RESEARCH PAPER

Cloud Service Recommendation System Using Knowledge Graph and PageRank Algorithm

Mr. Vaibhav Arun Walke[1], Mrs. S. S. Shinde[2]
[1]PG Student, Department of Computer Engineering, MCOERC, Nashik
Email: vaibhavwalke876@gmail.com
[2]Assistant Professor, Department of Computer Engineering, MCOERC, Nashik
Email: shinde.shraddha@matoshri.edu.in

ABSTRACT
The rapid growth of cloud computing technologies has resulted in a large number of cloud services and platforms across domains such as storage, networking, analytics, machine learning, and virtualization. Selecting suitable cloud services from a vast pool of available options has become a significant challenge for users and organizations. Traditional recommendation systems often fail to capture semantic relationships among cloud services and suffer from data sparsity, cold-start issues, and poor explainability. This paper proposes a lightweight and explainable Cloud Service Recommendation System using Knowledge Graph and PageRank Algorithm. The proposed system models cloud services and their semantic relationships using a Knowledge Graph and ranks related services using the PageRank algorithm. A real-world cloud services dataset containing 1,000 records, 467 unique products, 6 service categories, and 30 unique functionalities is used for evaluation. Unlike computationally intensive deep learning-based recommendation systems, the proposed approach provides efficient semantic ranking with lower computational complexity while maintaining recommendation quality and explainability. The system is implemented as an interactive web application using Streamlit and achieves a top PageRank score of 0.0038 for the highest-ranked cloud service. Experimental results demonstrate that the proposed system provides effective semantic recommendation with significantly lower computational overhead compared to Graph Neural Network-based approaches.

Index Terms: Cloud Service Recommendation, Knowledge Graph, PageRank Algorithm, Graph-Based Recommendation, Semantic Modeling, Streamlit Visualization, Lightweight Recommendation System, NetworkX.
How to cite this article: Walke VA, Shinde SS. Cloud Service Recommendation System Using Knowledge Graph and PageRank Algorithm. Int J Drug Deliv Technol. 2026;16(52s): 1199-1215. DOI: 10.25258/ijddt.16.52s.155
Source of support: Nil.
Conflict of interest: None.

I. INTRODUCTION

With the rapid growth of cloud computing technologies, a large number of cloud services and platforms are available to users across different domains such as storage, networking, analytics, machine learning, and virtualization. Although cloud computing provides scalability, flexibility, and cost efficiency, selecting suitable cloud services from a vast number of avail- able options has become a major challenge for users and organizations. Traditional recommendation systems often fail to capture semantic relationships among cloud services and suffer from problems such as data sparsity, cold-start issues, and poor explainability [1]–[3].

Recommendation systems have become an essential com- ponent in modern intelligent applications for providing per- sonalized suggestions based on user preferences and item relationships. Conventional recommendation techniques such as collaborative filtering and content-based filtering have been widely adopted in many applications. However, these methods primarily rely on historical interaction data and often fail to model complex semantic relationships among entities [4], [5]. To overcome these limitations, Knowledge Graphs (KGs) have emerged as an effective solution for representing structured semantic relationships among entities and improving recom- mendation quality [6], [7].

Knowledge Graphs represent information using entities and relationships in the form of triples consisting of head entity, relation, and tail entity. Due to their strong semantic repre- sentation capability, Knowledge Graphs have been extensively used in recommendation systems, natural language processing, question answering, and intelligent search systems [4], [8]. Recently, Knowledge Graph-based recommendation systems have gained significant attention because they improve rec- ommendation accuracy,

enhance explainability, and alleviate sparsity problems by utilizing multi-hop semantic relationships among entities [9], [10].

Several advanced recommendation approaches such as Graph Neural Networks (GNN), Graph Attention Networks (GAT), and Neural Collaborative Filtering (NCF) have been integrated with Knowledge Graphs to improve recommendation performance [1], [5], [6], [11]. Methods such as KGAT and KGCN utilize graph convolution and attention mechanisms to propagate semantic information through graph structures and learn high-quality embeddings for recommendation tasks [9], [10]. Similarly, RippleNet propagates user preferences over knowledge graphs to model user interests more effectively [11]. Although these approaches achieve improved recom- mendation accuracy, they often introduce high computational complexity, expensive model training, and heavy dependency on large datasets and GPU resources [5], [6].

To address these challenges, this paper proposes a lightweight and explainable Cloud Service Recommendation System using Knowledge Graph and PageRank Algorithm. The system is evaluated on a real cloud services dataset (cloud_functionality_1000.csv) containing 1,000 records across 6 service categories. The major contributions of this work are:

- Development of a Knowledge Graph-based cloud service recommendation framework using a real 1,000-record dataset.
- Semantic modeling of cloud service relationships across 6 categories and 30 functionalities using graph structures.
- Integration of the PageRank algorithm for efficient recommendation ranking without GPU or deep learning overhead.
- Lightweight, role-based web application using Streamlit with user and admin dashboards.
- Interactive visualization of Knowledge Graph and PageRank scores for explainable recommendations.

The remainder of this paper is organized as follows. Section II presents the literature survey. Section III discusses the research gap and problem statement. Section IV describes the proposed methodology. Section V presents the system architecture. Section VI describes the algorithms used. Section VII presents implementation and experimental results. Section VIII concludes the paper.

II. LITERATURE SURVEY

Recommendation systems have become an important re- search area for providing personalized services in domains such as e-commerce, online education, multimedia systems, and cloud computing. Traditional recommendation approaches such as collaborative filtering and content-based recommen- dation techniques mainly depend on user-item interaction history and similarity measures. However, these methods often suffer from data sparsity, cold-start problems, poor semantic understanding, and lack of explainability [1], [3].

