ISSN 2616-5775 Vol. 9, Issue 6: 9-14, DOI: 10.25236/AJCIS.2026.090602

Modeling Method and Implementation Mechanism for QoS Prediction Based on Multi-Source Data Fusion

Zhenzhen Liu

Xi'an Peihua University, Xi'an, 710100, China

Abstract: *Quality of Service (QoS) prediction in service-oriented computing supports candidate service identification and resource scheduling. However, the limited single-source data dimension fails to capture dynamic characteristics such as network fluctuations, spatiotemporal migration, and user behavior coupling. This study integrates three types of heterogeneous information to construct a unified embedding space, achieves cross-source alignment of semantically heterogeneous data through a hierarchical architecture, introduces differentiated dynamic weight allocation among three data sources, captures nonlinear evolution patterns through temporal joint representations, and employs multi-source context-guided sparse compensation to address missing entries in the service invocation matrix. Experimental results demonstrate that the proposed method significantly outperforms single-source models in both prediction accuracy and robustness under complex scenarios.*

Keywords: *QoS Prediction; Multi-Source Data Fusion; Dynamic Weight Allocation; Temporal Joint Representation; Sparse Compensation*

1. Introduction

The widespread adoption of microservices architecture has led to a rapid proliferation of network services with similar functionalities, making service quality the key criterion for distinguishing candidate services and enabling efficient scheduling. Due to network fluctuations, geographical migration, and user preference coupling, QoS values exhibit both dynamic and sparse characteristics; collaborative filtering methods relying on single-source records demonstrate insufficient feature coverage and struggle to adapt to the complexity of real-world environments. Multi-source fusion integrating network, spatiotemporal, and user behavior data leverages complementary information to address single-source data limitations, yet semantic heterogeneity, varying data contributions, and matrix sparsity remain significant challenges. This study develops modeling frameworks focusing on embedding alignment, dynamic weighting, and sparse compensation to explore feasible paths for accurate QoS prediction supported by multi-source collaboration.

2. QoS Representation Analysis and Fusion Architecture for Multi-Source Heterogeneous Data

2.1 QoS Coupling Characteristics of Networks, Spatiotemporal Data, and User Behavior Data

The QoS attributes of services primarily encompass a set of performance metrics including availability, response time, and throughput, each representing specific quality characteristics of the service [1]. The values of these attributes are not determined by isolated factors but are the result of coupled effects among network links, spatiotemporal environments, and user invocation behaviors. Network-level factors such as RTT, packet loss rate, and bandwidth jitter impose upper limits on response times and throughput, forming the physical foundation for quality fluctuations; spatiotemporal environments influence service availability through geographical distribution and load cycles associated with invocation times, with latency differences across regions often exceeding 40%; user behavior-related factors like preference distributions and invocation frequencies shape the subjective dimension of QoS perception, leading to varied experiences even with identical physical metrics. These three factors interact nonlinearly rather than simply summing up: network congestion is amplified by spatiotemporal loads during peak periods and manifests as uneven feedback due to varying user sensitivity levels. Quantifying these nonlinear cross-source coupling pathways provides a direct basis for setting differentiated weights.

ISSN 2616-5775 Vol. 9, Issue 6: 9-14, DOI: 10.25236/AJCIS.2026.090602

2.2 Hierarchical Fusion Architecture and Data Flow Organization for Semantic Heterogeneity

The three categories of data sources exhibit significant heterogeneity in measurement scales, sampling frequencies, and semantic meanings. Network metrics represent continuous physical measurements; spatiotemporal information combines discrete identifiers with continuous coordinates; and user behavior manifests as high-dimensional sparse interaction records. Direct concatenation often leads to semantic misalignment and information overload [2]. To address this challenge, a QoS-oriented hierarchical fusion architecture organizes data flows through a progressive structure consisting of data-layer cleansing and alignment, feature-layer semantic mapping, and decision-layer weighted aggregation. This enables heterogeneous signals to achieve scale consistency and semantic alignment during hierarchical abstraction. At the data layer, ETL processes perform noise removal and timestamp synchronization, converting raw samples into comparable tensor units. At the feature layer, source-specific encoders project the three categories of signals into a shared feature space, alleviating dimensional imbalance caused by high-dimensional sparsity and increasing effective feature density by nearly 30%. At the decision layer, fusion weights are dynamically allocated according to the contextual confidence of each source, preventing dominant sources from suppressing weaker sources semantically. Outputs from each layer are transmitted through unified interfaces, preserving source-specific information throughout the entire fusion process.

