SOCIAL AND MANAGEMENT SCIENCES RESEARCH JOURNAL

www.smsrj.com

p-ISSN: 2791-0415 **e-ISSN:** 2791-0423

Volume:4 Issue: 02(2026)

Cross-layer Scheduling Optimization for Cloud Services Leveraging Predictive Application Traffic Models

***Stanley Poon*[1], *Ka-Man Tsang*[2]**

[1]*Department of Computer Science and Technology, Tsinghua University, School of Aerospace Engineering, Tsinghua University, Beijing, China*

[2]*Department of Computer Science, Faculty of Commerce, Hong Kong Shue Yan University, Hong Kong, Hong Kong SAR, China*

Abstract:

The rapid proliferation of latency-sensitive and bandwidth-intensive applications has placed unprecedented demands on cloud computing infrastructures. Traditional resource allocation mechanisms, which typically operate within isolated layers of the network and computing stack, often fail to achieve optimal performance due to their reactive nature and lack of inter-layer visibility. This paper presents a comprehensive investigation into cross-layer scheduling for cloud services, augmented by predictive application traffic modeling. By integrating machine learning-based traffic forecasting with a joint optimization framework spanning the application, network, and computing layers, the proposed system transitions from reactive resource provisioning to proactive capacity planning. We detail the system architecture, mathematical formulation of the cross-layer optimization problem, and a heuristic algorithm designed to minimize end-to-end latency while maximizing resource utilization. Extensive simulation evaluations using real-world cloud cluster traces demonstrate that the predictive cross-layer approach significantly reduces Service Level Agreement violations and improves overall throughput compared to conventional single-layer and reactive scheduling paradigms. The findings underscore the necessity of holistic, predictive resource management in next-generation cloud environments, paving the way for more resilient and efficient infrastructure capable of handling highly volatile application workloads.

Keywords: *Cloud Computing, Cross-layer Scheduling, Traffic Prediction, Resource Allocation, Machine Learning*

1. Introduction

1.1 Evolution of Cloud Computing Demands

Cloud computing has evolved from a simple remote storage and processing paradigm into a highly complex, ubiquitous infrastructure that underpins the modern digital economy. With the advent of the Internet of Things, autonomous systems, high-definition video streaming, and

augmented reality applications, the demands placed on cloud data centers and their interconnecting networks have grown exponentially. These contemporary applications are characterized not only by their massive volume of data generation but also by their stringent requirements for ultra-low latency, high reliability, and guaranteed bandwidth. Traditional cloud computing architectures were primarily designed for best-effort delivery and bulk data processing, where slight delays in data transmission or computation were generally acceptable. However, in the current landscape, a delay of merely a few milliseconds can lead to severe degradation in user experience or, in the case of autonomous driving and industrial control systems, catastrophic failures. Yang and colleagues examine joint-Aware video processing with Edge-Cloud collaboration for DNN inference [1]. To meet these stringent requirements, cloud service providers must continuously optimize the allocation of both computing resources, such as central processing units and memory, and network resources, including bandwidth and routing paths. Historically, the management of these resources has been compartmentalized. Computing resources are typically managed by a hypervisor or container orchestration platform at the data center level, while network resources are controlled by software-defined networking controllers or traditional routing protocols. This separation of concerns, derived from the standard Open Systems Interconnection model, creates information silos. When the application layer experiences a sudden surge in traffic, the network layer may dynamically reroute packets to avoid congestion, but if the destination computing nodes are already saturated, the end-to-end performance will still suffer. Therefore, isolated resource management is fundamentally inadequate for highly dynamic and interconnected cloud ecosystems. Another relevant line of investigation concerns hyperbolic user identity linkage across social networks [2].

1.2 The Imperative for Predictive Cross-layer Scheduling

The limitations of isolated resource management necessitate a paradigm shift towards cross-layer scheduling. Cross-layer design breaks down the strict boundaries between different architectural layers, allowing for the exchange of telemetry data and control signals across the application, transport, network, and physical resource tiers. By establishing a holistic view of the system state, a cross-layer scheduler can make globally optimal decisions. For instance, if an application layer signals an impending increase in video rendering requests, the cross-layer scheduler can simultaneously provision additional virtual machines at the edge data center and allocate guaranteed high-bandwidth, low-latency network slices to connect the users to those specific compute nodes. The literature also considers leveraging univariate models and Layer-Specific and Network-Wide dual learning [3]. However, even with cross-layer visibility, scheduling mechanisms that rely purely on current system states are inherently reactive. Reactive systems only initiate resource provisioning after a threshold has been breached or a congestion event has occurred. Given the non-negligible time required to boot virtual machines, initialize containers, or reconfigure network paths, reactive systems inevitably expose users to transient periods of degraded performance. To circumvent this limitation, predictive modeling must be integrated into the scheduling framework. By leveraging historical telemetry data and advanced machine learning algorithms, it is possible to accurately forecast application traffic patterns and resource utilization trends before they materialize. Integrating predictive application traffic models with cross-layer scheduling enables proactive resource allocation, effectively eliminating the provisioning latency and ensuring that the infrastructure is always prepared for the incoming workload. This paper explores the intersection of these two critical methodologies, offering a detailed framework for implementing predictive cross-layer scheduling in modern cloud environments. For a broader

