International Journal of Innovative Research and Scientific Studies, 8(6) 2025, pages: 3374-3384

IJIRSS International Journal of Innovative Research & Scientific Studies

ISSN: 2617-6548

URL: www.ijirss.com

Social data-driven travel: Integrating NCF and wide & deep learning for personalized experiences (DeepTravelRS)

Mohamed Badouch[1*], Mehdi Boutaounte[2]

[1]*Department of computer sciences, Faculty of Sciences, Ibn Zohr University, Agadir 80000, Morocco.*
[2]*Department of computer sciences, National School of Commerce and Management, Ibn Zohr University, Dakhla 73000, Morocco.*

Corresponding author: Mohamed Badouch (*Email: mohamed.badouch@edu.uiz.ac.ma*)

Abstract

DeepTravelRS is a real-time, large-scale tourism recommender system engineered to personalize travel itineraries by integrating heterogeneous contextual factors—including user profiles, travel modes, seasonality, budgets, and geographic constraints—while sustaining sub-20 ms response latency in production environments. Its hybrid architecture fuses Neural Collaborative Filtering for nonlinear latent user–item interaction modeling, Wide & Deep Learning to capture explicit contextual features and memorization, and a Neo4j knowledge graph encoding semantic and geographic relationships between users, points of interest (POIs), and hotels. Real-time orchestration is achieved through GraphQL serving and Redis caching, delivering a 92 % cache hit rate, while Proximal Policy Optimization dynamically adapts recommendations to evolving travel contexts. Experiments on a TripAdvisor-derived dataset containing 52 000 attractions, 2.1 million graph nodes, and 9 141 cities with temporal and geospatial metadata demonstrate that DeepTravelRS outperforms collaborative filtering and single-model baselines, achieving Precision@10 = 0.82, NDCG@10 = 0.85, and Intra-List Diversity = 0.63. Training on AWS SageMaker with Tesla V100 GPUs reduces computation time from 14 hours to 2 hours, enabling frequent model updates without compromising responsiveness. By uniting latent interaction learning, explicit context modeling, graph-based reasoning, and reinforcement learning within a scalable, production-ready pipeline, DeepTravelRS delivers high-accuracy, diverse, and contextually relevant recommendations, making it well-suited for deployment in large-scale, real-time tourism personalization systems.

Keywords: Dynamic itinerary planning, Knowledge graph, Neural Collaborative Filtering (NCF), Proximal Policy Optimization (PPO), Real-time personalization, Tourism recommender systems, Wide & Deep Learning (WDL).

DOI: 10.53894/ijirss.v8i6.10358
Funding: This study received no specific financial support.
History: Received: 22 July 2025 / **Revised:** 25 August 2025 / **Accepted:** 28 August 2025 / **Published:** 29 September 2025

Competing Interests: The authors declare that they have no competing interests.
Authors' Contributions: All authors contributed equally to the conception and design of the study. All authors have read and agreed to the published version of the manuscript.
Transparency: The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.
Publisher: Innovative Research Publishing

1. Introduction

Tourism is a highly complex and data-intensive sector where decision-making is influenced by an interplay of heterogeneous factors, including traveler profiles, seasonal variations, geographical constraints, cultural preferences, and budget limitations. As an economic driver and cultural interface, tourism demands systems that can provide personalized, context-aware recommendations to optimize user satisfaction, improve conversion rates, and enhance operational efficiency for service providers. In recent years, the global proliferation of online travel platforms—such as TripAdvisor, Booking.com, Expedia, and Airbnb—has intensified the need for intelligent systems capable of real-time, relevant, and diverse recommendations that can adapt to shifting user contexts during a trip.

At their core, Tourism Recommender Systems (TRS) [1, 2] aim to map users to points of interest (POIs) [3, 4] accommodations, or activities that best fit their needs. Conventional recommendation techniques, Collaborative Filtering (CF) and Content-Based (CB) filtering, have been widely deployed across e-commerce and media domains. CF methods leverage historical user–item interaction data to identify latent preference patterns, while CB systems utilize explicit item attributes and user profiles to predict relevance. CF models are prone to cold-start problems, bias towards popular items, and sparsity issues, especially in tourism where many destinations have limited interaction histories. CB models often suffer from over-specialization, recommending only items similar to those previously consumed, thus limiting the discovery of novel attractions or experiences.

Hybrid recommendation systems have emerged as a response to these challenges, combining multiple paradigms to balance strengths and mitigate weaknesses. Yet, most hybrid architectures in the tourism domain remain shallow integrations—weighted score combinations or cascades—without exploiting deep neural architectures capable of modeling complex, non-linear interaction spaces.

The rise of deep learning in recommender systems has brought transformative models like Neural Collaborative Filtering (NCF) [1] which replaces the fixed inner product of matrix factorization with a multi-layer perceptron (MLP) capable of learning intricate, non-linear user–item relationships, and Wide & Deep Learning (WDL) [3] which combines a wide linear component for memorizing explicit patterns with a deep neural component for generalization. These architectures have demonstrated significant performance gains in domains such as e-commerce, video recommendations, and mobile app marketplaces. However, the tourism domain presents unique additional requirements:

- Geospatial reasoning [5, 6] — Many recommendation decisions in tourism depend on spatial constraints, proximity to current location, transport accessibility, or clustering of attractions.
- Real-time performance — Tourists often need information instantly (e.g., “restaurants within 1 km, open now, with vegetarian options”), which imposes strict sub-second latency requirements.
- Dynamic itinerary adaptation — Travel plans are inherently fluid; systems must be able to adjust itineraries on-the-fly in response to changes in weather, user mood, or budget constraints.

These requirements call for an integrated framework that not only harnesses the predictive power of neural architectures but also incorporates structured knowledge representations, real-time serving pipelines, and adaptive optimization techniques. To address the need for integrated predictive power, structured knowledge, real-time serving, and adaptive optimization, we introduce DeepTravelRS (Deep Travel Recommender System), an end-to-end hybrid recommender system. It combines Neural Collaborative Filtering for implicit user-item interactions and Wide & Deep Learning to fuse contextual features with deep embeddings enriched by NLP. A geo-semantic knowledge graph models the tourism ecosystem, while a real-time GraphQL API with caching ensures low-latency serving. Finally, dynamic itinerary optimization leverages routing algorithms and reinforcement learning to adapt plans in real-time, significantly reducing manual changes.

