RESEARCH ARTICLE OPEN ACCESS

Graph-Based Peer-to-Peer Skill Exchange Platform

Parth G. Gaikwad*, Srushti A. Rajkule**, Shreya Kolam***, Ganesh S. Patinge****, Rucha Thakre*****

*(Department of Computer Sciences and Engineering, Sandip University, Nashik, India Email: parth271gaikwad@gmail.com)
**(Department of Computer Sciences and Engineering, Sandip University, Nashik, India Email: rajkulesrushti2005@gmail.com)
***(Department of Computer Sciences and Engineering, Sandip University, Nashik, India Email: shreyakolam@gmail.com)
****(Department of Computer Sciences and Engineering, Sandip University, Nashik, India Email: ganeshpatinge81@gmail.com)
*****(Department of Computer Sciences and Engineering, Sandip University, Nashik, India Email: rucha.thakare@sandipuniversity.edu.in)

----------**********----------

Abstract:

Peer-to-peer (P2P) learning allows people to share skills directly with other learners instead of depending only on formal courses. Existing skill exchange platforms make this idea practical, but many of them still use keyword search or simple category matching. Such matching can miss useful relationships between related skills, users, learning goals, and previous interactions. This paper presents a proposed graph-based P2P skill exchange platform that uses Neo4j to model users, skills, interests, communities, sessions, and ratings as connected data. A graph-based recommendation process is then used to identify learning partners using skill similarity, shared interests, learning history, trust, ratings, and community participation. The work is based on a review of recent studies on skill exchange, knowledge graphs, graph learning, graph neural networks, and Neo4j-based recommendation. The main contribution is a unified framework that brings reciprocal learning and relationship-aware recommendation into one system. The paper focuses on system design and analytical discussion; it does not claim experimental accuracy because the proposed platform has not yet been experimentally implemented.

Keywords — **Peer-to-peer learning, skill exchange, Neo4j, graph database, recommendation system, knowledge graph, graph learning.**

----------**********----------

I. INTRODUCTION

The way people learn has changed quickly with the growth of the internet, smartphones, and online learning services. Learning is no longer limited to a classroom or a fixed teacher-student arrangement. People can learn from others through digital communities, discussion groups, video sessions, and skill-sharing platforms. In this setting, peer-to-peer learning is useful because a person can act as a learner in one topic and as a mentor in another. The survey that motivates this work describes P2P learning as direct sharing of knowledge, experience, and practical skills between participants. [1]-[4] Skill exchange platforms are a practical form of this model. A user can list the skills they already know and the skills they want to learn. For example, one user may know Python and want to learn UI design, while another user may know UI design and want to learn Python. In such a case, both users can benefit from each other without depending on a paid course. This makes the idea attractive for students, professionals, and local communities. The main problem is not only finding a skill, but finding the right person. Many existing systems depend on exact keywords or manually selected categories. A keyword such as "Artificial Intelligence" may fail to connect a learner with someone whose profile emphasizes Python, Machine Learning, or Data Science, even though these skills are strongly related.

 ©IJSRED: All Rights are Reserved

The survey reports that conventional systems also overlook factors such as trust, reputation, learning history, community participation, and reciprocal learning opportunities. Graph databases are a natural fit for this problem because they store entities and relationships together. Instead of forcing every relationship into separate relational tables and then joining them, a graph can directly represent connections such as User-KNOWS-Skill, User-WANTS_TO_LEARN-Skill,User-ATTENDED-Session, and User-RATED-User. Neo4j is therefore selected in the proposed framework as the graph database layer.

