

AI-Based Congestion Prediction and Traffic Engineering for High-Performance Computing Interconnects

Leandra Codina

Department of Computer Science, University of Central Florida, Orlando, FL, USA.
contactleandra@ucf.edu

Abstract

High-performance computing interconnects have become critical substrates for artificial intelligence training, scientific simulation, and data-intensive service workloads. As these systems scale, congestion ceases to be a marginal performance issue and becomes a structural challenge that couples network architecture, workload behavior, scheduling policy, energy use, and organizational governance. This paper examines artificial intelligence based congestion prediction and traffic engineering for high-performance computing interconnects from a system-level perspective. It argues that prediction models cannot be understood merely as algorithmic improvements; they are embedded in instrumentation regimes, control loops, topology choices, and operational policies that determine whether congestion management is effective, fair, and sustainable. The discussion first situates interconnect congestion within the evolution of data center and supercomputing fabrics, then analyzes workload characteristics that produce burstiness, incast, and tail latency. It subsequently evaluates how machine learning can be positioned across endpoints, switches, and centralized controllers, and how predictive signaling can inform routing, scheduling, and flow control. The paper also considers deployment trade-offs involving telemetry overhead, model drift, multi-tenant fairness, robustness under failure, energy consumption, and policy constraints. Case illustrations from cloud data centers, wide-area traffic engineering, and large-scale scientific computing are used to compare governance models and operational incentives. The conclusion emphasizes that AI-based congestion prediction is most credible when treated as a socio-technical capability rather than a standalone model, requiring observability, accountability, fallback mechanisms, and careful alignment with the physical and institutional limits of high-performance computing infrastructure.

Keywords

high-performance computing, interconnect congestion, traffic engineering, congestion prediction, machine learning, data center networks, socio-technical systems, sustainability.

1. Introduction

High-performance computing interconnects are no longer confined to tightly coupled scientific machines. They increasingly support artificial intelligence training, inference, large-scale simulation, real-time analytics, and hybrid cloud workflows. This expansion has transformed interconnect design from a specialized engineering concern into a central determinant of system capability. In conventional data center networks, congestion has long been recognized as a consequence of oversubscription, bursty traffic, and uneven demand [1]. Measurement studies have shown that production data center traffic exhibits complex flow-size distributions, synchronized bursts, and hot spots that challenge static provisioning

assumptions [2]. Similar investigations in large-scale clusters have demonstrated that traffic patterns are shaped by application semantics, scheduling decisions, and storage access behavior rather than by aggregate averages alone [3]. The economic pressure to sustain high utilization while controlling cost further intensifies these tensions [4]. For high-performance computing, the stakes are even higher because collective operations, checkpointing, and parameter synchronization can produce tightly correlated communication phases that propagate congestion across many nodes within a short interval.

Artificial intelligence workloads amplify these dynamics. Training jobs often involve frequent all-reduce, all-gather, and point-to-point exchanges whose timing is coupled to computation stages. When one stage slows, the entire job may stall, producing cascading inefficiency. The well-known tail-at-scale problem illustrates how rare slow components can dominate user-visible performance in large distributed systems [9]. Congestion is therefore not simply a throughput problem; it is a latency and synchronization problem with direct consequences for job completion time, resource utilization, and energy efficiency. As accelerators and specialized processors become more powerful, the network increasingly determines whether compute capacity can be kept busy [10]. This paper examines how artificial intelligence based congestion prediction and traffic engineering can be integrated into high-performance computing interconnects without reducing the problem to model accuracy alone. It emphasizes architecture, deployment, governance, fairness, robustness, sustainability, and policy implications.

The central argument is that AI-based congestion prediction should be understood as a socio-technical capability. Prediction models depend on telemetry, which depends on hardware support, which depends on procurement and standards. Control actions depend on trust, which depends on explainability, fallback behavior, and institutional accountability. Traffic engineering decisions affect tenants differently, and therefore raise questions about fairness, priority, and acceptable degradation. In high-performance computing environments, these questions are especially salient because resources are often shared among scientific communities, government agencies, universities, and commercial partners. A prediction system that improves average throughput while harming a minority of latency-sensitive jobs may be technically successful but operationally unacceptable. The sections that follow develop this perspective by examining interconnect evolution, congestion phenomena, model placement, control loops, deployment trade-offs, governance, robustness, sustainability, and comparative cases.

