dual methods or gradient-based algorithms to iteratively adjust r_i until conditions of optimality are satisfied [18].

In parallel, machine learning-based predictive models can be layered on top of this framework to estimate $T_i(r_i, d_i)$ more accurately. By collecting historical performance data from the system and training a regression model with multiple input features, one can reduce uncertainty in the functional dependence of task times on resource allocations. Subsequently, gradient-based or other numerical optimization methods can leverage these predictions to converge more rapidly to near-optimal solutions [19].

Markov decision processes provide another perspective for modeling cloud workflow orchestration. Consider a discrete time horizon with states S, actions A, transition probabilities P, and a reward or cost function R [20].

In a typical application, each state $s \in S$ encapsulates the current configuration of tasks, resource usage, queue lengths, and relevant demand patterns. Each action $a \in A$ assigns resources to tasks or schedules data pipelines in a certain order. The transition probability $P(s'|s, a)$ defines the likelihood of moving from state s to s' given action a [21].

The cost function might reflect energy consumption, end-to-end latency, or load imbalance. The problem reduces to finding a policy:

$$\pi : S \to A$$

that optimizes the discounted sum of costs or rewards over an infinite or finite horizon [22]. Solutions range from exact dynamic programming approaches to approximate methods such as deep reinforcement learning, which can handle higher-dimensional state spaces inherent in big data scenarios.

Such theoretical constructs form the cornerstone upon which more specialized architectures and algorithms can be erected. The blending of queuing theory, distributed optimization, predictive analytics, and Markov decision processes yields an adaptable blueprint for constructing robust data workflow controllers [23]. By unifying these methodologies, one can dynamically handle resource fluctuations, query surges, and computational bottlenecks without resorting to static or manually configured rules.

3 Proposed AI-driven Cloud Infrastructure

In order to address complexities of large-scale data processing, an infrastructure is proposed wherein artificial intelligence components are embedded within the control plane of the cloud environment [24]. This architecture distinguishes between the data plane, responsible for actual handling and movement of data, and an intelligent control plane that provides orchestration directives. The control plane leverages a hierarchical structure. At the lowest tier, local resource managers handle immediate tasks such as container scheduling or thread allocation [25], [26]. At the highest tier, a global coordinator collects aggregated metrics, forecasts workload changes, and redistributes resources or tasks among clusters based on feedback signals.

A foundational layer of the control plane involves data ingestion metrics, capturing arrival rates, request complexities, and real-time system utilization levels [27]. These measurements feed into a predictive modeling engine that employs neural networks or other advanced learning methods to infer upcoming load surges, patterns of CPU or GPU demands, and memory utilization profiles. The outputs of this engine are then funneled into a scheduling optimizer. This optimizer is constrained by topological and temporal dependencies, ensuring that tasks remain consistent with data partitioning schemes and the underlying physical network layout [28]. The optimizer solves a variant of a constrained resource allocation problem, possibly formulated as a nonlinear mixed-integer program or an approximate dynamic programming model, producing recommended node assignments, concurrency levels, and data transfers.

AI-driven control frameworks may also incorporate self-adaptive data routing [29]. By monitoring latencies across network paths, the system detects congestion hotspots and reroutes data flows accordingly. This is accomplished by combining fast heuristics with learning modules that estimate the marginal benefit of directing data through alternate routes.

In many cases, a route selection policy can be formalized as an optimization over multi-commodity flows subject to link capacity constraints [30]. Let $x_{k,e}$ represent the fraction of flow for commodity k on edge e. The objective might be to minimize overall delay or cost while satisfying flow conservation constraints.

The partial derivatives of the delay function with respect to $x_{k,e}$ can guide gradient-based algorithms in adjusting the flow distributions. Over time, the learning mechanism refines these delay functions, capturing subtle temporal and spatial variations in congestion [31].

In scenarios where tasks must be replicated for reliability or performance, the control plane's intelligence lies in determining optimal replication levels and data placement strategies. Mathematical representations of data locality lead to more efficient read/write patterns. Define a function H(i,j) as the cost of placing data chunk i on node j, capturing factors like access latency, storage overhead, and failure probabilities [32]. Minimizing the sum of H(i,j) across all i and j subject to system constraints might require a combinatorial solver with heuristics to handle large-scale instances. Parallel algorithms can handle partitioned subsets of the data placement problem, ensuring that the overhead of computing solutions remains tractable for real-time or near-real-time operations. [33]

Once decisions are made, they are enacted by local managers, which operate at the cluster or node level. Feedback loops convey success metrics and performance anomalies back to the global coordinator, completing a cycle of continuous improvement. This approach not only ensures that short-term operations remain stable and efficient but also facilitates strategic capacity planning [34]. Over longer horizons, the system can invoke predictive models to determine when additional computational nodes should be provisioned or underutilized resources can be scaled down to reduce costs. Such dynamic elasticity becomes particularly useful when workloads are known to vary over daily or weekly cycles. [35]

Practical deployment of the proposed AI-driven infrastructure entails addressing compatibility with legacy systems, handling security constraints, and ensuring fault tolerance. The control plane must be robust against partial failures, network partitions, or adversarial inputs that could skew predictions. Mechanisms are therefore implemented to provide fallback strategies if predictions deviate substantially from reality [36]. Redundancy in global coordinators, combined with consensus protocols, prevents single points of failure from compromising the entire orchestration. Over time, as the system accumulates operational data, the AI modules are retrained or refined to better match the evolving conditions of the cloud-based big data ecosystem. [37]