II. PROBLEM STATEMENT & RESEARCH GAP

A. Related Work From The Survey The first group of studies reviewed in the survey focuses directly on peer-to-peer skill exchange. SkillSwap demonstrates a communitydriven learning model in which users can list skills they can teach and skills they want to learn. Its main strength is the simple idea of mutual learning, but the survey notes that matching is largely based on keywords and manually selected categories. As the number of users grows, this approach can return less relevant matches because it does not understand indirect skill relationships or user reputation. [1] SkillSphere adds a local-community dimension to the same idea. It is useful when users want to meet people in the same geographical area for practical learning. The survey highlights the benefits of local collaboration, but also notes that the system does not fully consider previous collaborations, reputation, or hidden relationships between users. [2] SkillGrid treats skill exchange as a digital marketplace where knowledge can be exchanged without a direct financial transaction. Profiles, search, ratings, and feedback make the model more practical, but the survey again identifies keyword-based matching as a limitation. Two users may have related skills but different descriptions, preventing a match. [3] The Micro Skill Exchange Platform moves toward short learning activities instead of large courses. This is useful for skills such as resume writing, basic design, spreadsheet automation, cooking, or language practice. The survey notes that scheduling, session tracking, and reviews improve user interaction, but the recommendation mechanism remains relatively static. [12] The second group of studies provides the graph-based part of the research direction. The survey on knowledge graph-based recommender systems explains that knowledge graphs can combine user interests, item attributes, behavior, and semantic relationships in one connected structure. This richer representation can help recommendation systems find indirect relationships that simple matching may miss. [4]

The review of graph learning-based recommender systems discusses Graph Convolutional Networks, Graph Attention Networks, Graph Autoencoders, graph embeddings, and link prediction. The important idea for the proposed platform is that graph learning considers relationships as part of the data rather than treating every record as independent. [5] PinSage shows how graph convolution can be applied to very large recommendation graphs. The survey describes its use of graph sampling to manage a very large number of relationships. This provides a useful lesson for future scalability, although such deep models may be unnecessarily complex for an initial academic implementation. [7]

The GNN recommender survey similarly indicates that models such as GCN, GAT, GraphSAGE, GIN, and heterogeneous GNNs can learn structural information from recommendation graphs. For this research, the important point is not to deploy a deep neural network immediately, but to keep the graph model flexible enough for later graph-learning experiments. [8]

The third group contains graph database applications. Skills2Job and Skills2Graph show how graph representations can organize skills and relationships for recommendation and skill modeling, but their main focus is closer to employment and professional networking than reciprocal learning. Neo4j Graph Data Science studies described in the survey provide practical graph algorithms such as similarity search, community detection, link prediction, PageRank, shortest-path analysis, centrality measures, and embeddings. [6], [9]-[11]

 ©IJSRED: All Rights are Reserved

Taken together, the reviewed work suggests that the two sides of the problem are already understood separately. P2P platforms show how people can exchange knowledge, while graph-based recommenders show how connected data can improve recommendations. The missing step is a single educational platform that uses the graph itself to represent reciprocal skill exchange and then uses those relationships during partner selection.

The literature reviewed in the source survey can be grouped into three broad areas: P2P skill exchange platforms, graph-based recommendation systems, and graph database applications. SkillSwap, SkillSphere, SkillGrid, and the Micro Skill Exchange Platform show that community-based learning is practical. However, their matching mechanisms are described as simple, mainly based on keywords, categories, or static searches. [1]-[3], [12] On the other side, studies on knowledge graphs, graph learning, PinSage, GNN-based recommenders, Skills2Job, Skills2Graph, and Neo4j Graph Data Science show that graph structures can improve recommendation quality and reveal indirect relationships. However, these works mainly address domains such as employment, e-commerce, social networks, or general recommendation rather than reciprocal educational skill exchange. [4]-[11]

This creates a clear gap: there is limited work that combines community-driven P2P skill exchange with a graph representation and a relationship-aware recommendation process in one platform. The proposed work addresses this gap by treating users, skills, interests, communities, learning sessions, and ratings as connected entities. It also makes reciprocal learning a central feature instead of a secondary option.

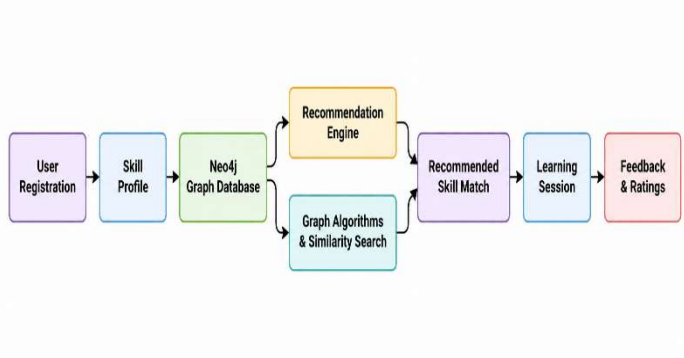


Fig. 1. Proposed graph-based peer-to-peer skill exchange platform.