2. High-Performance Computing Interconnects as a Socio-Technical Infrastructure

Modern high-performance computing interconnects are complex assemblies of switches, links, protocols, network interface cards, optical components, and management software. Their architectural choices reflect competing goals: low latency, high bandwidth, scalability, fault tolerance, cost control, and compatibility with legacy software. Fat-tree and related topologies were developed to provide predictable bisection bandwidth using commodity components [1]. Hybrid electrical and optical designs sought to reduce energy and increase reconfiguration flexibility [5]. Software-defined networking introduced centralized control and programmability, enabling operators to adjust routing and policy dynamically [20]. In practice, however, these technical capabilities are embedded in organizational processes. Procurement cycles, vendor ecosystems, standards bodies, and facilities planning all shape what can be deployed. An interconnect is therefore not merely a technical artifact; it is a socio-technical

infrastructure whose performance depends on governance, maintenance, and operational knowledge.

The physical layer remains consequential. Optical interconnects, wavelength switching, and hollow-core fiber research show that latency and synchronization can be improved through material and photonic innovation [17]. Such advances matter because congestion control operates within a latency budget: when propagation delay and switching delay are lower, feedback loops can respond more quickly and queue buildup can be reduced. Yet physical-layer improvements do not eliminate congestion by themselves. They alter the boundary conditions within which traffic engineering operates. If applications become more bursty, or if oversubscription increases to control cost, congestion can still emerge. Similarly, exascale computing ambitions intensify the need for coordinated design across processors, memory, storage, and network [11]. The interconnect cannot be optimized in isolation because application mapping, checkpointing strategy, and data locality decisions determine the traffic that the network must carry.

The socio-technical nature of interconnects also appears in benchmarking and evaluation. Scientific benchmarking of parallel systems has long emphasized the danger of reporting isolated peak numbers without context [19]. The same caution applies to AI-based congestion prediction. A model that performs well on one workload trace may fail on another because the trace reflects a particular scheduling policy, tenant mix, or hardware generation. Moreover, operational teams may resist systems they cannot interpret or override. Trust is built through transparent telemetry, reproducible evaluation, and clear escalation paths when automated decisions are uncertain. In high-performance computing, where jobs may run for days or weeks, a congestion prediction error can waste substantial compute time and funding. Consequently, deployment must be treated as an institutional process, not just a software update.

3. Congestion Phenomena in High-Performance Computing and AI Workloads

Congestion in high-performance computing interconnects arises from the interaction of workload communication patterns, topology, routing, flow control, and buffer management. Data center transport research produced important mechanisms such as Data Center TCP, which uses explicit congestion notification to maintain low queue occupancy [6]. Remote direct memory access deployments introduced new congestion control challenges because kernel bypass and hardware offload change the visibility of flow state [7]. Delay-based approaches such as TIMELY showed that round-trip time signals can support fine-grained control in high-bandwidth environments [8]. These mechanisms improve responsiveness, but they do not predict congestion before it occurs. They react to feedback, which may arrive too late when many nodes communicate simultaneously. Artificial intelligence workloads make this limitation more visible because their communication phases can be highly synchronized and short-lived.

Collective operations are a major source of congestion. In distributed deep learning, gradient synchronization can create periods of intense many-to-many communication [12]. Climate analytics and other scientific AI workloads combine large data movements with iterative computation, producing complex traffic matrices that shift over time [13]. When many ranks send to a small number of destinations, incast can overwhelm switch buffers. When routing is uneven, some links become hot while others remain underused. When storage or checkpointing is involved, traffic can collide with training traffic and disrupt latency-sensitive operations. The tail-at-scale effect means that even a small number of congested paths can

delay an entire job [9]. This is why prediction is attractive: if congestion can be anticipated, routing and scheduling decisions can be adjusted before queues grow and before synchronized workers stall.

However, prediction is difficult because congestion is emergent. It is not a property of a single flow but of a dynamic system in which flows, switches, and applications interact. Workload behavior may change because of algorithmic choices, hyperparameter tuning, failure recovery, or user decisions. A training job may alter its batch size, parallelism strategy, or checkpoint frequency. A simulation may enter a new phase with different communication requirements. The resulting traffic matrix is non-stationary and partially observable. Prediction models must therefore handle concept drift, missing telemetry, and delayed labels. They must also distinguish between short-term fluctuations and persistent structural bottlenecks. A model that predicts transient microbursts may be useful for flow control but less useful for routing. A model that predicts long-term hot spots may inform topology reconfiguration or job placement. The design question is not whether prediction is possible, but what decisions it should inform and at what time scale.