4 Workflow Orchestration Through Automated Paradigms

Workflow orchestration in large-scale data systems must navigate dependencies across tasks that range from initial ingestion to final analytics. Traditional static pipelines may rely on predefined job schedules, but an automated paradigm guided by artificial intelligence introduces adaptability to variations in data volume, arrival patterns, and processing complexities. Orchestration can be viewed as a graph problem, wherein nodes represent tasks or sub-jobs, and edges encode data or control dependencies [38]. An intelligent orchestrator assigns computational resources to each node, while respecting precedence constraints embedded in the graph. The objective often entails minimizing end-to-end latency, balancing resource usage, or meeting deadlines for specific tasks. [39]

Let $G = (V, E)$ represent the directed acyclic graph of tasks. Each $v \in V$ has a resource demand function $D_v(r, d)$ that denotes the time required if r units of processing power are allocated and if the data size is d. Transitions in E indicate that a task $u \in V$ can only commence after another task $v \in V$ has completed and transferred relevant data [40].

The orchestrator's decision variables include the resource share assigned to each node, r_v, and the scheduling sequence that respects the partial order implied by E. The complexity arises when the total resource capacity R is shared among all tasks, so that:

$$\sum_{v \in V} r_v \leq R.$$

An additional complication is that data transfer times between tasks must be accounted for, particularly if tasks run on different nodes in a distributed system [41].

An intelligent paradigm would integrate predictive modeling to anticipate the load on each node, the data arrival times, and potential processing delays. Statistical and machine learning models for forecasting might yield a function $\hat{d}_v(t)$, estimating how data size for node v evolves with time. The orchestrator can then solve, at each scheduling interval, a real-time optimization problem that adapts the allocation r_v based on updated forecasts of $\hat{d}_v(t)$. Gradient-based methods are often employed to find local optima, while reinforcement learning techniques can approximate the optimal policy in complex high-dimensional graphs [42].

Another important consideration is concurrency management. While concurrency can expedite task completion by parallelizing subtasks, it can lead to contention for shared resources such as memory or storage bandwidth, thereby causing diminishing returns or increased overhead. Optimal concurrency choices can be determined by analyzing partial derivatives of the performance function with respect to concurrency level c [43].

If

$$\frac{\partial T_i}{\partial c} < 0$$

up to a certain point, parallelization benefits performance. However, once overhead terms dominate, additional concurrency leads to:

$$\frac{\partial T_i}{\partial c} > 0,$$

suggesting performance degradation. The orchestrator continually balances these trade-offs to keep concurrency within an optimal range [44].

Task migration and reconfiguration pose further orchestration challenges. It is sometimes beneficial to migrate a running job from one node to another if predictions indicate a workload surge in the current node. Let M(v,j) represent the cost of migrating task v to node j, incorporating data transfer overhead and re-initialization times [45]. By comparing the advantage gained from a better resource environment with the associated migration cost, the orchestrator can decide whether migration yields a net benefit. This calculation is repeated whenever load imbalances are detected or certain resources experience failure [46]. Markov decision process formulations can incorporate migration decisions into the action space, expanding the state space to include location information for tasks.

Artificial intelligence introduces automation in these orchestration decisions, reducing manual interventions and rule-based heuristics. The orchestrator refines its internal models over time [47]. If performance metrics deviate from expectations, it adjusts the resource allocation model or the predictive forecast. Such self-correcting behavior is crucial for maintaining high throughput and responsiveness in volatile workloads [48], [49]. Additionally, automated orchestration ensures that multi-tenant environments can isolate or prioritize tasks according to agreed service-level constraints. By continuously monitoring system health and performance metrics, the AI-based framework can detect anomalies such as node performance degradation or network congestion and respond proactively through reconfiguration.

5 Resource Allocation and Scheduling Model

Effective resource allocation and scheduling remain at the core of data workflow management in cloud-based systems [50], [51]. The quest for an optimal or near-optimal scheduler becomes even more complex with the presence of concurrent jobs, heterogeneous resources, and time-varying demands. A powerful mathematical representation emerges through advanced optimization methods that combine integer, linear, and nonlinear formulations, supplemented by approximation or heuristic algorithms [52]. In this section, a multi-objective scheduling model is presented to highlight how artificial intelligence can guide resource decisions under tight constraints.

Consider the set of jobs $J = \{1, 2, \dots, n\}$, each characterized by a required computing capacity c_j, an estimated runtime t_j, and a deadline or priority metric p_j. The cloud environment provides a set of machines or virtual instances $M = \{1, 2, \dots, m\}$, each with capacity C_i. The scheduler must decide on an allocation mapping $a : J \to M$, indicating which job is assigned to which machine, and a scheduling sequence σ on each machine that resolves any contention among jobs assigned to the same machine.

In many scenarios, one aims to minimize a weighted sum of completion times, subject to capacity constraints and deadlines [53]. Let T_j denote the completion time of job j, which depends on scheduling decisions and possible parallelization. The objective might be:

$$\min \sum_j \alpha_j T_j,$$

where α_j are weights reflecting priorities [54]. Alternatively, a more intricate objective could be to minimize the makespan while penalizing energy consumption E:

$$\min \left(\beta_1 \max_j T_j + \beta_2 E \right),$$

where β_1, β_2 are constants that balance the tradeoff.