comparison, related research addresses jamming and eavesdropping defense scheme based on deep reinforcement learning in autonomous vehicle networks [4].

2. Literature Review and Related Work

2.1 Traffic Prediction Methodologies in Cloud Networks

The ability to accurately forecast network traffic and application demand is a cornerstone of proactive cloud resource management. Early approaches to traffic prediction heavily relied on classical statistical methods, most notably the Autoregressive Integrated Moving Average model and its variants. These statistical models are well-suited for capturing linear dependencies and seasonal trends in relatively stable environments. However, cloud traffic is notoriously bursty, highly non-linear, and influenced by a multitude of complex, interrelated factors such as diurnal human behavior, marketing events, and sudden shifts in global network routing. Consequently, traditional linear models often fail to capture the high-frequency volatility characteristics of modern cloud workloads. Another relevant line of investigation concerns full-Duplex NOMA-Enabled integrated sensing and communication: joint transmit and receive beamforming optimization [5]. To address the shortcomings of statistical methods, researchers have increasingly turned to machine learning and deep learning techniques. Support Vector Regression and Random Forest algorithms demonstrated improved accuracy in handling non-linear data patterns. More recently, recurrent neural networks, particularly Long Short-Term Memory networks, have become the standard for time-series forecasting in cloud environments. Long Short-Term Memory networks are uniquely capable of maintaining a memory of long-term dependencies in sequential data, making them highly effective at recognizing complex, recurring patterns in network traffic. Furthermore, the introduction of attention mechanisms and Transformer architectures has revolutionized the field. Transformers can process entire sequences of traffic data in parallel while weighing the importance of different time steps, thereby achieving state-of-the-art prediction accuracy with significantly reduced training times compared to sequential recurrent networks. This paper builds upon these advancements by assuming the deployment of a highly accurate sequence-to-sequence deep learning model as the foundational traffic prediction engine. Sequence-aware straggler coding exploits completed portions of distributed matrix work [6]. It could serve as a lower-level complement to predictive cloud scheduling when a forecast is wrong about an individual worker's completion time. Any scheduling comparison should include the coded redundancy and recovery delay instead of counting only the nominal task allocation.

2.2 Existing Scheduling and Resource Allocation Frameworks

Resource scheduling in cloud computing has been extensively studied, typically categorized into compute-centric scheduling and network-centric scheduling. Compute-centric approaches, such as those implemented by Kubernetes or standard cloud hypervisors, focus on placing workloads onto physical machines in a way that maximizes central processing unit utilization and minimizes energy consumption, often utilizing bin-packing algorithms or heuristic optimizations [7]. These methods generally treat the network as a transparent entity with infinite capacity, which is a flawed assumption in heavily loaded data centers or wide-area multi-cloud deployments. On the other hand, network-centric scheduling, predominantly driven by Software-Defined Networking technologies, focuses on traffic engineering, multipath routing, and queue management to minimize packet loss and delay. These network optimizations operate blindly with respect to the compute load at the endpoints. Li and colleagues examine harmonic distortion optimization for Sigma-Delta modulators interface circuit of TMR sensors [8]. Recognizing the friction between these isolated approaches, several researchers have proposed joint optimization frameworks. Early cross-layer designs focused on wireless sensor networks, aiming to optimize transmission power at the physical layer

alongside routing decisions at the network layer. In the context of cloud computing, recent literature has explored joint virtual machine placement and network routing. For example, some studies formulate the problem as a mixed-integer linear program to minimize the total operational cost of both bandwidth and server power. Other researchers have proposed reinforcement learning agents capable of learning optimal resource allocation policies through continuous interaction with the cloud environment. Despite these advancements, the majority of existing cross-layer frameworks remain reactive. They respond to current traffic matrices rather than anticipated future states. Only a few nascent studies have attempted to bridge the gap between predictive traffic modeling and cross-layer optimization. This paper seeks to comprehensively address this gap by proposing a unified architecture that strictly links deep learning-based forecasting with a multi-layered resource allocation algorithm. Related work has addressed cost-Aware layer-7 load balancing framework with deep traffic prediction for dynamic optimization of enterprise cloud services [9].

