Indian Journal of Computer Science and Technology
Volume 5, Issue2 (May-August 2026), PP: 52-57.
www.indjcst.com
ISSN No: 2583-5300

Autonomous AI Hiring Agent Using LLM, Machine Learning and Graph-Based Recommendation

Bandaru Siva[#1], Giri Kollati[#2], Bodhanam Dastha Giri[#3], G. Sasi Kumar[#4]
[#1,2,3]UG Student, Department of Computer Science and Engineering, School of Engineering and Technology, Dhanalakshmi Srinivasan University, Trichy, Tamilnadu, India.
[#4]Assistant Professor, Department of Computer Science and Engineering, Dhanalakshmi Srinivasan Institute of Technology, Trichy, Tamil Nadu, India.

To Cite this Article: *Bandaru Siva[#1], Giri Kollati[#2], Bodhanam Dastha Giri[#3], G. Sasi Kumar[#4], "Autonomous AI Hiring Agent Using LLM, Machine Learning and Graph-Based Recommendation", Indian Journal of Computer Science and Technology, Volume 05, Issue 02 (May-August 2026), PP: 52-57.*

Abstract: *The rapid growth of digital recruitment platforms has resulted in a massive volume of resumes for each job opening, making manual candidate screening time-consuming, inconsistent and difficult to scale. Automated hiring tools are increasingly required to help organizations shortlist suitable candidates quickly while maintaining fairness, transparency and alignment with job requirements. This paper presents an autonomous AI hiring agent that combines natural language processing, Large Language Models (LLMs), classical machine learning and graph-based modeling to support data-driven hiring decisions. The proposed system automatically parses resumes, extracts relevant technical skills and experience, and represents both candidates and job descriptions as structured feature vectors. A TF-IDF and cosine-similarity based matching engine is used to compute relevance scores between each candidate and multiple job roles, which are then represented in a candidate–job graph where nodes correspond to entities and weighted edges indicate match strength. On top of this graph, decision rules and configurable thresholds categorize candidates into levels such as strong hire, consider or reject, thereby imitating the behavior of an intelligent recruiter or modern Applicant Tracking System. The system is implemented using Python, scikit-learn, NetworkX and a Streamlit-based interface, enabling interactive visualization of matches and easy interpretation for HR users. The architecture is modular and can be extended with domain-specific LLMs, knowledge-graph integrations and real-time labour-market data, making it suitable for academic study as well as practical use in AI-driven recruitment automation.*

Key Words: *Autonomous hiring agent, resume screening, Large Language Model, machine learning, TF-IDF, cosine similarity, skills graph, recruitment automation.*

I. INTRODUCTION

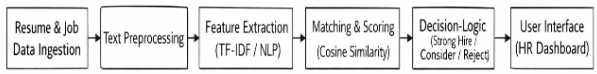
Recruitment has become increasingly complex as organizations receive a large number of applications for each vacancy through online job portals, social media and referral platforms. Hiring teams are expected to identify qualified candidates quickly while considering technical skills, experience, cultural fit and role expectations within limited time. Manual resume screening is often repetitive and subjective, and important candidate information can be overlooked when large volumes of unstructured text must be reviewed under time pressure. As a result, companies are looking for intelligent tools that can assist human recruiters by organizing candidate information, highlighting relevant profiles and supporting consistent decision-making across different job roles.

Traditional Applicant Tracking Systems (ATS) mainly rely on keyword matching and basic filters such as years of experience, education level or fixed skill lists. While these systems help to reduce the initial workload, they struggle with semantic understanding, varied resume formats and domain-specific terminology. Candidates who use different wording for the same skill may be incorrectly ranked lower, and resumes that are rich in context but poor in keywords may not be detected by simple rule-based engines. Recent advances in natural language processing and Large Language Models (LLMs) have shown that unstructured text can be analyzed more deeply, allowing systems to infer skills, responsibilities and domain knowledge beyond explicit keyword overlaps. However, many AI-based tools are still limited to isolated scoring functions and do not explicitly capture the relationships between multiple candidates and multiple job descriptions in a structured manner.