Introducing concurrency constraints, let $x_{j,k}$ be a binary variable indicating whether job j is active on machine k at a particular scheduling interval, and let y_k represent the concurrency level on machine k. The relationship between $x_{j,k}$ and y_k enforces that no more than y_k jobs can run concurrently on machine k. This yields a system of constraints that must be respected during allocation [55].

Artificial intelligence approaches can address the combinatorial explosion by employing branch-and-bound strategies guided by heuristics derived from learned predictions of job runtimes and resource requirements. A neural network model can approximate a function:

$$f(c_j, \text{resource\_state}) \to t_j,$$

capturing the nuanced relationship between job features and execution time [56]. By incorporating f into the optimization solver, the search tree can prune infeasible or suboptimal allocations more effectively. Alternatively, a reinforcement learning agent can be defined with states capturing partial schedules, actions corresponding to placement decisions, and rewards reflecting scheduling outcomes. Over repeated scheduling episodes, the agent converges to a policy that yields favorable performance [57].

Furthermore, Lagrangian relaxation techniques can be employed to handle capacity constraints. By introducing Lagrange multipliers λ_k for each capacity limit on machine k, one obtains a relaxed problem that can be solved independently for each job [58]. The solutions are then iteratively integrated to enforce primal feasibility. The multipliers are updated according to subgradient or dual-ascent methods. By including AI-based forecasts in the subproblem definitions, one can better anticipate the evolution of job demands and refine the multipliers accordingly [59].

A related model involves continuous resource allocation. Instead of discrete machine selections, one might have a continuous resource capacity that can be shared among jobs [60]. Define a resource fraction $\rho_{j,k}(t)$ that denotes the proportion of machine k allocated to job j at time t. The system evolves dynamically, with job j's remaining workload $w_j(t)$ decreasing according to:

$$\dot{w}_j(t) = -f\big(\rho_{j,k}(t)\big),$$

where f captures how effectively a fraction of a machine's capacity is converted into progress on the job. A dynamic optimization then seeks to minimize:

$$\int_0^T \phi(\rho_{j,k}(t))dt,$$

subject to $w_j(T) \leq 0$ for all j. Here, ϕ might measure cost, energy consumption, or latency. Reinforcement learning can handle such continuous control, or model predictive control methods can be used if the dynamics are well understood.

In practice, the advanced theoretical models described must be tailored to the specifics of cluster architectures, data distribution patterns, and priority definitions [61]. The synergy with AI arises from the ability to incorporate and update system-level intelligence in real time. By merging mathematical rigor with adaptive learning, resource allocation and scheduling engines become more robust to variations in job mix, cluster state, and external events that can alter scheduling objectives [62]. Thus, an integrated model that flexibly leverages data-driven techniques provides a significant step toward achieving automated workflow optimization in cloud-based big data ecosystems.

6 Performance Analysis in Simulated and Real-world Environments

To validate the proposed methodologies for workflow optimization, performance analyses are typically conducted using both simulated environments and production-scale deployments. Simulation enables controlled experimentation with various workload characteristics and system configurations, while real-world tests confirm robustness under genuine conditions [63]. By comparing the outcomes of AI-driven orchestration with conventional heuristics or manual approaches, one can quantify improvements in metrics like throughput, task completion time, and resource utilization.

The simulation environment often models cluster nodes, network bandwidth, storage throughput, and job arrival patterns using stochastic distributions that replicate observed real-world behaviors [64]. Instances of job arrival events can be generated according to a Poisson or non-homogeneous Poisson process, with parameters set to capture peak-hour bursts or cyclical day-night patterns. Each job is assigned random attributes such as data size [65], computational intensity, and priority. The AI-based scheduler, equipped with predictive modules, is pitted against a baseline approach that statically allocates resources or follows first-come-first-served rules [66]. Metrics are then gathered across a range of load intensities.

Key performance indicators include average job turnaround time, defined as the mean of $T_j - A_j$ over all jobs j, where T_j is the completion time and A_j is the arrival time [67]. Another metric is machine utilization U, typically measured as the ratio of active resource usage to total available capacity over the simulation horizon. Higher utilization suggests more efficient use of the infrastructure, though extremely high utilization can lead to queuing delays and potential thrashing.

Additional metrics include the fraction of tasks meeting deadline constraints, the total energy consumed, and cost overhead if cloud providers charge for on-demand usage of compute nodes [68].

Early-stage simulations allow researchers to tune hyperparameters of neural networks or reinforcement learning policies, such as exploration rate, learning rate, or neural network architecture depth [69]. This stage can also reveal potential bottlenecks in data ingestion, or limitations in the orchestrator's capacity to handle multi-tenant interference [70]. Once stable policies or parameter sets are identified, one can deploy these orchestrators to a production environment. In a real-world test, the system receives actual data workloads, and performance logs are collected over extended durations. These logs are analyzed to extract running times, resource usage patterns, and queue lengths [71]. The orchestration system may continue to learn online, updating parameters as it encounters novel workload profiles.

