International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 10 | Oct - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Advanced Algorithms and Implementations for Talent Matching and Team Formation in Crowdsourced Engineering

Nikita Gharate[1] , Isha Katariya[2], Shital Kendre[3], Shreyas Kothimbire[4], Prof. Nikhil Deshpande[5]

[1] *Department Of Information Technology, Sinhgad College of Engineering, Pune- 41*
[2] *Department Of Information Technology, Sinhgad College of Engineering, Pune- 41*
[3] *Department Of Information Technology, Sinhgad College of Engineering, Pune- 41*
[4] *Department Of Information Technology, Sinhgad College of Engineering, Pune- 41*
[5] *Department Of Information Technology, Sinhgad College of Engineering, Pune- 41*
Email: nikkigharte2004@gmail.com

Abstract - This project addresses the significant problem of high failure rates (approximately 15.7%) in crowdsourced engineering and hackathon environments, which are primarily caused by static, manual team formation processes that result in poorly matched skills and ineffective collaboration. We propose TalentBridge, an automated platform that optimizes talent matching and team formation by leveraging real-time data and a Hybrid Machine Learning (ML) pipeline. The methodology involves: API integration with platforms like GitHub and LinkedIn to fetch live data; KeyBERT (NLP) for contextualized skill extraction from profiles; and K-Means Clustering for grouping participants into well-balanced teams with complementary skill sets. The implemented system reduces manual effort and errors, enhances team balance and fairness, and is expected to lead to improved match quality and higher success rates for projects. This scalable, data-driven coordination mechanism is a unique and necessary solution for fostering innovation in dynamic crowdsourced environments.

- ***Key Words***: Talent Matching, Team Formation, Crowdsourced Engineering, Hackathons, Machine Learning (ML), KeyBERT, K-Means Clustering, Natural Language Processing (NLP), Real-Time Data, API Integration, Skill Extraction, Recommender Systems, Collaborative Filtering, Web Systems, Project Success**.**

1. INTRODUCTION

Crowdsourced engineering and hackathon platforms represent a critical, dynamic environment for rapid innovation and talent development, seeking to unite participants with complementary skills to improve project outcomes. However, this domain faces a significant challenge: high project failure rates, with statistics showing that over 15% of crowdsourced software projects fail due to inadequate team formation and a lack of adaptive matching.The core issue lies in the fact that manual or static team formation processes fail to consider real-time skill profiles, evolving event requirements, and adaptive team dynamics, resulting in mismatched teams and reduced project success. This project, titled Advanced Algorithms and Implementations for Talent Matching and Team Formation in Crowdsourced Engineering, directly addresses this problem with the aim of developing an automated platform that optimizes talent matching and team formation for these events. Our novel approach utilizes a Hybrid Machine Learning (ML) pipeline, specifically employing KeyBERT for contextualized skill extraction and K-Means Clustering for optimal team grouping, combined with real- time API integration from social and event platforms. This methodology provides a fully automated, data-driven solution that is currently unavailable in existing platforms, the enhancement of team balance and fairness, and the resulting improvement in overall success rates and participant experience in this dynamic field.

2. LITERATURE SURVEY

[1] This research presents a method for team assembly and developer recommendation utilizing **KeyBERT for skill embeddings** and fuzzy logic for task matching. The approach offers a framework to categorize pairings as "Strong," "Average," or "Weak". However, the primary limitation is that this system is **static**, lacking **real-time API adaptation**, and it is confined to structured datasets. It does not address the dynamic nature of user relationships or evolving hackathon requirements.

[2] This paper explores the use of Graph Neural Networks (GNNs) in recommender systems to model complex relationships between users, tasks, and skills. While GNNs offer a sophisticated approach to task assignment in crowdsourcing scenarios, the model's complexity results in a high computational cost, making it unsuitable for small-scale projects like the one proposed.

[3] This foundational work utilizes Reinforcement Learning (RL) to create policies for matching, aiming to learn from simulated environments to improve decision-making over time. The major constraint here is that the RL method requires a huge amount of interaction data for training and adaptation, making it impractical for use with hackathon APIs and small datasets.

[4] This paper discusses the use of **Large Language Models (LLMs)** for skill extraction from web-scale datasets, highlighting their ability to capture semantic relationships and process dynamic data. Despite their power, LLMs are **difficult to fine-tune** for specific tasks and can often **struggle with niche engineering terms**, which limits their direct application in precise developer skillmatching.

3. OVERVIEW

1. Problem Statement and Motivation

The project addresses the significant issue of high failure rates (approximately 15.7%) in crowdsourced software projects and student hackathons.The core problem is that manual or static team formation processes fail to consider real- time skill profiles, evolving event requirements, and adaptive team dynamics.This results in mismatched teams and an absence of automated, data-driven coordination, leading to reduced project success.

