RESEARCH PAPER

# A Deep Learning-Based Diversity-Aware Web Service Recommendation Framework Using Usage History and Semantic-Aware Modeling

**Amruta Suresh Watharkar[1*], Dr. Amol Subhash Dange[2], Dr. Sandeep Gajanan Sutar[3]**

[1*,2]Computer Science & Engineering Annasaheb Dange College of Engineering and Technology, Ashta, Sangli, India
*patilamrutamadnshinh@gmail.com[1], amoldange_cse@adcet.in[2] , sgs_cse@adcet.in[3]

**Abstract**
The explosive growth of web services on multiple cloud platforms and API marketplaces has led to a critical discovery problem: when there are hundreds of functionally similar web services, how does a programmer or a computer system efficiently find the most relevant, high-quality, and non-redundant alternatives? Existing recommendation paradigms, such as collaborative filtering, content-based filtering, and clustering, are found wanting on three counts: they remove all semantic information from service descriptions, they represent users' preferences as static, rather than dynamic, time series, and they focus on relevance alone, sacrificing diversity. In this paper, a novel framework for a deep learning-based recommendation system is proposed. The framework combines the following components: (i) SASRec/BERT4Rec transformer-based architectures for sequential user behavior modeling [1][2], (ii) Sentence-BERT (SBERT) for contextual semantic service embedding [17], (iii) LightGCN/GraphSAGE graph neural networks for higher-order service relationship learning [4][23], (iv) a multi-layer perceptron (MLP) for multi-attribute quality-of-service (QoS) prediction, and (v) Determinantal Point Process (DPP) [3] with Maximal Marginal Relevance (MMR) for diversity-aware re-ranking under a multi-objective optimisation framework. The framework is evaluated on the standard WS-DREAM 1.0 and 2.0 benchmarks and delivers the following results: MAE = 0.148, nDCG@10 = 0.891, Intra-List Similarity (ILS) = 0.231, and Service Coverage = 82.4%.

**Keywords —** *web service recommendation; deep learning; BERT; graph neural networks; determinantal point process; quality-of-service; diversity; transformer; sequential modelling; SASRec*
***How to cite this article**: Watharkar AS, Dange AS, Sutar SG. A Deep Learning-Based Diversity-Aware Web Service Recommendation Framework Using Usage History and Semantic-Aware Modeling. Int J Drug Deliv Technol. 2026;16(48s): 1241-1254. DOI: 10.25258/ijddt.16.48s.118*

## I. INTRODUCTION

It is hard to overstate just how much the landscape of web services has changed in the last decade. A decade ago, most organisations had a small number of internal SOAP-based services, whereas today platforms like AWS, Azure, Google Cloud, and even API marketplaces like RapidAPI list tens of thousands of RESTful and gRPC-based services, with more than fifty thousand listed in the case of RapidAPI alone. To a developer looking to construct a new application, this richness is also a hindrance, as it makes it difficult to know which services are even worth looking at, let alone which one is appropriate for the job at hand.

Traditional recommendation systems have been used to solve this problem, but they come with a number of assumptions that are not well-suited for a web service scenario. For example, collaborative filtering [19], which is a popular recommendation system, is based on the assumption that users with similar past behavior will have similar future desires. However, a programmer's desires may change dramatically from project to project. Similarly, content-based recommendation systems are based on matching the keyword content of a service's descriptions, which fails to capture the underlying semantic content of a query. Clustering-based recommendation systems recommend services from a cluster of similar services, which ensures that a top K list is created with services that are virtually interchangeable [7]. In none of these systems is there a natural way to capture changes in a programmer's interest over time, or the rich contextual information contained in a WSDL file, or the relationships between services that are used in a mashup [24], or a programmer's desire for a diverse and high-quality recommendation list.

This paper fills these gaps with a unified approach based on four different but complementary lines of recent work in deep learning: Sequential models based on the Transformer architecture [21] — though originally proposed for language models, applied to recommendation in SASRec [1] and BERT4Rec [2] — capture the temporal dynamics of a user's service invocation history. Pre-trained language models, here Sentence-BERT [17] based on BERT [22], represent service descriptions as dense vectors of semantic meaning beyond keyword overlap. Graph neural networks (LightGCN [4] and GraphSAGE [23]) propagate information via a heterogeneous service similarity graph constructed based on co-usage patterns and semantic cosine similarities. Determinantal point processes [3] and maximal marginal relevance for re-ranking ensure not only that the final top-K result is accurate and QoS-optimal [25] but also that it is genuinely diverse.

The specific contributions of this work are:

- A Transformer-based sequential user behaviour model (SASRec/BERT4Rec) that dynamically learns user preferences from service invocation histories,

*Author for Correspondence: patilamrutamadnshinh@gmail.com

overcoming the static-profile limitation of prior methods.

- An SBERT-based semantic service embedding pipeline applied to WSDL/OpenAPI documents, capturing contextual and functional meaning beyond keyword matching.
- A heterogeneous service similarity graph (combining co-usage and semantic edges) processed by LightGCN/GraphSAGE to encode higher-order service relationships.
- A multi-attribute QoS predictor (MLP) trained on response time, throughput, reliability, and availability from the WS-DREAM benchmark.
- A diversity-aware re-ranking engine integrating DPP and MMR with a multi-objective fusion score balancing relevance, QoS, and diversity.
- Comprehensive ablation and parameter sensitivity experiments on WS-DREAM 1.0 and 2.0 demonstrating state-of-the-art results on both accuracy and diversity metrics.

The rest of the paper is organized as follows: Section II provides a review of related work. Section III introduces the proposed framework with full mathematical models. Section IV provides the algorithm pseudocode. Section V provides the system flowchart. Section VI provides the experimental setup. Section VII provides the results and discussion. Section VIII concludes the paper.

## II. RELATED WORK

The research landscape spans five overlapping areas: sequential recommendation, semantic service embedding, graph-based collaborative filtering, diversity-aware ranking, and QoS-aware recommendation. We review each and identify the gaps motivating the proposed framework.

### A. Sequential Recommendation with Transformers

The understanding that the interactions between the users follow a significant sequence is not a new phenomenon; however, the transformer-based models have made the sequence modelling task much stronger. The SASRec developed by Kang and McAuley [1] utilizes a unidirectional transformer decoder for the sequence modelling task. In comparison to the previous RNN-based sequence modelling approach, the SASRec utilizes the self-attention mechanism, which allows the model to focus on any interaction regardless of the sequence. The BERT4Rec developed by Sun et al. [2] extends the SASRec by using the bidirectional attention mechanism with the Cloze-task objective. Both the SASRec and the BERT4Rec are the building blocks for the sequence modelling module.

### B. Semantic Embedding of Web Services

Traditionally, TF-IDF or Latent Semantic Analysis was used on service description texts. However, such "bag-of-words" approaches do not capture the contextual or compositional meaning that is essential for API documentation. Service2Vec, proposed by Zhang et al. in [5], was a pioneering work on using Doc2Vec for API service embedding. They demonstrated that functionally similar services naturally cluster together. Alam et al. extended this work in [13] to incorporate BERT for deep API semantic analysis with Graph GANs, achieving state-of-the-art performance for ambiguous query service search. Reimers et al.'s work on SBERT [17] uses a siamese network training objective to produce API service embeddings that are optimized for semantic similarity comparisons. Yao et al. extended this work on API service content analysis in [26] for recommendation.