3. Design of the QoS Prediction Modeling Method Based on Multi-Source Data Fusion

3.1 Unified Embedding Mapping and Static Cross-Source Alignment for Heterogeneous Features

The unified embedding mapping configures differentiated encoding pathways for the three source signals. Network features are compressed into a 128-dimensional dense vector via fully connected projection. Spatiotemporal identifiers are transformed into embeddable symbols through position encoding and Geohash discretization of geographic grids. User behaviors are compressed from tens of thousands of sparse interactions into a 128-dimensional latent representation using look-up tables. The outputs of all three branches are uniformly projected into a 256-dimensional semantic subspace, where shared linear transformation matrices eliminate numerical biases caused by differences in measurement scales and dimensions before being reduced back to a 128-dimensional standard embedding [3]. Static cross-source alignment is adopted, with alignment anchors predefined according to offline Pearson correlation coefficients among sources. Maximum Mean Discrepancy (MMD) is employed to constrain marginal consistency among source distributions within the embedding space, causing semantically similar states—such as network congestion and low user activity—to become closer in distance metrics. The alignment process incorporates L2 normalization to confine vectors to the unit hypersphere and uses cosine similarity instead of Euclidean distance to measure cross-source similarity, yielding standardized embeddings with consistent scales for dynamic weighting applications.

3.2 Dynamic Adaptive Weight Allocation Mechanism for Differentiated Contributions from Three Sources

Dynamic weight allocation is centered around Multi-Head Attention, employing eight attention heads to map the three-source embeddings respectively into query, key, and value triples. The relevance strength of each source in the current invocation context is calculated via scaled dot-product attention, yielding context-dependent normalized weights that enable the three sources to participate in fusion according to real-time relevance rather than fixed proportions during each invocation [4]. This weight calculation can be expressed as follows:

$$a_i = \mathrm{Softmax}\left(\frac{Q_i \cdot K_i^{\mathrm{T}}}{\sqrt{d_k}}\right), \quad \mathbf{z} = \sum_{i=1}^{3} a_i \cdot V_i \quad (1)$$

Where , Q_i, K_i and V_i represent the query, key, and value vectors obtained through linear transformation of the embedding from the i-th source, respectively; d_k represents the scaling factor; a_i represents the normalized source weight; and z represents the weighted fusion representation. The design introduces a gating unit to apply secondary modulation to the attention output, dynamically

ISSN 2616-5775 Vol. 9, Issue 6: 9-14, DOI: 10.25236/AJCIS.2026.090602

suppressing distorted sources according to trigger signals such as abrupt RTT increases or sudden declines in user activity. Consequently, the weight of the network source is adaptively increased during network congestion periods, while low-activity user sources are suppressed, preventing noisy sources from misleading the final representation. The contribution scores of the three sources are re-estimated at sample granularity using a four-dimensional context vector, with weights normalized via Softmax to remain within a probability simplex, ensuring their sum remains constant and enabling balanced competition among them. The attention output incorporates residual connections to mitigate gradient attenuation in deep architectures, while Dropout is applied at a rate of 0.3 to randomly deactivate neurons and prevent single-source overfitting. Weight re-estimation occurs independently at each invocation step, ensuring the fusion results align with instantaneous contexts rather than rigidly adhering to offline priors.

3.3 Temporal Joint Representation Modeling for Network-Spatiotemporal-Behavior Coupling

The weighted fusion output solely captures inter-source relationships at a single time step, while the evolution of QoS metrics across invocation sequences requires temporal modeling support. The three-source weighted embeddings are restructured into window sequences of length 16 with a sliding step size of 1 based on invocation timestamps, then scanned bidirectionally through GRUs to capture temporal dependencies between cumulative network congestion and user behavior drift. Coupling modeling incorporates four-head temporal self-attention mechanisms at the sequence level, establishing long-range correlations between distant peak loads and quality declines while mitigating forgetting effects caused by circular structures in remote dependencies [5]. The three sources evolve non-independently; the module explicitly models second-order coupling terms between network states and spatiotemporal states via a bilinear interaction layer, encoding interaction effects amplified by geographic loads during specific periods into joint representations. The hidden dimension is set to 64. After two stacked GRU layers, layer normalization is applied to stabilize feature distributions. The temporal output converges into a 128-dimensional context vector that simultaneously carries instantaneous inter-source weights and cross-temporal evolution trajectories, providing temporally structured conditional representations for the missing-value compensation stage (as shown in Figure 1).