III. PROPOSED SYSTEM

3.1 SYSTEM OPERATION WITH A SIMPLE EXAMPLE

Consider a learner who knows Python and wants to learn UI design. The learner creates a profile with these two pieces of information. Another learner knows UI design and wants to learn Python. In a simple keyword system, the two profiles may be found because the skill names match. In the proposed graph system, the same information is stored as relationships, so the platform can recognize that the two users have complementary goals. This makes reciprocal matching a direct part of the model.

The graph becomes more useful when the profiles contain additional information. Suppose the UI-design learner has previously completed successful sessions, has good ratings, and belongs to an active design community. These facts can be connected to the same user node. When the Python learner asks for a partner, the recommendation process can use these connections together instead of treating the profile as a single text record.

The design also supports one user having several learning paths at the same time. A student may know programming and database skills, want to learn cloud computing, and belong to a college technical community. The graph does not require a rigid table structure for every combination. New relationships can be added as the platform grows, which is one of the reasons the survey identifies graph storage as a flexible approach.

3.2 SYSTEM MODULES

User Management Module: This module handles account creation and the basic user profile. The profile acts as the starting point for the graph. It includes information such as educational or professional background and the user's learning goals.

Skill Management Module: Users enter the skills they can teach and the skills they want to learn. Skill proficiency and interests can also be stored. Related skill categories can later be connected so that the recommendation system does not depend only on exact wording.

Graph Database Module: The Neo4j layer stores users, skills, interests, communities, sessions, ratings,

 ©IJSRED: All Rights are Reserved

and their relationships. It is the central representation used by the recommendation process.

Recommendation Engine: This module searches the graph, identifies candidate users, applies similarity and trust-related criteria, and ranks the candidates. Its purpose is to reduce manual searching and provide more relevant learning partners.

Feedback and Rating Module: After a session, users provide feedback and ratings. These records become part of the graph and can influence future recommendations. In this way, the system can gradually learn which types of matches are successful.

The proposed system is a conceptual framework derived from the research gap identified above. A user creates an account and builds a profile containing existing skills, desired skills, skill proficiency, areas of interest, preferred learning mode, and optional location. The profile is then represented in the Neo4j graph database. The system uses graph relationships to search for suitable learning partners instead of depending only on exact text matches.

The platform has five major functional parts: user management, skill management, graph data storage, recommendation, and feedback/rating. The user-management and skill-management parts collect the information required for matching. The graph layer stores the relationships. The recommendation layer searches and ranks possible partners. After a learning session, feedback and ratings are returned to the graph so that future recommendations can be improved.

Reciprocal learning is central to the proposed design. A user is not permanently classified as only a teacher or only a student. The same person may teach one skill, learn another skill, and participate in several communities. This two-way model is expected to encourage continuous learning and stronger community interaction.

IV. GRAPH DATA MODEL

The graph data model contains six main entities: User, Skill, Interest, Community, Session, and Rating. The relationships described in the survey include KNOWS, WANTS_TO_LEARN, ATTENDED, RATED, BELONGS_TO, and COMPLETED_SESSION. The simplified paper-ready view in Fig. 2 keeps the model easy to read while retaining the important entities. [1]-[12]

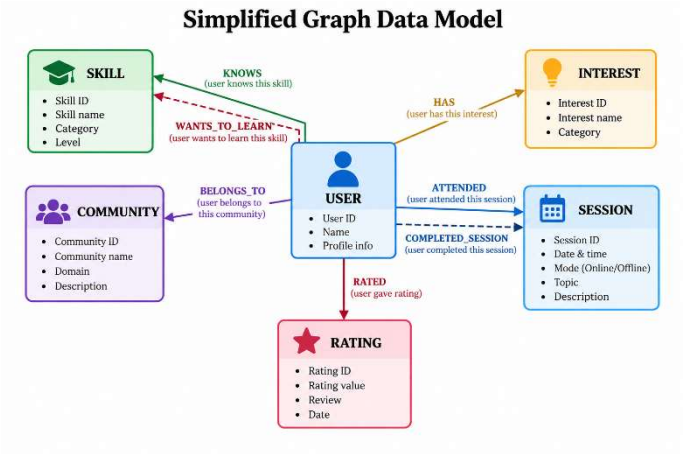


FIG. 2. SIMPLIFIED GRAPH DATA MODEL OF THE PROPOSED PLATFORM.