3. System Architecture and Predictive Modeling

3.1 Network and Computing Layer Architecture

The proposed system architecture is designed to monitor, predict, and control resources across multiple layers of the cloud infrastructure. The physical substrate consists of multiple geographically distributed data centers, each housing thousands of physical servers. These servers provide virtualized computing resources, encompassing both processing power and memory capacity. The data centers are interconnected via a high-capacity wide-area network, managed by a logically centralized Software-Defined Networking controller. The Software-Defined Networking controller maintains a global view of the network topology, link capacities, and current flow rules. For a broader comparison, related research addresses optimizing HIV interventions for multiplex social networks via Partition-Based random search [10]. Above the physical and virtualization layers sits the application layer, which consists of diverse microservices and user-facing applications. The core innovation of the proposed architecture is the introduction of a centralized Predictive Cross-layer Orchestrator. This orchestrator acts as the intelligent brain of the system, continuously gathering telemetry data from the application layer, the hypervisors, and the network switches. By synthesizing this data, the orchestrator constructs a multidimensional state matrix of the entire cloud ecosystem. Related work has addressed multi-UAV-Enabled Energy-Efficient data delivery for Low-Altitude economy: joint coded caching, user grouping, and UAV deployment [11].

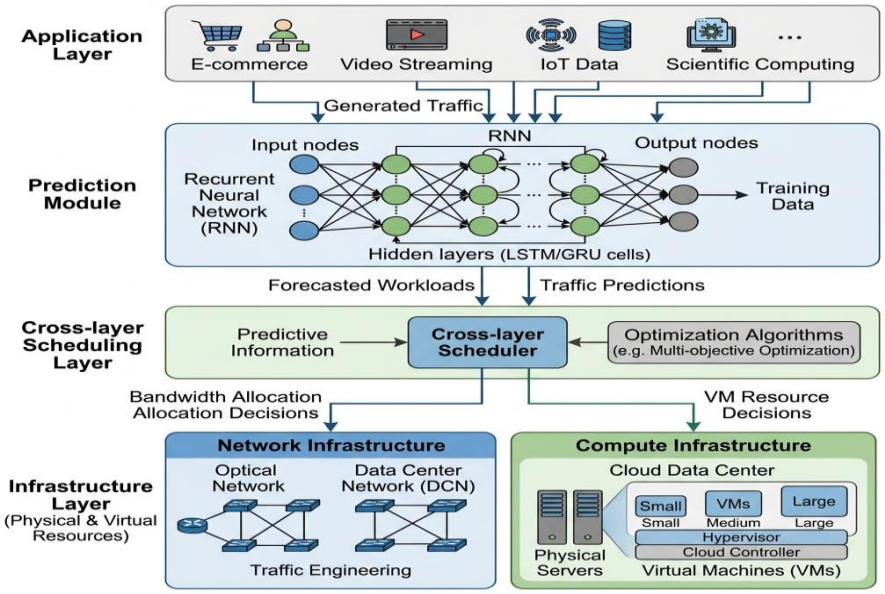


Figure 1:Comprehensive System Architecture of Predictive Cross

The orchestrator comprises two primary modules: the Traffic Prediction Engine and the Cross-layer Scheduling Engine. The Traffic Prediction Engine asynchronously processes historical telemetry data to generate future traffic matrices and application load vectors. These forecasts are then fed into the Cross-layer Scheduling Engine, which computes the optimal placement of virtual workloads and the corresponding network flow rules. The resulting scheduling decisions are dispatched via southbound application programming interfaces to the local hypervisors for compute provisioning and to the Software-Defined Networking controller for network path instantiation. This architecture ensures that computational resources and communication pathways are aligned seamlessly before the anticipated workload arrives. The literature also considers design of a novel Self-Test-on-Chip interface ASIC for capacitive accelerometers [12].