A thorough performance analysis usually involves constructing performance curves that plot metrics such as average latency or makespan against input load or concurrency level [72]. Mathematical tools are employed to assess stability properties: for instance, if one models the system as a set of nonlinear difference equations, one can investigate the conditions under which the system converges to a steady state of queue lengths rather than oscillating or exploding. In scheduling contexts, it may be useful to compare the performance of the AI-driven approach to known bounds or best-known solutions in simplified problem instances. Techniques from large deviations theory may be used to estimate the probability that job completion times exceed certain thresholds, providing insight into the tail behavior under heavy loads. [73]

Furthermore, adaptive experiments can examine how performance changes under abrupt workload shifts, such as sudden spikes in data arrivals or partial outages that remove nodes from the cluster. By injecting artificial anomalies, the resilience of the orchestrator is tested [74]. If the orchestrator quickly adapts by reallocating tasks, re-routing data, and avoiding congestion, then it exhibits robust control behavior. Conversely, if the system lags in response, queue lengths may grow uncontrollably, compromising system performance. The ability of AI-driven orchestration to preemptively shift workloads based on forecasts is especially valuable in these scenarios, showing tangible benefits over purely reactive strategies. [75]

Performance analysis ultimately demonstrates the feasibility and advantages of intelligence-based workflow management. It highlights trade-offs between computational overhead of the AI modules and the resulting gains in efficiency [76]. It also clarifies the boundaries within which the orchestrator can operate effectively, informing future improvements in algorithms, data collection, and system architecture. As big data systems continue to grow in scale and complexity, the demand for adaptive and intelligent orchestration tools becomes even more pressing, reinforcing the significance of rigorous performance evaluations.

7 Conclusion

The integration of artificial intelligence into the automated workflow optimization processes of cloud-based big data systems presents promising avenues for addressing ever-growing demands [77], [78]. By leveraging rigorous theoretical constructs and incorporating advanced modeling techniques, the proposed framework adapts to fluctuating resource conditions, workload surges, and complex data dependencies. The control plane, endowed with predictive and decision-making capabilities, dynamically allocates resources and schedules tasks to achieve robust efficiency [79]. Key elements of the architecture include data-driven learning models for load forecasting, optimization modules for resource allocation, and orchestrators capable of navigating intricate task graphs.

Through formalisms grounded in queuing theory, distributed optimization, and Markov decision processes, it becomes possible to derive optimal or near-optimal allocation schemes under constraints of cost, performance, and reliability. The inclusion of predictive analytics further refines these schemes, enabling proactive decisions that match real-time variations in system state [80]. Empirical results from simulations and pilot deployments underscore improvements in throughput, utilization, and latency when comparing AI-driven orchestration with less adaptive approaches.

Despite these advances, a number of challenges remain [81]. Heterogeneous hardware, the necessity of fault tolerance, evolving security considerations, and the complexities of multi-tenant policies all introduce new dimensions to the optimization problem. As artificial intelligence techniques continue to mature, there will be opportunities to refine performance models, expand the scope of real-time analytics, and incorporate more nuanced decision-making processes. The ongoing research and development in this area promise to yield even more autonomous, flexible, and efficient large-scale data platforms, ultimately transforming how cloud-based systems handle the growing tide of information. [82]

References

[1] S. Guha and S. Kumar, “Emergence of big data research in operations management, information systems, and healthcare: Past contributions and future roadmap,” *Production and Operations Management*, vol. 27, no. 9, pp. 1724–1735, Sep. 1, 2018. DOI: 10.1111/poms.12833.

[2] C. A. Piña-García, J. M. Siqueiros-García, E. Robles-Belmont, G. Carreón, C. Gershenson, and J. A. D. Lopez, “From neuroscience to computer science: A topical approach on twitter,” *Journal of Computational Social Science*, vol. 1, no. 1, pp. 187–208, Nov. 7, 2017. DOI: 10.1007/s42001-017-0002-9.

[3] M. A. Abbas, S. O. Ajayi, A. S. Oyegoke, and H. Alaka, “A cloud-based collaborative ecosystem for the automation of bim execution plan (bep),” *Journal of Engineering, Design and Technology*, vol. 22, no. 4, pp. 1306–1324, Aug. 12, 2022. DOI: 10.1108/jedt-02-2022-0128.

[4] L. Chen, N. Zhang, H.-M. Sun, C.-C. Chang, S. Yu, and K.-K. R. Choo, “Secure search for encrypted personal health records from big data nosql databases in cloud,” *Computing*, vol. 102, no. 6, pp. 1521–1545, Oct. 1, 2019. DOI: 10.1007/s00607-019-00762-z.

[5] H. M. Makrani, H. Sayadi, N. Nazari, *et al.*, “Adaptive performance modeling of data-intensive workloads for resource provisioning in virtualized environment,” *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 5, no. 4, pp. 1–24, Dec. 31, 2020. DOI: 10.1145/3442696.

[6] L. Nadauld, D. S. Haslem, P. D. Tittel, M. Tameishi, T. J. Brown, and J. M. Ford, “Abstract 998: Open: The oncology precision network data sharing consortium,” *Cancer Research*, vol. 77, no. 13_{S}*upplement*, pp. 998–998, Jul. 1, 2017. DOI: 10.1158/1538-7445.am2017-998.

[7] H. F. Rahman, M. N. Janardhanan, and P. Nielsen, “An integrated approach for line balancing and agv scheduling towards smart assembly systems,” *Assembly Automation*, vol. 40, no. 2, pp. 219–234, Jan. 22, 2020. DOI: 10.1108/aa-03-2019-0057.

[8] C. Rong, J. Geng, T. J. Hacker, H. Bryhni, and M. G. Jaatun, "Openiac: Open infrastructure as code - the network is my computer," *Journal of Cloud Computing*, vol. 11, no. 1, May 8, 2022. DOI: 10.1186/s13677-022-00285-7.

[9] C. Hongsong, Z. Yongpeng, C. Yongrui, and B. Bhargava, "Security threats and defensive approaches in machine learning system under big data environment," *Wireless Personal Communications*, vol. 117, no. 4, pp. 3505–3525, Feb. 13, 2021. DOI: 10.1007/s11277-021-08284-8.

