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

Design And Development of an AI Powered Open-Source GitHub Repository Finder

Abhishek Jagtap[1], Aditya Hood[2], Gunjan Choukade[3] , Rohit Katore[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*

Email:adityahood13@gmail.com

Abstract - The rapid progress in Artificial Intelligence (AI) and Natural Language Processing (NLP) has enabled the automation of intelligent decision-making processes across various domains, including software development. This research focuses on the design and implementation of an AI-driven Open Source GitHub Repository Finder (OSGRF), developed to assist developers in identifying open-source projects that align with their technical skills and experience. The proposed system utilizes NLP techniques to extract key skills from a user's resume and employs Sentence-BERT embeddings for semantic matching with relevant GitHub repositories. By ranking repositories according to factors such as similarity, popularity, and difficulty level, and by providing personalized AI-based contribution recommendations, the system ensures a seamless onboarding experience for developers entering the open-source community. The overall architecture integrates multiple layers built using ReactJS, Node.js, MongoDB, and Python, forming a scalable and intelligent framework for efficient open-source exploration.

Key Words **:** AI Repository Recommendation, GitHub, Open Source Contribution, Natural Language Processing (NLP), Sentence-BERT (SBERT), Semantic Matching, Cosine Similarity, Resume Skill Extraction, Large Language Models (LLMs), Automated Contribution Guidance, Repository Ranking, Machine Learning, Recommender Systems, GitHub API Integration, Developer Skill Mapping**.**

1.INTRODUCTION

Open-source software has become a fundamental aspect of modern software engineering, offering developers a collaborative environment to learn, innovate, and contribute to projects with global impact. Despite this, new contributors often struggle to identify repositories that align with their technical skills and experience. Existing GitHub search mechanisms primarily depend on keyword-based matching, which lacks the ability to interpret the semantic relationships between user skills and project requirements.

To address this limitation, the present research proposes an AI-driven system capable of understanding the contextual meaning of both skills and repository descriptions. By integrating Natural Language Processing (NLP) and Large Language Models (LLMs), the system automates repository discovery, provides intelligent contribution recommendations, and performs issue classification with improved accuracy. The motivation behind this work lies in the growing demand for streamlined contributor onboarding and the desire to harness intelligent automation to enhance engagement within the open-source ecosystem.

2. LITERATURE SURVEY

[1] This paper, *"Sequential Recommendations on GitHub Repository"* (Applied Sciences, 2021), proposes a deep learning approach for suggesting repositories based on user activity such as stars, follows, and contributions. Using models like GRU4Rec and SASRec, it predicts future repository interests by learning behavioral patterns. However, it performs poorly in cold-start

situations where users lack prior activity and does not leverage resume-based skills or semantic understanding for personalized recommendations.

[2] This study explores the use of artificial intelligence and machine learning to automate various aspects of GitHub repository management, including issue classification, documentation generation, and pull request handling with large language models such as LLaMA. While it enhances repository maintenance and workflow efficiency, it lacks features that help new developers discover relevant repositories to contribute to, as its focus remains on internal automation rather than personalized recommendations.

[3] A machine learning-based system was introduced in 2022 to automatically identify "Good First Issues" (GFIs) in GitHub repositories by analyzing features such as issue content, reporter behavior, and project activity. Using an XGBoost classifier, the model effectively predicts beginner-friendly issues with a strong accuracy score. However, its scope is limited to general issue classification and does not account for individual developer skills or semantic matching. In contrast, the proposed system advances this concept by utilizing Sentence-BERT embeddings for skill-based repository matching, integrating large language models for contribution guidance, and delivering real-time, personalized recommendations with automated pull request generation.

[4] This paper presents a machine learning-based method for identifying "Good First Issues" (GFIs) in GitHub repositories using features like issue content, reporter activity, and project engagement. An XGBoost classifier predicts beginner-friendly issues effectively but lacks personalization or semantic understanding of developer skills. Building on this, the proposed system enhances the approach by applying Sentence-BERT embeddings for skill-based repository matching, integrating LLMs for contribution guidance, and offering real-time, user-focused recommendations with automated pull request generation.

3.OVERVIEW

The AI-Powered Open Source GitHub Repository Finder (OSGRF) is designed to intelligently bridge the gap between developers and the open-source community. It utilizes data-driven insights and AI-based automation to streamline the process of discovering open-source projects that align with a developer's skill set and interests. By simplifying project discovery and contribution matching, OSGRF promotes a culture of collaboration, knowledge sharing, and continuous learning within the global developer ecosystem.