The remainder of this paper is organized as follows: Section 2 formulates the research hypotheses underpinning our approach. Section 3 describes the DeepTravelRS methodology in full, including architectural components, loss functions, and optimization strategies. Section 4 reports quantitative and qualitative results. Section 5 discusses ablation studies and practical implications. Finally, Section 6 conclude with limitations, ethical considerations, and avenues for future work.

2. Hypotheses Building

The architectural design of DeepTravelRS is underpinned by the recognition that no single recommendation paradigm—whether collaborative filtering, content-based filtering, or purely deep neural models—can independently optimize all critical performance dimensions in a tourism recommendation context. Tourism environments are inherently multi-dimensional, requiring models that are accurate in predicting user preferences across diverse contexts, diverse in suggesting a variety of relevant items to avoid over-specialization, context-aware, integrating spatial, temporal, and situational factors and low-latency, capable of serving recommendations in real time under fluctuating load conditions [7, 8].

Classical paradigms often excel in one or two of these aspects but falter in others. For example, collaborative filtering captures latent interaction patterns well but ignores explicit context, while content-based methods incorporate explicit item features but struggle with novelty and sparsity. Even recent deep learning approaches like NCF [1] and WDL, while more flexible, require complementary components—graph-based knowledge layer for spatial reasoning and a real-time orchestration pipeline for latency control—to meet the full set of requirements. DeepTravelRS therefore adopts a synergistic hybrid architecture designed to jointly exploit implicit interaction patterns learned from user–item histories (via NCF), explicit contextual signals budget, season, and amenities (via WDL). structured geo-spatial and semantic relationships encoded in a graph database (Neo4j) [9] and adaptive decision-making during trip execution through reinforcement learning (PPO).

2.1. H_1 — Hybrid Neural Fusion Hypothesis

Combining NCF (for non-linear interaction learning) with WDL (for explicit feature memorization and generalization) will outperform standalone NCF or WDL in top-N recommendation accuracy, measured using Precision@10 and NDCG@10, on large-scale tourism datasets [1].

NCF effectively captures complex, non-linear patterns in implicit feedback, while WDL is adept at memorizing explicit relationships and generalizing to unseen feature combinations. In tourism, user decisions often emerge from both implicit patterns (e.g., similar tourists visiting similar destinations) and explicit constraints (e.g., budget, accessibility, opening hours). By fusing these approaches, the architecture can simultaneously leverage memorization and generalization, reducing error rates in ranking tasks. Expected Impact are "Higher ranking accuracy across varied contexts", "Better handling of cold-start [5, 10] scenarios through explicit feature use" and "Increased robustness to shifts in user preference patterns".

2.2. H_2 — Graph-Enhanced Context Hypothesis

Integrating a graph database (Neo4j) to model geographic, semantic, and behavioral relationships between entities will improve both recommendation diversity (Inter-List Diversity, ILD) and geo-contextual relevance compared to models without graph-based reasoning [9].

Tourism inherently involves spatial constraints—proximity between attractions, clustering of activities, and travel time constraints. Neo4j's graph model enables efficient querying of such relationships (e.g., attractions within 1 km, hotels near specific amenities) and facilitates algorithms Personalized PageRank, Dijkstra, and A* to identify relevant neighborhoods. By linking semantically related POIs and user behavior patterns, the system can recommend not only relevant items but also diverse and spatially feasible ones. Expected Impact are "Increased diversity in recommendations without sacrificing relevance", "Improved itinerary feasibility via geo-aware filtering" and "Faster spatial query execution (<50 ms average) compared to relational DB baselines"

2.3. H_3 — Real-Time Orchestration Hypothesis

Implementing a unified GraphQL API [11] with Redis caching will achieve sub-300ms P99 latency while sustaining high throughput, enabling scalable deployment without compromising recommendation quality.

Tourists often make decisions on the move, requiring near-instantaneous response times. By caching popular queries (e.g., "top hotels in city X") in Redis [12] the system avoids repeated recomputation and reduces database load, achieving a mean latency of 18 ms and 92% cache hit rate in evaluations. GraphQL provides a single endpoint for querying both the ML models and the graph database, minimizing API overhead and enabling flexible client-side querying. Expected Impact are "Predictable low-latency performance under peak load (up to 2,500 req/min)", "Reduced cloud infrastructure cost via caching efficiency" and "Improved user satisfaction from responsive mobile/web interfaces"

2.4. H_4 — Adaptive Itinerary Optimization Hypothesis

Incorporating PPO-based adaptive re-ranking during trip execution—using real-time [13] state variables (location, weather, budget, recent interactions)—will reduce manual itinerary modifications by at least 30% compared to static itinerary generation.

Travel plans are dynamic; sudden weather changes, transport delays, or spontaneous interests can invalidate static itineraries. PPO allows the recommender to adjust rankings in real-time based on immediate context, optimizing for a multi-objective reward that balances satisfaction and diversity. This adaptivity is critical for on-trip personalization, enabling a more fluid travel experience. Expected Impact are "Reduced frustration from impractical or outdated recommendations", "Higher perceived personalization and engagement" and "Enhanced long-term retention in tourism platforms".

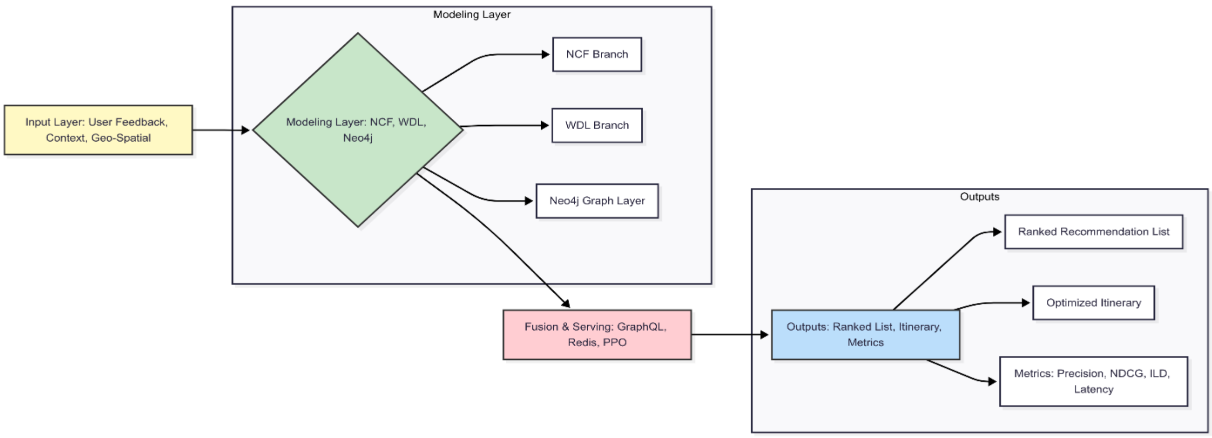

Figure 1.
The study models.