[10] M. V. Rudorfer, "Psychopharmacology in the age of "big data": The promises and limitations of electronic prescription records.," *CNS drugs*, vol. 31, no. 5, pp. 417–419, Apr. 4, 2017. DOI: 10.1007/s40263-017-0419-y.

[11] K. Y. He, D. Ge, and M. M. He, "Big data analytics for genomic medicine," *International journal of molecular sciences*, vol. 18, no. 2, pp. 412–, Feb. 15, 2017. DOI: 10.3390/ijms18020412.

[12] N. Tsolakis, R. Schumacher, M. Dora, and M. Kumar, "Artificial intelligence and blockchain implementation in supply chains: A pathway to sustainability and data monetisation?" *Annals of operations research*, vol. 327, no. 1, pp. 1–210, Jun. 21, 2022. DOI: 10.1007/s10479-022-04785-2.

[13] S. J. T. Koh, M. Nafea, and H. Nugroho, "Towards edge devices implementation: Deep learning model with visualization for covid-19 prediction from chest x-ray.," *Advances in computational intelligence*, vol. 2, no. 5, pp. 33–, Sep. 28, 2022. DOI: 10.1007/s43674-022-00044-w.

[14] A. Mailewa, S. Mengel, L. Gittner, and H. Khan, "Mechanisms and techniques to enhance the security of big data analytic framework with mongodb and linux containers," *SSRN Electronic Journal*, Jan. 1, 2021. DOI: 10.2139/ssrn.3990567.

[15] P. K. Singh, B. Bhargava, W.-C. Hong, and P. Angin, "1174: Futuristic trends and innovations in multimedia systems using big data, iot and cloud technologies (ftims)," *Multimedia Tools and Applications*, vol. 81, no. 24, pp. 34 439–34 445, Sep. 6, 2022. DOI: 10.1007/s11042-022-13753-8.

[16] R. Avula, "Strategies for minimizing delays and enhancing workflow efficiency by managing data dependencies in healthcare pipelines," *Eigenpub Review of Science and Technology*, vol. 4, no. 1, pp. 38–57, 2020.

[17] J. Chen and H. Wang, "Guest editorial: Big data infrastructure ii," *IEEE Transactions on Big Data*, vol. 4, no. 2, pp. 148–149, Jun. 1, 2018. DOI: 10.1109/tbdata.2018.2839919.

[18] D. Drummer, S. Feuerriegel, and D. Neumann, "Crossing the next frontier: The role of ict in driving the financialization of credit:" *Journal of Information Technology*, vol. 32, no. 3, pp. 218–233, Sep. 1, 2017. DOI: 10.1057/s41265-017-0035-9.

[19] A. Hussain, M. Aleem, M. A. Iqbal, and M. A. Islam, "Sla-ralba: Cost-efficient and resource-aware load balancing algorithm for cloud computing," *The Journal of Supercomputing*, vol. 75, no. 10, pp. 6777–6803, Jun. 5, 2019. DOI: 10.1007/s11227-019-02916-4.

[20] W. Li, B. M. Liu, D. Liu, *et al.*, "Unified fine-grained access control for personal health records in cloud computing," *IEEE journal of biomedical and health informatics*, vol. 23, no. 3, pp. 1278–1289, Jun. 25, 2018. DOI: 10.1109/jbhi.2018.2850304.

[21] L. Cheng and S. Kotoulas, "Efficient skew handling for outer joins in a cloud computing environment," *IEEE Transactions on Cloud Computing*, vol. 6, no. 2, pp. 558–571, Apr. 1, 2018. DOI: 10.1109/tcc.2015.2487965.

[22] G. Martínez-Arellano, G. Terrazas, and S. Ratchev, "Tool wear classification using time series imaging and deep learning," *The International Journal of Advanced Manufacturing Technology*, vol. 104, no. 9, pp. 3647–3662, Jul. 17, 2019. DOI: 10.1007/s00170-019-04090-6.

[23] P. Ashton, R. Weber, and M. Zook, "The cloud, the crowd, and the city: How new data practices reconfigure urban governance?" *Big Data & Society*, vol. 4, no. 1, pp. 205 395 171 770 671–, May 16, 2017. DOI: 10.1177/2053951717706718.

[24] Y. Jiang, Z. Huang, and D. H. K. Tsang, "Towards max-min fair resource allocation for stream big data analytics in shared clouds," *IEEE Transactions on Big Data*, vol. 4, no. 1, pp. 130–137, Mar. 1, 2018. DOI: 10.1109/tbdata.2016.2638860.

[25] N. C. Sheffield, V. R. Bonazzi, P. E. Bourne, *et al.*, "From biomedical cloud platforms to microservices: Next steps in fair data and analysis.," *Scientific data*, vol. 9, no. 1, pp. 553–, Sep. 8, 2022. DOI: 10.1038/s41597-022-01619-5.

[26] M. Kansara, "A comparative analysis of security algorithms and mechanisms for protecting data, applications, and services during cloud migration," *International Journal of Information and Cybersecurity*, vol. 6, no. 1, pp. 164–197, 2022.

[27] A. Mardani, E. K. Zavadskas, H. Fujita, and M. Köppen, "Big data-driven large-scale group decision-making under uncertainty (bigdm-u).," *Applied intelligence (Dordrecht, Netherlands)*, vol. 52, no. 12, pp. 13 341–13 344, Sep. 9, 2022. DOI: 10.1007/s10489-022-04113-y.