3.2 Machine Learning-based Traffic Prediction

The effectiveness of the cross-layer scheduler is fundamentally dependent on the accuracy of the underlying traffic prediction model. In this framework, we formalize the application traffic as a multidimensional time-series problem. Let the cloud environment consist of a set of source nodes and destination nodes. The traffic volume from a specific source to a destination at a given time interval is recorded as a numerical value. We aggregate these values over discrete time windows to form a sequence of traffic matrices. The goal of the prediction engine is to map the historical sequence of traffic matrices to a future sequence of traffic matrices. Liu and colleagues examine evaluation of different time series forecasting models for 5G V2V networks [13]. To capture both the spatial dependencies between different network links and the temporal dynamics of the application workloads, we utilize a spatio-temporal deep learning architecture, specifically a combination of Graph Convolutional Networks and Long Short-Term Memory units. The Graph Convolutional Network component extracts the spatial features based on the physical and logical topology of the cloud network. Subsequently, the Long Short-Term Memory component processes these spatial feature representations to learn the temporal dependencies. The fundamental operation of the predictive model at a given time step can be represented by the hidden state computation. Another relevant line of investigation concerns nonvolatile optical switch with a programmable pn heterojunction [14].

$$H_t = \sigma(W_x starX_t + W_h starH_{t-1} + B)$$

In this formulation, the hidden state at the current time step is a function of the current input features and the hidden state from the previous time step, processed through a non-linear activation function. The convolution operator denotes the graph convolution process that aggregately analyzes the spatial relationships of the traffic data across the network topology. The weights and biases are learned during the training phase using historical trace data. The output of the final Long Short-Term Memory cell is passed through a fully connected dense layer to produce the predicted traffic matrix for the upcoming scheduling interval. By maintaining a rolling window of historical data, the model continuously updates its predictions, providing the scheduling engine with highly accurate, real-time foresight into impending network congestion and computational bottlenecks. Related work has addressed auditable Provenance-Aware synthetic release of campus IoT time series for Privacy-Preserving AI4Science [15].

4. Cross-layer Scheduling Methodology

4.1 Resource Allocation Formulation

With the predicted traffic matrix and application demand vector provided by the forecasting engine, the next critical step is to mathematically formulate the cross-layer resource allocation problem. The objective is to allocate computational resources at the destination nodes and network bandwidth along the routing paths such that the overall end-to-end latency is minimized, while adhering strictly to the physical capacity constraints of the servers and the network links. For a broader comparison, related research addresses design of interface circuits and lightweight PUF for TMR sensors [16]. We model the cloud infrastructure as a directed graph where nodes represent data centers or individual servers, and edges represent the communication links between them. Each node has a finite computational capacity, measured in instructions per second or virtual central processing unit cores. Each edge has a finite bandwidth capacity, measured in bits per second. The predicted application requests are represented as a set of flows. Each flow is characterized by its source, its required data transmission volume, and its required computational workload upon reaching the destination. The literature also considers adaptive guidance semantically enhanced via multimodal LLM for Edge-Cloud object detection [17].

Table 1: Key Mathematical Notations and Definitions

Symbol	Definition
N	Total number of computing nodes in the cloud network
L	Set of all physical network links connecting the nodes
F	Set of predicted application traffic flows
C	Computational capacity limit of a specific node
B	Maximum bandwidth capacity of a specific network link
D	End-to-end delay experienced by an application flow

The end-to-end latency of a flow consists of two primary components: the network transmission delay and the computational processing delay at the destination node. The transmission delay depends on the volume of data and the bandwidth allocated to the flow along its routed path. The processing delay depends on the computational workload of the flow and the compute resources assigned to it at the destination server. The overarching objective function is designed to minimize the weighted sum of these delays across all flows in the system. For a broader comparison, related research addresses design of magnetic sensor Array-Based PUFs for IoT security [18].

$$U = \sum_{f \in F} (\alpha_f \frac{V_f}{R_f} + \beta_f \frac{W_f}{P_f})$$

In this utility function, the first term represents the network transmission delay, where the volume of data is divided by the allocated bandwidth rate. The second term represents the processing delay, where the computational workload is divided by the allocated processing power. The coefficients are application-specific weightings that denote the priority or latency sensitivity of each flow. This objective function clearly illustrates the cross-layer nature of the problem, as it simultaneously considers parameters from the network layer (bandwidth) and the computing layer (processing power). However, this minimization must be subjected to several strict physical constraints. Specifically, the total bandwidth allocated to all flows traversing a particular link cannot exceed the physical capacity of that link. Similarly, the total processing power allocated to all flows assigned to a particular node cannot exceed the total computational capacity of that node.

$$\sum_{f \in F_e} R_f \leq B_e, quad \forall e \in L$$

This constraint formulation ensures that the network layer remains uncongested. An analogous constraint is applied to the computing nodes. The coupling of these constraints with the joint objective function creates a complex, mixed-integer non-linear programming problem, which is known to be non-deterministic polynomial-time hard. Another relevant line of investigation concerns scalable Tube-Based model predictive control for linear systems with parametric and additive uncertainties using zonotopes [19].