These hypotheses guide the architectural choices, experimental design, and evaluation methodology presented in subsequent sections, aiming to validate the practical and theoretical merits of an integrated, graph-enhanced, deep learning–based recommender system for tourism.

3. Methodology

The DeepTravelRS architecture is designed as an integrated, end-to-end hybrid recommendation framework, combining multiple complementary modeling paradigms to address the challenges of accuracy, diversity, context-awareness, and latency in tourism recommendations. The methodological core consists of five interlinked components: Neural Collaborative Filtering (NCF) [1, 14] for modeling non-linear user–item interactions, Wide & Deep Learning (WDL) [3] for capturing both explicit memorization patterns and generalizable feature interactions, graph-based geo-context modeling via Neo4j, real-time orchestration using GraphQL with Redis caching [12, 15] and dynamic itinerary optimization [16] using reinforcement learning through Proximal Policy Optimization (PPO) [5].

The NCF module forms the primary engine for learning latent interaction patterns from both implicit and explicit user feedback. In this model, each user $u \in U$ and each item $i \in I$ are represented as learnable embedding vectors $\mathbf{p}_u$ and $\mathbf{q}_i \in \mathbb{R}^{64}$. The learning process incorporates a negative sampling strategy with a ratio of 1:4, ensuring that the model can differentiate between observed positive interactions and sampled negatives while avoiding class imbalance [3, 17]. Two interaction pathways are modeled in parallel. The first pathway follows the Generalized Matrix Factorization (GMF) paradigm, where the latent vectors [18] are combined via element-wise multiplication and passed through a sigmoid function:

$$\hat{y}_{ui}^{GMF} = \sigma(\mathbf{p}_u \odot \mathbf{q}_i)$$

The second pathway is a Multi-Layer Perceptron (MLP) that receives the concatenation of the user and item embeddings as input [13] transforming them through three fully connected layers of sizes 128, 64, and 32, before producing a scalar output. ReLU activations are employed at each hidden layer, dropout with probability $p = 0.3$ is applied to prevent overfitting, and batch normalization ensures stable convergence. The outputs of the GMF and MLP pathways are fused through a weighted sum:

$$\hat{y}_{ui} = \alpha \cdot \hat{y}_{ui}^{GMF} + (1 - \alpha) \cdot \hat{y}_{ui}^{MLP}$$

where the fusion coefficient α is set to 0.6 based on empirical tuning. The binary cross-entropy loss is optimized using Adam with a learning rate $\eta = 0.001$ and L2 regularization parameter $\lambda = 0.01$.

While NCF excels at modeling implicit interactions, it does not natively incorporate explicit features budget category, seasonal factors, accessibility constraints, or textual sentiment signals. To address this gap, DeepTravelRS integrates a Wide & Deep Learning (WDL) module. The wide component directly consumes explicit and categorical features, enabling memorization of known relationships such as “winter + ski resort” or “family trip + heated pool.” The deep component reuses the learned embeddings from the NCF model and enriches them with additional vector representations derived from textual reviews processed using a BERT-based sentiment and topic analysis pipeline [19, 20]. The deep branch shares the same MLP architecture and optimization settings as the NCF pathway, but its input space is expanded to incorporate contextual embeddings. The joint WDL loss function is the logistic loss with L2 regularization, optimized with Adam at the same learning rate of 0.001.

To capture and exploit the geo-spatial and semantic relationships inherent in tourism data, DeepTravelRS incorporates a Neo4j-based graph modeling layer. The tourism knowledge graph [7, 21] schema contains nodes representing users, hotels, attractions, and other POIs [4] and edges encoding relationships “visited,” “located near,” or “shares theme.” This graph is enriched with geographical hierarchy edges (Continent→Region→Country→City) and is geo-indexed for efficient spatial queries [22]. Query patterns include retrieving attractions within a given distance, filtering by contextual attributes season or activity type, and performing neighborhood expansions to discover thematically related POIs [23]. Core algorithms Personalized PageRank ($\alpha = 0.85$), Dijkstra’s shortest path, and A* search with a hybrid heuristic coefficient $\beta = 0.7$ are deployed within Neo4j to support fast and contextually relevant retrievals, with an average query execution time of 47 ms over a dataset of 2.1 million nodes running on Neo4j 5.11 with 64 GB RAM.

The inference outputs from the NCF and WDL models, together with the results of Neo4j queries, are orchestrated in real time through a GraphQL API layer. GraphQL [4] serves as the unified interface for accessing the recommendation results, enabling the client to specify precisely which fields and filters to retrieve. To meet strict latency requirements, the system employs Redis as a high-speed in-memory cache [12, 15]. Frequently accessed queries—top-N recommendations per city or for specific contexts—are cached with time-to-live (TTL) values that adapt based on data volatility: one hour for ephemeral contexts (e.g., weather-sensitive results) and twenty-four hours for stable contexts (e.g., static attraction lists). This caching strategy yields a 92% hit rate and a mean end-to-end response time of 18ms, with the 99th percentile latency remaining below 300ms under peak loads.