[28] J. Sun, B. Zhu, J. Qin, J. Hu, and J. Ma, "Confidentiality-preserving publicly verifiable computation schemes for polynomial evaluation and matrix-vector multiplication," *Security and Communication Networks*, vol. 2018, pp. 1–15, Jun. 21, 2018. DOI: 10.1155/2018/5275132.

[29] M. Wazid, A. K. Das, K.-K. R. Choo, and Y. Park, "Scs-wot: Secure communication scheme for web of things deployment," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 10 411–10 423, Jul. 1, 2022. DOI: 10.1109/jiot.2021.3122007.

[30] M. T. González-Aparicio, A. Ogunyadeka, M. Younas, J. Tuya, and R. Casado, "Transaction processing in consistency-aware user's applications deployed on nosql databases," *Human-centric Computing and Information Sciences*, vol. 7, no. 1, pp. 1–18, Apr. 5, 2017. DOI: 10.1186/s13673-017-0088-3.

[31] J. Li, R. Zhang, and Y. Zheng, "Qos-aware and multi-objective virtual machine dynamic scheduling for big data centers in clouds," *Soft Computing*, vol. 26, no. 19, pp. 10 239–10 252, Jul. 25, 2022. DOI: 10.1007/s00500-022-07327-x.

[32] C. Shen, W. Tong, J.-N. Hwang, and Q. Gao, "Performance modeling of big data applications in the cloud centers," *The Journal of Supercomputing*, vol. 73, no. 5, pp. 2258–2283, Mar. 16, 2017. DOI: 10.1007/s11227-017-2005-y.

[33] B. Bohar, D. Fazekas, M. Madgwick, *et al.*, "Sherlock: An open-source data platform to store, analyze and integrate big data for computational biologists.," *F1000Research*, vol. 10, pp. 409–409, May 21, 2021. DOI: 10.12688/f1000research.52791.3;10.12688/f1000research.52791.2.

[34] E. E. Abel and M. S. A. Latiff, "The utilization of algorithms for cloud internet of things application domains: A review," *Frontiers of Computer Science*, vol. 15, no. 3, pp. 1–27, Jan. 22, 2021. DOI: 10.1007/s11704-019-9056-6.

[35] W. Serrano, "Digital systems in smart city and infrastructure: Digital as a service," *Smart Cities*, vol. 1, no. 1, pp. 134–154, Nov. 6, 2018. DOI: 10.3390/smartcities1010008.

[36] S. Jiang, Y. Qian, H. Tang, *et al.*, "Examining computational thinking processes in modeling unstructured data," *Education and Information Technologies*, vol. 28, no. 4, pp. 4309–4333, Oct. 14, 2022. DOI: 10.1007/s10639-022-11355-3.

[37] G. Andrienko, N. Andrienko, C. Boldrini, *et al.*, "So) big data and the transformation of the city," *International Journal of Data Science and Analytics*, vol. 11, no. 4, pp. 311–340, Mar. 31, 2020. DOI: 10.1007/s41060-020-00207-3.

[38] X. Wei, Q. Sun, X. Liu, C. Yue, S. Y. Liang, and L. Wang, "Research on parallel distributed clustering algorithm applied to cutting parameter optimization," *The International Journal of Advanced Manufacturing Technology*, vol. 120, no. 11-12, pp. 7895–7904, May 5, 2022. DOI: 10.1007/s00170-022-09252-7.

[39] A. Ullah, H. Dagdeviren, R. C. Ariyattu, J. DesLauriers, T. Kiss, and J. Bowden, "Micado-edge: Towards an application-level orchestrator for the cloud-to-edge computing continuum," *Journal of Grid Computing*, vol. 19, no. 4, pp. 47–, Nov. 2, 2021. DOI: 10.1007/s10723-021-09589-5.

[40] A. Khalighifar, B. S. Gotthold, E. Adams, *et al.*, "¡i¿nabat ml¡/i¿: Utilizing deep learning to enable crowd-sourced development of automated, scalable solutions for documenting north american bat populations," *Journal of Applied Ecology*, vol. 59, no. 11, pp. 2849–2862, Sep. 20, 2022. DOI: 10.1111/1365-2664.14280.

[41] N. Vithanwattana, G. Mapp, and C. George, "Developing a comprehensive information security framework for mhealth: A detailed analysis," *Journal of Reliable Intelligent Environments*, vol. 3, no. 1, pp. 21–39, Apr. 27, 2017. DOI: 10.1007/s40860-017-0038-x.

[42] A. N. Richter and T. M. Khoshgoftaar, "Sample size determination for biomedical big data with limited labels," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 9, no. 1, pp. 1–13, Jan. 11, 2020. DOI: 10.1007/s13721-020-0218-0.

[43] Z. Wang, K. Wang, J. Liu, J. Huang, and H. Chen, "Measuring the innovation of method knowledge elements in scientific literature," *Scientometrics*, vol. 127, no. 5, pp. 2803–2827, Mar. 25, 2022. DOI: 10.1007/s11192-022-04350-5.

[44] J. Lever and R. Arcucci, "Sentimental wildfire: A social-physics machine learning model for wildfire nowcasting," *Journal of Computational Social Science*, vol. 5, no. 2, pp. 1427–1465, Jul. 22, 2022. DOI: 10.1007/s42001-022-00174-8.