A major advantage of this representation is that the recommendation system can move through several connected entities. A user can be connected to a known skill, a desired skill, an interest area, a community, previous sessions, and ratings. These paths can reveal information that a simple keyword search cannot easily express.

V. RECOMMENDATION METHODOLOGY

5.1 RECOMMENDATION CRITERIA IN DETAIL

Skill similarity is the first criterion because the recommended partner should be able to teach the requested skill or a closely related skill. The graph makes it possible to move beyond exact names and explore connected or semantically related skills.

Shared interests are useful because learning is often more successful when users have similar goals or topic interests. For example, two users interested in data science may be compatible even when one is stronger in Python and the other in statistics.

Learning history gives the system some context about what has already happened. Previous sessions and completed exchanges can help the engine avoid repeatedly recommending the same poor match and

 ©IJSRED: All Rights are Reserved

can highlight users who have a history of useful interactions.

Ratings and trust add a quality filter. A person with suitable skills may not be the best recommendation if previous sessions received poor feedback. Conversely, a highly rated user with relevant skills may deserve a higher ranking. The survey therefore includes ratings, reputation, and trust among the important graph signals.

Community participation can act as another indication of activity. An active learner who participates in communities and sessions may be more likely to respond and engage than a profile that has not interacted with the platform recently. These signals can be combined rather than treated as separate manual searches.

5.2 TRUST, FEEDBACK, AND THE COLD-START PROBLEM

Trust is important in a peer-learning platform because users are interacting directly with other people. The survey proposes using ratings and reputation information as part of the graph. A practical implementation should avoid treating one rating as enough evidence. Reputation can be updated from repeated completed sessions so that the recommendation engine uses a broader history.

New users create a different problem. At the beginning, a new profile may contain only a few skills and no session history. This is known as the cold-start problem, and the survey identifies it as a limitation of recommendation systems in general. A future implementation can handle this by asking new users for richer initial preferences, using basic skill similarity first, and gradually incorporating trust and history as interactions accumulate.

5.3 WHY NEO4J IS SUITABLE

The proposed database choice is based on the structure of the problem. A user is connected to many skills, interests, communities, sessions, and ratings. Each skill may also be related to other skills. In a relational representation, the number of joins needed to explore these paths can grow as the data model becomes more connected. Neo4j stores the relationships directly and allows graph traversal through those connections. The survey identifies faster relationship queries, flexible data modeling, scalability, and easier visualization as practical advantages. [10], [11]

Neo4j also provides Cypher, a query language designed for graph patterns. This is useful for research because a rule such as "find users who know a required skill and are connected to related interests" can be expressed as a graph pattern rather than a long sequence of relational joins. The exact Cypher queries and algorithm choices remain part of the implementation phase of this research.

The recommendation process starts when a user specifies a learning requirement. The system then performs graph traversal to find candidate users and related skills. Rather than stopping at an exact keyword match, the process considers multiple graph signals. The source survey identifies skill similarity, common interests, learning history, user ratings, community participation, previous successful exchanges, and trust as important recommendation criteria.

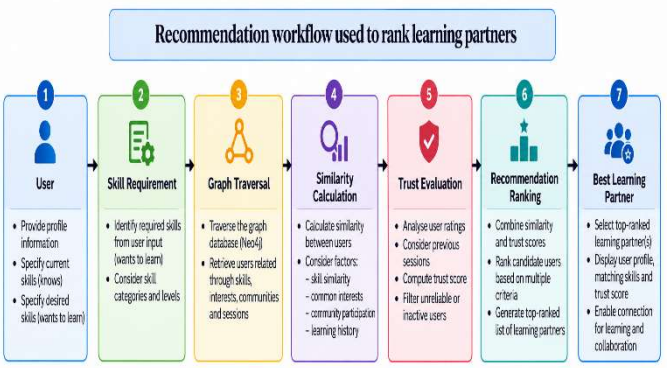


Fig. 3. Recommendation workflow for selecting a learning partner.

The process can be understood in six simple steps. First, the system receives the user and the required skill. Second, graph traversal identifies connected users and skills. Third, similarity is considered to remove weak matches. Fourth, trust and reputation are checked using ratings and past interactions. Fifth, candidates are ranked using the available relationship information. Finally, the highest-ranked compatible user is presented as a recommended learning partner. This workflow follows the conceptual sequence described in the survey.