4. AI-Based Congestion Prediction: Models, Data, and Architectural Placement

Machine learning for networking has evolved from early classification and regression tasks toward deep learning and reinforcement learning approaches [14]. Surveys of the field emphasize that data quality, feature engineering, and deployment constraints often matter as much as model architecture [15]. Deep learning has been applied to traffic control, anomaly detection, and resource management, but operational adoption remains uneven because network environments are unforgiving of errors [16]. In high-performance computing interconnects, prediction models can be placed at endpoints, switches, or centralized controllers. Endpoint-based prediction can use application-level knowledge, such as job phase, communication pattern, and rank placement. Switch-based prediction can use queue occupancy, link utilization, and per-flow statistics at line rate. Centralized prediction can aggregate telemetry across the fabric and optimize global objectives. Each placement has trade-offs in latency, overhead, visibility, and governance.

Endpoint placement offers rich context because the application knows when a collective operation will begin. For example, a training framework can signal an impending all-reduce, allowing the network to prepare routing or rate limits. However, endpoint prediction may lack a global view. It may not know about competing jobs on other nodes. Switch placement offers a more complete local view but is constrained by hardware resources and telemetry bandwidth. Centralized placement can combine information from many switches and endpoints, but it introduces communication delay and a single point of analytical authority. A hybrid architecture may be most practical: lightweight local inference for immediate control, and centralized learning for long-term policy improvement. This mirrors broader trends in software-defined networking, where centralized control coexists with distributed data planes [20].

Data is the foundation of prediction. Telemetry can include packet and flow counters, queue depths, link utilization, round-trip time, explicit congestion notification marks, and application-level events. High-fidelity telemetry is expensive. It consumes switch resources, network bandwidth, and storage. Sampling reduces overhead but may miss microbursts. Aggregation reduces volume but may hide tail behavior. Privacy and security concerns also arise when telemetry includes tenant identifiers or application details. In multi-tenant high-performance computing environments, tenants may be sensitive about exposing

communication patterns. Governance must therefore address what data is collected, how long it is retained, who can access it, and how it is used. A prediction system that is technically accurate but institutionally unacceptable will not be deployed at scale.

Model choice is secondary to problem formulation. Supervised learning can predict future queue occupancy or link utilization if labeled data is available. Unsupervised learning can detect anomalies or cluster traffic patterns. Reinforcement learning can learn control policies through interaction, but it requires safe exploration and simulation. Graph neural networks can represent topology and flow relationships. Sequence models can capture temporal dependencies. Yet high-performance computing interconnects are safety-critical in the sense that poor control can cause job failures, data corruption, or deadline misses. Therefore, predictive models must be paired with conservative fallback mechanisms. They should propose actions rather than dictate them, or operate within bounded authority. They should expose uncertainty and allow operators to understand why a decision was made. The most credible systems will combine prediction with classical control, using AI to anticipate and classical mechanisms to guarantee stability.

5. Traffic Engineering and Control Loops

Traffic engineering in high-performance computing interconnects involves routing, rate control, scheduling, and topology reconfiguration. Predictive models can inform each of these functions. Predictive routing can shift flows away from links that are likely to become congested. Predictive rate control can reduce sending rates before queues overflow. Predictive scheduling can delay or reorder jobs to avoid synchronized communication phases. Predictive topology reconfiguration can adjust optical circuits or wavelength assignments based on expected demand [5]. These actions operate at different time scales and have different risks. Routing changes can cause packet reordering and transient loops if not carefully managed. Rate reductions can slow applications even when congestion does not materialize. Job delays can violate fairness or service-level expectations. Topology reconfiguration can disrupt existing flows. Therefore, prediction must be integrated into control loops with clear stability criteria and rollback procedures.

The control loop should include observation, prediction, decision, action, and verification. Observation gathers telemetry. Prediction estimates future congestion with uncertainty. Decision selects an action based on policy, fairness constraints, and risk tolerance. Action applies the change through routing tables, rate limiters, schedulers, or optical switches. Verification checks whether the action improved conditions or caused harm. This loop is not merely technical; it encodes institutional priorities. For example, a policy might prioritize deadline-driven jobs over best-effort jobs, or protect small flows from starvation. It might cap the authority of automated agents during business hours or when human operators are unavailable. It might require logging of all automated decisions for audit. These choices determine whether AI-based traffic engineering is perceived as legitimate.