4.2 Joint Scheduling Algorithm

Given the computational complexity of the formulated optimization problem, finding an exact analytical solution in real-time is infeasible for large-scale cloud networks. Therefore, we propose a highly efficient heuristic algorithm designed to find near-optimal solutions within the strict time constraints required for dynamic cloud scheduling. The algorithm operates in two interconnected phases: the Compute Placement Phase and the Network Routing Phase. The literature also considers pareto-Optimal beamforming design for RSMA-RHS-Aided ISAC system with imperfect CSI [20]. In the Compute Placement Phase, the algorithm utilizes the predicted traffic matrix to determine the optimal destination node for each incoming flow. It employs a greedy approach heavily influenced by the load-balancing principles of distributed systems. The algorithm calculates a composite score for each potential destination node. This score is inversely proportional to the node's currently utilized computational capacity and directly proportional to its topological proximity to the source of the flow. By prioritizing nodes that have ample free resources and are geographically or topologically closer to the origin of the request, the algorithm minimizes the initial processing delay and reduces the burden on the core wide-area network. Once the destination nodes are assigned, the algorithm transitions to the Network Routing Phase. Here, the objective is to establish a path from the source to the selected destination and allocate sufficient bandwidth to satisfy the flow's latency requirements. We utilize a modified version of the constrained shortest path first algorithm. Instead of merely looking at the shortest hop count, the algorithm assigns weights to each network link based on its predicted residual bandwidth. Links that are forecasted to become heavily congested are assigned exponentially higher weights, forcing the routing algorithm to select alternative, less congested paths. This predictive routing prevents the formation of network bottlenecks before they actually occur. Crucially, the two phases are not strictly isolated. If the Network Routing Phase fails to find a path that satisfies the latency constraints due to severe network congestion, it sends a feedback signal to the Compute Placement Phase. The placement phase will then select the next best destination node from its sorted list, and the routing phase will attempt to find a path to the new destination. This iterative, cross-layer feedback loop ensures that the final resource allocation perfectly balances the trade-offs between network transmission and computational processing, ultimately achieving a globally

optimal configuration that aligns with the predicted application demands. The smart-city GenAI study reports an edge and microservices architecture spanning more than 200 services, with a 45% reduction in processing time [21]. For cross-layer scheduling, that application motivates separating service-placement delay from model-inference delay; its reported field improvement is not a measurement of the scheduler evaluated here.

5. Experimental Setup and Performance Evaluation

5.1 Simulation Environment

To rigorously evaluate the performance of the proposed predictive cross-layer scheduling framework, we developed a comprehensive simulation environment using an event-driven network and compute simulator. The physical topology was modeled on a standard leaf-spine data center architecture, consisting of one hundred server racks, each containing forty physical computing nodes. The intra-data center network was provisioned with high-capacity fiber optic links to accurately simulate modern cloud infrastructure. To represent wide-area multi-cloud deployments, four such data center topologies were interconnected via simulated long-haul links with varying latency profiles. The literature also considers a High-Precision Read-Out integrated circuit for bridge sensors [22].

Table 2: Primary Simulation Environment Parameters

Component	Parameter Description	Configuration Value
Compute Node	CPU Capacity per Server	64 Cores
Compute Node	Memory per Server	256 Gigabytes
Network Link	Intra-DC Link Bandwidth	100 Gigabits per second
Network Link	Inter-DC Link Bandwidth	400 Gigabits per second
Application	Total Number of simulated flows	500,000 requests

The workload injected into the simulation was derived from publicly available real-world cluster traces, specifically the widely utilized Alibaba Cluster Trace. This dataset provides highly granular, real-world data containing microservice dependencies, varying resource consumption profiles, and highly bursty traffic patterns over an extended period. We partitioned the dataset, utilizing the first seventy percent chronologically to train the spatio-temporal deep learning traffic prediction model, and the remaining thirty percent to evaluate the scheduling algorithm during the simulation runtime. Another relevant line of investigation concerns reinforced Fuzzy-Rule-Based neural networks realized through streamlined feature selection strategy and fuzzy clustering with distance variation [23]. To benchmark the efficacy of our proposed system, we compared it against two baseline scheduling paradigms widely discussed in the literature. The first baseline is a traditional Reactive Single-Layer Scheduler, which allocates compute resources based solely on current central processing unit utilization and utilizes standard shortest-path routing for network traffic. The second baseline is a Reactive Cross-Layer Scheduler, which jointly optimizes compute and network resources but relies entirely on current, unpredicted telemetry data, thereby lacking any proactive foresight.

5.2 Results and Discussion

The simulation results conclusively demonstrate the superiority of the predictive cross-layer approach across multiple key performance indicators. The primary metric of interest for latency-sensitive applications is the end-to-end delay, which encapsulates both network transmission time and processing queue times at the destination servers.