Neo4j provides graph algorithms that can support this type of analysis, including similarity search, community detection, link prediction, PageRank, shortest-path analysis, centrality measures, and

 ©IJSRED: All Rights are Reserved

embeddings. The research reviewed in the survey suggests that these techniques can be useful for relationship-aware recommendation. The proposed system does not claim to use every algorithm at once. The choice of algorithm can depend on the size of the graph, available data, and future experimental evaluation.

VI. LITERATURE-BASED COMPARISON

The main difference between the traditional and proposed approaches is the way information is connected. A relational approach can store users and skills effectively, but recommendation queries may become more complicated as the number of relationships grows. A graph model directly expresses these relationships and allows the system to traverse them. The following comparison summarizes the framework proposed in this paper.

Scalability: The graph structure is intended for highly connected data and can be extended with new entities and relationships.

Learning model: The proposed platform supports reciprocal learning, so one user can teach one skill and learn another.

Trust: The proposed model keeps ratings and reputation as graph information that can influence ranking.

Relationships: Traditional matching mainly checks direct similarity; graph matching can follow direct and indirect relationships.

Personalization: Traditional systems use fewer profile signals; the proposed design can consider skills, interests, history, ratings, trust, and participation.

Matching: Traditional matching depends mainly on keywords or categories; the proposed system uses graph traversal and similarity.

Database: Traditional systems use relational/SQL storage, while the proposed model uses Neo4j to keep connected information together.

VII. RESULTS AND DISCUSSION

7.1 EXPECTED EVALUATION PLAN

A complete implementation should be evaluated with real or carefully prepared user profiles. The first experiment should compare keyword-based matching with graph-based matching using the same set of users and skills. Useful measures would include how often the top recommendations are relevant, how many recommended pairs are reciprocal, and how much manual searching users need to perform. The current paper does not report such measurements because the proposed platform has not yet been implemented.

A second experiment can test the effect of additional graph signals. Recommendations can be generated using only skill similarity and then compared with recommendations that also include ratings, learning history, and community participation. This would help identify which signals actually improve partner selection instead of assuming that every available feature is beneficial.

A third experiment can study scalability. The number of users, skills, relationships, and recommendation requests can be increased gradually while recording query time and resource usage. This would provide evidence for the survey's expectation that graph structures are suitable for highly connected data. It would also reveal the point at which more advanced graph-learning methods become useful.

Finally, user feedback should be included in the evaluation. Recommendation quality is not only a technical issue. A mathematically close match may still be a poor learning partner if communication, availability, or expectations are different. The feedback module therefore provides a way to evaluate whether the graph-based recommendations are actually useful to people.

7.2 PRACTICAL IMPLEMENTATION CONSIDERATIONS

A practical implementation can be divided into stages. The first stage is basic web or mobile user registration and profile creation. The second stage is the Neo4j graph model and data loading. The third stage is a simple graph-traversal recommendation engine. Only after this foundation works reliably should more complex graph learning or GNN models be introduced.

 ©IJSRED: All Rights are Reserved

Data quality is another important consideration. Skill names may contain spelling differences, abbreviations, or different levels of detail. The system may therefore need a controlled skill vocabulary or a method for connecting related skill nodes. The survey's discussion of semantic relationships and knowledge graphs supports this direction, but the exact ontology or taxonomy is still a future implementation decision. [4], [9]

Privacy and user control should also be considered in a real deployment. The graph may contain information about skills, learning history, reputation, and communities. Not every relationship needs to be publicly visible. A practical system should allow users to control which profile details are visible to other participants while still providing enough information for meaningful recommendations.

The result of this research is a unified conceptual architecture rather than a measured software implementation. The literature reviewed in the source survey consistently points to the usefulness of graph-based representation for recommendation. P2P platforms demonstrate the value of mutual learning, while graph recommendation studies demonstrate the value of modeling relationships. Combining these two ideas provides a stronger basis for intelligent partner selection than simple search.

The first expected improvement is better recommendation relevance. A user interested in Artificial Intelligence may benefit from users connected to Python, Machine Learning, or Data Science. A graph can represent these relationships and therefore widen the useful search space without depending on identical words. This is the main reason a graph structure is important for the proposed platform.