1.Objective:

The primary objective of the Open Source GitHub Repository Finder (OSGRF) is to develop a seamless and intelligent recommendation system that connects developers with open-source repositories matching their genuine skills and interests. Through the automation of skill extraction, repository discovery, and AI-driven contribution guidance, the system aims to minimize the challenges faced by newcomers when initiating open-source contributions. Ultimately, OSGRF encourages skill-oriented community growth and fosters a more inclusive and efficient open-source collaboration environment.

2.Core Components:

The Open Source GitHub Repository Finder (OSGRF) consists of five key components that work together to provide intelligent and personalized recommendations:

- Resume Analysis Engine: Uses NLP techniques to extract technical skills from a developer's resume.
- GitHub Integration Module: Connects to the GitHub API to fetch and update repository data.
- Semantic Similarity Engine: Applies Sentence-BERT embeddings to match user skills with repository descriptions.
- AI-Powered Guide Generator: Provides context about repositories and step-by-step contribution suggestions.
- User Interface Module: Displays ranked recommendations and allows users to explore results through interactive filters.

Together, these components create a multi-layered system capable of delivering real-time intelligent suggestions and adaptive learning based on user behavior and profile data.

3.Working Principle:

The system operates through the integration of **Natural Language Processing (NLP)**, **semantic similarity scoring**, and **AI-based content generation**. After a user logs in via GitHub

and uploads their resume, the system extracts relevant technical skills and converts them into **vector embeddings**. These embeddings are then matched with open-source repositories retrieved from GitHub. Using **cosine similarity**, the system ranks repositories based on **similarity**, **popularity**, and **recency**. Finally, an **LLM-powered module** generates concise summaries and contribution guidelines, helping users quickly understand project contexts and initiate meaningful contributions.

4. Accessibility and Innovation:

The system features an intuitive web interface that enables developers from diverse backgrounds to explore open-source projects without extensive manual searching or technical analysis. Its key innovation lies in combining NLP-based skill extraction with LLM-generated contribution guidance, forming a fully automated recommendation pipeline. By accommodating various skill levels and offering filtered task suggestions, OSGRF effectively reduces entry barriers and promotes greater participation among students and developers worldwide.

5.Social Impact:

Beyond its technical capabilities, the OSGRF project fosters community-driven innovation within the open-source ecosystem. It promotes inclusivity by guiding new developers toward suitable mentorship opportunities and collaborative projects. Through intelligent matchmaking, the system facilitates skill sharing, accelerates project development, and enhances knowledge exchange among contributors. Over time, such interactions can significantly boost global participation in open-source initiatives, leading to a more connected, collaborative, and sustainable developer community.

4.METHODOLOGY

A. System Architecture and Layered Design

The Open Source GitHub Repository Finder (OSGRF) is structured around a four-tier modular architecture to ensure scalability, maintainability, and efficient component interaction.

- Presentation Layer: A responsive web interface developed using ReactJS and TailwindCSS, providing users with a seamless experience for authentication, resume upload, and repository exploration.
- Application Layer: Built with Node.js and ExpressJS, this middleware manages OAuth authentication, API routing, and communication with the GitHub API.
- AI/NLP Layer: Implemented in Python, this engine integrates spaCy for skill extraction, Sentence-BERT for generating semantic embeddings, and LLMs for contribution guide generation.
- Data Layer: Utilizes MongoDB for efficient data storage, indexing, and caching of user profiles, repository data, and generated outputs.

B. Data Acquisition and Synchronization

The OSGRF system acquires data through three parallel pipelines designed to ensure accuracy, scalability, and continuous synchronization. The first pipeline focuses on user data, which is collected through GitHub OAuth authentication. This process securely retrieves verified details such as the user's name, email, and publicly available repositories. The second pipeline handles resume data, where spaCy's Named Entity Recognition (NER) model is employed to extract relevant technical skills. To enhance precision, custom labels are trained to identify domain-specific entities such as programming languages, frameworks, and software tools, enabling an accurate correlation between a developer's expertise and potential repository topics. The third pipeline manages repository data acquisition through the GitHub REST API, gathering metadata including repository descriptions, star counts, and recent activity levels. This information is stored and periodically refreshed to maintain relevance. To support real-time responsiveness, the system incorporates asynchronous data synchronization, ensuring that updates occur seamlessly without interrupting active user sessions.

C. Processing and Core Logic

1)Skill Extraction

Once the user uploads a resume, the text is cleaned and parsed to remove unnecessary elements. The spaCy Named Entity Recognition (NER) model is then used to identify and extract skill-related entities such as programming languages, frameworks, and tools. These extracted skills are converted into vector embeddings using Sentence-BERT, enabling the system to represent them semantically for accurate similarity analysis.