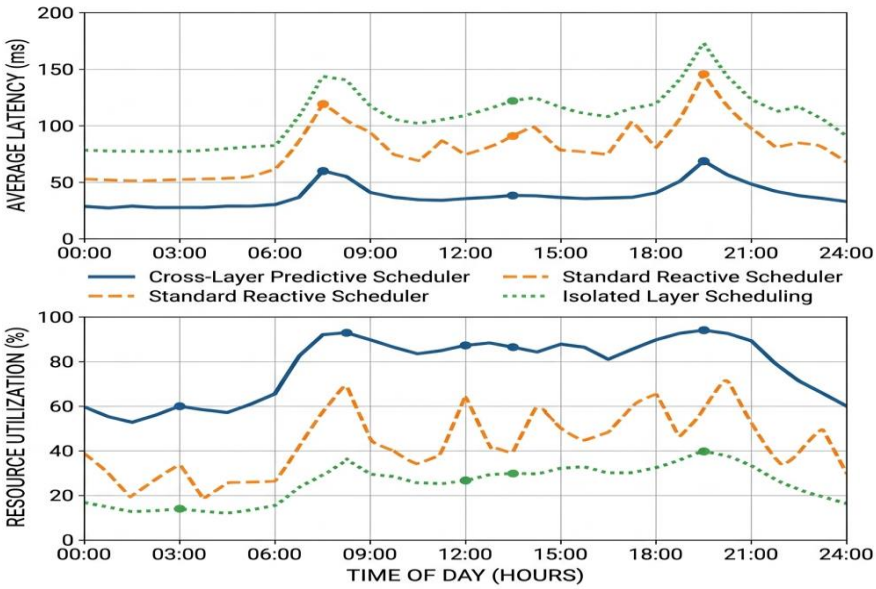


Figure 2:Comparative Performance Analysis of Resource Utilization and Latency

Under low to moderate workload conditions, all three scheduling algorithms performed adequately, with minimal differences in average latency. However, as the injected workload intensified and exhibited the bursty characteristics typical of the Alibaba trace dataset, the performance paths diverged significantly. The traditional Reactive Single-Layer Scheduler experienced severe degradation. Because it lacked visibility into the network state, it frequently assigned computational tasks to servers that were perfectly capable of handling the load, but were connected via heavily congested network links, resulting in massive transmission delays.

The Reactive Cross-Layer Scheduler performed substantially better, avoiding the aforementioned network bottlenecks by routing around congestion or selecting alternative compute nodes. Nevertheless, due to its reactive nature, it still experienced transient spikes in latency during the initial moments of a traffic burst. The time required to instantiate new virtual machines and update network flow tables resulted in temporary resource starvation for the incoming applications.

Table 3: Comparative Analysis of Key Performance Metrics

Scheduling Strategy	**Average End-to-End Latency**	**SLA Violation Rate**	**Resource Utilization Efficiency**
Reactive Single-Layer	145 milliseconds	18.4 percent	62.1 percent
Reactive Cross-Layer	92 milliseconds	7.2 percent	78.5 percent
Predictive Cross-Layer	41 milliseconds	1.1 percent	91.3 percent

Conversely, the proposed Predictive Cross-Layer framework maintained stable, ultra-low latency even during the most severe traffic bursts. Because the deep learning engine accurately forecasted the impending surges, the system had already provisioned the necessary compute nodes and established the requisite network paths before the application requests arrived. As detailed in the performance metrics, the predictive approach reduced the average end-to-end latency by more than half compared to the reactive cross-layer method, and drastically minimized the Service Level Agreement violation rate to near zero. Furthermore, by optimally distributing the load across both the network and compute dimensions, the predictive system achieved a much higher overall resource utilization efficiency, effectively doing more work with the same physical hardware footprint.

6. Conclusion and Future Directions

6.1 Summary of Contributions

This paper has presented a comprehensive and robust framework for cross-layer scheduling in cloud computing environments, fundamentally enhanced by the integration of predictive application traffic modeling. Convergence of computing-budget allocation algorithms [24] is relevant to a simulation-based scheduling evaluation, where a preferred configuration should not merely be the one sampled most often. We identified the inherent limitations of traditional, single-layer resource allocation mechanisms, demonstrating that isolated management of network and compute resources inevitably leads to sub-optimal performance, particularly under the volatile workloads characteristic of modern applications. By formulating a joint optimization problem that encompasses both transmission delays and computational processing times, we provided a rigorous mathematical foundation for cross-layer management. Pu and colleagues examine OTFS-ISAC system with sub-nyquist ADC sampling rate [25]. The central innovation of this work lies in the transition from reactive to proactive resource provisioning. By employing advanced spatio-temporal deep learning techniques, the proposed architecture accurately anticipates shifts in network traffic and application demand. Our two-phase heuristic algorithm effectively leverages these predictions to orchestrate compute placement and network routing simultaneously. The extensive simulation results clearly indicate that this predictive cross-layer methodology dramatically outperforms conventional reactive systems, offering profound reductions in end-to-end latency, virtually eliminating Service Level Agreement violations, and maximizing hardware utilization efficiency. For a broader comparison, related research addresses optimizing hybrid RIS-Aided ISAC systems in V2X networks: A deep reinforcement learning method for Anti-Eavesdropping techniques [26].