In this work, an autonomous AI hiring agent is proposed to address these limitations by combining LLM-based text understanding, classical machine learning techniques and graph-based modeling. The system first extracts text from candidate resumes and job descriptions, then applies natural language processing methods to identify skills, technologies and role-specific terms that characterize each profile. These profiles are transformed into feature representations and compared using TF-IDF and cosine similarity to obtain match scores between candidates and available positions. The resulting scores are modeled as a candidate–job graph where nodes represent candidates and jobs, and edges represent the strength of their match, which enables more flexible analysis and visualization of the recruitment landscape. On top of this graph, decision rules classify candidates into

categories such as strong hire, consider or reject, thereby simulating the behavior of an intelligent assistant that can support human recruiters in making faster and more consistent hiring decisions.

II.SYSTEM DEELOPMENT FRAMEWORK



System Development Framework of the **Autonomous AI Hiring Agent**

- The development of an autonomous AI hiring agent requires a structured framework that connects recruitment requirements, data processing components and decision-making logic into a single coherent system. In the initial phase, the functional requirements of the hiring workflow are identified, such as resume ingestion, job description management, skill extraction, candidate–job matching, ranking and recommendation presentation to the recruiter. These requirements are gathered from common hiring practices where HR teams manually download resumes, scan them for key skills and experience, compare them with job descriptions and then prepare a shortlist of suitable candidates for interview. Once these needs are clearly defined, they are translated into software modules that can automate or assist each step in a reliable and repeatable way.
- The overall framework is organized into multiple stages that mirror the life cycle of recruitment data within the system. At the data acquisition stage, resumes in formats such as PDF and job descriptions stored in tabular or text files are collected and stored in a structured repository. The preprocessing stage handles tasks such as text extraction, lowercasing, stop-word removal and basic normalization to ensure that the downstream models receive consistent input. In the feature construction stage, natural language processing and vectorization techniques, such as TF-IDF, are applied to represent both candidate and job texts as numerical vectors in a common feature space. These vectors serve as the basis for computing similarity scores and building the candidate–job graph.
- On top of the feature layer, the framework defines a matching and scoring stage, a graph-construction stage and a decision-logic stage. In the matching stage, similarity measures such as cosine similarity are used to estimate how closely each candidate aligns with each job description, producing a matrix of scores across all candidate–job pairs. The graph-construction stage then interprets these scores as weighted edges in a bipartite graph that connects candidate nodes and job nodes, allowing the system to reason about the overall structure of matches and highlight strong connections. Finally, at the decision-logic stage, rule-based thresholds and configurable policies classify candidates into categories such as strong hire, consider or reject, and generate human-readable recommendations that can be reviewed by recruiters inside a user interface. This stepwise framework ensures that the AI hiring agent remains transparent, modular and easier to extend with more advanced LLM-based components in future work.

III.EXISTING SYSTEM OF AI-BASED HIRING

- In many organizations, the existing hiring process is supported by basic Applicant Tracking Systems that mainly focus on storing candidate profiles, filtering applications by predefined criteria and searching resumes using simple keyword queries. These systems can automatically reject candidates who do not meet hard constraints, such as required degree or minimum years of experience, and they help recruiters manage large volumes of applications by organizing them into stages such as applied, shortlisted and interviewed. However, the underlying logic is usually rule-based and dependent on exact keyword matches, which limits the system's ability to understand the context of skills, synonyms or transferable experience described in natural language on resumes and job descriptions.
- Several AI-assisted tools for resume screening and candidate matching have emerged that use natural language processing to extract entities such as skills, job titles, organizations and durations from unstructured text. These tools typically compute

similarity scores between candidate profiles and job descriptions and then rank candidates based on these scores to support shortlisting. While such systems improve ranking quality compared to purely manual screening, they often operate as black-box scoring engines, provide limited transparency about why a candidate was rated highly or poorly, and rarely offer a global view of how multiple candidates relate to multiple job roles in a recruitment campaign. In many cases, AI components are plugged into existing ATS platforms without a clear model of interactions between candidates, roles and required skills.