Classical congestion control provides a foundation. Data Center TCP, RDMA congestion control, and TIMELY demonstrate that feedback-based mechanisms can manage queues effectively [6][7][8]. AI-based prediction should complement, not replace, these mechanisms. A predictive layer can tune parameters, select among available paths, or anticipate phase changes. Classical control can handle immediate backpressure and guarantee fairness. This division of labor is important because machine learning models can be brittle under distribution shift. If a new workload, hardware failure, or security event changes traffic patterns, a learned policy may produce poor actions. A hybrid design can detect out-of-

distribution conditions and fall back to conservative behavior. It can also use simulation and digital twins to test policies before deployment, reducing the risk of operational surprises.

Optical and physical-layer innovations interact with traffic engineering. Wavelength switching and hollow-core fiber can reduce latency and improve synchronization, which changes the feasible response time of control loops [17]. Lower latency allows faster feedback, but it also means that congestion can form more quickly if applications are tightly synchronized. Traffic engineering must therefore consider the entire system, including application scheduling and collective communication libraries. In high-performance computing, the Message Passing Interface and related libraries mediate many communication patterns. If those libraries expose phase information or allow adaptive algorithms, the network can coordinate more effectively. If they do not, prediction must infer intent from traffic alone. The more the system is designed for observability and cooperation, the more effective AI-based congestion management becomes.

6. Deployment, Governance, and Fairness

Deployment of AI-based congestion prediction raises questions of governance. Who owns the telemetry? Who is accountable when an automated decision causes a job slowdown? How are tenants informed about data collection? How are conflicts resolved between different communities? In cloud data centers, the operator typically has broad authority and can centralize policy [4]. In high-performance computing centers, governance may be distributed across funding agencies, university leadership, vendor contracts, and user committees. This fragmentation can slow adoption but also provides checks and balances. A prediction system that is developed without stakeholder engagement may be technically sophisticated yet politically fragile. Governance must therefore include transparency, appeal mechanisms, and clear boundaries on automated action.

Fairness is a central concern. Congestion prediction can optimize aggregate throughput while disadvantaging specific workloads. For example, a model might learn to deprioritize small flows because they contribute little to total bytes, even though they are latency-sensitive. It might favor large training jobs because they dominate resource usage, while starving interactive debugging sessions. It might allocate bandwidth based on historical priority, reinforcing existing inequalities. Fairness in high-performance computing is not just about equal bandwidth; it is about equitable access to scientific opportunity. A job that is delayed may miss a conference deadline, a grant milestone, or a time-sensitive observation. Thus, traffic engineering policies should be evaluated not only by network metrics but also by impact on users and missions. Fairness constraints should be explicit in the optimization objective and auditable after deployment.

Multi-tenancy intensifies these challenges. When multiple projects share an interconnect, their traffic is mixed and may be difficult to attribute. Telemetry may reveal confidential information about data sizes, model architectures, or collaboration patterns. Prediction models trained on one tenant mix may not generalize to another. Security is also a concern: an adversary might manipulate traffic to confuse the predictor or exploit automated control actions. Therefore, deployment should include access control, data minimization, adversarial testing, and anomaly detection. The system should be robust to poisoning, evasion, and feedback manipulation. In high-performance computing, security and performance are often treated as separate concerns, but AI-based traffic engineering brings them together. A compromised predictor could become a denial-of-service vector or a covert channel.

7. Robustness, Sustainability, and Policy Implications

Robustness is essential because high-performance computing systems are expected to run continuously and reliably. Components fail; links degrade; workloads change. A prediction system that only works under nominal conditions is not operationally useful. Robustness can be improved through redundancy, uncertainty quantification, ensemble methods, and conservative fallback policies. It can also be improved through graceful degradation: if telemetry is missing, the system reverts to classical control; if a model is uncertain, it takes no action; if an action is harmful, it is quickly reversed. Verification mechanisms should monitor whether predicted congestion actually occurs and whether control actions improve outcomes. Over time, the system can learn from these outcomes, but it must avoid feedback loops that reinforce biased or unstable behavior.

Sustainability adds another dimension. Data centers and supercomputers consume significant energy, and network congestion can increase energy use by prolonging job runtimes and keeping resources idle [11]. AI-based traffic engineering can reduce energy consumption by improving utilization and avoiding unnecessary retransmissions. However, prediction models themselves consume energy. They require telemetry collection, computation, and cooling. The net sustainability benefit depends on whether congestion savings exceed model overhead. In some cases, a simple heuristic may be more energy-efficient than a complex model. Therefore, evaluation should include energy per completed job, carbon intensity of electricity, and embodied emissions of additional hardware. Policy implications follow: procurement rules may favor systems with telemetry support, standards may require interoperability, and funding agencies may demand sustainability reporting. These requirements shape what kinds of AI-based congestion management are feasible and desirable.