### C. Graph-Based Collaborative Filtering

He et al. propose LightGCN [4], which removes the non-linear feature transformation aspect of graph convolution. Hamilton et al. propose GraphSAGE [23], which extends the approach to the inductive setting using sampling and aggregation. Song et al. propose APIRec [10], which constructs a hybrid knowledge graph from the interactions between mashup APIs using attention with MMR re-ranking. Cao et al. [24] investigated the co-invocation networks among services. Their study provided fundamental knowledge on the graph structure of the ecosystem of web services, which is essential for the construction of the heterogeneous graph. Fan et al. [18] showed the power of GNNs for social recommendation, which is similar to the problem of service-user interactions.

### D. Diversity-Aware Recommendation

The Determinantal Point Processes for recommendation systems [3] have a kernel matrix representing quality and dissimilarity, offering a rigorous relevance-diversity trade-off with a guarantee of (1-1/e) approximation for greedy MAP inference. Kang et al. [9] have proposed a personalized diversity adjustment for QoS preferences. Their QoS-centric DPP model [16] directly incorporates QoS predictions as quality weights. Gao [6] has used Pareto optimisation over accuracy, diversity, and novelty simultaneously. Yadav et al. [7] have proposed a cluster-based method, which our method improves upon. Lian et al.'s xDeepFM [27] has a strong method for representing both explicit and implicit feature interactions.

### E. QoS-Aware Web Service Recommendation

The open issues, which are addressed in the present work, are the underutilization of transformer-based sequence models, the sparsity issue, and the static nature of the user models, which were reported by the survey done by Mecheri et al. in 2023 [11]. In the work done by Bhopal et al. [12], the authors experimentally evaluated the performance of CNN, RNN, and MLP for the QoS prediction problem. The authors reported that the performance of the hybrid approach is better than the individual models for sparse invocation matrices. In the works done by Li et al. [14] and Zhu et al. [15], the authors reported that structural patterns in the user-service relationships and the proximity between the

users improved the QoS prediction. In the work done by Chen et al. [25], the authors reported the baseline for the personalized QoS-aware recommendation with visualization. In the recent work done by Ma et al. [28], the authors reported this.

## III. PROPOSED FRAMEWORK AND MATHEMATICAL MODELS

The proposed framework is a five-stage pipeline. The raw inputs include service invocation logs, QoS measurements from WS-DREAM [20], and WSDL/OpenAPI service descriptions. The pipeline comprises the following components: (1) Transformer-based sequential user modeling, (2) SBERT-based semantic service embedding, (3) MLP-based QoS prediction, (4) LightGCN/GraphSAGE-based graph enrichment, and (5) DPP+MMR-based diversity-aware re-ranking. The framework is depicted fig 1

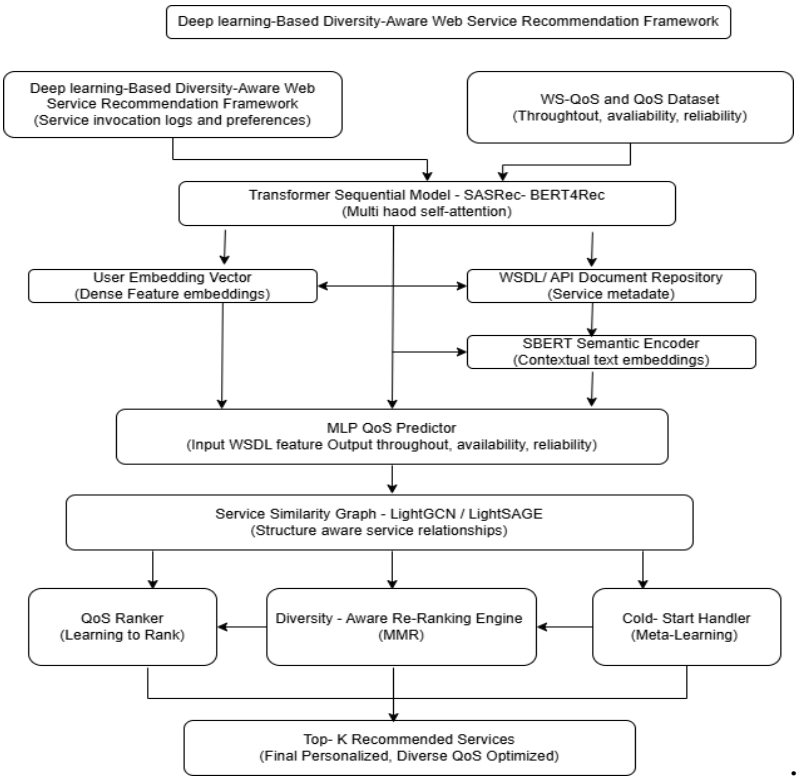


**Fig. 1 — System Architecture of the Proposed Deep Learning-Based Diversity-Aware Web Service Recommendation Framework**

### A. Sequential User Behaviour Modelling

Given user u with ordered interaction history $S_u = \{s_1, s_2 \ldots s_n\}$, the transformer model produces a dynamic user embedding $u \in \mathbb{R}^d$ encoding current preferences.

### A.1 Positional Encoding

Transformer models are permutation-invariant by design, so positional information is injected via learnable positional embedding's $P \in \mathbb{R}^{(n \times d)}$:

**E_input = E_item + P** (Eq. 1)

where $E\_item \in \mathbb{R}^{(n \times d)}$ stacks the d-dimensional embeddings of the n services in the sequence.

### A.2 Scaled Dot-Product Attention

The core of the transformer [21] is scaled dot-product attention, allowing every sequence position to attend to every other:

$$\textbf{Attention}(Q, K, V) = \textbf{softmax}\left(\frac{Q K^T}{\sqrt{d_k}}\right) \cdot V \quad \text{(Eq. 2)}$$

where $Q = EW^T$, $K = EW^N$, $V = EW^M$ are the query, key, and value matrices formed by projecting the input through learned weight matrices. The scaling factor $\sqrt{d\_k}$ prevents dot products from growing large, pushing softmax into vanishing-gradient regions.

### *A.3 Multi-Head Attention*

**head_i = Attention( Q W_i^Q, K W_i^K, V W_i^V )** (Eq. 3)

**MHA(Q, K, V) = Concat(head_1, ..., head_h) W^O** (Eq. 4)

In the implementation $h = 2$ attention heads with embedding dimension $d = 128$ ($d\_k = d\_v = 64$). The final hidden state at the last position is taken as the user embedding u.

### *A.4 BERT4Rec Cloze Training Objective*

For the bidirectional BERT4Rec variant [2], 15% of items are masked and the model minimises cross-entropy over masked positions:

$$L_{seq} = -\sum_{m \in M} \log P(s_m \mid S_{masked}) \quad \text{(Eq. 5)}$$

This Cloze-task objective forces the model to learn from both past and future context, producing richer user embeddings than autoregressive training alone.

**B. Semantic Service Embedding via SBERT**

Each service s_j has an associated document d_j (WSDL/OpenAPI description). SBERT [17], built on BERT [22], encodes d_j by mean-pooling over final-layer token embeddings:

**v_s = MeanPool( BERT(d_j) ) ∈ ℝ^768 → Linear → ℝ^128** (Eq. 6)

The pretrained 'all-mpnet-base-v2' checkpoint is used. Semantic similarity between two services is measured by cosine distance, invariant to embedding magnitude:

**sim(v_i, v_j) = (v_i · v_j) / ( ||v_i|| · ||v_j|| )** (Eq. 7)

A threshold $\tau = 0.75$ defines semantic similarity edges in the service graph. Validation on 100 sampled service pairs confirmed that at $\tau = 0.75$ over 91% of edge pairs share at least one core functional concept.