- Another limitation of current systems is the lack of integrated representation of relationships between candidates and jobs over time. Most platforms treat each application as an independent record and do not construct explicit structures that show shared skills, competing candidates for the same role or potential fit of a candidate across different openings. As a result, hiring teams may miss opportunities to reassign candidates to alternative roles for which they are better suited, and they may find it difficult to visualize the overall talent pool. Existing solutions also struggle to combine different AI techniques, such as LLM-based understanding, classical machine learning and graph-based reasoning, into a unified framework that can provide both strong performance and human-interpretable recommendations.

IV.PROPOSED AUTONOMOUS AI HIRING AGENT

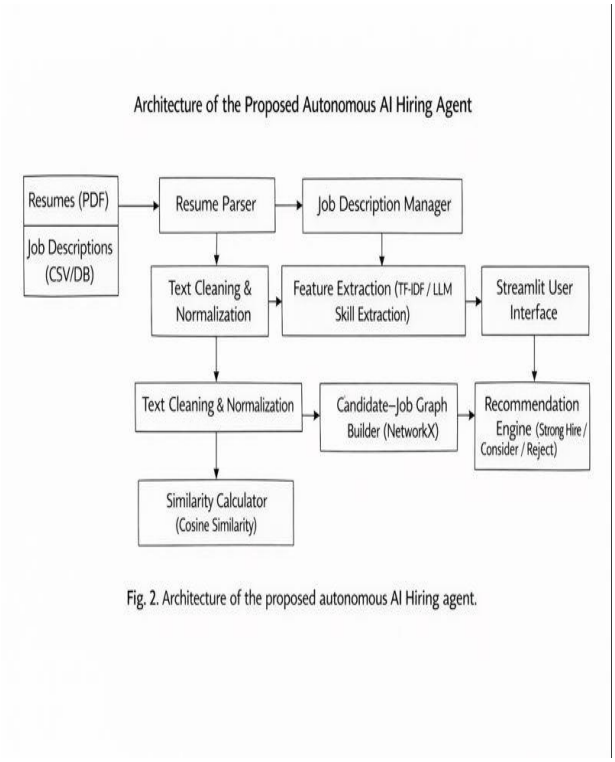


Fig. 2. Architecture of the proposed autonomous AI Hiring agent.

- The proposed system is designed as an autonomous AI hiring agent that supports recruiters throughout the candidate evaluation process by integrating resume analysis, job understanding, similarity computation, graph modeling and recommendation generation into a single pipeline. The main objective is not to replace human judgment but to provide a structured, data-driven assistant that can highlight the most relevant candidates for each role and present clear reasoning for its suggestions. The system follows a modular architecture in which each stage has a well-defined responsibility, making it easier to implement, test and extend according to the specific needs of different organizations.
- At a high level, the proposed architecture begins with a data ingestion module that collects resumes and job descriptions from predefined sources. Resumes are typically uploaded as PDF documents, while job descriptions can be stored in a spreadsheet or database containing job titles and corresponding descriptions. A text extraction module converts the PDF resumes into plain text, which is then cleaned and normalized. In parallel, the job descriptions are preprocessed to remove unnecessary symbols and stop-words so that both candidate and job texts share a similar representation format before further analysis.
- The next part of the architecture is responsible for representing the textual information as numerical features suitable for machine learning. A natural language processing layer identifies important terms related to skills, technologies and roles, and a TF-IDF vectorizer transforms the processed text into feature vectors in a common space. Using these vectors, the system computes cosine similarity scores between each candidate resume and each job description, resulting in a set of match scores that reflect how closely the candidate's profile aligns with the requirements of each role. These scores form the basis for both ranking and graph construction.
- To better capture relationships across the entire recruitment scenario, the system models candidates and jobs as nodes in a bipartite graph where edges are weighted by the corresponding similarity scores. This graph-based representation allows the autonomous agent to view the recruitment process as a network, where strong edges indicate good fits and weaker edges indicate marginal suitability. On top of this graph, a decision module applies configurable rules and thresholds to assign labels such as strong hire, consider or reject to each candidate–job pair. The final output is presented to the recruiter through a simple user