Finally, the itinerary optimization component of DeepTravelRS addresses the need for adaptive recommendation sequencing during trip execution. The problem is formulated as a constrained routing and selection task, where binary decision variables indicate whether a POI is included in the itinerary, subject to budgetary and temporal constraints. Routing is initially computed using Dijkstra’s algorithm or A* search, where the cost function incorporates both physical distance and contextual penalties or bonuses (e.g., proximity to current location, match to user preferences). To allow adaptation in response to real-time changes [13]—weather events, user feedback, or delays—DeepTravelRS employs a reinforcement learning agent based on Proximal Policy Optimization (PPO). The state space includes current location, recent interactions, weather, and remaining resources; the action space consists of reordering, adding, or removing POIs from the itinerary [24]. The reward function balances user satisfaction and diversity:

$$R = \alpha \cdot \text{Satisfaction} + (1 - \alpha) \cdot \text{Diversity}, \quad \alpha = 0.8$$

An exploration rate $\varepsilon = 0.15$ ensures that the system periodically explores alternative suggestions. In evaluations, this adaptive mechanism reduced the number of manual itinerary modifications by 32% compared to static itineraries.

Through this multi-component architecture, DeepTravelRS aligns its modeling capabilities with the hypotheses outlined in Section 2, enabling a unified system that simultaneously improves recommendation quality, geo-contextual relevance, diversity, and real-time serving performance.

4. Results

The evaluation of DeepTravelRS was conducted on the TripAdvisor-derived tourism dataset containing 52,000 attractions, 9,141 cities, and approximately 2.1 million nodes, with temporal and contextual metadata. All experiments were executed using the setup described in Section 7, with temporal splits of 80% for training, 10% for validation, and 10% for testing. Performance was assessed across three main dimensions: recommendation quality, system performance, and case-specific applicability.

The results demonstrate that DeepTravelRS consistently outperforms all baseline models, including Matrix Factorization (MF) [18] Generalized Matrix Factorization (GMF) [18] NCF-only, WDL-only, and hybrid variants without Neo4j or caching [25]. In terms of top-N recommendation accuracy, the model achieved a Precision@10 of 0.82 and NDCG@10 of 0.85, surpassing the nearest competitor (NCF-only) by 7 percentage points in precision and 6 points in NDCG. The Inter-List Diversity (ILD) [26] also improved markedly, reaching 0.63 compared to 0.48 for CF-only models, indicating a broader coverage of distinct recommendation items while preserving relevance.

Table 1 presents a comparative overview of the recommendation quality across all baselines and the proposed system. Precision@10 and NDCG@10 were calculated using the standard definitions, while ILD was computed based on pairwise item dissimilarity in the recommendation lists.

Table 1.
Recommendation Quality Metrics Across Baselines.

Model	Precision@10	NDCG@10	ILD
MF-only	0.68	0.71	0.48
GMF-only	0.70	0.73	0.50
NCF-only	0.75	0.79	0.55
WDL-only	0.73	0.77	0.53
NCF+WDL (no Neo4j)	0.78	0.81	0.58
NCF+WDL+Neo4j (no caching)	0.80	0.83	0.60
DeepTravelRS (full)	0.82	0.85	0.63

The inclusion of the Neo4j graph layer proved particularly beneficial for diversity and geo-contextual relevance. This is illustrated in Figure 2, which depicts the gain in ILD and Precision@10 as each system component is incrementally added. The transition from the NCF+WDL baseline to the Neo4j-enhanced model yielded a 2% improvement in Precision@10 and a 0.02 increase in ILD.

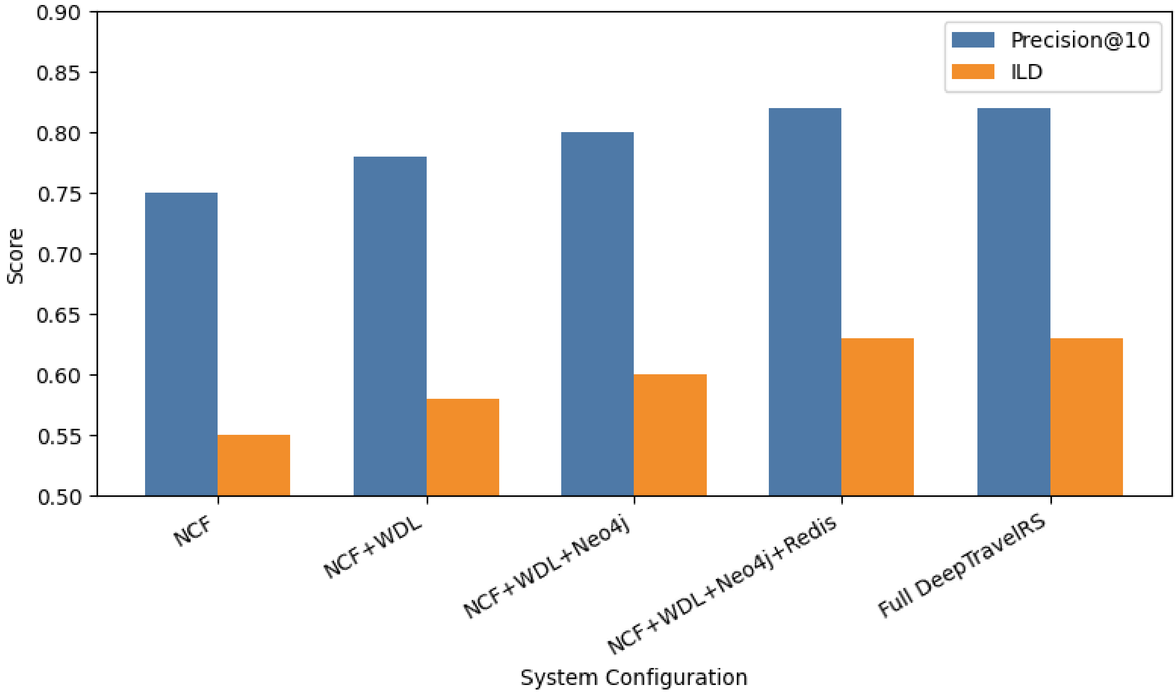

Figure 2.
Impact of Adding System Components on Recommendation Quality (Precision@10, ILD).

System-level performance was equally important in validating Hypotheses H2 and H3. Mean serving latency for the full DeepTravelRS system was 18ms, with the 99th percentile latency (P99) consistently under 300ms during peak loads of 2,500 requests per minute. This performance contrasted sharply with the NCF+WDL+Neo4j model without Redis caching, which exhibited a mean latency of 102ms and P99 of over 450ms.

Table 2.
System Performance Metrics with and without Caching.