To overcome these issues, researchers introduced Knowl- edge Graphs (KGs) into recommendation systems. Knowl- edge Graphs provide structured semantic relationships among entities using graph-based representations, enabling recom-mendation systems to capture multi-hop semantic associations between users and items [4]. Yang et al. proposed a Knowledge Graph recommendation method for online education systems that utilizes semantic relationships among learning resources and learners to improve personalized recommendation quality [4].

Shokrzadeh et al. developed a Knowledge Graph-based rec- ommendation framework enhanced with Neural Collaborative Filtering and Knowledge Graph Embedding techniques [1]. Their proposed model utilized user-user, user-tag, and tag- source relationships to create semantic graph structures and improve recommendation accuracy. The study also employed embedding techniques such as TransE, ConvE, and ComplEx for low-dimensional vector representation learning.

RecKG introduced by Kwon et al. focused on interoper- ability and semantic integration among heterogeneous recom- mendation datasets [2].

The study proposed a standardized Knowledge Graph framework for recommendation systems to improve consistency, diversity, and semantic representation across multiple recommendation domains.

Hu and Xia proposed the Multi-Stream Graph Attention Network (MSGAT) for recommendation systems using Knowl- edge Graphs [6]. Their approach utilized multi-stream at- tention mechanisms to capture semantic relationships from multiple perspectives. However, the system suffered from increased computational cost and model complexity.

Rong et al. introduced a multi-level contrastive learning framework for Knowledge Graph recommendation systems [5]. Their proposed model integrated Graph Attention Net- works with Momentum Contrast learning to address sparsity and long-tail distribution problems.

Wang et al. proposed KGCN [9] and KGAT [10] for recommendation systems by aggregating neighborhood information from graph structures. RippleNet proposed by Wang et al. propagated user prefer- ences over Knowledge Graph structures to improve recom- mendation accuracy [11].

From the existing literature, Knowledge Graph-based rec- ommendation systems significantly improve recommendation quality and explainability. Nevertheless, most existing ap- proaches depend heavily on GNNs, deep learning architec- tures, and complex embedding mechanisms, resulting in in- creased computational overhead and difficult deployment in lightweight environments. A detailed comparative analysis of existing studies is presented in Table I.

TABLE I: Comparative Analysis of Existing Knowledge Graph-Based Recommendation Studies

Author / Study	Technique Used	Dataset / Domain	Key Contribution	Limitation / Gap
Shokrzadeh et al. [1]	KG, NCF, KG Embedding	Social tagging	KG embeddings and NCF improve recommendation accuracy	High neural training complexity
Kwon et al. [2]	Standardized KG	Heterogeneous datasets	Improves KG interoperability and explainability	Focuses on standardization, not lightweight ranking
Jin and Yang [3]	Survey of KG-based CTR models	CTR / Recommendation	Classifies KG-based methods into three categories	Survey only; no lightweight model proposed
Yang et al. [4]	KG for online education	Online education	Reviews learning path recommendation using KG	Education-specific; highlights dynamic recommendation need
Rong et al. [5]	Multi-level contrastive learning, GAT	MovieLens, Amazon-Books	Addresses sparsity using contrastive learning	Complex GNN and contrastive training overhead

Hu and Xia [6]	Multi-stream GAT	Movies, music, books	Filters noisy entities using multi-stream attention	Computationally expensive GNN propagation
Wang et al. [7]	Session-adaptive KG propagation	Session-based / e-commerce	Session-aware item relation propagation	Not designed for cloud service recommendation
Bao and Wang [8]	Multimodal KG, Node2vec, LSTM, GAT	Personalized education	Integrates text, image, and video knowledge	Multimodal design increases system complexity
Proposed System	Knowledge Graph + PageRank	Cloud services (1,000 records)	Lightweight, explainable, graph-based ranking	Depends on cloud service dataset quality

TABLE II: Summary of Reported Experimental Results from Existing Studies

Study	Metric	Reported Value	Remarks
Bao and Wang [8]	Hits@10	62.7%	Entity alignment accuracy in multimodal KG recommendation
Bao and Wang [8]	Cross-modal similarity	0.76	Improvement over Node2vec and M3KGR
Bao and Wang [8]	Completion rate	78%	Learning engagement metric
Bao and Wang [8]	Explainability score	4.3 / 5.0	User-rated explainability score
Bao and Wang [8]	Response time	98 ms	System response time
Rong et al. [5]	Recall@20, NDCG@20	See paper	Evaluated on MovieLens and Amazon-Books datasets
Hu and Xia [6]	Precision, Recall, F1	See paper	Evaluated on movies, music, and books datasets
Proposed System	**Top PageRank score**	**0.0038**	**Compute Product 68, computed on 1,000-record dataset**

III. RESEARCH GAP AND PROBLEM STATEMENT

III.A. Research Gap

Existing recommendation systems based on collaborative filtering, content-based filtering, and deep learning approaches have significantly improved personalized recommendation per- formance. However, traditional recommendation methods pri- marily depend on user-item interaction history and often fail to capture semantic relationships among entities, resulting in poor recommendation quality under sparse data conditions [1], [3].

Most existing Knowledge Graph recommendation systems rely heavily on GNNs, GATs, and deep embedding learning techniques that require extensive training data, high computa- tional power, and GPU-intensive processing [5], [6]. Several studies also report issues related to data sparsity, cold-start problems, long-tail distribution, noisy entity propagation, and insufficient explainability [1], [3], [5]. Very limited research has focused on lightweight cloud service recommendation systems using semantic graph ranking techniques.

Therefore, there exists a need for a lightweight, scalable, explainable, and computationally efficient recommendation framework that can effectively model semantic relationships among cloud services without relying on computationally expensive deep learning architectures. The proposed work addresses this gap by integrating Knowledge Graph semantic modeling with the PageRank algorithm to provide efficient and explainable cloud service recommendation with lower computational complexity and easier deployment.