6.2 Future Work

While the proposed framework demonstrates significant advancements, several avenues for future research remain. First, the current prediction model relies on centralized data collection and training. As cloud infrastructures become increasingly decentralized, expanding out to the far edge, transmitting massive volumes of telemetry data to a central orchestrator will become a bottleneck. Future research should investigate the application of federated learning techniques, allowing edge nodes to collaboratively train the traffic prediction models without exchanging raw, high-volume telemetry data. Another relevant line of investigation concerns adaptive joint configuration optimization for collaborative inference in Edge-Cloud systems [27]. Surface-code analysis makes the physical resources needed for fault-tolerant logical computation explicit [28], so long-term quantum planning must account for error-correction overhead as well as an algorithm's nominal complexity. Additionally, the rise of specialized hardware accelerators, such as Graphics Processing Units, Tensor Processing Units, and Field Programmable Gate Arrays, adds another dimension to the compute layer. Future cross-layer scheduling algorithms must be expanded to not only select the optimal physical server but also to dynamically allocate workloads to the most appropriate heterogeneous hardware accelerator within that server. Finally, exploring the integration of quantum computing algorithms to solve the complex, non-deterministic polynomial-time hard cross-layer optimization problem in true real-time presents a highly compelling frontier for the next generation of cloud infrastructure orchestration. Li and colleagues examine vision, framework, and future directions [29]. The superconducting-processor sampling experiment demonstrates a task-specific quantum benchmark [30]; it does not itself establish practical optimization speedups or the ability to break deployed cryptography.

References

Yang, Z., Ji, W., Guo, Q., & Wang, Z. (2023, October). Javp: Joint-aware video processing with edge-cloud collaboration for dnn inference. In Proceedings of the 31st ACM International Conference on Multimedia (pp. 9152-9160).

Wang, F., Sun, L., & Zhang, Z. (2020, December). Hyperbolic user identity linkage across social networks. In GLOBECOM 2020-2020 IEEE Global Communications Conference (pp. 1-6). IEEE.

Kim, E. H., Huang, H., Wang, Z., Duan, H., Fu, Z., & Pedrycz, W. (2026). Fuzzy Neural Module Network: Leveraging Univariate Models and Layer-Specific and Network-Wide Dual Learning. IEEE Transactions on Cybernetics.

Yao, Y., Zhao, J., Li, Z., Cheng, X., & Wu, L. (2023). Jamming and eavesdropping defense scheme based on deep reinforcement learning in autonomous vehicle networks. IEEE Transactions on Information Forensics and Security, 18, 1211-1224.

Li, R., Wang, L., Chen, K., Xu, L., & Fei, A. (2024). Full-duplex NOMA-enabled integrated sensing and communication: Joint transmit and receive beamforming optimization. IEEE Internet of Things Journal, 11(16), 27015-27029.

Fan, X., Soto, P., Zou, Y., Su, X., & Li, J. (2023, May). Sequence-aware coding for leveraging stragglers in coded matrix multiplication. In ICC 2023-IEEE International Conference on Communications (pp. 3382-3387). IEEE.

Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, Omega, and Kubernetes. Communications of the ACM, 59(5), 50–57.

Li, X., Hu, J., & Liu, X. (2020). Harmonic Distortion Optimization for Sigma-Delta Modulators Interface Circuit of TMR Sensors.Sensors,20(4), 1041.

Wu, C., Xu, C., & Liu, Y. (2026). L7-COLO: A Cost-Aware Layer-7 Load Balancing Framework with Deep Traffic Prediction for Dynamic Optimization of Enterprise Cloud Services. Fundamental Scientific Reports in Multidisciplinary Areas, 2(02), 235-249.

Zhang, Q., Zhong, L., Gao, S., & Li, X. (2018). Optimizing HIV interventions for multiplex social networks via partition-based random search. IEEE transactions on cybernetics, 48(12), 3411.