2. Project Aim and Objectives

To develop an automated platform that optimizes talent matching and team formation for hackathons and crowdsourced engineering events.

- Key Objectives:
- To extract and analyze participant skills from profiles and files.
- To fetch and sync live hackathon/event data using external APIs.
- To recommend the best-matched developers and teams via ML algorithms (NLP and clustering).
- To deliver personalized dashboards, badges, and notifications for engagement.
- To ensure the system meets requirements for scalability, security, and a user-friendly interface.

3. Proposed Methodology (Hybrid ML Engine)

The platform integrates with various Hackathon and Event APIs, along with Social APIs such as GitHub, LinkedIn, and Devpost, to acquire and display live, dynamic participant and event data. It utilizes Natural Language Processing (NLP) techniques, specifically the KeyBERT algorithm, to analyze profile text and extract domain-specific skills through contextualized embeddings. To recommend well-balanced and diverse teams, K-Means Clustering is applied to the extracted skill embeddings, grouping participants based on complementary skill sets. The system is designed using a layered architecture, which flows from the Web User Interface (UI) to the API Integration Layer, then to the Machine Learning Engine, and finally to the Team Recommendation Module, where optimal team suggestions are generated.

4. Technology Stack and Implementation Details

The platform's frontend is developed using React.js (v18+), providing a dynamic and responsive user interface. The backend is powered by Node.js (v20+) and Express (v4+), ensuring efficient handling of API requests and smooth server-side operations. MongoDB (v6+) serves as the database for secure and scalable data storage. The Machine Learning Engine is implemented in Python, leveraging libraries such as KeyBERT for skill extraction and Scikit-learn for clustering and recommendation processes. In terms of non-functional requirements, the system is optimized for fast performance with a response time of under two seconds and incorporates strong security measures, including OAuth authentication and data encryption, to protect user information and maintain system integrity.

5. Expected Outcomes (Immediate and Future)

The immediate outcome of the platform is to enable faster and more effective team formation, leading to improved project outcomes and greater participant satisfaction. By leveraging intelligent matching

algorithms, the system enhances match quality, which is expected to increase the overall success rate of hackathon events. Looking ahead, future developments aim to further optimize the machine learning models by incorporating advanced techniques such as Reinforcement Learning (RL) and Graph Neural Networks (GNNs). Additionally, the platform's scope will be expanded beyond hackathons to support corporate hiring and large- scale engineering competitions, making it a versatile solution for diverse collaboration and recruitment scenarios.

4. METHODOLOGY

1. System Architecture and Layers

The system is designed with a layered architecture to ensure modularity, scalability, and real-time processing. The **API Integration Layer** is responsible for fetching and synchronizing live event and participant data from external sources. The **Machine Learning Engine** serves as the core intelligence of the platform, handling skill extraction and clustering processes. The **Team Recommendation Module** utilizes the processed data to generate optimal team match suggestions based on skill complementarity and diversity. Finally, **MongoDB** is used for data storage, maintaining user profiles, extracted skills, and hackathon-related data securely and efficiently.

2. Data Acquisition and Synchronization

This layer ensures that the system functions with the most current and accurate information, effectively overcoming the limitations of static models. **Profile Data** is collected as participants create their profiles or link existing accounts, with all information securely stored in the database. The **API Integration** mechanism enables the platform to dynamically fetch and synchronize data from external sources. **Hackathon/Event APIs** are used to gather live event details such as project themes, requirements, and deadlines, ensuring real-time relevance. Additionally, **Social APIs** from platforms like GitHub, LinkedIn, and Devpost are integrated to acquire up-to-date and diverse information on participant skills, experience, and achievements, enriching the dataset for more precise recommendations.

3. Hybrid Machine Learning Engine (Processing and Core Logic)

The ML Engine employs a two-step hybrid approach, chosen for its efficiency and suitability for small- scale projects.

A. Skill Extraction (NLP using KeyBERT)

The primary goal of this component is to accurately extract domain-specific skills and areas of expertise from unstructured text sources such as resumes and profile descriptions. The KeyBERT algorithm is employed for this purpose, as it is particularly effective in generating meaningful keyword representations from short, domain-focused text like developer profiles. Unlike traditional TF-IDF, which lacks contextual understanding, and Word2Vec, which can produce overly broad semantic associations, KeyBERT provides contextualized embeddings that capture the precise relevance of skills. The final output of this process is a vector embedding that represents each participant's core skills and interests, serving as a foundation for clustering and team recommendation.