2) Repository Collection

Based on the extracted skills, the system retrieves relevant repositories from GitHub using the REST API. The search focuses on language tags and project topics related to the user's skills. For each repository, details such as the description, open issues, and recent activity are collected and stored in MongoDB for further ranking and comparison.

3) Recommendation and Interaction

The system computes cosine similarity between the user's skill embeddings and the repository embeddings to assess how closely they match. A weighted ranking mechanism then generates a prioritized list of repositories by considering factors like similarity score, star count, and recent commit activity. When a user selects a repository, an LLM-based module generates a concise summary along with beginner-friendly issue recommendations. The system also tracks user interactions to improve future recommendations through adaptive learning.

D. Design and Architecture

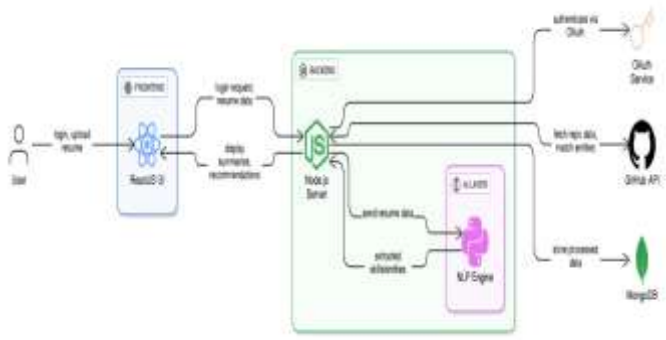

Fig 1. Architecture of System

E. Model Training and Validation

The proposed Sentence-BERT model will be fine-tuned using a curated dataset of resume excerpts and their corresponding GitHub repositories to enhance the accuracy of semantic matching. A text classification model will also be developed using labeled GitHub issues categorized as Easy, Medium, and Hard, supporting skill-appropriate task recommendations.

The system's performance will be evaluated using Mean Squared Error (MSE) for embedding validation and F1-score for classification accuracy. Additional parameters such as cosine similarity, classification precision, and API response latency will be analyzed to assess the efficiency of the model. Future evaluations will include manual validation by developers to verify the relevancy and accuracy of the recommended repositories once the implementation is complete.

F. User Interface Design

The proposed system interface is designed to emphasize clarity, responsiveness, and interactivity. It will include three primary pages:

- Login Page: Implements GitHub OAuth authentication to ensure secure and seamless user access.
- Dashboard: Dynamically displays extracted skills and recommended repositories based on the user's profile.
- Repository Detail View: Presents concise project summaries, AI-generated contribution guides, and issue filters to help users identify suitable tasks.

A responsive design approach will be followed to ensure compatibility across various devices. The use of TailwindCSS will maintain a consistent visual appearance, while React Hooks will efficiently manage dynamic rendering and application state for a smooth user experience.

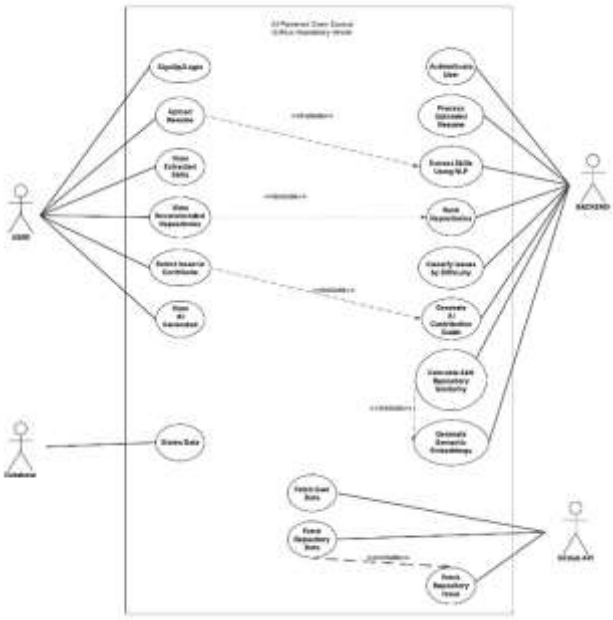

Fig 2. Use Case Diagram

G. Implementation Details

The system implementation will be carried out in clearly defined stages to ensure modular development and smooth integration. The process will begin with GitHub-based user authentication, where users will log in through OAuth, enabling secure and seamless access to their public repositories. Authentication

tokens will allow data retrieval while keeping sensitive credentials protected.