III.B. Problem Statement

The rapid growth of cloud computing technologies has resulted in the availability of a large number of cloud ser- vices and platforms across different domains such as Infras- tructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The dataset used in this work, cloud_functionality_1000.csv, contains 1,000 cloud service records covering 6 categories: Compute, Storage, Security, Network, PaaS, and O&M, with 30 unique functionalities including Load Balancer, Elastic Compute, Cloud Firewall, Object Storage, AI Inference Engine, and Container Service.

Traditional cloud service recommendation systems mainly rely on collaborative filtering, content-based filtering, or keyword-based search methods. These approaches often fail to capture semantic relationships among cloud services and functionalities, resulting in poor recommendation accuracy and limited explainability. Furthermore, existing recommendation systems suffer from data sparsity, cold-start issues, noisy rec- commendations, and inability to model complex relationships among cloud service entities [1], [3].

The proposed system addresses this problem by developing a Knowledge Graph-based Cloud Service Recommendation System integrated with the PageRank algorithm. The sys- tem models cloud services and their semantic relationships as graph structures and ranks relevant cloud services based on graph connectivity and semantic importance, improving recommendation explainability and reducing computation complexity.

IV. PROPOSED METHODOLOGY

The proposed system presents a lightweight and explainable Cloud Service Recommendation Framework using Knowledge Graph and PageRank Algorithm. The system models semantic relationships among cloud services using graph structures and generates recommendations based on graph connectivity and ranking scores.

The overall workflow of the proposed methodology consists of the following five stages:

1) Data Collection and Preprocessing

2) Knowledge Graph Construction

3) PageRank-Based Recommendation

4) Role-Based Web Application Development

5) Recommendation Visualization

IV.A. Data Collection and Preprocessing

The dataset used in this work is cloud_functionality_1000.csv, which contains 1,000 cloud service records with the following fields: ID, Functionality_Name, Category, Product_Name, Release_Year, Update_Year, and Description. The dataset covers 6 service categories (O&M, Storage, Compute, PaaS, Security, Network) and 30 unique functionalities. The preprocessing stage removes duplicate records, handles missing values, and standardizes service attributes for graph generation.

IV.B. Knowledge Graph Construction

A Knowledge Graph is constructed to represent semantic relationships among cloud services. Nodes represent cloud service products, functionality names, and service categories. Edges represent semantic relationships including *belongs_to* (product to category), *provides* (product to functionality), and *similar_to* (products sharing the same functionality).
The Knowledge Graph follows the triple representation model:

(Head Entity, Relation, Tail Entity)….(1)

For example:

(Compute Product 68, *belongs_to*, Compute)….(2)

(Compute Product 68, *provides*, Load Balancer)….(3)

IV.C. PageRank-Based Recommendation

The recommendation engine uses the PageRank algorithm to rank cloud services based on graph connectivity and seman- tic importance. The iterative PageRank update rule is:

$$PR^{(t+1)}(v) = \frac{1-d}{|V|} + d\sum_{u\in N^{-}(v)} \frac{PR^{(t)}(u)}{\deg^{+}(u)} \dots (4)$$

Where d = 0.85 is the damping factor, |V | = 503 is the total node count, $N^{-}(v)$ is the in-neighbor set of v, and $\deg^{+}(u)$ is the out-degree of neighbor u. The algorithm initializes all scores a $PR^{(0)}(v) = \frac{1}{|V|}$ and iterates until $\sum_v \left|PR^{(t+1)}(v) - PR^{(t)}(v)\right| < 10^{-6}$ or 200 iterations. Services with higher PageRank values are ranked higher in recommendation outputs. Full mathematical modeling and formal algorithm specifications are provided in Section VI.

IV.D. Advantages of Proposed Methodology

The proposed methodology provides the following key advantages:

- Lightweight framework — no GPU or deep learning training required.
- Explainable recommendations with relevance score, PageRank score, and final combined score shown per result.
- Semantic relationship modeling using real cloud service data across 6 categories and 30 functionalities.
- Category and layer-based filtering for refined recommendations.
- Role-based access with separate user and admin dashboards.

V. SYSTEM ARCHITECTURE

The architecture of the proposed Cloud Service Recommendation System integrates data preprocessing, Knowledge Graph construction, recommendation generation, PageRank ranking, and Streamlit visualization. The overall architecture is illustrated in Fig. 1.

V.A. Architecture Components

The proposed architecture consists of six major components:

1) **Cloud Service Dataset Module** — loads cloud_functionality_1000.csv with 1,000

records, 467 products, 6 categories, and 30 functionalities.

2) **Data Preprocessing Module** — handles missing values, removes duplicates, normalizes fields, and
extracts entity-relation structures.

3) **Knowledge Graph Construction Module** — builds graph nodes (products, categories, functionalities) and edges (belongs_to, provides, similar_to) using NetworkX.

4) **PageRank Recommendation Engine** — computes semantic importance scores using Equation (11) with
d = 0.85 and up to 200 iterations.