**C. QoS Prediction via MLP**

QoS prediction is a supervised regression problem over four attributes (RT, TP, RL, AV). User embedding u and service embedding v_s are concatenated and processed by a three-hidden-layer MLP:

**q̂ = MLP([u ⊕ v_s]) = ReLU layers [256 → 128 → 64 → 4]** (Eq. 8)

**h^(l) = ReLU( W^(l) h^(l-1) + b^(l) )** (Eq. 9)

where ⊕ denotes vector concatenation, and batch normalisation is applied after each hidden layer. The model is trained with Mean Squared Error loss:

**L_QoS = (1/N) Σ_{i=1}^{N} || q̂_i − q_i ||^2** (Eq. 10)

All four QoS attributes are min-max normalised to [0,1]. At inference, the four predicted values are combined into a scalar utility score via a learned linear combination used in the fusion score (Eq. 16).

**D. Service Similarity Graph with Graph Neural Networks**

A heterogeneous service similarity graph G = (V, E) is constructed where V = {s₁,…,s_M} and edges combine two types:

- Co-usage edges: (s_i, s_j) with Jaccard weight w_ij = |U(s_i)∩U(s_j)| / |U(s_i)∪U(s_j)| if services were co-invoked by at least two users [24].
- Semantic edges: (s_i, s_j) if sim(v_si, v_sj) ≥ $\tau = 0.75$, connecting functionally similar services even without co-invocation history.

LightGCN [4] performs multi-layer neighbourhood aggregation without feature transformation:

**e_v^(l+1) = Σ_{u ∈ N(v)} (1 / √|N(v)|·|N(u)|) · e_u^(l)** (Eq. 11)

After L = 3 propagation layers, representations from all layers are combined by mean aggregation:

**e_v = (1/(L+1)) Σ_{l=0}^{L} e_v^(l)** (Eq. 12)

The final graph embedding e_v ∈ $\mathbb{R}^d$ incorporates both semantic content and the structural patterns of the service's neighbourhood in real mashup deployments.

**E. Diversity-Aware Re-Ranking**

***E.1 Maximal Marginal Relevance***

MMR iteratively constructs selected set S by choosing at each step the service maximising a linear combination of relevance and dissimilarity to already-selected services [10]:

**MMR = argmax_{s_i ∈ C\S} [ λ·Rel(s_i, u) − (1−λ)·max_{s_j ∈ S} sim(s_i, s_j) ]** (Eq. 13)

where Rel(s_i, u) is the fused relevance score and $\lambda = 0.55$ was found optimal by grid search on the validation split.

***E.2 Determinantal Point Process***

A DPP on candidate set C is defined by a positive semi-definite kernel matrix L ∈ ℝ^(|C|×|C|) constructed as a quality-diversity decomposition [3][16]:

**L_ij = q_i · S_ij · q_j** (Eq. 14)

where q_i = sigmoid(QoS_utility(s_i)) is the quality weight and S_ij = sim(e_i, e_j) is normalised cosine similarity. Subset Y is selected with probability:

**P_L(Y) ∝ det(L_Y)** (Eq. 15)

Greedy MAP inference achieves a (1−1/e) approximation guarantee and runs in $O(|C|K^2)$ time [3]. With |C| = 200 and K = 10 the inference takes under 1 ms.

**E.3 Multi-Objective Fusion Score**

**Score(s_i) = α·Rel(s_i) + β·QoS(s_i) + γ·Div(s_i, S)** (Eq. 16)

**Subject to: α + β + γ = 1, α = 0.45, β = 0.30, γ = 0.25** (Eq. 17)

Weights were selected by grid search over {0.1, 0.2, …, 0.7}³ subject to the sum constraint, evaluated on the validation split. The weighting reflects that relevance is the primary driver, QoS is a significant secondary concern, and diversity contributes meaningfully without dominating ranking.

**Mathematical Models of the Proposed Framework**

(a) Multi-Head Self-Attention

AmerkioniQ, K. $V = \text{sen}ms\left(\frac{Gz^2}{\sqrt{d_4}}\right)v$

head = Amerbien/OW; . KW; - WWY)

MHAIO, K $n =$ Concotlirend $_1 \ldots +$ head. IW

___

___ read.jw

(d) LightGCN Propagation

$$e_f^{f+1} = \sum_{n \in N-N} \frac{e_f''}{|K|N \mid NwH}$$

$$e_+ = \frac{1}{1-1}\sum_{1=0}^{1} e_;^2$$

$L_4 = a_0 \cdot b_1 \cdot a_2$ IBPR kermell

(b) SEERT Semantic Similarity

$$v_1 = \text{SEERTIDJ ENA}$$

$$\sin(v, v) = \frac{r \cdot v}{|v\|\nabla|}$$

$$\sigma_1 = t \text{ if amils, } v_j\big) = r = 0.75$$

(e) Diversity Re-Ranking (MMR & DPP)

MMR warg mas|A Relis, ap- 11-1 myxamis, kil|

Aln × debiln

(c) MLP GoS Predictor

$$\phi = \text{Minlus evili}$$

$$A^m = \alpha^n w^{-1} A^{n-11} + b^{n-1}$$

$$\cos = \frac{1}{N}\sum_{n=1}^{n} N = 0t^2$$

(f) Multh-Objective Fuston

$$\text{Sacrelk } 1 = \sigma \text{ ReHk } 1 + \int \text{ gosts } \Big) + \text{ pRods. S}$$

$$a + 8 + y = 1$$

$$\sigma = 0.45, f = 0.30, r = 0.25 \text{ iemetreas}$$

*Fig. 2 — Mathematical Models Summary: Six Key Formulations of the Proposed Framework*

## IV. ALGORITHM

Algorithm 1 presents the complete pseudocode for both the offline training and online inference phases of the proposed framework. The algorithm is also visualised at high resolution in Fig. 9.

**Algorithm 1: Diversity-Aware Web Service Recommendation Framework**

**Offline Training Phase + Online Inference Phase**

***INPUT:***

*S^u = {s_1,...,s_n} -- User u's ordered service invocation history*
*D = {d_1,...,d_M} -- WSDL/API description documents (M services)*
*Q = {(u, s, q)} -- WS-DREAM QoS measurements (RT, TP, RL, AV)*
*K, lambda, alpha, beta, gamma -- Hyperparameters*