B. Team Formation (Clustering using K-Means)

The goal of this component is to recommend well- balanced teams by grouping developers who possess complementary skill sets. The **K-Means Clustering** algorithm is employed for this purpose, as it efficiently organizes participants into clusters based on their skill embeddings. **K-Means** is chosen due to its simplicity, speed, and scalability, making it an ideal choice for small- to medium-scale projects with limited computational resources. Unlike more complex methods such as Graph Neural Networks (GNNs) or Reinforcement Learning (RL), which require extensive data and processing power, K- Means provides a practical and effective approach for generating meaningful team groupings in real time.

4. Recommendation and Interaction

After the Machine Learning Engine generates the candidate groups, the system leverages several modules to deliver results and maintain user engagement. The **Team Recommendation Module** maps the clustered groups to available hackathon projects and produces a ranked list of potential teams based on skill compatibility and project requirements. The **Dashboard Service** enhances user interaction by displaying personalized information such as badges, scores, and notifications, fostering active participation. The **Messaging & Feedback System** facilitates collaboration through chat and group communication tools while also enabling post-event feedback and ratings to support continuous

the platform employs **OAuth 2.0 authentication** and **data encryption**, securing both user accounts and API access.

5. Design and Architecture

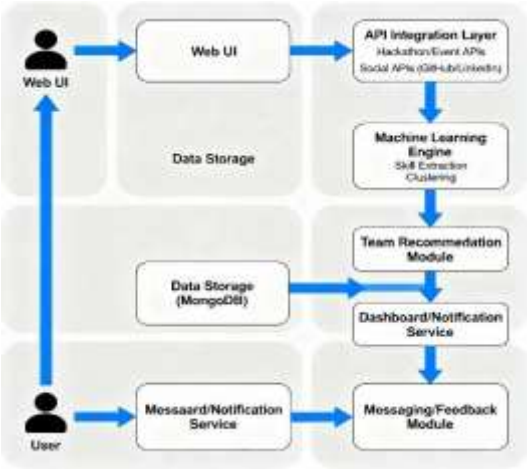

Fig 1. Architecture Of System

6. Model Training and Validation

The talent matching system is built on a **Hybrid Machine Learning (ML) pipeline** that combines Natural Language Processing (NLP) and clustering techniques to optimize team formation. For **skill extraction**, the **KeyBERT** model processes participant profile text to generate contextualized embeddings, accurately identifying relevant skills and expertise areas. During the **team formation** phase, **K-Means Clustering** groups developers into balanced teams based on the extracted skill embeddings, ensuring diversity and complementarity within each group. The **training process** involves curated datasets of developer skills and profiles, allowing the models to learn meaningful patterns. To achieve **optimal performance**, key hyperparameters—such as the learning rate for KeyBERT and the number of clusters for K-Means—are carefully tuned. Additionally, **robust validation techniques** are implemented to prevent overfitting, ensuring that the model remains generalizable and effective in small-scale, real-world scenarios.

7. User Interface Design

The platform's design focuses on creating a seamless and engaging user experience through a modern web interface built with **React.js**. The **frontend application** emphasizes **intuitive navigation**, ensuring that hackathon participants can easily explore events, view team recommendations, and manage their profiles. To foster collaboration, the system incorporates **communication pathways**, including a **Messaging and Collaboration module** and a **Feedback System** that enable smooth interaction among team members. Additionally, the platform enhances user engagement through a **Dashboard** that displays personalized elements such as badges, scores, and notifications, promoting active participation and continuous involvement throughout hackathon activities.

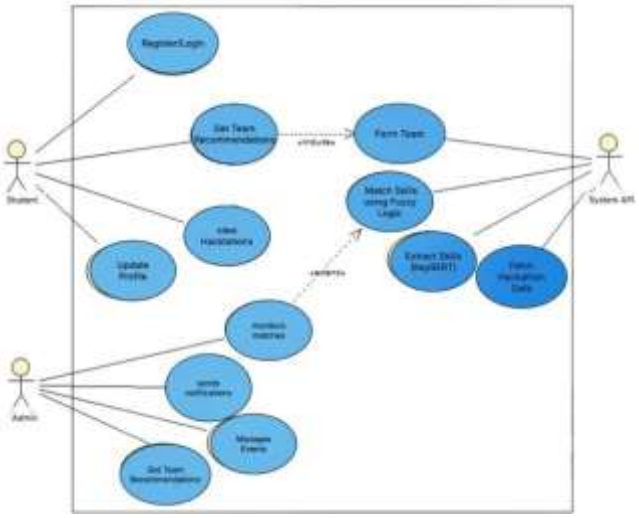

Fig 2. Use case Diagram

8. Implementation Details

The platform's backend is built using Node.js and Express, providing a robust and scalable server-side framework. The Machine Learning engine, implemented in Python with libraries like KeyBERT and Scikit-learn, handles model inference for skill extraction and team clustering. MongoDB is used as the database, securely storing user profiles, event data, and participant information. For deployment, the system leverages scalable cloud infrastructure, ensuring low latency, high availability, and smooth performance under varying loads. Security protocols such as OAuth authentication and data encryption are implemented to safeguard user profiles and API access, maintaining data privacy and integrity across the platform.