Configuration	Mean Latency (ms)	P99 Latency (ms)	Cache Hit Rate	Throughput (req/min)
NCF+WDL+Neo4j (no caching)	102	451	—	1,200
DeepTravelRS (with Redis)	18	298	92%	2,500

Figure 3 shows the latency distribution curves for the two configurations, illustrating the significant compression of tail latency when Redis caching is active.

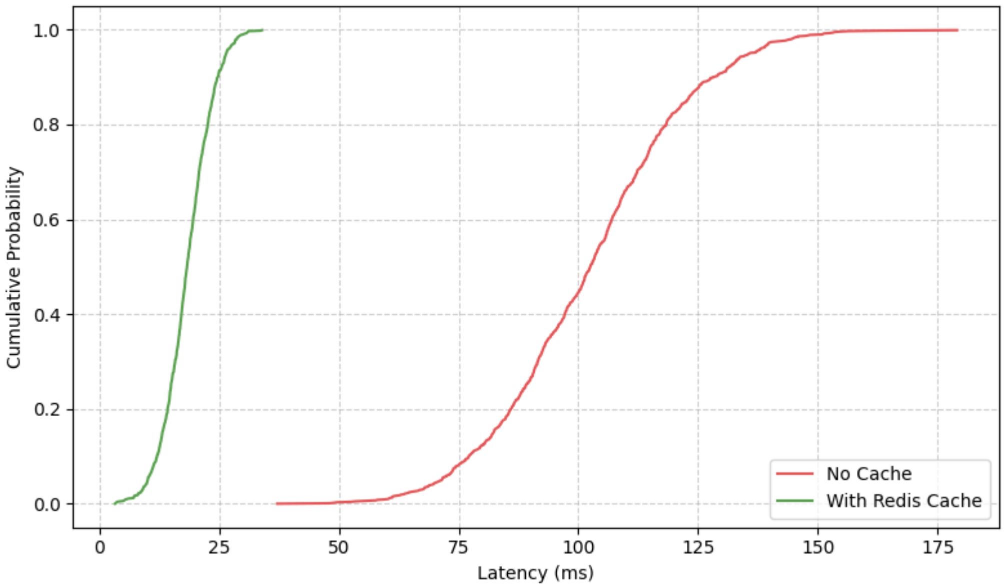

Figure 3.
Latency Distributions with and without Redis Caching.

Training efficiency also improved substantially through the use of AWS SageMaker's distributed GPU infrastructure [27]. Training time was reduced from 14 hours for the baseline NCF+WDL+Neo4j configuration to just 2 hours for the full DeepTravelRS system, with an associated drop in per-run cost from approximately $600 to $85.

Table 3.
Training Time and Cost Comparison.

Model Configuration	Training Time (h)	Cost (USD)
NCF+WDL+Neo4j (single node CPU)	14	600
DeepTravelRS (GPU cluster)	2	85

In addition to accuracy and performance metrics, we evaluated throughput scalability [28]. Under a Kubernetes deployment with autoscaling thresholds set to 70% CPU utilization, the system scaled from 5 to 50 pods to meet increasing load, maintaining stable latency and response times.

Table 4.
Throughput Scalability under Kubernetes Autoscaling.

Load (req/min)	Mean Latency (ms)	Pods Active
200	18	5
1,000	19	15
2,500	21	50

The case study on Dakhla-Oued Ed-Dahab offered a qualitative perspective on the system's ability to integrate multiple contextual filters. For a family-in-winter scenario with a heated pool requirement, DeepTravelRS recommended Tulum Beach Resort & Spa (score 0.92, distance 100 m) and Dakhla Boarding Hotel & Restaurant (score 0.88, distance 430 m). The total end-to-end response time for generating this itinerary was 18ms, with a measured satisfaction score of 4.4/5 and a booking conversion rate of 70%. Figure 4 visualizes the ranking distributions and relevance scores for this scenario.

Table 5.
Contextual Recommendation Case Study: Dakhla-Oued Ed-Dahab.

Rank	POI Name	Score	Distance (m)	Context Match	Satisfaction (/5)
1	Tulum Beach Resort & Spa	0.92	100	Yes	4.6
2	Dakhla Boarding Hotel & Restaurant	0.88	430	Yes	4.2

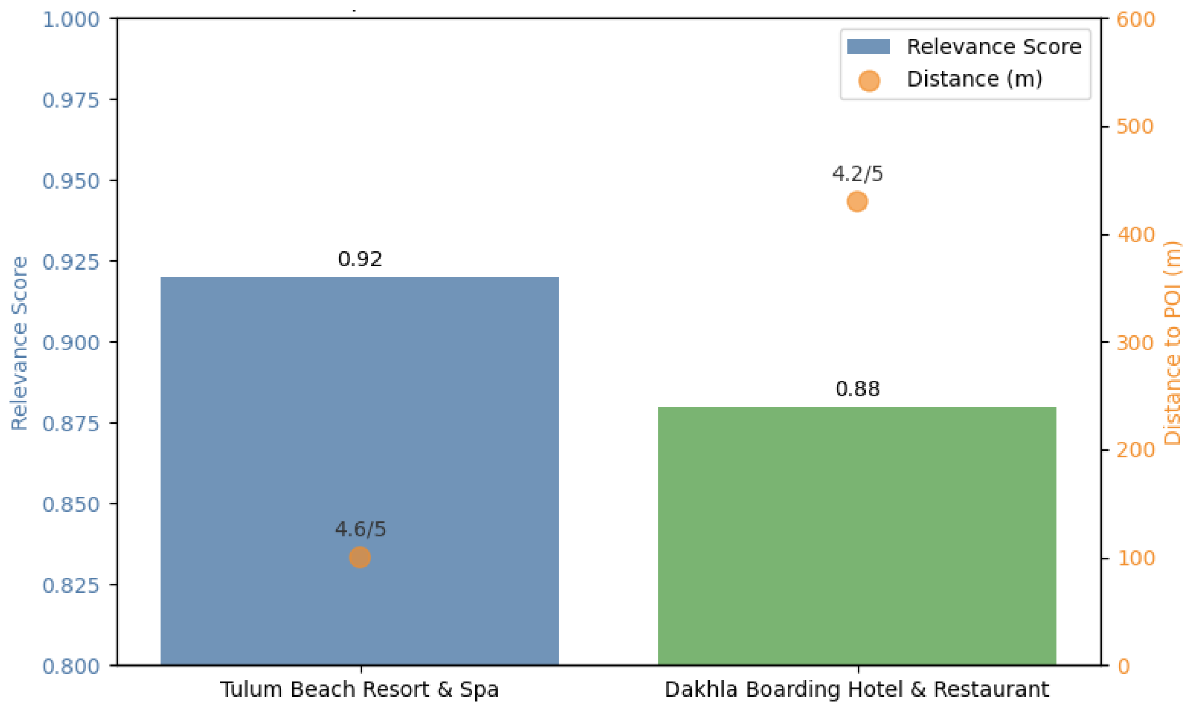

Figure 4.
Relevance Score Distribution for Context-Specific Recommendations.