The second expected improvement is personalization. Two learners asking for the same skill may not need the same mentor. One may prefer a nearby user, another may value a highly rated mentor, and another may benefit from someone with a similar learning history. The proposed criteria provide a way to include these differences in the ranking process.

The third expected improvement is more efficient relationship handling. Neo4j stores relationships directly, which can reduce the complexity of multi-table joins for relationship-heavy queries. The survey also identifies flexibility and scalability as important advantages of graph databases for connected datasets. However, these claims should be validated experimentally in future work rather than treated as measured results of the current framework.

The proposed platform also supports community building. Users can belong to communities, attend sessions, exchange feedback, and gradually build a reputation. This turns the platform from a simple search tool into a learning network. The same design could later be used in educational institutions, professional groups, corporate training, and local communities.

VIII. LIMITATIONS

The proposed framework has important limitations. First, it is conceptual and has not yet been experimentally implemented. Therefore, no numerical accuracy, response-time, or scalability values are reported in this paper. Second, recommendation quality depends on the quality of user profiles and interaction data. New users may suffer from a cold-start problem because little information is available about their skills or preferences. Third, large graph structures and graph algorithms can require additional storage and computing resources. Finally, trust and reputation should be designed carefully so that ratings are meaningful and resistant to manipulation.

IX. FUTURE ENHANCEMENT

Future implementation can extend the proposed system in several directions. Graph Neural Networks and other graph-learning methods can be added after enough interaction data becomes available. Machine learning can be used to update recommendation behavior from feedback. An AI assistant can help users describe their skills, identify learning resources, and understand recommendations. Multilingual support can improve accessibility, while a mobile application can make skill exchange easier to use from smartphones.

Blockchain can also be considered for verification of completed learning sessions or skill credentials, especially when certificates need to be trusted across

 ©IJSRED: All Rights are Reserved

organizations. Gamification features such as badges and points may increase participation. Integration with online meeting tools could support scheduling and virtual sessions. These enhancements are consistent with the future directions identified in the source survey. [1]-[12]

X. SOCIAL IMPACT

The proposed platform has a clear social value because it can lower the cost of learning. People who cannot afford many paid courses could exchange knowledge with peers. Students can learn from senior students, professionals can exchange practical knowledge, and local communities can share digital and entrepreneurial skills. The survey particularly highlights the potential value for rural and semi-urban areas, where access to quality learning resources may be more limited. The platform therefore supports the broader idea of affordable and continuous learning.

XI. CONCLUSION

This paper presented a graph-based peer-to-peer skill exchange framework derived from a review of recent research on P2P learning, knowledge graphs, graph learning, GNNs, and Neo4j-based recommendation. The key problem addressed is the weakness of keyword-based matching when the relationship between users and skills is richer than the text stored in a profile.

The proposed framework represents users, skills, interests, communities, sessions, and ratings as connected graph entities. Neo4j is used as the conceptual graph storage layer, while graph traversal, similarity analysis, trust evaluation, and ranking form the recommendation process. This design supports reciprocal learning, where users can both teach and learn. The literature indicates that the approach has strong potential for improved personalization, relationship analysis, and scalable knowledge sharing, but experimental testing is still required.

The main value of this work is therefore not a claimed numerical improvement, but a clear and implementable research direction that joins P2P learning with graph-based recommendation. Future implementation and evaluation can determine which graph algorithms, ranking strategies, and trust models provide the best results for real learning communities.

REFERENCES

[1] SkillSwap: A Peer-to-Peer Skill Exchange Platform for Community-Driven Learning.
[2] SkillSphere: A Local Skill Exchange Platform for Communities.
[3] SkillGrid - Skill Exchange Platform.
[4] A Survey on Knowledge Graph-Based Recommender Systems.
[5] Graph Learning Based Recommender Systems: A Review.
[6] Skills2Job: A Recommender System Using Graph Databases.
[7] PinSage: Graph Convolutional Neural Networks for Web-Scale Recommender Systems.
[8] Graph Neural Networks in Recommender Systems.
[9] Skills2Graph.
[10] Neo4j Graph Data Science for Recommendation Systems.
[11] A Graph Learning Framework for Recommendation.
[12] SkillSwap - Micro Skill Exchange Platform.

 ©IJSRED: All Rights are Reserved