[45] B. Everman, N. Rajendran, X. Li, and Z. Zong, "Improving the cost efficiency of large-scale cloud systems running hybrid workloads - a case study of alibaba cluster traces," *Sustainable computing : informatics and systems*, vol. 30, pp. 100 528–, Mar. 3, 2021. DOI: 10.1016/j.suscom.2021.100528.

[46] H. Liu, H. Du, D. Zeng, and Q. Tian, "Cloud detection using super pixel classification and semantic segmentation," *Journal of Computer Science and Technology*, vol. 34, no. 3, pp. 622–633, May 10, 2019. DOI: 10.1007/s11390-019-1931-y.

[47] B. Azam, "Big data's evolution: From storage to cloud-driven insights," *Zenodo (CERN European Organization for Nuclear Research)*, Oct. 12, 2022. DOI: 10.5281/zenodo.8415726.

[48] P. Ping, H. Hermjakob, J. S. Polson, P. V. Benos, and W. Wang, "Biomedical informatics on the cloud: A treasure hunt for advancing cardiovascular medicine.," *Circulation research*, vol. 122, no. 9, pp. 1290–1301, Apr. 27, 2018. DOI: 10.1161/circresaha.117.310967.

[49] R. Avula, "Architectural frameworks for big data analytics in patient-centric healthcare systems: Opportunities, challenges, and limitations," *Emerging Trends in Machine Intelligence and Big Data*, vol. 10, no. 3, pp. 13–27, 2018.

[50] F. Yin, X. Jiao, J. Zhou, *et al.*, "Fintech application on banking stability using big data of an emerging economy," *Journal of Cloud Computing*, vol. 11, no. 1, Sep. 14, 2022. DOI: 10.1186/s13677-022-00320-7.

[51] M. Kansara, "A structured lifecycle approach to large-scale cloud database migration: Challenges and strategies for an optimal transition," *Applied Research in Artificial Intelligence and Cloud Computing*, vol. 5, no. 1, pp. 237–261, 2022.

[52] V. Prakash, A. Williams, L. Garg, P. Barik, and R. K. Dhanaraj, "Cloud-based framework for performing digital forensic investigations," *International Journal of Wireless Information Networks*, vol. 29, no. 4, pp. 419–441, Jul. 4, 2022. DOI: 10.1007/s10776-022-00560-z.

[53] Y. Zhu, G. Sun, G. Ding, *et al.*, "Large-scale field phenotyping using backpack lidar and cropquant-3d to measure structural variation in wheat.," *Plant physiology*, vol. 187, no. 2, pp. 716–738, Jul. 16, 2021. DOI: 10.1093/plphys/kiab324.

[54] P. M. Regan and J. Jesse, "Ethical challenges of edtech, big data and personalized learning: Twenty-first century student sorting and tracking," *Ethics and Information Technology*, vol. 21, no. 3, pp. 167–179, Dec. 3, 2018. DOI: 10.1007/s10676-018-9492-2.

[55] V. E. Owen and R. S. Baker, "Fueling prediction of player decisions: Foundations of feature engineering for optimized behavior modeling in serious games," *Technology, Knowledge and Learning*, vol. 25, no. 2, pp. 225–250, Nov. 22, 2018. DOI: 10.1007/s10758-018-9393-9.

[56] M. R. Jabbarpour, A. Marefat, A. Jalooli, and H. Zarrabi, "Could-based vehicular networks: A taxonomy, survey, and conceptual hybrid architecture," *Wireless Networks*, vol. 25, no. 1, pp. 335–354, Jul. 31, 2017. DOI: 10.1007/s11276-017-1563-5.

[57] A. Rahman and A. T. Asyhari, "The emergence of internet of things (iot): Connecting anything, anywhere," *Computers*, vol. 8, no. 2, pp. 40–, May 17, 2019. DOI: 10.3390/computers8020040.

[58] M. S. Kettouch and C. Luca, "Linkd: Element-based data interlinking of rdf datasets in linked data," *Computing*, vol. 104, no. 12, pp. 2685–2709, Jul. 16, 2022. DOI: 10.1007/s00607-022-01107-z.

[59] A. Yilmaz, J. D. Wegner, F. Remondino, T. Fuse, and I. Toschi, "Preface: Technical commission ii," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 52, pp. 7–7, Jun. 28, 2021. DOI: 10.5194/isprs-archives-xliii-b2-2021-7-2021.

[60] F. Lordan, J. Jensen, and R. M. Badia, "Towards mobile cloud computing with single sign-on access," *Journal of Grid Computing*, vol. 16, no. 4, pp. 627–646, Oct. 30, 2017. DOI: 10.1007/s10723-017-9413-3.

[61] Y. Zeng, J. Tan, and C. H. Xia, "Fork and join queueing networks with heavy tails," *ACM SIGMETRICS Performance Evaluation Review*, vol. 46, no. 1, pp. 122–124, Jun. 12, 2018. DOI: 10.1145/3292040.3219668.

[62] M. Chen, J. Yuan, D. Liu, and T. Li, "An adaption scheduling based on dynamic weighted random forests for load demand forecasting," *The Journal of Supercomputing*, vol. 76, no. 3, pp. 1735–1753, Dec. 30, 2017. DOI: 10.1007/s11227-017-2223-3.

[63] Q. Liu, X. Wu, X. Liu, Y. Zhang, and Y. Hu, "Near-data prediction based speculative optimization in a distribution environment," *Mobile Networks and Applications*, vol. 27, no. 6, pp. 2339–2347, Jun. 18, 2022. DOI: 10.1007/s11036-021-01793-7.