Policy also affects resilience and equity. Governments and institutions may impose data sovereignty rules, export controls, or critical infrastructure protections that restrict where telemetry can be stored and who can analyze it. In collaborative research, different partners may have different legal constraints. A centralized prediction system that crosses jurisdictional boundaries may be difficult to deploy. Federated learning or local inference may offer alternatives, but they introduce coordination challenges. Policy can also promote openness and reproducibility. If models are trained on proprietary traces, independent verification is difficult. If control policies are opaque, users cannot challenge decisions. A sustainable and legitimate ecosystem requires a balance between innovation, privacy, security, and accountability.

8. Case Illustrations and Cross-Domain Comparisons

Cloud data center networks provide one comparative case. Their operators control the physical infrastructure, the workload scheduler, and the tenant platform, enabling tight integration between congestion prediction and traffic engineering [1][4]. They can deploy telemetry across switches and endpoints, and they can adjust routing and rate limits dynamically [20]. However, cloud environments are also highly diverse and commercially sensitive. Predictions must handle many different applications and service-level agreements. High-performance computing centers differ because jobs are often batch-oriented, users are researchers, and scheduling is governed by allocation policies. The network may be shared by simulations, AI training, and data analysis. Prediction models must therefore respect scientific priorities and allocation agreements, not just commercial performance metrics.

Wide-area traffic engineering offers another comparison. Software-defined wide-area networks have demonstrated the value of centralized optimization and dynamic path selection [20]. They manage inter-datacenter traffic with global objectives such as cost, latency, and utilization. High-performance computing interconnects operate at shorter distances but with more tightly coupled communication patterns. The time scales are faster, and the consequences of jitter are more severe for collective operations. Lessons from wide-area traffic engineering include the importance of measurement, simulation, and gradual rollout. However, direct translation is limited because high-performance computing applications often have synchronized phases that do not resemble elastic web traffic. Prediction must be informed by application semantics and parallel programming models [12].

Scientific AI workloads provide a third case. Training climate models or large language models involves massive data movement and iterative synchronization [13]. These workloads may run for days and involve thousands of accelerators [10]. Congestion can waste energy and delay scientific results. AI-based prediction can help by anticipating communication phases, rebalancing flows, and coordinating checkpointing. Yet these workloads also have unique constraints: they may be tightly coupled to specific libraries, they may require deterministic performance for reproducibility, and they may have strict checkpoint deadlines. Benchmarking practices must therefore be rigorous and contextual [19]. A prediction system should be evaluated across multiple workloads, scales, and failure scenarios. It should report not only average improvement but also worst-case behavior, fairness, and operational overhead.

9. Conclusion

AI-based congestion prediction and traffic engineering for high-performance computing interconnects represent a promising but demanding direction. The technical opportunity is clear: prediction can shift congestion management from reactive control to anticipatory coordination, potentially reducing tail latency, improving utilization, and lowering energy consumption. The challenge is equally clear: prediction models depend on telemetry, control authority, and institutional trust. They must operate in non-stationary environments with incomplete information, adversarial risks, and multiple stakeholders. They must respect fairness, privacy, security, and sustainability constraints. They must degrade gracefully when conditions change. Therefore, the most credible path forward is not to treat AI as a replacement for established congestion control but as a complementary layer within a carefully governed control loop.

Future research should focus on hybrid architectures that combine local inference with centralized learning, uncertainty-aware decision making, and verifiable fallback mechanisms. It should develop benchmarks that capture realistic high-performance computing and AI workload phases, including failures and multi-tenancy. It should study the energy and carbon implications of both congestion and prediction. It should examine governance models that allow operators, users, and policymakers to audit automated decisions. Finally, it should recognize that interconnect performance is inseparable from application design, scheduling, and physical infrastructure. Optical advances, protocol improvements, and AI-based prediction must be co-designed rather than optimized in isolation. Only then can congestion prediction become a durable capability that strengthens high-performance computing as a shared scientific and economic resource.