interface, for example using a Streamlit application, which displays the best-matched job for a given resume, the associated similarity score and a list of alternative roles along with their scores. This combination of text analysis, numerical scoring and graph-based reasoning forms the core of the proposed autonomous AI hiring agent and provides a flexible foundation for integrating more advanced LLM-based components in future versions.

V.SYSTEM MODULES OF THE AUTONOMOUS AI HIRING AGENT

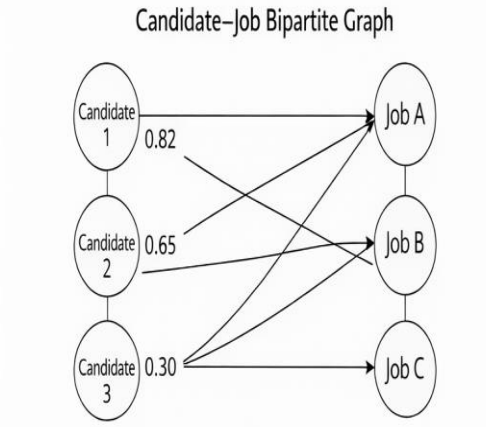


Fig. 3. Candidate–job bipartite graph based on similarity scores.

- The autonomous AI hiring agent is organized into several interconnected modules, each responsible for a specific function in the recruitment pipeline. This modular structure makes the system easier to understand, implement and maintain, and allows individual components to be improved or replaced without changing the overall design. The major modules include the Resume Parser, Job Description Manager, Feature Extraction and Skill Representation, Matching Engine, Graph Builder and Recommendation Engine.
- The Resume Parser module handles the ingestion and processing of candidate resumes. Resumes uploaded in PDF format are read using a PDF processing library, and the textual content is extracted page by page. The extracted text is then converted to a uniform format by applying operations such as lowercasing, removal of special characters and basic tokenization. This module ensures that resumes with different layouts and templates are converted into comparable text representations that can be used by downstream components.
- The Job Description Manager module stores and manages information about available positions, such as job titles and detailed descriptions of required skills and responsibilities. In a simple implementation, these descriptions are maintained in a CSV file or database table where each row corresponds to a specific role. The text of each job description is cleaned and normalized in a manner similar to the resume text so that both types of documents can be processed by the same feature extraction techniques. This module provides a consistent view of the job space that the system will use for matching.
- The Feature Extraction and Skill Representation module converts the processed text into numerical vectors suitable for similarity computation. A TF-IDF vectorizer is applied to the combined corpus of resume text and job descriptions, producing feature vectors that capture the importance of terms relative to all documents in the dataset. These vectors implicitly encode skills, technologies and domain concepts appearing in resumes and job descriptions. In more advanced versions, this module can be extended with LLM-based skill extraction, where an LLM is prompted to explicitly list technical skills, which are then mapped into a structured skill space for refined matching.
- The Matching Engine module computes similarity scores between candidates and jobs using the feature vectors produced in the previous stage. Cosine similarity is used as a measure of how closely the direction of the candidate vector aligns with the job vector, where higher values indicate stronger matches. For each candidate, the engine calculates scores against all job descriptions and stores the results in a matrix or table. This module produces the numerical evidence that will be used later to rank candidates and build the candidate–job graph.
- The Graph Builder module interprets the similarity scores as edges in a bipartite graph that connects candidate nodes and job nodes. Each candidate is represented as a node on one side of the graph, each job as a node on the other side, and edges are added with weights equal to the corresponding similarity scores. Graph libraries such as NetworkX can be used to construct and analyze this network, providing a visual and analytical view of how candidates relate to different roles. This structure makes it easier to identify strong connections, overlapping opportunities and potential alternative roles for a given candidate.