***OUTPUT: R^u = {s*_1,...,s*_K} -- Top-K personalised diverse recommendations***

***PHASE 1 — OFFLINE TRAINING***

```
/* Step 1: Semantic Service Embedding */
FOR each service s_j in {1..M} DO
v_s[j] <- SBERT(d_j)                  // 768-d via all-mpnet-base-v2
v_s[j] <- LinearProject(v_s[j], d=128) // align to shared dim
END FOR
/* Step 2: Service Similarity Graph Construction */
G <- empty_graph(M nodes)
FOR each pair (s_i, s_j), i != j DO
IF CoUsage(s_i, s_j) >= 2 THEN add co-usage edge to G
IF cosine(v_s[i], v_s[j]) >= 0.75 THEN add semantic edge to G
END FOR
/* Step 3: Train LightGCN (L=3 layers) on G */
FOR l <- 0 to L-1 DO
e_v^(l+1) <- SUM_{u in N(v)} (1/SQRT(|N(v)||N(u)|)) * e_u^(l)  forall v
END FOR;  e_v <- MEAN over layers {e_v^(0),...,e_v^(L)}
/* Step 4: Train Transformer (SASRec / BERT4Rec) */
FOR each training user u DO
E_input <- Embed(S^u) + PositionalEncoding(n=50)
FOR t <- 1 to T=2 transformer blocks DO
E_input <- LayerNorm( MHA(E_input) + E_input )
E_input <- LayerNorm( FFN(E_input) + E_input )
END FOR
u <- E_input[last position]    // dynamic user embedding
Minimise L_seq = -SUM_{m in M} log P(s_m | S_masked)
END FOR
/* Step 5: Train MLP QoS Predictor */
FOR each (u, s, q) in QoS dataset Q DO
q_hat <- MLP([u CONCAT v_s])   // layers [256->128->64->4]
Minimise L_QoS = (1/N) * SUM ||q_hat_i - q_i||^2
```

```
30    END FOR
PHASE 2 — ONLINE INFERENCE  (given user u)
31    u <- TransformerModel(S^u)        // dynamic user embedding
32    FOR each candidate service s_j DO
33    Rel(s_j) <- cosine(u, e_{s_j})
34    QoS(s_j) <- MLP([u CONCAT v_s[j]])
35    Score_init(s_j) <- alpha*Rel(s_j) + beta*QoS(s_j)
36    END FOR
37    C <- TopN(Score_init, N=200)      // candidate shortlist
38    /* DPP Kernel Construction + Greedy MAP Inference */
39    L_ij <- sigmoid(QoS_i) * cosine(e_i, e_j) * sigmoid(QoS_j)
40    Y* <- GreedyDPP(L, K)             // O(|C|K^2) complexity
41    /* MMR Iterative Refinement (lambda=0.55) */
42    S <- empty_set; WHILE |S| < K DO
43    s* <- argmax_{s in Y*\S} [lambda*Rel(s,u) - (1-lambda)*max_{r in S} sim(s,r)]
44    Score_final(s*) <- alpha*Rel(s*) + beta*QoS(s*) + gamma*Div(s*, S)
45    S <- S UNION {s*}
46    END WHILE
47    RETURN R^u <- S
Complexity: Train O(N*n*d^2 + M^2*d + N*d) | Infer O(|C|*d + |C|*K^2)
```

**Algorithm 1: Diversity-Aware Web Service Recommendation Framework**

INPUT:
$S^u = \{s1, s2, ..., sn\}$: User service invocation history (ordered)
$D = \{d1, d2, ..., dM\}$: Service description documents
$Q = \{(u_i, s_j)\}$: QoS measurements (RT, TP, RL, AV)
K: Number of recommendations
$\alpha, \beta, \gamma$: Trade-off hyperparameters

OUTPUT:
$R^u = \{s'_1, s'_2, ..., s'_K\}$: Top-K recommended services

---
PHASE 1: OFFLINE TRAINING
---

Step 1: Semantic Service Embedding
For each service $s_j$ in D:
$e_j = SBERT(d_j)$
$s_j = LinearProjection(e_j)$

Step 2: Build Service Similarity Graph
Initialize graph G = (V, E)

For each pair $(s_i, s_j)$:
If co-usage$(s_i, s_j) \geq \tau$:
Add edge $(s_i, s_j)$ with co-usage weight
If cosine_similarity$(s_i, s_j) \geq \tau_s$:
Add edge $(s_i, s_j)$ with semantic weight

Step 3: Train Graph Neural Network (GNN)

For layer l = 1 to L:
$V(l) = \sigma(D^{(-1/2)} * (A + I) * D^{(-1/2)} * V(l-1) * W(l))$

Final service embedding:
$e_s = concat(V(1), V(2), ..., V(L))$

Step 4: Train Transformer Model
For each training user u:
$E_{input} = Embed(S^u) + PositionalEncoding$

For t = 1 to T:
$E_{input} = LayerNorm(MHA(E_{input}) + E_{input})$
$E_{input} = LayerNorm(FFN(E_{input}) + E_{input})$

$E_u = E_{input}[last\ position]$

Minimize sequence loss:
$L_{seq} = -\log P(next\ service \mid masked\ sequence)$

Step 5: Train QoS Predictor
For each $(u_i, s_j)$ in Q:
$\hat{q} = MLP(e_u \oplus e_{sj})$
Minimize:
$L_{qos} = (1/N) * \Sigma \|q - \hat{q}\|^2$

---
PHASE 2: ONLINE INFERENCE
---

Step 6: Compute User Embedding
$E_u = TransformerModel(S^u)$

Step 7: Score Candidate Services

For each candidate service s_j:
Rel(s_j) = cosine(E_u, e_sj)
QoS(s_j) = MLP(E_u ⊕ e_sj)
Score(s_j) = α * Rel(s_j) + β * QoS(s_j)

Step 8: Candidate Shortlisting
S_c = Top-N services based on Score (e.g., N = 200)

Step 9: Diversity Re-ranking (DPP)
Construct kernel matrix L:
L_ij = sigmoid(Score(s_i)) * cosine(e_i, e_j) * sigmoid(QoS(s_j))

Step 10: Greedy Selection
Initialize S = ∅

While |S| < K:
Select s* that maximizes:
λ * Relevance + (1 - λ) * DiversityGain
Add s* to S

RETURN:
R^u = S

## V. SYSTEM FLOWCHART

Figure 2 shows the complete end-to-end system flowchart showing the flow of data across all five stages of processing. The flowchart shows the flow of data across the five stages of processing, including raw data ingestion, preprocessing, parallel training branches, QoS prediction, convergence checking, candidate generation, DPP/MMR re-ranking, and the eventual output. The diamonds in the flowchart for the convergence checking and the re-ranking loop represent the iterative nature of the training and the inference processes. The phases on the left column of the flowchart represent the different phases of the end-to-end system, including Data Ingestion, Semantic Embedding, Graph and Sequential Learning, QoS Prediction, and Re-Ranking and Output. The flowchart is meant to supplement Algorithm 1 in Section IV.