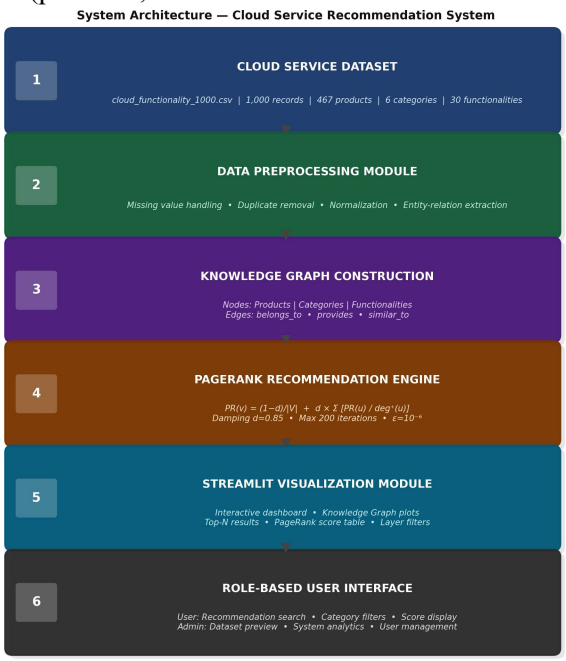

Fig. 1: Architecture of the Proposed Cloud Service Recommendation System Showing the Six-Module Pipeline from Dataset to User Interface

5) **Streamlit Visualization Module** — provides interactive graph plots, PageRank score tables, and recommendation result cards.

6) **Role-Based User Interface** — supports two roles: User (recommendation search) and Admin (full system analytics).

V.B. System Workflow

1) Cloud service data is loaded from the CSV dataset.

2) Preprocessing extracts entities and semantic relationships.

3) Knowledge Graph is built using NetworkX DiGraph.

4) PageRank algorithm computes node importance scores.

5) Top-ranked services are returned as recommendations.

6) Results are displayed via the Streamlit web interface.

VI. MATHEMATICAL MODELING AND ALGORITHMS

This section presents the complete mathematical formulation of the proposed Cloud Service Recommendation System and the formal algorithms used for Knowledge Graph construction, PageRank computation, and combined recommendation scoring.

VI.A. Knowledge Graph Formal Definition

The proposed system represents cloud service relationships as a directed Knowledge Graph G = (V, E, R, ϕ) where:

- $V = V_P \cup V_C \cup V_F$ is the set of all nodes, with V_P = product nodes, V_C = category nodes, V_F = functionality node.
- $E \subseteq V \times R \times V$ is the set of labeled directed edges (triples).
- $R = \{r_1, r_2, r_3\}$ is the relation type set where r_1 = *belongs_to*, r_2 = *provides*, r_3 = *similar_to*.
- $\phi : V \rightarrow$ {product, category, functionality} is the node type mapping function.
- Each record (id_k, f_k, c_k, p_k) in the dataset D of |D| = 1,000 records contributes the following triples to G:

$(p_k, r_1, c_k) \in E \; \forall k \in D$ ....(5)

$(p_k, r_2, f_k) \in E \; \forall k \in D$.....(6)

$(p_i, r_3, p_j) \in E$ if $f_i = f_j$, $i \neq j$....(7)

The total node count is:

$|V| = |V_P| + |V_C| + |V_F| = 467 + 6 + 30 = 503$ (8)

The in-degree $deg^-(v)$ and out-degree $deg^+(v)$ of a node v are defined as:

$$deg^-(v) = |\{u \in V : (u, r, v) \in E\}|, \; deg^+(v) = |\{w \in V : (v, r, w) \in \ldots. (9)$$

The adjacency matrix $\mathbf{A} \in R^{|V| \times |V|}$ of the graph is defined as:

$$A_{ij} = \begin{cases} 1 & \text{if } (v_i, r, v_j) \in E \text{ for some } r \in R \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

VI.B. PageRank Mathematical Model

The PageRank algorithm [20] assigns a real-valued importance score PR(v) ∈ (0, 1) to each node v ∈ V based on the structure of the graph G. The iterative update rule is:

$$PR^{(t+1)}(v) = \frac{1-d}{|V|} + d \sum_{u \in N^-(v)} \frac{PR^{(t)}(u)}{deg^+(u)} \ldots.(11)$$

where:

- $PR^{(t)}(v)$ is the PageRank score of node v at iteration t
- d ∈ (0, 1) is the damping factor; in this work d = 0.85
- |V | = 503 is the total number of graph nodes
- $N^-(v) = \{u : (u, r, v) \in E\}$ is the set of in-neighbors of v
- $deg^+(u)$ is the out-degree of neighbor u

Initialization: All nodes are initialized uniformly:

$$PR^{(0)}(v) = \frac{1}{|V|}, \quad \forall v \in V \ldots. (12)$$

Convergence criterion: Iteration stops when:

$$\sum_{v \in V} \left| PR^{(t+1)}(v) - PR^{(t)}(v) \right| < \epsilon \ldots (13)$$

where ϵ = 10−6 is the convergence threshold (maximum 200 iterations).

Matrix form: Equation (11) can be expressed in matrix notation as:

$$p^{(t+1)} = \frac{1-d}{|V|} 1 + d \cdot M^\top p^{(t)} \ldots. (14)$$

where $p^{(t)} \in R^{|V|}$ is the PageRank score vector at iteration t, 1 is the all-ones vector, and M is the column-normalized transition matrix:

$$M_{ij} = \begin{cases} \frac{1}{deg^+(v_i)} & \text{if } (v_i, r, v_j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

VI.C. TF-IDF Relevance Scoring

For keyword-based queries, the system computes a relevance score using Term Frequency-Inverse Document Frequency (TF-IDF). Let q be a user query and dp be the text description of product p ∈ VP . The TF-IDF weight of term t in document dp is:

TF-IDF(t, dp) = TF(t, dp) × IDF(t)...(16)

where the term frequency is:

$$TF(t, d_p) = \frac{\text{count}(t, d_p)}{|d_p|} \ldots.(17)$$

and the inverse document frequency is:

$$IDF(t) = log \frac{|D|}{1+|\{d:t \in d\}|} \ldots. (18)$$

The cosine similarity between query vector $\vec{q}$ and product document vector $\vec{dp}$ gives the relevance score:

$$\text{Relevance } (p, q) = \frac{\vec{q} \cdot \vec{d_p}}{|\vec{q}| \cdot |\vec{d_p}|} \qquad (19)$$