9. Performance Evaluation

The platform's performance is evaluated using multiple metrics, including match quality, project success rates, and participant satisfaction, to ensure the system effectively supports team formation. It is designed for fast performance, with a target response

time of under 2 seconds to provide a seamless user experience. Scalability testing is conducted to verify stable operation and readiness for increasing user and event loads. Additionally, the Feedback System collects post-event ratings and comments, which are analyzed to enhance future team recommendations and gauge overall user satisfaction, ensuring continuous improvement of the platform.

5. APPLICATIONS

The platform has diverse applications across educational, competitive, and professional contexts.

1. Student Hackathons and Competitions:
The core application of the platform is to automate and optimize the formation of balanced student teams for university-level and external hackathons. By ensuring that teams consist of participants with complementary skills and diverse expertise, it directly addresses challenges such as project failure due to mismatched team composition.

2. Crowdsourced Engineering Projects:
The system is highly applicable to platforms handling crowdsourced software or engineering tasks, accurately matching individual talents to project requirements. This ensures effective team formation for collaborative student competitions in fast-paced and dynamic environments.

3. Future Expansion – Corporate Hiring and Team Building:
The platform's ML algorithms, including KeyBERT for skill extraction and K-Means for clustering, can be adapted for corporate applications. It can assist in identifying candidates with the required skills for hiring and facilitate the assembly of functional, distributed remote teams.

4. Future Expansion – Large-Scale Competitions:
Designed with scalability in mind, the system can be extended to support large-scale engineering competitions and events involving high participant volumes without compromising performance or match quality.

5. Educational and Departmental Use:
University departments, such as Information Technology, can leverage the platform to form project groups, ensuring balanced workload distribution and fostering team synergy based on data-driven insights.

6. CONCLUSION AND FUTURE SCOPE

1. Conclusion

This project successfully developed and implemented TalentBridge, an automated platform that efficiently optimizes hackathon team matching and skill extraction. By leveraging a Hybrid Machine Learning (ML) pipeline (KeyBERT for NLP and K-Means Clustering) and real-time APIs, the system directly achieves its core aims and objectives. The system streamlines the complex team formation process, making it scalable and user-friendly, and enhances overall participation. The technologies clearly demonstrate the potential for improved match quality and higher success rates for hackathon projects compared to conventional static methods.

2. Future Scope

Future work will focus on enhancing the platform's capabilities and improving the performance of its underlying machine learning models. **Model optimization** will involve integrating advanced techniques such as **Reinforcement Learning (RL)** and **Graph Neural Networks (GNNs)** to enable more adaptive and accurate team formation. In addition, **new features** will be added to improve user engagement, including integration with social collaboration tools like **Slack** and **Discord**, as well as enhanced gamification elements such as badges and leaderboards.

The platform's **application scope** will also be broadened beyond student hackathons to encompass **corporate hiring**, **remote team building**, and **large- scale engineering competitions**, making it suitable for diverse collaborative environments. Finally, **scalability enhancements** will be implemented through robust cloud infrastructure, allowing the system to handle larger user bases, increased data volumes, and a wider variety of hackathon ecosystems while maintaining performance and reliability.

7. REFERENCES

[1] Yasir Munir et al., "Developer Recommendation and Team Formation in Collaborative Crowdsourcing Platforms," IEEE Access, vol. 2025, pp. xx-xx, 2025. DOI: 10.1109ACCESS.2025.3558557

[2] Aymen Hamrouni, Hakim Ghazzai, Turki Alelyani, Yehia Massoud, "Low Complexity Recruitment for Collaborative Mobile Crowdsourcing Using Graph Neural Networks," arXiv, 2021. arXiv2106.00717

[3] Various authors, "Team Formation in Software Engineering: A Systematic Mapping Study," IEEE Access, 2021.

[4] Hamrouni A., Ghazzai H., Alelyani T., Massoud Y., "A Stochastic Team Formation Approach for Collaborative Mobile Crowdsourcing," arXiv/IEEE Conf., 2020. arXiv2004.13881

[5] Xinqiang Xie, Bin Wang, Xiaochun Yang, "SoftRec Multi-Relationship Fused Software Developer Recommendation," MDPI Applied Sciences, 2020.

[6] Farhadi, Sorkhi, Hashemi, Hamzeh, "An Effective Expert Team Formation in Social Networks Based on Skill Grading," IEEE ICDMW, 2011.

[7] Liangyue Li, Hanghang Tong, et al., "Enhancing Team Composition in Professional Networks: Problem Definitions and Fast Solutions," IEEE TKDE, 2016.

[8] Brown et al., "OpenAI Language Models are Few-ShotLearners.