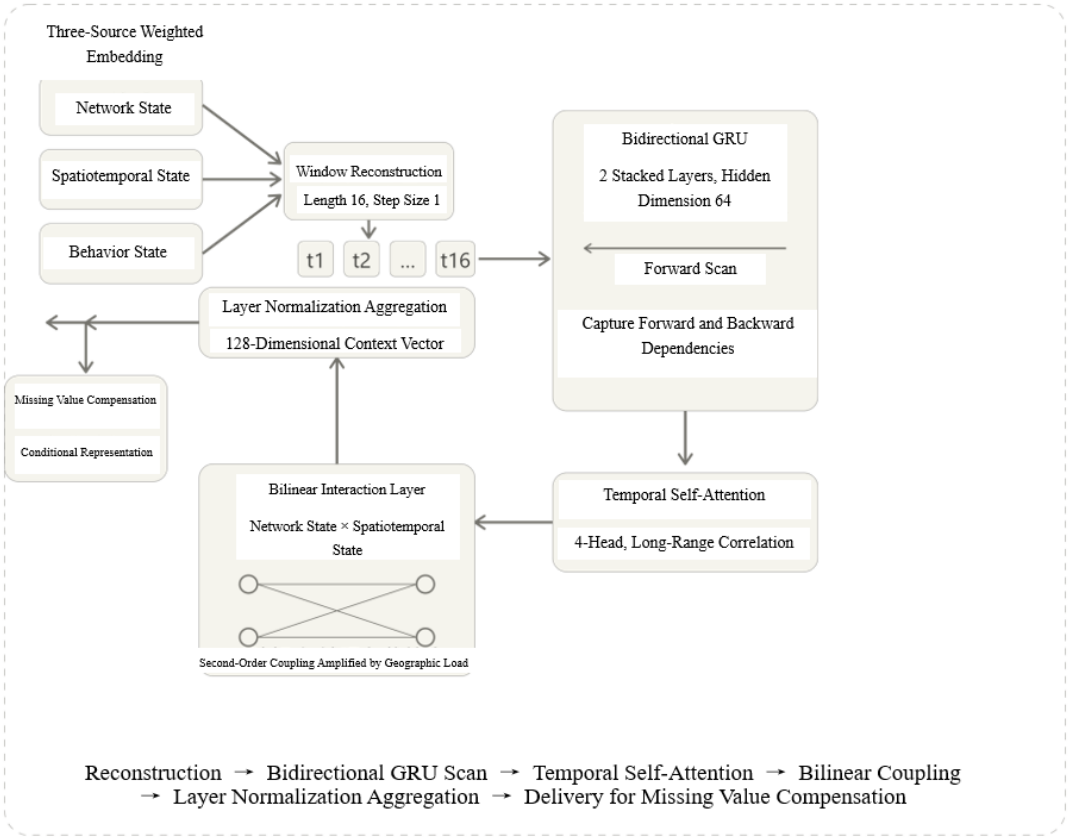

Figure 1 Structural Diagram of Temporal Joint Representation Modeling for Network-Spatiotemporal-Behavior Coupling

ISSN 2616-5775 Vol. 9, Issue 6: 9-14, DOI: 10.25236/AJCIS.2026.090602

3.4 Multi-Source Context-Guided Sparse Invocation Missing Value Compensation and Low-Rank Constraint

The evolutionary prior provided by temporal representation can be transformed into contextual inference conditions during the compensation stage. Instead of using crude zero-value filling, the missing-value compensation mechanism employs the 128-dimensional context vector generated by the joint representation module as a conditional driver for completion, inferring the conditional expectations of missing entries based on known invocations from similar users and neighboring spatiotemporal contexts. The compensation framework incorporates a nuclear norm-based low-rank constraint on top of matrix factorization, compressing the latent factors of users and services into a subspace with a rank not exceeding 20 while leveraging the inherent low-dimensional structure of QoS data to suppress noise amplification. The entire compensation process is solved through joint objective-driven optimization:

$$\min_{X}\left\|\Omega\odot\left(M-X\right)\right\|_{F}^{2}+\lambda\left\|X\right\|_{*}+\beta tr\left(X^{\mathrm{T}}LX\right) \quad (2)$$