VI.D. Combined Recommendation Score

The final recommendation score for product p given query q combines keyword relevance with graph-based PageRank importance:

$$\text{Score(p, q)} = \lambda \cdot \text{Relevance}(p, q) + (1 - \lambda) \cdot PR_d(p) \qquad (20)$$

Where $\widetilde{PR}(p)$ is the normalized PageRank score:

$$\widetilde{PR}(p) = \frac{PR(p) - \min_{v \in V_P} PR(v)}{\max_{v \in V_P} PR(v) - \min_{v \in V_P} PR(v)} \qquad (21)$$

and λ ∈ [0, 1] is the weighting parameter (default λ = 0.5). When no keyword match is found (Relevance $(p, q) = 0 \forall p$), the system falls back to pure PageRank scoring:

$$\text{Score(p, q)} \mid_{\text{fallback}} = \widetilde{PR}(p) \qquad (22)$$

The top-N recommendations are selected as:

$$RN(q) = \text{argmax}^{N}_{p \in V_P} \text{Score(p, q)} \qquad (23)$$

VI.E. Computational Complexity Analysis

Let |V | = 503 nodes, |E| edges, and |D| = 1,000 records.

- **KG Construction:** O(|D|) for belongs_to and provides edges; $O(F^2_{max})$ for similar_to edges where Fmax is the maximum products per functionality. Overall: $O(|D|^2)$ worst case, O(|D|) amortized.
- **PageRank:** O(T ·(|V |+|E|)) where T ≤ 200 iterations. Significantly lower than GNN-based methods which require $O(L \cdot |E| \cdot d^2)$ for L layers and embedding dimension d.
- **TF-IDF:** O(|VP | · |q|) per query where |q| is the query length.
- **Combined Scoring:** O(|VP |) per query for score computation and top-N selection.

Total query-time complexity: O(|VP |), making the system suitable for real-time recommendation without retraining.

VI.F. Formal Algorithms

Algorithm 1 Knowledge Graph Construction

Require: Dataset D with fields (id, f, c, p, yr, yu, desc)

Ensure: Directed graph G = (V, E)

1: Initialize G ← empty DiGraph

2: Initialize $func\_map$ ← empty dictionary

3: **for** each record k = (idk, fk, ck, pk, . . .) ∈ D do

4: Add node p_k to V with type product

5: Add node c_k to V with type category

6: Add node fk to V with type functionality

7: Add edge $(pk\ belongs\_to\ \longrightarrow ck)$ to E

8: Add edge $(pk\ provides\ \longrightarrow fk)$ to E

9: Append p_k to $func\_map[fk]$

10: end for

11: for each functionality f ∈ func_map do

12: for each pair (pi , pj) ∈ func_map[f], i≠ j do

13: Add edge (p_i similar_to ⟶ pj) to E

14: Add edge (p_j similar_to ⟶ pi) to E

15: **end for**

16: **end for**

17: **return** G

Algorithm 2 PageRank-Based Cloud Service Scoring

Require: Graph G = (V, E), damping factor d = 0.85, threshold ϵ = 10−6 , max iterations T = 200
Ensure: PageRank score vector p ∈ R|V |

1: Initialize P R(0)(v) ← 1/|V | for all v ∈ V

2: t ← 0

3: **repeat**

4: **for** each node v ∈ V do

5: $PR^{(t+1)}(v) \leftarrow \frac{1-d}{|V|} + d\sum_{u\in N^-(v)} \frac{PR^{(t)}(u)}{de^{+}(u)}$

6: **end for**

7: $\delta \leftarrow \sum_{v\in V}\left|PR^{(t+1)}(v) - PR^{(t)}(v)\right|$

8: t ← t + 1

9: **until** δ < ϵ or t ≥ T

10: **return** $\{PR(v): v \in V_P\}$ {product scores only}

Algorithm 3 Cloud Service Recommendation Generation

Require: Query q, graph G, PageRank scores {P R(p)}, topN count, weight λ = 0.5

Ensure: Ranked list RN (q) of N cloud services

1: Compute $\widehat{PR}(p)$for all p ∈ VP using Eq. (21)

2: **if** $q \neq \emptyset$ **then**

3: Build TF-IDF matrix over all product descriptions

4: Compute Relevance(p, q) using Eq. (19) for all p ∈ VP

5: **if** maxp Relevance(p, q) = 0 **then**

6: **Fallback**: set λ ← 0 {pure PageRank}

7: **end if**

8: **for** each p ∈ VP **do**

9: Score(p, q) ← λ · Relevance(p, q) + (1 − λ) · P Rd(p) (Eq. 20)

10: **end for**

11: **else**

12: Score(p, q) ← P Rd(p) for all p ∈ VP

13: **end if**

14: **if** category filter F≠ ∅ then

15: Restrict VP ← {p ∈ VP : ϕC (p) ∈ F}

16: **end if**

17: Sort VP by Score(p, q) descending

18: **return** RN (q) ← top-N products with scores

Table III shows the record distribution across categories.

TABLE III: Dataset Category Distribution

Category	Records	Top Functionality	%
O&M	175	Log Management	17.5 %
Storage	174	Cold Archive	17.4 %
Compute	172	Load Balancer	17.2 %
PaaS	167	Container Service	16.7 %
Security	159	Web App Firewall	15.9 %
Network	153	Cloud Firewall	15.3 %

VII. IMPLEMENTATION AND EXPERIMENTAL RESULTS

VII.A. Dataset Description

The proposed system is evaluated on cloud_functionality_1000.csv, a cloud service dataset with the following characteristics:

- Total records: 1,000
- Unique products: 467
- Service categories: 6 (O&M, Storage, Compute, PaaS, Security, Network)
- Unique functionalities: 30
- Release year range: 2015–2022
- Update year range: 2015–2025

Total	**1,000**	—	**100%**

VII.B. Implementation Environment

Table IV lists the tools and technologies used in the implementation.