- Finally, the Recommendation Engine module applies decision rules on top of the graph and similarity scores to generate human-readable hiring suggestions. Thresholds are defined to categorize matches into levels such as strong hire for high scores, consider for medium scores and reject for low scores. The module selects the best-matched job for each candidate and can also present a ranked list of alternative jobs with their corresponding scores. The output is displayed through a user interface, for example using a Streamlit application, where recruiters can upload resumes, view recommended roles, examine scores and understand the reasoning behind the system's decisions in a clear and transparent way.

VI.FUTURE EVOLUATION OF THE AUTONOMOUS AI HIRING AGENT

- The autonomous AI hiring agent described in this work represents an initial step towards more intelligent and transparent recruitment systems, but there are several directions in which the architecture can be extended and refined. One important area is the integration of advanced Large Language Models that can better understand complex resume narratives, summarize candidate experience and infer soft skills such as communication or leadership from descriptive text. By incorporating fine-tuned LLMs, the system could move beyond surface-level term matching and capture deeper semantic relationships between candidate backgrounds and job expectations, especially for roles that require diverse and evolving skill sets.
- Another direction for future evolution is the development of richer skill and knowledge graphs that connect technologies, roles, industries and learning paths. At present, the system models candidates and jobs in a bipartite graph based on similarity scores, but this could be extended into a multi-layer graph where skills and concepts also appear as nodes. Such a graph would allow the system to reason about related skills, emerging technologies and alternative career routes, and to suggest training or upskilling recommendations for candidates who are close to meeting job requirements. This approach would make the hiring agent not only a screening tool but also a guidance tool for both recruiters and applicants.
- The evolution of the system can also include the incorporation of real-time labour-market data and feedback signals from actual hiring decisions. By connecting to job portals and market analytics, the agent could adjust its weighting of skills and experience according to current demand trends, making recommendations more aligned with industry needs. Similarly, feedback from recruiters, such as which candidates were interviewed or eventually hired, can be used to retrain and calibrate the matching models over time, improving accuracy and reducing bias. These feedback loops would enable the system to become more adaptive and robust in dynamic hiring environments.
- In addition, future versions of the autonomous AI hiring agent should address fairness, transparency and regulatory considerations. Techniques such as bias detection in training data, explanation of model decisions and controlled use of sensitive attributes are necessary to ensure that the system supports equitable hiring practices and complies with emerging guidelines for AI in recruitment. Providing interpretable explanations, such as which skills and experiences contributed most to a candidate's score, will help build trust among HR professionals and candidates and encourage responsible deployment of AI in the hiring process.

VII.FUTURE RESEARCH

- Future research on autonomous AI hiring agents can explore more sophisticated combinations of LLMs, classical machine learning and graph-based reasoning for candidate evaluation. One promising direction is the design of multi-agent systems in which different AI agents specialize in tasks such as skill extraction, cultural-fit assessment and salary recommendation, and then collaborate to produce a final hiring suggestion. Coordinating these agents through shared representations and protocols could lead to more nuanced and flexible decision-making compared to a single monolithic model.
- Another research area involves the development of standardized benchmarks and datasets for evaluating AI hiring systems. At present, many recruitment solutions are tested on proprietary or limited datasets, making it difficult to compare performance fairly or to understand how models behave across different industries and job levels. Creating anonymized, representative datasets with labeled outcomes such as interview selection and hiring decisions would enable systematic evaluation of matching quality, bias and robustness, and would support the comparison of different modeling approaches.
- Explainability and fairness also remain important topics for future investigation. Methods for generating human-understandable explanations of candidate scores, such as highlighting influential skills, experiences and sections of the resume, can help HR professionals verify and trust AI recommendations. Research into fairness-aware learning algorithms and de-biasing techniques is needed to reduce the risk that historical biases in recruitment data are reproduced or amplified by the system. These approaches should be complemented by user studies with recruiters and candidates to understand how explanations and fairness controls are perceived in real hiring contexts.