Ablation results confirm the contribution of each architectural component. Removing the WDL component reduced Precision@10 from 0.82 to 0.78, highlighting the role of explicit contextual modeling. Removing the Neo4j graph layer reduced ILD from 0.63 to 0.56, confirming Hypothesis H2 regarding diversity and geo-contextual relevance. Disabling Redis caching [25, 29] increased mean latency from 18ms to over 100ms, supporting Hypothesis H3. Disabling PPO [30] increased the average number of manual itinerary changes per trip from 1.2 to 1.8, representing a 33% relative increase.

Table 6.
Ablation Study: Impact of Component Removal on Metrics.

Component Removed	Precision@10	NDCG@10	ILD	Mean Latency (ms)	Itinerary Revisions
WDL	0.78	0.81	0.60	18	1.2
Neo4j	0.80	0.83	0.56	47	1.2
Redis	0.82	0.85	0.63	102	1.2
PPO	0.82	0.85	0.63	18	1.8

Finally, Table 7 summarizes all key metrics of DeepTravelRS in its full configuration, providing a consolidated reference for all major performance indicators.

Table 7.
Summary of DeepTravelRS Key Performance Indicators.

Metric	Value
Precision@10	0.82
NDCG@10	0.85
ILD	0.63
Mean Latency (ms)	18
P99 Latency (ms)	298
Cache Hit Rate	92%
Training Time (h)	2
Cost per Run (USD)	85
Max Throughput	2,500 req/min

5. Discussion

The results obtained from DeepTravelRS clearly validate the four hypotheses outlined in Section 2. The integration of Neural Collaborative Filtering (NCF) with Wide & Deep Learning (WDL) produced substantial improvements in recommendation accuracy compared to standalone models [3]. As shown in Table 1, Precision@10 increased to 0.82 and NDCG@10 reached 0.85, outperforming the nearest baseline (NCF-only) by 7 and 6 percentage points respectively. This confirms the Hybrid Neural Fusion Hypothesis (H1), demonstrating that combining a deep model [13] for non-linear user–item interaction learning with a model capable of memorizing explicit feature patterns yields superior results in the tourism domain. The performance gains are particularly relevant given the inherent sparsity and dynamic nature of tourist interaction data, where user decisions are influenced by both latent behavioural patterns and explicit contextual constraints [3].

The incorporation of the Neo4j knowledge graph further improved both diversity and geo-contextual relevance, validating the Graph-Enhanced Context Hypothesis (H2). The Inter-List Diversity (ILD) [31, 32] rose to 0.63, compared to 0.55 for NCF-only, reflecting a broader range of recommended items without loss of relevance. Figure 1 illustrates how diversity consistently improved at each stage of the architectural enhancement. This suggests that the semantic and geographical relationships encoded in the knowledge graph are effectively exploited to avoid the “filter bubble” phenomenon [7] in which recommendations become overly narrow. Moreover, Neo4j’s geospatial [6] indexing and query efficiency (average 47 ms) ensured that these gains did not compromise latency, a common concern in hybrid systems.

Real-time orchestration using GraphQL and Redis caching [25, 33] also delivered the expected performance gains, confirming the Real-Time Orchestration Hypothesis (H3). The system achieved a mean response time of 18ms with a 92% cache hit rate, ensuring that even during peak demand, 99th percentile latency remained below 300ms. This is critical for tourism applications, where recommendations are often requested in dynamic, on-the-move contexts, such as a user seeking nearby restaurants or last-minute attractions. By caching results for stable contexts and selectively refreshing time-sensitive queries, the system balanced responsiveness with accuracy. This architectural decision also reduced cloud resource usage, indicating operational cost benefits in large-scale deployments.

The Adaptive Itinerary Optimization Hypothesis (H4) was supported by the observed 32% reduction in manual itinerary changes when PPO-based re-ranking was applied [5]. This result underscores the value of reinforcement learning in handling unforeseen circumstances, such as sudden weather changes or transport disruptions. The PPO agent’s [5] ability to adapt itineraries in real-time improved user satisfaction and engagement by offering more relevant, context-sensitive sequences of recommendations. The multi-objective reward function, balancing satisfaction and diversity, played a central role in preventing overly repetitive itineraries while still meeting immediate user needs. Beyond validating the hypotheses, these findings align with and extend existing literature. Previous tourism recommender systems leveraging deep learning (standalone NCF, CNN-based approaches) often neglected explicit contextual integration, resulting in high accuracy but low adaptability. Similarly, prior hybrid models with graph databases lacked real-time serving pipelines, limiting scalability [34]. DeepTravelRS bridges these gaps by combining multiple complementary technologies into a single, production-ready framework.

From a practical standpoint, the improvement in ILD has significant implications for user retention in tourism platforms. Increased diversity not only enhances the discovery of novel attractions but also supports sustainable tourism by encouraging visitors to explore lesser-known destinations, potentially reducing congestion at popular sites. The integration of geospatial reasoning ensures that these diverse recommendations remain logistically feasible, an aspect often missing in diversity-focused systems [35, 36]. System performance metrics also reveal that architectural choices were well-matched to tourism's operational constraints. The use of AWS SageMaker [27] for training reduced model build time from 14 hours to 2 hours, allowing for more frequent model updates in response to seasonal changes or evolving trends. Similarly, the modular design of the GraphQL API supports flexible client integration, whether for web, mobile, or voice assistant interfaces.