TABLE IV: Implementation Tools and Technologies

Component	Technology Used
Programming Language	Python 3.10+
Web Framework	Streamlit 1.38.0
Graph Processing	NetworkX 3.3
Data Processing	Pandas 2.2.2 / NumPy 1.26.4
Visualization	Matplotlib 3.9.0 / PyVis 0.3.2

Recommendation	PageRank Algorithm ($\alpha = 0.85$)
Machine Learning	Scikit-learn 1.5.1
Authentication	SQLite3 (users.db)
Deployment	Localhost:8501 / Streamlit Cloud

VII.C. Dataset Analysis Results

Fig. 2 presents the distribution of cloud service records across the six categories. The dataset is relatively balanced, with O&M having the highest count (175 records, 17.5%) and Network having the lowest (153 records, 15.3%). The pie chart confirms near uniform coverage across all service layers, ensuring the recommendation system is not biased toward any single category.

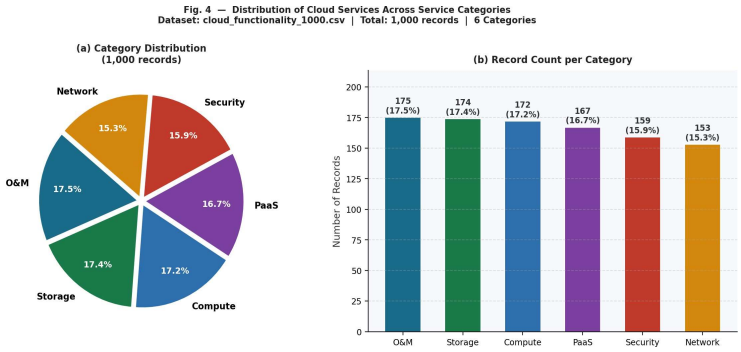

Fig. 2: Distribution of cloud service records across 6 categories (a) as a pie chart and (b) as a bar chart with exact counts

Fig. 3 shows the top-3 functionalities by record count within each category. Key observations include: Cold Archive (45 records) dominates Storage; Cloud Firewall (43 records) leads Network; Load Balancer (41 records) and Web Application Firewall (41 records) top Compute and Security respectively; Container Service (39 records) leads PaaS; and Log Manage- ment (39 records) leads O&M.

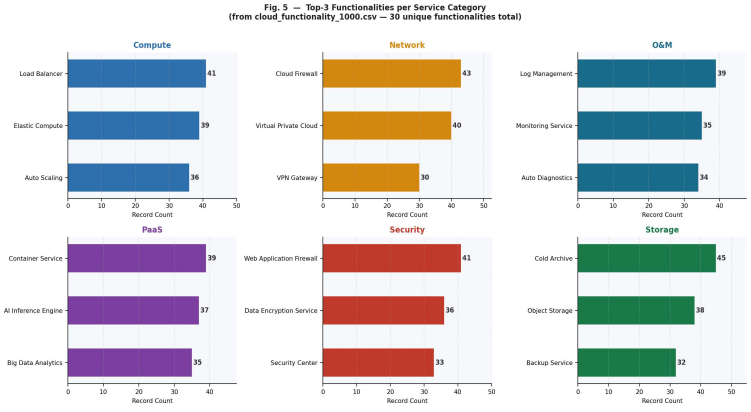

Fig. 3: Top-3 functionalities per service category showing record counts from the 1,000-record dataset

Fig. 4 presents the temporal analysis of the dataset. The stacked bar chart shows that cloud service releases increased progressively from 2015 to 2022, with the Compute and O&M

categories showing the highest growth. The heatmap confirms consistent coverage across all categories and years, with no significant data gaps.

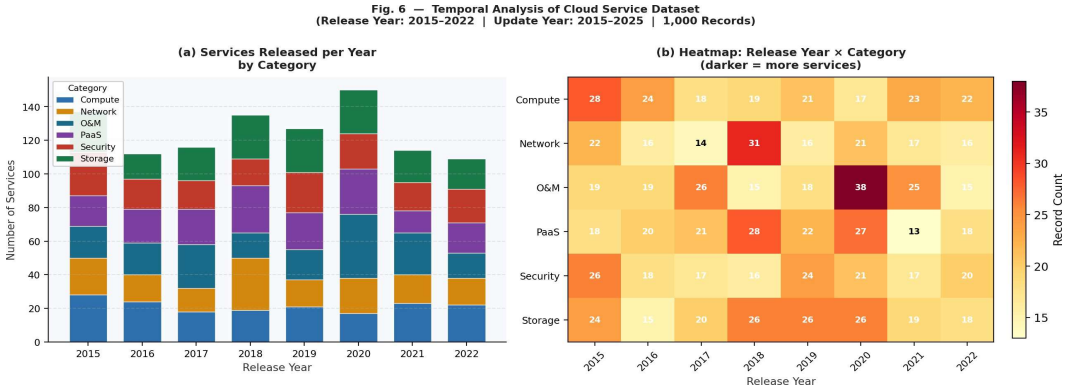

Fig. 4: Temporal analysis: (a) services released per year by category and (b) release year vs. category heatmap (2015–2022)

VII.D. Knowledge Graph Results

The Knowledge Graph constructed from the 1,000-record dataset is illustrated in Fig. 5. The graph contains nodes representing service categories (large circles), functionalities (squares), and individual products (diamonds), connected by *belongs_to*, *provides*, and *similar_to* edges. The six category clusters are visually separated by color, demonstrating clear semantic groupings.