The frontend will be developed using ReactJS, providing a responsive and interactive interface capable of real-time dynamic updates. The backend will utilize Node.js with Express.js to manage routing, handle API communication, and serve as the middleware connecting the frontend and AI components. The AI Layer, built in Python, will integrate tools such as spaCy for entity extraction, Sentence-BERT for generating semantic embeddings, and OpenAI's language models for producing natural-language guidance and summaries.

For data management, MongoDB Atlas will serve as the cloud-hosted database solution, ensuring scalability, availability, and efficient querying. Future deployment plans include hosting the backend on platforms like Render or Heroku, and the frontend on Netlify, maintaining a clear separation of concerns for easier maintenance and updates.

All communication between components will occur securely over encrypted HTTPS connections, with data exchanged via RESTful APIs to ensure consistency and reliability.

H. Performance Evaluation

Once the system prototype is developed, performance testing will be conducted to assess its efficiency and reliability. The evaluation will primarily focus on response time, skill extraction accuracy, repository ranking relevancy, and model scalability across multiple user sessions.

To measure real-world performance, load simulation tests will be performed by gradually increasing concurrent user requests and monitoring the system's stability and throughput. Preliminary targets include achieving an average API response time below 2.5 seconds and maintaining high recall in semantic similarity matching between extracted skills and repositories.

Insights obtained from these tests will guide further optimization, including caching strategies, asynchronous API handling, and enhancements in embedding retrieval efficiency to improve overall responsiveness and scalability of the system.

5. APPLICATIONS

The OSGRF system demonstrates wide-ranging applicability across educational, professional, and open-source domains.

In **educational settings**, it can assist students in identifying live open-source projects that align with their recently acquired skills. This promotes hands-on, experiential learning and encourages early participation in real-world development environments.

For **recruitment and talent platforms**, OSGRF can be adapted to evaluate and recommend candidates based on their verified GitHub activities. Organizations may also employ it to locate developers proficient in specific technologies, streamlining the hiring process.

Within **open-source communities**, the system offers an efficient onboarding tool for newcomers. By automatically classifying and recommending beginner-friendly issues, it helps maintainers guide fresh contributors and sustain project momentum.

In the **freelancing and upskilling ecosystem**, OSGRF can support professionals and learners in selecting projects that build credible public portfolios, thereby strengthening their visibility, technical expertise, and employability in competitive markets.

6. CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The AI-Powered Open Source GitHub Repository Finder (OSGRF) represents a forward-looking initiative designed to simplify and enhance open-source contributions through the integration of artificial intelligence. By combining natural language processing, semantic embeddings, and generative AI, the system aims to automate key stages such as skill analysis, repository discovery, and contribution guidance.

Once fully implemented, OSGRF will significantly lower the entry barrier for new contributors, foster global collaboration, and accelerate individual developer growth. Ultimately, it aspires to strengthen the culture of shared learning and innovation within the open-source ecosystem

6.2 Future Scope

The future development of the OSGRF system envisions expanding its capabilities beyond individual recommendations to foster collaborative engagement and broader platform integration.

A Collaborative Contribution Recommendation feature can be introduced to enable users to form teams and receive project suggestions that align with their combined technical skills. This would encourage teamwork and collective problem-solving within the open-source ecosystem.

Additionally, cross-platform expansion is proposed to extend support to other major code-hosting platforms such as GitLab, Bitbucket, and SourceForge. Such integration will provide users with wider coverage and more diverse contribution opportunities, enhancing the system's adaptability and overall impact.

7. REFERENCES

[1] J. Kim, J. Wi, and Y. Kim, "Sequential Recommendations on GitHub Repository," Applied Sciences, vol. 11, no. 4, p. 1585, 2021. DOI: 10.3390/app11041585.

[2] L. Barhate, P. Chhajed, and S. Vibhute, "AI Powered GitHub Project Management," International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS), vol. 07, no. 06, pp. 1–6, Jun. 2025. DOI: 10.56726/IRJMETS78747.

[3] W. Xiao, Y. Sun, D. Lo, and J. Jiang, "RecGFI: Recommending Good First Issues in GitHub OSS Projects," Proceedings of the IEEE/ACM 44th International Conference on Software Engineering (ICSE), pp. 1–12, 2022.

[4] G. Mitrov, B. Stanoev, S. Gievska, G. Mirceva, and E. Zdravevski, "Combining Semantic Matching, Word Embeddings, Transformers, and LLMs for Enhanced Document Ranking: Application in Systematic Reviews," Big Data and Cognitive Computing, vol. 8, no. 9, p. 110, 2024. DOI: 10.3390/bdcc8090110.