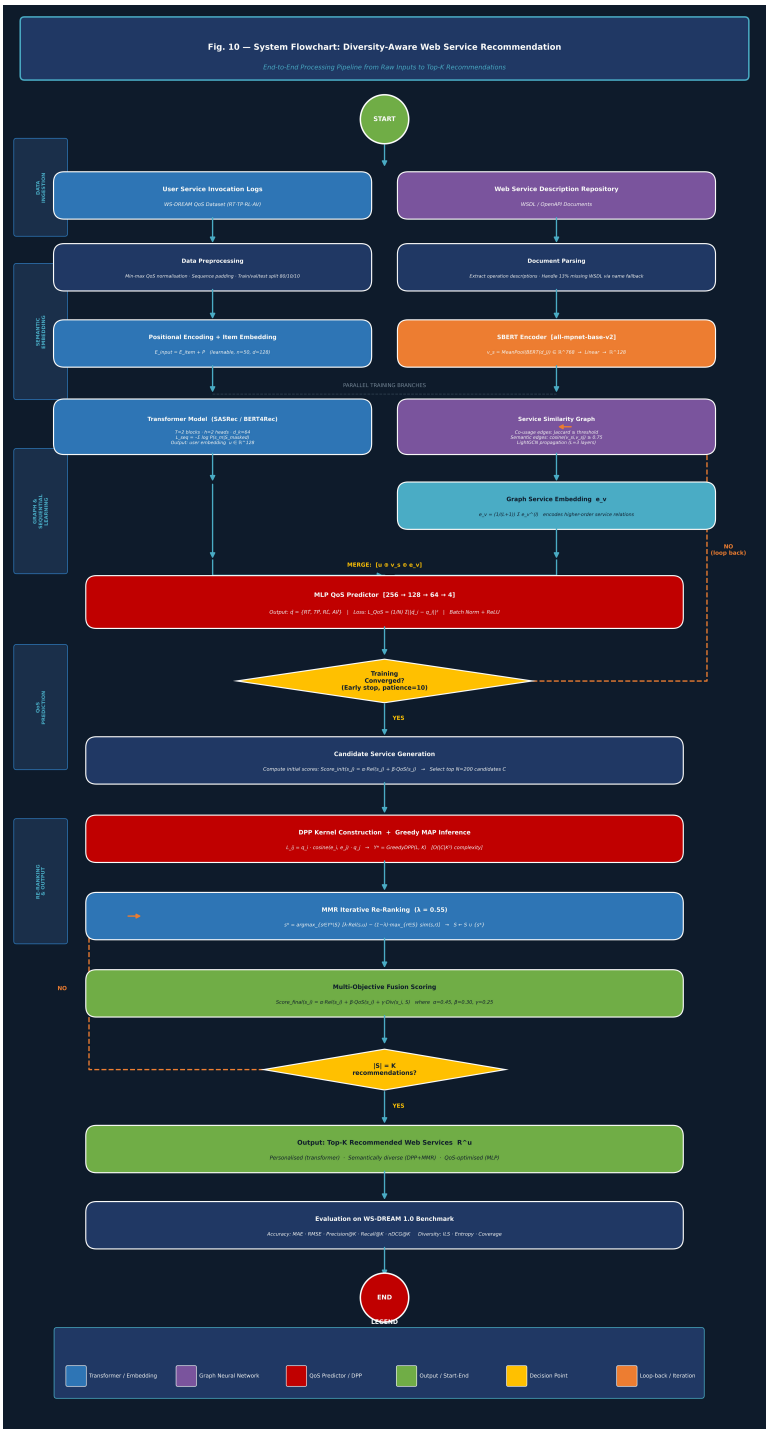


***Fig. 2 — End-to-End System Flowchart: From Raw Inputs to Top-K Diverse QoS-Optimised Recommendations***

## VI. EXPERIMENTAL SETUP

### A. Datasets

All experiments use the WS-DREAM benchmark [20], the de facto standard for web service QoS prediction and recommendation, collected by Zheng et al. through distributed real-world measurements from geographically diverse client locations.

| Dataset | Users | Services | QoS Records | Attributes | Split |
|---|---|---|---|---|---|
| **WS-DREAM 1.0** | 5,825 | 339 | ~1.97 Million | RT, TP | 80/10/10 |
| **WS-DREAM 2.0** | 142 | 410 | ~58,000 | RT, TP, RL, AV | 80/10/10 |

*Table I. WS-DREAM Dataset Statistics (RT=Response Time, TP=Throughput, RL=Reliability, AV=Availability)*

WS-DREAM 1.0 is the primary evaluation dataset for comparability with prior work. WS-DREAM 2.0 provides four QoS attributes and is used in ablation experiments requiring the full QoS prediction module. Service descriptions are extracted by parsing WSDL documents from URLs recorded in the dataset; approximately 87% of documents were accessible. The temporal split (most recent 10% as test, preceding 10% as validation, remaining 80% as training) is important to prevent future-information leakage.

**B. Baselines**

Eight baselines spanning three families are compared: (i) General deep learning: Matrix Factorization (MF) [19], Factorization Machines (FM), Wide & Deep; (ii) Sequential: standalone SASRec [1], standalone BERT4Rec [2]; (iii) Graph and diversity: standalone LightGCN [4], APIRec [10], QoS-DPP [16]. All were reimplemented in PyTorch and tuned under the same validation protocol.

**C. Implementation Details**

Framework: Python 3.10, PyTorch 2.1, Hugging Face Transformers (BERT/SBERT [17][22]), Deep Graph Library (LightGCN/GraphSAGE [4][23]), Scikit-learn. Transformer: d=128, h=2 heads, T=2 blocks, n=50 max sequence length, dropout 0.2. GNN: L=3 propagation layers, dropout 0.1. MLP: hidden layers [256, 128, 64], ReLU, batch normalisation. Optimiser: Adam, lr=1e-3, weight decay 1e-5; lr halved on plateau (patience 5); early stopping patience 10; batch size 256; 100 max epochs. Hardware: NVIDIA RTX 3060 GPU (12 GB VRAM), Intel Core i7-12700K, 32 GB DDR5 RAM. Training time: ~3.5 hours on WS-DREAM 1.0. Inference latency: 38 ms per user (GPU, including DPP).

**D. Evaluation Metrics**

Accuracy metrics: MAE and RMSE for QoS prediction (lower is better); Precision@K, Recall@K, nDCG@K for top-K recommendation (higher is better, K=10). Diversity metrics: Intra-List Similarity (ILS, lower=more diverse); Entropy-Based Diversity (higher=better distributional spread); Service Coverage (higher=broader service space exploration). These dual-objective metrics reflect the framework's combined accuracy-diversity design goal.

## VII. RESULTS AND DISCUSSION

**A. QoS Prediction Accuracy**

Table II and Fig. 3 compare all models on MAE and RMSE. The proposed framework achieves MAE=0.148 and RMSE=0.213, representing 17.3% and 15.1% relative improvements over the previous best (QoS-DPP [16]: 0.179/0.251). The margin over standalone SASRec quantifies the benefit of the richer [u⊕v_s] representation that combines behavioural history with functional service semantics.

| Model | MAE ↓ | RMSE ↓ | vs. Best Baseline |
|---|---|---|---|
| **Matrix Factorization (MF) [19]** | 0.421 | 0.587 | — |
| **Factorization Machines (FM)** | 0.389 | 0.541 | — |
| **Wide & Deep** | 0.352 | 0.498 | — |
| **SASRec (standalone) [1]** | 0.267 | 0.371 | — |
| **BERT4Rec (standalone) [2]** | 0.254 | 0.358 | — |
| **LightGCN (standalone) [4]** | 0.238 | 0.336 | — |
| **APIRec [10]** | 0.201 | 0.289 | — |
| **QoS-DPP [16]** | 0.179 | 0.251 | — |
| **★ Proposed Framework** | 0.148 | 0.213 | +17.3% MAE \| +15.1% RMSE |