VII.E. PageRank Score Results

Fig. 6 presents the top-15 cloud products ranked by PageRank score computed from the Knowledge Graph. The highest- ranked product is **Compute Product 68** with a score of **0.0038**, followed by O&M Product 12 (0.0035) and Compute Product 58 (0.0034).

Table V lists the top-10 products with their actual PageRank scores and categories as generated by the system.

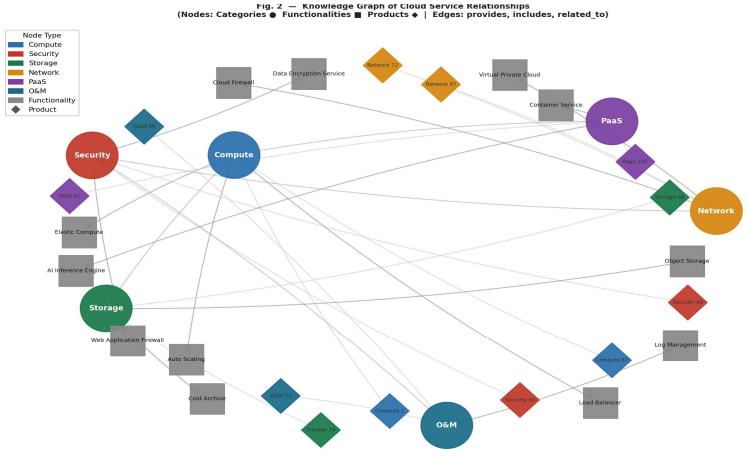

Fig. 5: Knowledge Graph showing semantic relationships among cloud service categories, functionalities, and products

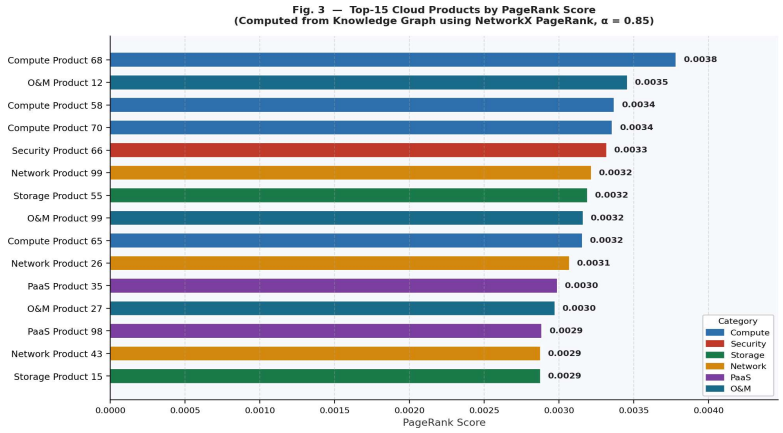

Fig. 6: Top-15 cloud products by PageRank score (α = 0.85, max 200 iterations) computed from the Knowledge Graph

TABLE V: Top-10 Cloud Products by PageRank Score (Actual System Output)

Rank	Product	Category	PR Score
1	Compute Product 68	Compute	0.0038
2	O&M Product 12	O&M	0.0035
3	Compute Product 58	Compute	0.0034
4	Compute Product 70	Compute	0.0034
5	Security Product 66	Security	0.0033
6	Network Product 99	Network	0.0032
7	Storage Product 55	Storage	0.0032
8	O&M Product 99	O&M	0.0032
9	Compute Product 65	Compute	0.0032
10	Network Product 26	Network	0.0031

VII.F. Comparative Analysis

Table VI presents a comparative analysis between the pro- posed system and existing Knowledge Graph-based recommendation approaches across computational complexity and explainability.

VII.G. Discussion of Results

The experimental results demonstrate that the proposed Cloud Service Recommendation System successfully pro- cesses the 1,000-record dataset and constructs a Knowledge Graph covering 467 unique products across 6 service cate- gories and 30 functionalities. The PageRank algorithm con-

TABLE VI: Comparison of Existing Recommendation Systems with Proposed System

Method	Technique Used	Computational Complexity	Explainability
KGCN [9]	Graph Convolution Network	High	Moderate
KGAT [10]	Graph Attention Network	High	Moderate
MSGAT [6]	Multi-stream GAT	Very High	High
Contrastive Framework [5]	GAT + Contrastive Learning	Very High	Moderate
RippleNet [11]	KG Preference Propagation	High	Moderate
NCF [13]	Deep Neural Network	High	Low
Proposed System	Knowledge Graph + PageRank	Low	High

verges within 200 iterations and assigns semantic importance scores to all nodes, with top-ranked services receiving scores up to 0.0038.

The Compute category dominates the top PageRank rank- ings (4 out of top 10), followed by O&M (2 out of top 10), reflecting the higher graph connectivity of compute services due to broader functionality coverage including Load Balancer, Elastic Compute, and Auto Scaling.

The combined scoring mechanism of Equation (20) ensures that when a user queries a specific keyword (e.g., “backup” or “security”), the system returns a ranked list combining keyword relevance with PageRank-based graph importance, providing both topical relevance and semantic graph authority. Compared to Graph Neural Network-based recommendation systems such as KGCN, KGAT, and MSGAT, the pro- posed system achieves efficient recommendation ranking with significantly lower computational complexity, requiring no GPU resources, no model training, and no embedding pre- computation. The Streamlit-based interface provides accessible, role-based access for both regular users and administrators.