VIII.CONCLUSION

- The autonomous AI hiring agent presented in this work demonstrates how natural language processing, classical machine learning and graph-based modeling can be combined to support modern recruitment processes. By automatically parsing resumes and job descriptions, transforming them into numerical feature representations and computing similarity scores, the system is able to highlight candidates whose profiles align closely with specific job roles in a transparent and systematic way. The use of a candidate–job graph provides an additional structural view of the hiring landscape, enabling the identification of strong matches and alternative opportunities while maintaining an interpretable representation for recruiters.
- The modular design, which includes components such as the Resume Parser, Job Description Manager, Feature Extraction module, Matching Engine, Graph Builder and Recommendation Engine, makes the system practical to implement and extend. A simple user interface allows HR professionals to upload resumes, inspect similarity scores and understand the reasoning behind hiring suggestions, thereby positioning the agent as an assistant rather than a replacement for human decision-makers.

Although the current prototype relies primarily on TF-IDF and cosine similarity, the architecture is well suited for integration with more advanced LLM-based skill extraction and domain-specific knowledge graphs in future work.

- As recruitment practices continue to evolve and the volume of candidate data grows, intelligent tools that can provide consistent, fair and explainable assistance will become increasingly important. The concepts and framework discussed in this paper contribute to ongoing efforts to design AI-driven hiring systems that are both technically effective and aligned with organizational and ethical requirements. Further research on multi-agent architectures, standardized evaluation datasets and fairness-aware learning methods can help transform such autonomous hiring agents into reliable components of next-generation talent acquisition platforms.

REFERENCES

1. M. C. Ng and J. Chan, "Resume Screening with Natural Language Processing," *International Journal of Computer Applications*, vol. 187, no. 5, pp. 1–7, 2024.
2. Coresignal, "How to Build an AI Candidate Matching Agent?," 2025. [Online]. Available: https://coresignal.com/blog/candidate-matching/
3. S. Sharma, "AI Resume Filtering Agent," Lyzr, 2025. [Online]. Available: https://www.lyzr.ai/blueprints/hr/ai-resume-filtering-agent/
4. "Resume Screening AI Agent: Automate Hiring & Improve Shortlisting," AdvansAppz, 2025. [Online]. Available: https://advansappz.com/resume-screening-ai-agent/
5. "Beyond the CV: The Global Rise of AI-Powered Skills Graphs in Hiring," VirtualEmployee, 2025. [Online]. Available: https://www.virtualemployee.com/blog/beyond-the-cv-the-global-rise-of-ai-powered-skills-graphs-in-hiring
6. J. B. Jibin et al., "AI-Resume-Matcher: An AI-powered Resume Matching System using TF-IDF and Cosine Similarity," GitHub repository, 2025. [Online]. Available: https://github.com/jibz33on/AI-Resume-Matcher
7. D. Kuvvetli, "Smart Resume Screening using TF-IDF | NLP Python Project," YouTube, 2025. [Online]. Available: https://www.youtube.com/watch?v=-VIkOgzXRxI
8. R. K. Singh and P. Gupta, "Resume Ranker Powered by LLM," *International Journal of Novel Research and Development*, vol. 10, no. 6, pp. 1–6, 2025.
9. S. Kumar et al., "Personalized Job Search with AI: A Recommendation System Integrating Real-Time Data and Skill-Based Matching," *International Journal of Engineering Research & Technology*, 2025.
10. Lee and M. Kim, "Application of LLM Agents in Recruitment," arXiv preprint arXiv:2401.08315, 2024