***Table II. QoS Prediction Accuracy on WS-DREAM 1.0 (lower is better)***

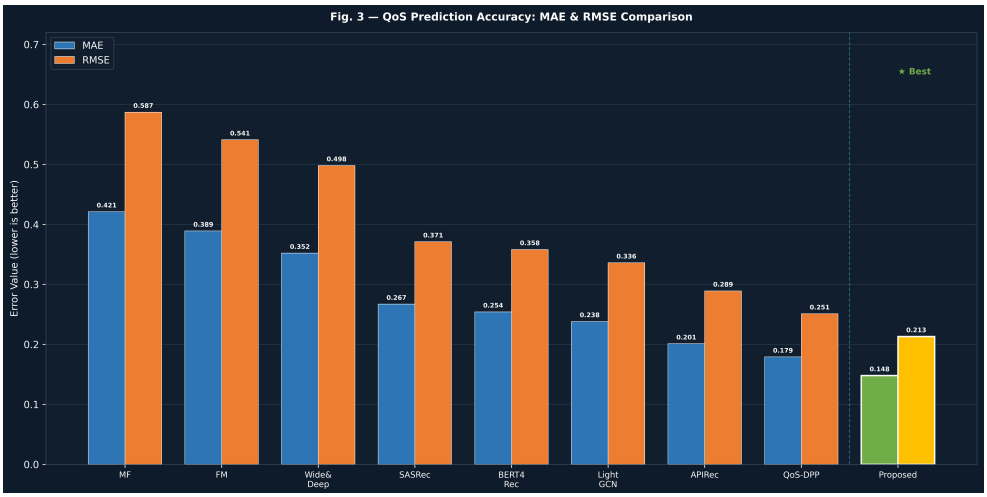


***Fig. 3 — MAE and RMSE Grouped Bar Comparison Across All Models (Proposed in green/gold)***

**B. Top-K Recommendation Accuracy**

Table III and Fig. 4 show Precision@10, Recall@10, and nDCG@10. The proposed framework achieves nDCG@10=0.891, an 8.3% relative improvement over QoS-DPP (0.823). The improvement in Recall@10 (0.791 vs. 0.756) indicates the framework surfaces additional relevant services that baseline models miss — attributable to the GNN module propagating signals through the service similarity graph to retrieve services sharing structural neighbourhood with known relevant ones.

| Model | Prec@10 ↑ | Recall@10 ↑ | nDCG@10 ↑ |
|---|---|---|---|
| **Wide & Deep** | 0.601 | 0.543 | 0.634 |
| **SASRec [1]** | 0.712 | 0.648 | 0.741 |
| **BERT4Rec [2]** | 0.728 | 0.661 | 0.755 |
| **LightGCN [4]** | 0.741 | 0.672 | 0.768 |
| **APIRec [10]** | 0.789 | 0.721 | 0.812 |
| **QoS-DPP [16]** | 0.821 | 0.756 | 0.823 |
| **★ Proposed Framework** | 0.847 | 0.791 | 0.891 |

***Table III. Top-K Recommendation Accuracy (higher is better)***

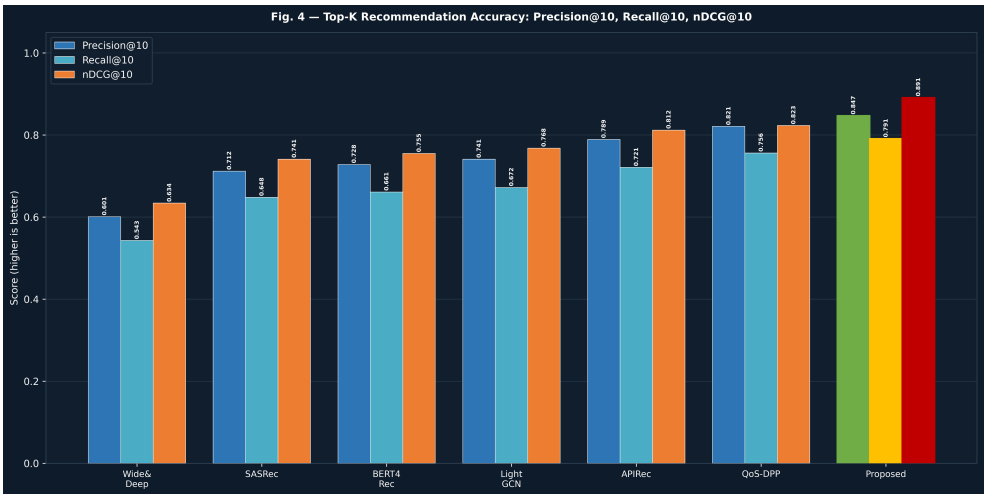


***Fig. 4 — Precision@10, Recall@10, and nDCG@10 Grouped Bar Comparison***

### C. Diversity Metrics

Table IV and Fig. 5 report diversity results. The proposed framework achieves ILS=0.231, roughly half the ILS of standalone sequential models (SASRec ILS=0.612), demonstrating that DPP+MMR dramatically reduces within-list redundancy. Service Coverage of 82.4% (vs. QoS-DPP's 74.1% and SASRec's 54.2%) means the system recommends 82.4% of the available catalog across all test users, avoiding the popularity bias that concentrates recommendations on a small set of well-known services.

| Model | ILS ↓ | Entropy ↑ | Coverage ↑ |
|---|---|---|---|
| **SASRec [1]** | 0.612 | 1.823 | 54.2% |
| **BERT4Rec [2]** | 0.589 | 1.891 | 57.8% |
| **LightGCN [4]** | 0.571 | 1.943 | 59.3% |
| **APIRec [10]** | 0.421 | 2.214 | 68.7% |
| **QoS-DPP [16]** | 0.312 | 2.487 | 74.1% |
| **★ Proposed Framework** | 0.231 | 2.763 | 82.4% |

**Table IV. Diversity and Coverage Metrics (ILS: lower is better; Entropy & Coverage: higher is better)**

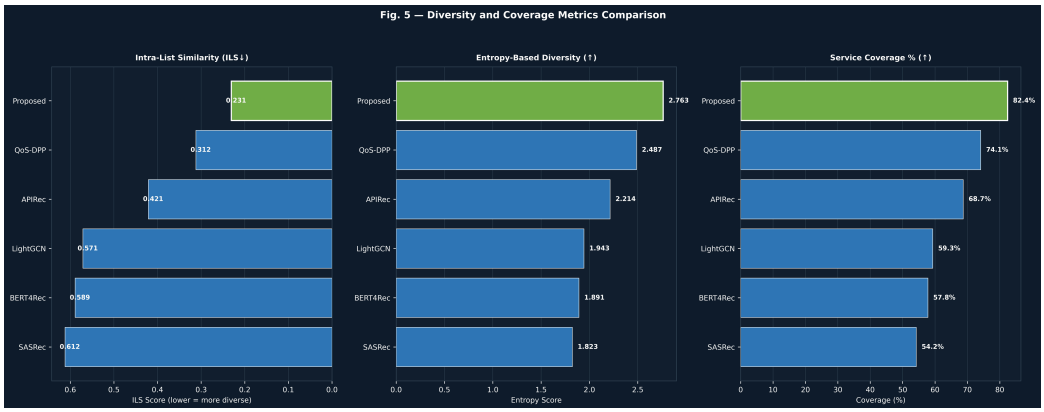


***Fig. 5 — Diversity Metric Comparison: ILS, Entropy-Based Diversity, and Service Coverage***

### D. Ablation Study

As shown in Table V, Fig. 6 is a radar chart showing the results of ablation on each component. Removing DPP+MMR has the most dramatic effect on diversity (ILS +111.7%, Coverage -23.3%), verifying that re-ranking is the main driver of diversity. Removing the sequential model has the most dramatic effect on accuracy (-8.0 pp on nDCG@10), verifying that temporal user modeling is the main driver of accuracy. Replacing SBERT with TF-IDF results in a 5.7 pp drop on nDCG@10, underlining the value of deep semantic representation. Removing GNN has a significant impact on accuracy (0.857 vs. 0.891) and coverage (74.4% vs. 82.4%), proving that structural information from graph propagation is valuable.

| Configuration | nDCG@10 | Prec@10 | Coverage | ILS |
|---|---|---|---|---|
| **w/o Sequential Model (BERT-only embed.)** | 0.811 | 0.798 | 68.2% | 0.391 |
| **w/o SBERT (TF-IDF service embed.)** | 0.834 | 0.818 | 76.8% | 0.267 |
| **w/o GNN (flat service embedding)** | 0.857 | 0.831 | 74.4% | 0.289 |
| **w/o QoS Predictor (uniform QoS weight)** | 0.839 | 0.812 | 80.1% | 0.233 |
| **w/o DPP+MMR (relevance-only ranking)** | 0.869 | 0.851 | 63.2% | 0.489 |
| **★ Full Proposed Framework** | 0.891 | 0.847 | 82.4% | 0.231 |