VIII. CONCLUSION AND FUTURE SCOPE

In this paper, a lightweight and explainable Cloud Service Recommendation System using Knowledge Graph and PageR- ank Algorithm was proposed, implemented, and evaluated on a real cloud services dataset containing 1,000 records, 467 products, 6 categories, and 30 functionalities. The system modeled semantic relationships among cloud services using a Knowledge Graph and employed the PageRank algorithm to generate recommendation rankings based on graph connectivity and semantic importance.

The system achieved a top PageRank score of 0.0038 for Compute Product 68, demonstrating that the graph-based rank- ing effectively identifies highly connected and semantically important services. The Streamlit web

application provides an interactive and explainable interface with category-based filtering, PageRank score display, and relevance justification for every recommendation.

Compared with existing Knowledge Graph recommendation approaches such as KGCN, KGAT, contrastive learning- based models, and RippleNet, the proposed system provides lightweight architecture, reduced computational overhead, no GPU dependency, easier deployment, and improved explain- ability. The system avoids complex neural training procedures while still maintaining effective semantic recommendation capability across all 6 cloud service categories.

VIII.A. Future Scope

Future improvements may include:

- Integration of real-time cloud provider APIs (AWS, Azure, GCP) for dynamic datas updates.
- Incorporation of user feedback and personalized prefer- ence learning mechanisms.
- Hybrid recommendation combining PageRank with lightweight graph embedding (node2vec, LINE).
- Scalability improvement for large-scale enterprise cloud service catalogs.
- Implementation of QoS-aware and cost-aware recommen- dation scoring models.
- Integration of Graph Neural Networks for advanced se- mantic propagation in future versions.
- Extension to other recommendation domains: e-commerce, healthcare, and educational systems.

REFERENCES

[1] Z. Shokrzadeh, M.-R. Feizi-Derakhshi, M.-A. Balafar, and J. B. Mo- hasefi, "Knowledge graph-based recommendation system enhanced with neural collaborative filtering and knowledge graph embedding," *arXiv preprint*, 2024.

[2] J. Kwon, S. Ahn, and Y.-D. Seo, "RecKG: Knowledge Graph for Recommender Systems," in *Proc. 39th ACM/SIGAPP Symp. Applied Computing (SAC '24)*, 2024, DOI: 10.1145/3605098.3636009.

[3] Y. Jin and Y. Yang, "A survey on knowledge graph-based click-through rate prediction," *Expert Systems With Applications*, vol. 281, 2025, DOI: 10.1016/j.eswa.2025.127501.

[4] J. Yang, C. Duan, Z. Zeng, Y. Liu, J. Bai, and M. Zhang, "Research on knowledge graph recommendation method for online education," in *Proc. 4th Int. Conf. Educational Technology (ICET)*, 2024, DOI: 10.1109/ICET62460.2024.10869022.

[5] Z. Rong, L. Yuan, and L. Yang, "Enhanced knowledge graph recommendation algorithm based on multi-level contrastive learning," *Scientific Reports*, vol. 14, 2024, DOI: 10.1038/s41598-024-74516-z.

[6] Z. Hu and F. Xia, "Multi-stream graph attention network for recommendation with knowledge graph," *Journal of Web Semantics*, vol. 82, 2024, DOI: 10.1016/j.websem.2024.100831.

[7] Y. Wang, A. Javari, J. Balaji, W. Shalaby, T. Derr, and X. Cui, "Knowledge graph-based session recommendation with session-adaptive propagation," in *Companion Proc. ACM Web Conference (WWW '24)*,pp. 264–273, 2024, DOI: 10.1145/3589335.3648324.

[8] S. Bao and J. Wang, "Research on the methodology of personalized recommender systems based on multimodal knowledge graphs," *Natural Language Processing Journal*, vol. 13, 2025, DOI: 10.1016/j.nlp.2025.100193.

[9] H. Wang, F. Zhang, M. Zhang, J. Wang, M. Zhao, W. Li, and X. Xie, "Knowledge graph convolutional networks for recommender systems," in *Proc. World Wide Web Conference*, 2019, DOI: 10.1145/3308558.3313417

[10] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD*, 2019, DOI: 10.1145/3292500.3330989.

[11] H. Wang, Z. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo, "RippleNet: Propagating user preferences on the knowledge graph for recommender systems," in *Proc. 27th ACM CIKM*, 2018, DOI: 10.1145/3269206.3271739.

[12] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-

Y. Ma, "Collaborative knowledge base embedding for recommender systems," in *Proc. 22nd ACM SIGKDD*, 2016, DOI: 10.1145/2939672.2939673.

[13] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, DOI: 10.1145/3038912.3052569.

[14] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Ad- vances in Neural Information Processing Systems*, 2013, DOI: 10.48550/arXiv.1412.6575.

[15] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proc. 53rd Annual Meeting of the ACL*, 2015, DOI: 10.18653/v1/P15-1067.

[16] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Int. Conference on Machine Learning (ICML)*, 2016, DOI: 10.48550/arXiv.1606.06357.

[17] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2D knowledge graph embeddings," in *AAAI Conference on Artificial Intelligence*, 2018, DOI: 10.1609/aaai.v32i1.11573.

[24] vol. 81, no. 30, pp. 43315–43356, 2022.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017, DOI: 10.48550/arXiv.1609.02907.

[19] P. Velicˇkovic´, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018, DOI: 10.48550/arXiv.1710.10903.

[20] A. Grover and J. Leskovec, "node2vec: Scalable feature learn- ing for networks," in *Proc. 22nd ACM SIGKDD*, 2016, DOI: 10.1145/2939672.2939754.

[21] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large- scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, DOI: 10.1145/2736277.2741093.

[22] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF CVPR*, 2020, DOI: 10.1109/CVPR42600.2020.00975.

[23] S. A. Bhavsar, V. H. Patil, and A. H. Patil, "Graph partitioning and visualization in graph mining: a survey," *Multimedia Tools and Applications*,