Where, M represents the observed invocation matrix; Ω represents the indicator mask of missing positions; X represents the matrix to be completed; $\left\|X\right\|_{*}$ represents the nuclear norm imposing the low-rank constraint; L represents the Laplacian matrix of the user similarity graph, and λ and β represent the weights of the low-rank term and graph regularization term, respectively. The low-rank term and multi-source context term are jointly optimized via a weighted objective function, while the regularization coefficient λ is set to 0.1 to balance global structure and local correction. Both components converge to a stable solution through 30 rounds of alternating ALS iterations. The framework further incorporates the adjacency relationships in the user similarity graph into graph regularization, using a smoothing weight of 0.08 to ensure smooth transitions between adjacent elements during completion. The low-rank skeleton and multi-source context are updated alternately under the same objective, with missing values filled element-wise to yield a dense feature matrix subject to rank constraints.

4. Implementation Mechanism and Performance Validation of Predictive Modeling

4.1 Parallel Collection and Unified Preprocessing of Multi-source Data

At the implementation layer, a distributed data collection framework is employed to collect signals from three sources in parallel. Network probes collect RTT and packet loss rates at intervals of 500 ms, while spatiotemporal identifiers are obtained through dual channels consisting of GPS and base station positioning and aligned to a unified temporal reference. User behavior data is extracted from invocation logs at session granularity. The three data streams are buffered in Kafka message queues before entering the preprocessing pipeline, where eight parallel processing nodes perform deduplication, outlier removal, and timestamp alignment, transforming asynchronously arrived samples into synchronized tensors. Using the ETL paradigm, preprocessing applies Z-score normalization to raw measurements with different measurement scales, standardizing values across all sources for comparability. A sliding window mechanism stores nearly one hour of invocation sequences to support subsequent temporal modeling. To address approximately 15% of out-of-order and jitter issues during collection, the pipeline introduces a threshold mechanism to tolerate bounded delays, filling missing samples with previous sequence means when timeouts occur. Processing results from each source are stored in feature storage using a unified schema organized as user-service tuples, providing an indexing foundation for parallel retrieval and batch loading.

4.2 Collaborative Implementation Mechanism for Incremental Fusion and Online Updates

In online service scenarios, invocation data streams continuously flow in. The fusion model employs an incremental mechanism to avoid the high costs of full retraining. New samples are extracted and integrated into a sliding buffer, with the model performing small-batch gradient updates only on recent data within the window (each batch size set at 256). Parameters smoothly transition along an exponential moving average trajectory to mitigate abrupt perturbations. Incremental updates and offline full-scale training are coordinated through a dual-track collaborative mechanism: the offline track re-estimates the global low-rank skeleton and distributes baseline parameters every 24 hours,

ISSN 2616-5775 Vol. 9, Issue 6: 9-14, DOI: 10.25236/AJCIS.2026.090602

while the online track performs real-time fine-tuning between consecutive full-training cycles. Version number alignment ensures parameter consistency across both tracks. The model utilizes a dual-buffer architecture to isolate updates from inference processes—after backend replicas complete parameter updates, they seamlessly take over online requests via atomic switching, ensuring update operations do not delay millisecond-level prediction responses. To address concept drift, the mechanism monitors distribution deviations in the sliding window; when local divergence exceeds a threshold, local weight re-estimation is triggered, updating only affected source pathways rather than the entire network, reducing single-update duration to seconds for sustained online operation in large-scale service grids.