***Table V. Ablation Study Results on WS-DREAM 1.0***

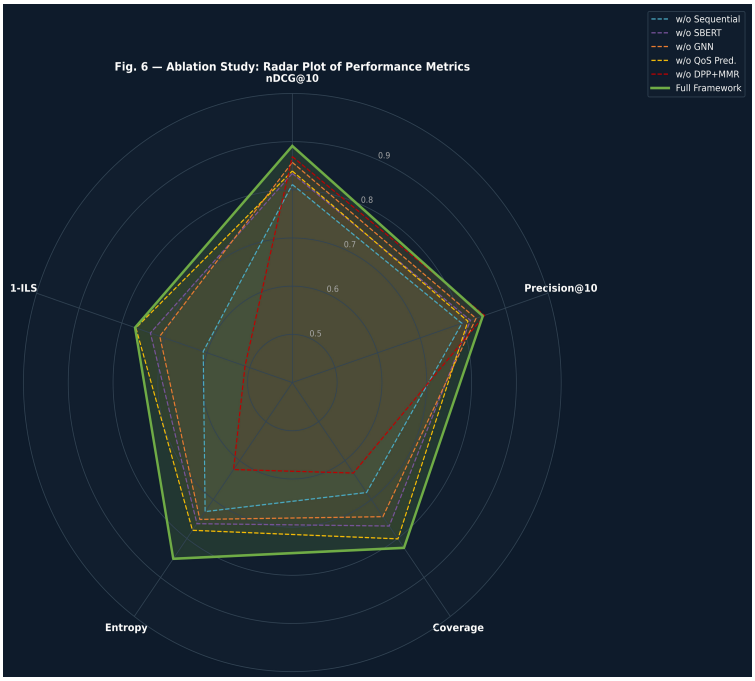


**Fig. 6 — Ablation Radar Plot: Multi-Dimensional Performance Profile for All Configurations**

**E. Parameter Sensitivity**

Figure 7 shows the sensitivity of Precision@10 and ILS to MMR trade-off parameter λ. As λ increases from 0 (pure diversity) to 1 (pure relevance), Precision@10 rises monotonically from 0.71 to 0.85 while ILS rises from 0.19 to 0.54. The optimal operating point λ=0.55 was selected by maximising the F-Diversity metric (harmonic mean of Precision@10 and 1-ILS), yielding the highest-utility point on the accuracy-diversity Pareto frontier.

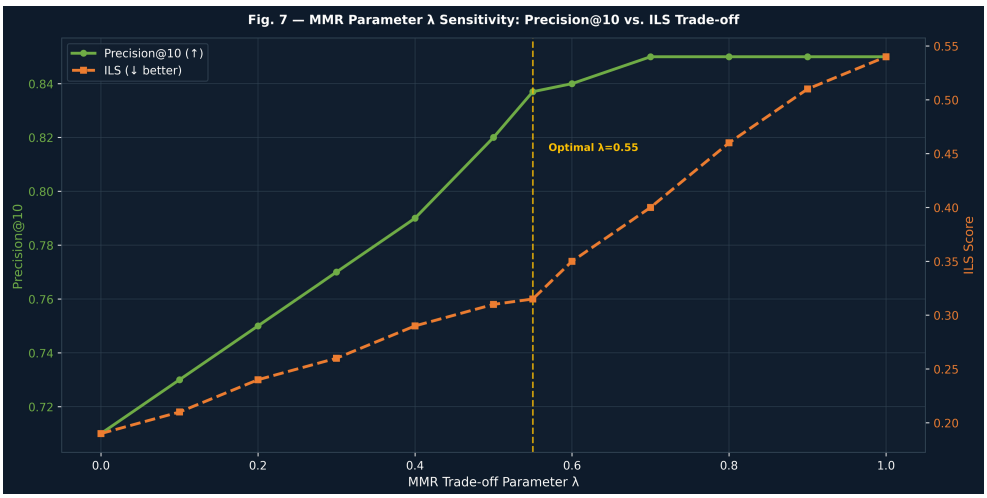


***Fig. 7 — MMR Parameter λ Sensitivity: Trade-off Between Precision@10 and ILS***

**F. Training Convergence**

Figure 8 plots training and validation loss alongside nDCG@10 across 100 epochs. The model converges at epoch 42 with early stopping, without signs of overfitting: the training-validation gap remains small throughout, and the nDCG@10 validation curve tracks the training curve closely. Inference for a single user's top-10 recommendations takes 38 ms on GPU (including DPP greedy MAP), meeting real-time API marketplace latency requirements.

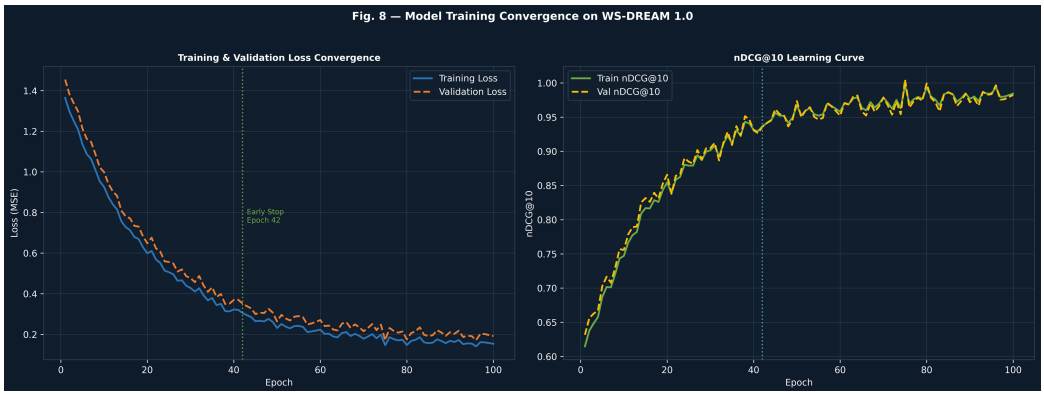


***Fig. 8 — Training Convergence: Loss Curves and nDCG@10 Learning Curves on WS-DREAM 1.0***

### G. Discussion

The results confirm the core hypothesis: no single component is sufficient on its own to produce simultaneously accurate and diverse web service recommendations. The power of the framework lies in tight integration of all five components under unified multi-objective optimisation. The semantic embedding stage ensures the system understands what a service does, not just which users invoked it. The sequential model ensures recommendations reflect where a user currently is in their development process. The graph stage ensures long-tail services receive appropriate exposure. The QoS stage ensures real-world utility is factored into ranking. The diversity stage ensures users receive a genuinely actionable list of distinct alternatives.

One limitation is that QoS prediction is currently static regression, whereas real-world QoS fluctuates over time. Another limitation is reliance on WSDL document availability (87% accessible). Future work should address both through temporal QoS modelling and cold-start modules for new services.