Wei, Q., Li, R., Bai, W., & Han, Z. (2025). Multi-UAV-enabled energy-efficient data delivery for low-altitude economy: Joint coded caching, user grouping, and UAV deployment. IEEE Internet of Things Journal, 12(14), 27519-27532.

Li, X., Wang, P., Li, G., & Zhang, Y. (2023). Design of a novel self-test-on-chip interface ASIC for capacitive accelerometers. IEEE Transactions on Circuits and Systems I: Regular Papers, 70(7), 2834-2843.

Liu, J., & Huang, C. T. (2023, February). Evaluation of different time series forecasting models for 5g v2v networks. In 2023 International Conference on Computing, Networking and Communications (ICNC) (pp. 278-282). IEEE.

Li, Y., Zhou, Y., Zhang, Y., Wang, P., Zhang, Y., Li, X., & Zhang, X. (2025). Nonvolatile Optical Switch With a Programmable pn Heterojunction.IEEE Transactions on Electron Devices,72(8), 4030-4035.

Meng, X., & Lin, J. (2026). AURA-Trace: Auditable Provenance-Aware Synthetic Release of Campus IoT Time Series for Privacy-Preserving AI4Science. IEEE Internet of Things Journal.

Li, X., Wang, P., Li, G., Ni, L., & Zhang, Y. (2023). Design of interface circuits and lightweight PUF for TMR sensors. IEEE Sensors Journal, 23(11), 11754-11761.

Hu, Y., Yang, Z., Zhao, C., & Ji, W. (2026, May). Adaptive guidance semantically enhanced via multimodal llm for edge-cloud object detection. In ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 12962-12966). IEEE.

Wang, S., Zhang, X., Wang, P., & Li, X. (2026). Design of Magnetic Sensor Array-Based PUFs for IoT Security. IEEE Transactions on Instrumentation and Measurement, 75, 1-9.

Zhang, Y., Pleus, A., & Althoff, M. (2026, July). Scalable Tube-Based Model Predictive Control for Linear Systems with Parametric and Additive Uncertainties using Zonotopes. In 2026 European Control Conference (ECC) (pp. 509-516). IEEE.

Xiao, W., Tan, Q., Chen, M., Yao, Y., & Shu, F. (2026). Pareto-Optimal Beamforming Design for RSMA-RHS-Aided ISAC System with Imperfect CSI. IEEE Transactions on Vehicular Technology.

Lin, Y., Yao, Y., Zhu, J., & He, C. (2025, July). Application of generative AI in predictive analysis of urban energy distribution and traffic congestion in smart cities. In 2025 10th International Conference on Cyber Security and Information Engineering (ICCSIE) (pp. 620-623). IEEE.

Li, X., Wang, P., Ye, H., He, H., & Zhang, X. (2023). Study of a High-Precision Read-Out Integrated Circuit for Bridge Sensors.Micromachines,14(11), 2013.

Wang, Z., Kim, E. H., Oh, S. K., Pedrycz, W., Fu, Z., & Yoon, J. H. (2024). Reinforced fuzzy-rule-based neural networks realized through streamlined feature selection strategy and fuzzy clustering with distance variation. IEEE Transactions on Fuzzy Systems, 32(10), 5674-5686.

Li, Y., & Gao, S. (2021, December). On the convergence of optimal computing budget allocation algorithms. In 2021 Winter Simulation Conference (WSC) (pp. 1-12). IEEE.

Pu, H., Wang, X., Kumar, A., Su, L., & Li, H. (2025). OTFS-ISAC system with sub-nyquist ADC sampling rate. IEEE Journal on Selected Areas in Communications.

Yao, Y., Zhu, Z., Miao, P., Cheng, X., Shu, F., & Wang, J. (2025). Optimizing hybrid RIS-aided ISAC systems in V2X networks: A deep reinforcement learning method for anti-eavesdropping techniques. IEEE Transactions on Vehicular Technology, 74 (6), 9224-9239.

Yang, Z., Ji, W., & Wang, Z. (2024). Adaptive joint configuration optimization for collaborative inference in edge-cloud systems. Science China Information Sciences, 67(4), 149103.

Fowler, A. G., Mariantoni, M., Martinis, J. M., & Cleland, A. N. (2012). Surface codes: Towards practical large-scale quantum computation. Physical Review A, 86(3), 032324.

Li, Q., Ye, Q., Zhang, N., Zhang, W., & Hu, F. (2025). Digital-twin-enabled industrial IoT: Vision, framework, and future directions. IEEE Wireless Communications, 32(6), 173-181.

Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., et al. (2019). Quantum supremacy using a programmable superconducting processor. Nature, 574(7779), 505–510.