[64] X. Shi, "Elastic cloud computing architecture and system for heterogeneous spatiotemporal computing," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-4/W2, pp. 115–119, Oct. 19, 2017. DOI: 10.5194/isprs-annals-iv-4-w2-115-2017.

[65] R. Avula, "Optimizing data quality in electronic medical records: Addressing fragmentation, inconsistencies, and data integrity issues in healthcare," *Journal of Big-Data Analytics and Cloud Computing*, vol. 4, no. 5, pp. 1–25, 2019.

[66] X. Liu, L. Zhang, H. Yu, *et al.*, "Bridging multiscale characterization technologies and digital modeling to evaluate lithium battery full lifecycle," *Advanced Energy Materials*, vol. 12, no. 33, Jun. 15, 2022. DOI: 10.1002/aenm.202200889.

[67] C. Mario, "Recent books and journals articles in public opinion, survey methods, survey statistics, big data, data science, and user experience research. 2019 update," *Survey Practice*, vol. 14, no. 1, pp. 1–6, Dec. 16, 2021. DOI: 10.29115/sp-2021-0013.

[68] H. Yennapusa and R. R. Palle, "An optimal secure and verifiable education content searching scheme for cloud-assisted edge computing," *Scholars Journal of Engineering and Technology*, vol. 6, no. 12, pp. 444–453, Dec. 30, 2018. DOI: 10.36347/sjet.2018.v06i12.011.

[69] A. Sharma and K. M. Goolsbey, "Simulation-based approach to efficient commonsense reasoning in very large knowledge bases," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1360–1367.

[70] O. Malomo, D. B. Rawat, and M. Garuba, "Next-generation cybersecurity through a blockchain-enabled federated cloud framework," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 5099–5126, May 2, 2018. DOI: 10.1007/s11227-018-2385-7.

[71] G. Somani, X. Zhao, S. N. Srirama, and R. Buyya, "Integration of cloud, internet of things, and big data analytics," *Software: Practice and Experience*, vol. 49, no. 4, pp. 561–564, Nov. 11, 2018. DOI: 10.1002/spe.2664.

[72] M. M. Baig and S. Sonekar, "An approach toward standardization of emr with efficient accessibility of data on demand," *International Journal of Next-Generation Computing*, Nov. 26, 2021. DOI: 10.47164/ijngc.v12i5.464.

[73] C. Gonzalez, "Connected automobiles the cloud drives with you," *ATZelektronik worldwide*, vol. 13, no. 3, pp. 50–52, May 27, 2018. DOI: 10.1007/s38314-018-0033-x.

[74] B. Chen, Y. Ye, C. Tong, J. Deng, K. Wang, and Y. Hong, "A novel big data mining framework for reconstructing large-scale daily maiac aod data across china from 2000 to 2020," *GIScience & Remote Sensing*, vol. 59, no. 1, pp. 670–685, Mar. 21, 2022. DOI: 10.1080/15481603.2022.2051382.

[75] T. C. Vance, M. Wengren, E. Burger, *et al.*, "From the oceans to the cloud: Opportunities and challenges for data, models, computation and workflows," *Frontiers in Marine Science*, vol. 6, May 21, 2019. DOI: 10.3389/fmars.2019.00211.

[76] L. Chen, C. Zhai, L. Wang, X. Hu, and X. Huang, "Modular structure construction progress scenario: A case study of an emergency hospital to address the covid-19 pandemic," *Sustainability*, vol. 14, no. 18, pp. 11 243–11 243, Sep. 8, 2022. DOI: 10.3390/su141811243.

[77] A. Bonfrer, E. Podolski, and M. Veeraraghavan, "The effect of cloud cover on consumer retail spending: Evidence from big data," *SSRN Electronic Journal*, Jan. 1, 2022. DOI: 10.2139/ssrn.4250806.

[78] M. Kansara, "A framework for automation of cloud migrations for efficiency, scalability, and robust security across diverse infrastructures," *Quarterly Journal of Emerging Technologies and Innovations*, vol. 8, no. 2, pp. 173–189, 2023.

[79] M. A. N. Saif, S. K. Niranjan, and H. D. E. Al-ariki, "Efficient autonomic and elastic resource management techniques in cloud environment: Taxonomy and analysis," *Wireless Networks*, vol. 27, no. 4, pp. 2829–2866, Apr. 22, 2021. DOI: 10.1007/s11276-021-02614-1.

[80] C. Ferreira and M. Janssen, "Shaping the future of shared services centers: Insights from a delphi study about ssc transformation towards 2030," *Journal of the Knowledge Economy*, vol. 14, no. 4, pp. 4828–4847, Oct. 10, 2022. DOI: 10.1007/s13132-022-01072-0.

[81] K. K. Punjani, K. Mahadevan, A. Gunasekaran, V. V. R. Kumar, and S. Joshi, "Cloud computing in agriculture: A bibliometric and network visualization analysis," *Quality & Quantity*, vol. 57, no. 4, pp. 3849–3883, Oct. 6, 2022. DOI: 10.1007/s11135-022-01535-1.

[82] D.-K. Kim, B. Lee, D. Kim, and H. Jeong, "Multi-label classification of historical documents by using hierarchical attention networks," *Journal of the Korean Physical Society*, vol. 76, no. 5, pp. 368–377, Mar. 13, 2020. DOI: 10.3938/jkps.76.368.