4.3 Experimental Environment, Dataset Partitioning, and Baseline Configuration for Comparison

The experiment was conducted on a server equipped with dual Intel Xeon processors and four NVIDIA V100 graphics cards. The model was implemented using PyTorch 1.12, employing Adam as the optimizer with a learning rate of 0.001. Validation was performed using the publicly available WS-DREAM real-world invocation dataset, which contains invocation records from 339 users across 5,825 services, with response time selected as the QoS metric. Data were divided into training, validation, and test sets in a ratio of 7:2:1, with random masking applied to simulate varying invocation densities (5%–20%) to replicate real-world sparse distribution patterns. Comparison baselines included three categories: collaborative filtering (represented by matrix factorization), single-source recurrent networks utilizing only network features, and multi-source fusion with equal-weight concatenation—all trained under identical data partitioning and hardware configurations as the proposed method. Evaluation metrics employed Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), with results averaged via five-fold cross-validation. Hyperparameters were optimized using grid search to ensure fair and reproducible comparisons.

4.4 Verification Results of Prediction Accuracy Improvement and Robustness in Complex Scenarios

During the performance evaluation stage, under a unified experimental setup, the proposed method must be compared horizontally with three baselines using identical data partitioning and evaluation metrics. By analyzing error metrics and performance at different sparsity levels, the comprehensive effectiveness of the method in terms of prediction accuracy and robustness can be assessed.

Table 1 Comparison of Prediction Performance of Different Methods on the WS-DREAM Dataset

Method Category	MAE (Response Time)	RMSE (Response Time)	MAE (20% Sparsity)	MAE (5% Sparsity)
Matrix Factorization (Collaborative Filtering)	0.612	1.483	0.658	0.847
Single-Source Recurrent Network	0.548	1.327	0.591	0.762
Equal-Weight Concatenation Fusion	0.493	1.205	0.532	0.694
Proposed Multi-Source Fusion Method	0.387	0.961	0.415	0.538

As shown in Table 1, the proposed method achieves an MAE of 0.387 and an RMSE of 0.961, representing reductions of approximately 21% and 20%, respectively, compared to equal-weighted concatenation fusion method. When the invocation density is reduced from 20% to 5%, baseline errors significantly increase; however, the proposed method maintains an MAE of 0.538 even at the 5% sparsity level, demonstrating the robustness of multi-source context compensation and temporal coupling modeling in sparse environments.

5. Conclusion

The modeling paradigm relying solely on single-source data exhibits limited accuracy in complex service invocation environments, whereas multi-source collaboration establishes a more adaptive technical pathway for QoS prediction. The deep coupling of three types of heterogeneous information significantly expands feature coverage, while differentiated weighting and temporal joint representation

ISSN 2616-5775 Vol. 9, Issue 6: 9-14, DOI: 10.25236/AJCIS.2026.090602

enhance the model's ability to capture dynamic environments. Sparse compensation further mitigates prediction biases caused by insufficient input data. The overall effectiveness of the proposed framework has been comprehensively validated in terms of both prediction accuracy and robustness, providing a reliable foundation for precise service recommendations and efficient scheduling. Future efforts should expand the variety and scale of data sources, explore lightweight deployment mechanisms for fusion algorithms in large-scale real-time applications, and leverage federated learning to drive continuous optimization and widespread adoption of quality prediction systems while ensuring data security.

References

[1] Ren Lifang, Wang Wenjian. Method for QoS Prediction in Mobile Edge Computing Environment [J]. Journal of Chinese Computer Systems, 2020, 41 (6): 1176–1181.

[2] Chen Manman, Wang Junfeng, Li Xiaohui, et al. Temporal QoS Prediction Based on Multi-Source Features and Multitask Learning in Cloud Service Recommendation [J]. Journal of Sichuan University (Natural Science Edition), 2024, 61 (4): 140–150.

[3] Tan Hefei, Zong Rong, Wu Hao, et al. Multi-task Web Service QoS Prediction Based on Graph Convolution Network [J]. Computer Engineering and Design, 2024, 45 (1): 47–54.

[4] Xu Jianlong, Lin Jian, Li Yusen, et al. Distributed User Privacy Preserving Adjustable Personalized QoS Prediction Model for Cloud Services [J]. Chinese Journal of Network and Information Security, 2023, 9 (2): 70–80.

[5] Liu Jianxun, Ding Linghang, Kang Guosheng, et al. Joint QoS Prediction for Web Services Based on Deep Fusion of Features [J]. Journal on Communications, 2022, 43 (7): 215–226.