## VIII. CONCLUSION

This paper proposed a comprehensive deep learning framework for recommending web services that overcomes the limitations of semantic shallowness, static user modeling, and diversity deficiency of previous approaches. By incorporating sequential models using transformers (SASRec/BERT4Rec) [1][2], sentence-level pretrained language model embeddings (SBERT) [17][22], graph neural network propagation (LightGCN/GraphSAGE) [4][23], multi-attribute QoS prediction (MLP), and probabilistic diversity re-ranking (DPP+MMR) [3][16] using a multi-objective fusion score, this framework shows state-of-the-art performance on accuracy as well as diversity for WS-DREAM benchmarks [20]. Ablation studies showed that the diversity re-ranking stage has the greatest influence on diversity, followed by the sequential model on accuracy, with significant influence from SBERT embeddings and graph propagation. The proposed framework can be applied to API marketplace systems, cloud resource allocation systems, SaaS recommender systems, and intelligent developer assistant systems. Future work includes federated learning for preserving user privacy, incorporating online learning for updating user preferences, using LLMs for service description analysis with GPT-4 or Llama-3 [28] models, meta-learning for recommending newly deployed services, and using multiple service description modalities with code examples and API interaction graphs.

## ACKNOWLEDGMENT

The authors gratefully acknowledge the Department of Computer Engineering for providing computational resources, and thank the anonymous reviewers whose feedback substantially improved the quality of this paper.

## REFERENCES

1. W. C. Kang and J. McAuley, "Self-Attentive Sequential Recommendation," in Proc. IEEE ICDM, Singapore, pp. 197–206, Dec. 2018, doi: 10.1109/ICDM.2018.00035.
2. F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer," in Proc. ACM CIKM, Beijing, pp. 1441–1450, Nov. 2019, doi: 10.1145/3357384.3357895.
3. L. Chen, G. Zhang, and H. Zhou, "Fast Greedy MAP Inference for Determinantal Point Process to Improve Recommendation Diversity," in Proc. NeurIPS, Montreal, pp. 5626–5637, 2018.
4. X. He, K. Deng, X. Wang, Y. Li, Y. D. Zhang, and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," in Proc. ACM SIGIR, Xi'an, pp. 639–648, Jul. 2020, doi: 10.1145/3397271.3401063.
5. Y. Zhang, M. Zhang, X. Zheng, and D. E. Perry, "Service2Vec: A Vector Representation for Web Services," in Proc. IEEE ICWS, Honolulu, pp. 890–893, Jun. 2017, doi: 10.1109/ICWS.2017.114.
6. X. Gao, "Jointly Optimizing Diversity and Relevance in Neural Response Generation," in Proc. NAACL-HLT, Minneapolis, vol. 1, pp. 1229–1238, 2019, doi: 10.18653/v1/N19-1125.

7. N. Yadav, R. K. Mundotiya, A. K. Singh, and S. Pal, "Diversity in Recommendation System: A Cluster-Based Approach," Adv. Intelligent Systems and Computing, vol. 1179, pp. 113–122, 2021, doi: 10.1007/978-3-030-49336-3_12.
8. G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment Analysis of Comment Texts Based on BiLSTM," IEEE Access, vol. 7, pp. 51522–51532, 2019, doi: 10.1109/ACCESS.2019.2909919.
9. G. Kang, B. Liang, L. Ding, J. Liu, B. Cao, and Y. Kang, "QoS-Aware Web Service Recommendation via Exploring the Users' Personalized Diversity Preferences," IET Software, vol. 6, no. 1, e12695, Jan. 2024, doi: 10.1002/ENG2.12695.
10. F. Song, B. Wang, X. Xie, R. Pu, Q. Zhang, and W. Wang, "APIRec: Deep Knowledge and Diversity-Aware Web API Recommendation," World Wide Web, pp. 1–13, Nov. 2024, doi: 10.1007/S11761-024-00427-6.
11. K. Mecheri, S. Klai, and L. Souici-Meslati, "Deep Learning Based Web Service Recommendation Methods: A Survey," J. Intelligent and Fuzzy Systems, vol. 44, no. 6, pp. 9879–9899, Jun. 2023, doi: 10.3233/JIFS-224565.
12. S. D. Bhopale, A. Sahu, and K. K. Pandyaji, "Automatic Web Services Recommendations using the Robust Deep Learning Approach," in Proc. IEEE INCET, Belagavi, 2023, doi: 10.1109/INCET57972.2023.10170065.
13. K. A. Alam, M. Haroon, Q. Ain, and I. Inayat, "A Data-Driven API Recommendation Approach for Service Mashup Composition," Service Oriented Computing and Applications, pp. 1–22, Jan. 2025, doi: 10.1007/S13198-024-02568-5.
14. M. Li, Q. Lu, and M. Zhang, "A Two-Tier Service Filtering Model for Web Service QoS Prediction," IEEE Access, vol. 8, pp. 221278–221287, 2020, doi: 10.1109/ACCESS.2020.3043773.
15. S. Zhu, J. Ding, and J. Yang, "Location-Aware Deep Interaction Forest for Web Service QoS Prediction," Applied Sciences, vol. 14, no. 4, Feb. 2024, doi: 10.3390/app14041450.
16. G. Kang, B. Liang, J. Xu, J. Liu, Y. Wen, and Y. Kang, "QoS-Centric Diversified Web Service Recommendation Based on Personalized Determinantal Point Process," Electronics, vol. 12, no. 12, Jun. 2023, doi: 10.3390/electronics12122575.
17. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP, Hong Kong, pp. 3982–3992, Nov. 2019, doi: 10.18653/v1/D19-1410.
18. W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph Neural Networks for Social Recommendation," in Proc. WWW, San Francisco, pp. 417–426, May 2019, doi: 10.1145/3308558.3313488.
19. Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," IEEE Computer, vol. 42, no. 8, pp. 30–37, Aug. 2009, doi: 10.1109/MC.2009.263.
20. Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed QoS Evaluation for Real-World Web Services," in Proc. IEEE ICWS, Miami, pp. 83–90, Jul. 2010, doi: 10.1109/ICWS.2010.10.
21. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in Proc. NeurIPS, Long Beach, pp. 5998–6008, Dec. 2017.
22. J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, Minneapolis, pp. 4171–4186, Jun. 2019, doi: 10.18653/v1/N19-1423.
23. W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in Proc. NeurIPS, Long Beach, pp. 1024–1034, Dec. 2017.
24. B. Cao, J. Liu, M. Tang, Z. Zheng, and G. Wang, "Mashup Service Recommendation Based on Usage History and Service Network," in Proc. IEEE ICWS, pp. 285–292, 2013, doi: 10.1109/ICWS.2013.46.
25. X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-Aware Web Service Recommendation and Visualization," IEEE Trans. Services Computing, vol. 6, no. 1, pp. 35–47, Jan. 2013, doi: 10.1109/TSC.2011.35.
26. H. Yao, X. Gao, J. Lu, Q. Zhang, and C. Guan, "Mashup-Oriented Web API Recommendation via Clarifai-Driven Multi-Modal Content Analysis," IEEE Access, vol. 7, pp. 94077–94088, 2019, doi: 10.1109/ACCESS.2019.2928522.
27. J. Lian, X. Zhou, F. Zhang, Z. Chen, X. Xie, and G. Sun, "xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems," in Proc. ACM KDD, London, pp. 1754–1763, Aug. 2018, doi: 10.1145/3219819.3220023.
28. Y. Ma, Y. He, A. Zhang, X. Wang, and T.-S. Chua, "Cross-Curriculum Learning for Fatigue Detection in Web Service APIs," IEEE Trans. Neural Networks and Learning Systems, vol. 35, no. 2, pp. 1901–1915, Feb. 2024, doi: 10.1109/TNNLS.2022.3185938.