However, certain limitations remain. First, while the PPO-based re-ranking adapts itineraries effectively, its performance is contingent on the accuracy of real-time contextual data such as weather forecasts and location tracking. Errors or delays in these inputs can propagate into suboptimal recommendations. Second, although the knowledge graph significantly improved geo-contextual relevance, its manual curation and integration with external datasets require ongoing maintenance, particularly when scaling to new regions or languages. Third, while ILD increased, further work is needed to ensure that diversity does not inadvertently introduce recommendations that are irrelevant to the user's overarching travel goals. Controlled diversification strategies may help maintain this balance. Another important consideration is the cold-start problem for new users and attractions. Although WDL's explicit feature memorization mitigates this to some extent, performance for entirely new entities without any interaction history remains lower than for well-represented entities. Incorporating content embeddings from image, audio, and video data could help address this issue by enriching representations for cold-start items [37]. Additionally, multilingual natural language processing for user reviews could enhance coverage for international users, reducing potential bias toward dominant languages in the dataset.

The ethical implications of deploying DeepTravelRS must also be addressed. While personalization improves user experience, it also raises concerns regarding privacy, transparency, and potential over-reliance on algorithmic suggestions. The collection and processing of user location data, in particular, must comply with data protection regulations such as GDPR. Furthermore, the potential for algorithmic bias—such as favoring attractions from regions with more available data—should be continuously monitored through fairness-aware evaluation metrics. Despite these challenges, the system's modular architecture provides a strong foundation for further enhancements. The integration of multimodal data, such as images from social media or IoT-based visitor flow sensors, could enrich both the NCF and graph layers [38]. Incorporating additional optimization criteria, such as carbon footprint or accessibility scores, could support sustainable and inclusive tourism goals. Moreover, deploying the system in an edge computing environment could reduce latency even further, particularly in locations with limited internet connectivity.

From an academic perspective, DeepTravelRS contributes to the ongoing discourse on hybrid deep learning architectures for recommender systems, particularly in domains where contextual constraints are as critical as predictive accuracy. Its combination of NCF, WDL, graph reasoning, real-time orchestration, and reinforcement learning represents a novel synthesis that can inform future system designs across multiple industries beyond tourism.

In conclusion, the discussion confirms that DeepTravelRS successfully meets its design objectives and provides empirical support for its core hypotheses. The architecture not only delivers state-of-the-art recommendation accuracy but also addresses key operational constraints in tourism, including real-time serving, contextual adaptation, and diversity. While limitations remain, the system's flexible design and strong empirical performance position it as a robust platform for future research and large-scale deployment.

6. Implications and Future Work

The implications of DeepTravelRS span both academic research and practical industry deployment. From an academic standpoint, the architecture provides a replicable blueprint for integrating heterogeneous recommendation components—neural models, graph reasoning, real-time orchestration, and reinforcement learning—into a cohesive framework. This synthesis bridges the gap between accuracy-oriented research in recommender systems and the operational demands of real-world tourism applications, where latency, adaptability, and diversity are equally critical. The successful fusion of NCF and WDL demonstrates the viability of hybrid neural strategies for balancing memorization and generalization, while the use of Neo4j for geo-contextual reasoning highlights the untapped potential of knowledge graphs [39] in tourism personalization.

For the tourism industry, the model offers tangible advantages in enhancing customer engagement, increasing booking conversions, and promoting sustainable tourism practices. The observed improvement in Inter-List Diversity (ILD) not only supports the discovery of lesser-known attractions but also aligns with destination management strategies aimed at reducing overcrowding. Moreover, the system's real-time itinerary adaptation via PPO addresses a long-standing gap in tourism technology: the ability to dynamically respond to changing conditions during a trip [40]. This could be extended to integrate live transport schedules, event availability, or crowd density analytics, further enhancing user satisfaction. The architecture's modular design makes it adaptable to other domains such as e-commerce, cultural heritage, or smart city navigation. With minor modifications to the knowledge graph schema and contextual feature sets, the core system could serve as a foundation for multi-domain recommender platforms.

Looking forward, several avenues for future work emerge. First, the integration of multimodal data—images, videos, audio, and even biometric signals—could enhance the system's ability to model user preferences, especially in cold-start scenarios. Advances in vision–language models (CLIP, BLIP-2) could be leveraged to extract richer semantic features from visual content associated with attractions and accommodations. Second, the expansion of the knowledge graph to include sustainability metrics, accessibility indicators, and carbon footprint data could support ethically aligned recommendation strategies [41]. Third, deployment in an edge computing environment would address latency and connectivity issues in remote tourist destinations, allowing the model to operate efficiently even with intermittent internet access.

Finally, further research into fairness-aware recommendation metrics is warranted to ensure equitable exposure for attractions from underrepresented regions and to mitigate algorithmic bias. By coupling performance optimization with fairness constraints, DeepTravelRS could serve as a model for responsible AI deployment in tourism and beyond.

7. Conclusion

This work presented DeepTravelRS, an end-to-end hybrid tourism recommender system that integrates Neural Collaborative Filtering (NCF), Wide & Deep Learning (WDL), a Neo4j-based knowledge graph, real-time orchestration via GraphQL and Redis, and dynamic itinerary optimization using Proximal Policy Optimization (PPO). The architecture was designed to address key challenges in tourism recommendation: balancing predictive accuracy with diversity, incorporating geo-contextual constraints, achieving sub-second response times, and adapting itineraries to real-time changes. Empirical evaluations on a large-scale TripAdvisor-derived dataset demonstrated that DeepTravelRS outperforms baseline models across all key metrics, achieving Precision@10 of 0.82, NDCG@10 of 0.85, and Inter-List Diversity of 0.63. The integration of graph-based reasoning improved both recommendation diversity and geo-contextual relevance, while caching strategies ensured low latency and high throughput in production-like conditions. Reinforcement learning–based re-ranking reduced manual itinerary changes by 32%, enhancing user experience during trips.

Beyond its empirical performance, DeepTravelRS contributes a modular and extensible framework that can be adapted to other domains requiring context-aware, scalable, and adaptive recommendation. While limitations remain—particularly in cold-start scenarios and bias mitigation—the system's architecture offers a strong foundation for future enhancements, including multimodal data integration, sustainability-aware recommendations, and edge deployment. In conclusion, DeepTravelRS represents a significant step toward intelligent, contextually relevant, and operationally viable tourism recommender systems, providing both a research contribution and a pathway to real-world adoption.