References

1. Al-Fares, M., Loukissas, A., & Vahdat, A. (2008). A scalable, commodity data center network architecture. *ACM SIGCOMM Computer Communication Review*, 38(4), 63-74. <https://doi.org/10.1145/1402946.1402967>
2. Benson, T., Akella, A., & Maltz, D. A. (2010). Network traffic characteristics of data centers in the wild. *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, 267-280. <https://doi.org/10.1145/1879141.1879175>
3. Kandula, S., Sengupta, S., Greenberg, A., Patel, P., & Chaiken, R. (2009). The nature of data center traffic: Measurements and analysis. *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, 202-208. <https://doi.org/10.1145/1644893.1644918>
4. Greenberg, A., Hamilton, J., Maltz, D. A., & Patel, P. (2008). The cost of a cloud: Research problems in data center networks. *ACM SIGCOMM Computer Communication Review*, 39(1), 68-73. <https://doi.org/10.1145/1496091.1496103>
5. Farrington, N., Porter, G., Radhakrishnan, S., Bazzaz, H. H., Subramanya, V., Fainman, Y., Papen, G., & Vahdat, A. (2010). Helios: A hybrid electrical/optical switch architecture for modular data centers. *ACM SIGCOMM Computer Communication Review*, 40(4), 339-350. <https://doi.org/10.1145/1851182.1851223>
6. Alizadeh, M., Greenberg, A., Maltz, D. A., Padhye, J., Patel, P., Prabhakar, B., Sengupta, S., & Sridharan, M. (2010). Data center TCP (DCTCP). *ACM SIGCOMM Computer Communication Review*, 40(4), 63-74. <https://doi.org/10.1145/1851182.1851192>
7. Zhu, Y., Eran, H., Firestone, D., Guo, C., Lipshteyn, M., Liron, Y., Padhye, J., Raindel, S., Yahia, M. H., & Zhang, M. (2015). Congestion control for large-scale RDMA deployments. *ACM SIGCOMM Computer Communication Review*, 45(4), 523-536. <https://doi.org/10.1145/2785956.2787484>
8. Mittal, R., Shpiner, A., Panda, A., Zahavi, E., Krishnamurthy, A., Ratnasamy, S., & Shenker, S. (2015). TIMELY: RTT-based congestion control for the datacenter. *ACM SIGCOMM Computer Communication Review*, 45(4), 537-550. <https://doi.org/10.1145/2785956.2787510>
9. Dean, J., & Barroso, L. A. (2013). The tail at scale. *Communications of the ACM*, 56(2), 74-80. <https://doi.org/10.1145/2408776.2408794>
10. Jouppe, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., ... & Yoon, D. H. (2017). In-datacenter performance analysis of a tensor processing unit. *ACM SIGARCH Computer Architecture News*, 45(2), 1-12. <https://doi.org/10.1145/3079856.3080246>
11. Reed, D. A., & Dongarra, J. (2015). Exascale computing and big data. *Communications of the ACM*, 58(7), 56-68. <https://doi.org/10.1145/2699414>
12. Ben-Nun, T., & Hoefler, T. (2019). Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *ACM Computing Surveys*, 52(4), 1-43. <https://doi.org/10.1145/3320060>
13. Kurth, T., Treichler, S., Romero, J., Mudigonda, M., Luehr, N., Phillips, E., Mahesh, A., Mattingly, M., Podhorszki, N., Ruiter, J., & Houston, M. (2018). Exascale deep learning

for climate analytics. SC18: International Conference for High Performance Computing, Networking, Storage and Analysis, 649-660. <https://doi.org/10.1109/SC.2018.00054>

14. Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., & Caicedo, O. M. (2018). A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(1), 1-99. <https://doi.org/10.1186/s13174-018-0087-2>
15. Wang, M., Cui, Y., Wang, X., Xiao, S., & Jiang, J. (2018). Machine learning for networking: Workflow, advances and opportunities. *IEEE Network*, 32(2), 92-99. <https://doi.org/10.1109/MNET.2017.1700200>
16. Fadlullah, Z. M., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T., & Mizutani, K. (2017). State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Communications Surveys & Tutorials*, 19(4), 2432-2455. <https://doi.org/10.1109/COMST.2017.2707140>
17. Zhou, Zichuan, et al. "Low-latency wavelength-switched clock-synchronized intra-data center interconnects enabled by hollow core fiber." *Optics Express* 31.15 (2023): 24739-24748.
18. Zhang, C., Patras, P., & Haddadi, H. (2019). Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(3), 2224-2287. <https://doi.org/10.1109/COMST.2019.2904897>
19. Hoefler, T., & Belli, R. (2015). Scientific benchmarking of parallel computing systems: Twelve ways to tell the masses when reporting performance results. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 1-12. <https://doi.org/10.1145/2807591.2807644>
20. Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1), 14-76. <https://doi.org/10.1109/JPROC.2014.2371999>