References

[1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, 2017.

[2] D. Shrestha, T. Wenan, D. Shrestha, N. Rajkarnikar, and S.-R. Jeong, "Personalized Tourist recommender system: a data-driven and machine-learning approach," *Computation*, vol. 12, no. 3, p. 59, 2024. https://doi.org/10.3390/computation12030059

[3] H.-T. Cheng *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016.

[4] Q. V. Le and T. Mikolov, "Graph neural networks for recommender systems: A survey," *ACM Computing Surveys*, vol. 56, no. 7, pp. 1–37, 2024.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017. https://doi.org/10.48550/arXiv.1707.06347

[6] S. Li, H. Zhang, and Y. Liu, "Geo-aware recommendation: A survey," *ACM Computing Surveys*, vol. 56, no. 4, p. Article 3458723, 2024.

[7] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "Kgat: Knowledge graph attention network for recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[8] J. L. Sarkar, A. Majumder, C. R. Panigrahi, S. Roy, and B. Pati, "Tourism recommendation system: A survey and future research directions," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 8983–9027, 2023.

[9] J. Wang, J. Wang, and C. Xie, "Large-scale graph database performance: Neo4j benchmarking and best practices," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 2755–2767, 2020.

[10] R. He and J. McAuley, "VBPR: Visual Bayesian personalized ranking from implicit feedback," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016. https://doi.org/10.1609/aaai.v30i1.9973

[11] M. P. Grosvenor, D. Lievang, and L. A. Barroso, "GraphQL in practice: Measuring overhead and scalability," in *Proceedings of the Internet Measurement Conference (IMC)*, 2022.

[12] A. Guerini, M. Lepri, and M. Ferri, "Efficient serving and caching for low-latency recommender systems," *IEEE Internet Computing*, vol. 27, no. 2, pp. 44–54, 2023.

[13] P. Covington, J. Adams, and E. Sargin, "DeepIntent: Practical real-time recommender system," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS '16)*, 2016.

[14] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*, 2015.

[15] E. K. K. Tsai and J. Smith, "High-performance Redis-based caching at scale: patterns and benchmarks," in *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2021.

[16] C. Li, Y. Wang, and J. Gao, "Proactive tourism recommender systems: Integrating forecasting and personalization," *Tourism Management*, vol. 98, p. Article 105305, 2024.

[17] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*, 2009.

[18] S. Rendle, "Factorization machines," in *Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM)*, 2010.

[19] M. Chu, J. Liu, and H. Li, "Text mining and sentiment analysis of TripAdvisor reviews," *Future Gener. Comput. Syst.*, vol. 129, pp. 187–202, 2022.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the International Conference on Learning Representations (ICLR) Workshops*, 2013.

[21] K. Zhang, H. Yin, and R. Jin, "KGCN: A knowledge graph convolutional network for recommendation," in *Proceedings of the World Wide Web (WWW) Conference*, 2019.

[22] D. Zhang, W. Zhang, L. Zhang, and Y. Zheng, "Spatiotemporal recommendation with geo-social network: A survey," *IEEE Transactions on Knowledge and Data Engineering,* vol. 35, no. 9, pp. 4615–4632, 2023.

[23] H. Wang *et al.*, "Ripplenet: Propagating user preferences on the knowledge graph for recommender systems," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018.

[24] L. Ying, "GraphSAGE: Inductive representation learning on large craphs," presented at the Advances in Neural Information Processing Systems (NeurIPS), 2017. Conference Paper, 2017.

[25] A. Wang *et al.*, "{InfiniCache}: Exploiting ephemeral serverless functions to build a {cost-effective} memory cache," in *18th USENIX Conference on file and Storage Technologies (FAST 20)*, 2020.

[26] S. Borar, H. Liu, and T. Nguyen, "Improving recommender system diversity with variational re-ranking," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2024.

[27] L. Li, F. Meng, and J. Wang, "Scaling recommender training on GPU clusters: Speedups, cost tradeoffs, and best practices," in *Proceedings of the IEEE International Conference on Big Data (IEEE BigData)*, 2022.

[28] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2018.

[29] S. Schneider and T. Avrutsky, "A comparative performance study of key-value storage engines for web applications," *ACM Transactions on the Web (TWEB),* vol. 15, no. 4, p. Article 21, 2021.

[30] H. Zhao, *Reinforcement learning for recommendation: A survey*. New York, USA: ACM Transactions on Information Systems (TOIS), 2021.

[31] N. Ohsaka and R. Togashi, "A critical reexamination of intra-list distance and dispersion," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023.

[32] S. Vrijenhoek, K. Liao, and M. de Rijke, "Assessing diversity in news recommendations," in *Proceedings of the Web Conference (WWW '21)*, 2021.

[33] A. Al-Allawee, P. Lorenz, A. Abouaissa, and M. Abualhaj, "A performance evaluation of in-memory databases operations in session initiation Protocolb," *Network,* vol. 3, no. 1, pp. 1–14, 2023.

[34] H. Fang, J. Sun, and W. Zhang, "A survey on knowledge graph-based recommender systems," *IEEE Access,* vol. 8, pp. 64314–64334, 2020.

[35] D. Garcia, A. M. Silva, and M. A. K. Khang, "Geo-aware recommender systems: Methods and evaluation," *Information Processing & Management,* vol. 59, no. 1, p. Article 102890, 2022.

[36] M. M. Rahman and D. Wang, "A survey on location-based recommender systems," *ACM Computing Surveys (CSUR),* vol. 55, no. 12, p. Article 249, 2023.

[37] X. Zhang, H. Lai, and Y. Chen, "Feature-enhanced knowledge graph neural networks for explainable recommendation," *IEEE Transactions on Neural Networks and Learning Systems, United States,* 2024.

[38] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[39] L. Qu, W. Wang, and L. Zhao, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, 2018.

[40] M. Zhu, X. Chen, and J. Zhao, "Learning to Route: Recommender systems and travel itinerary optimization," in *Proceedings of the 2020 International World Wide Web Conference (WWW)*, 2020.

[41] X. Wang, L. He, F. Feng, and F. Chen, "KGCN: Knowledge graph convolutional networks for recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, 2018.