ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AI-POWERED RESUME SCREENING SYSTEM FOR SMART HIRING: LEVERAGING NLP AND LARGE LANGUAGE MODELS FOR EFFICIENT AND FAIR RECRUITMENT

[1]ANGEL HEPZIBAH R, [2]JEYASANTHI J, [3]DURGA DEVI G, [4]MARIAPPAN E, [5]KALIAPPAN M, [6]PREETHY REBECCA P, [7]SENTHI S, [8]RAMNATH M

[1]Assistant Professor, Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India.
[2]Associate Professor, Department of Electrical and Electronics Engineering, Sethu Institute of Technology, Kariapatti, Tamil Nadu, India.
[3]Assistant Professor, Department of Artificial Intelligence and Data Science, SriEshwar College of Engineering, Coimbatore, Tamil Nadu, India.
[4]Associate Professor, Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India.
[5]Professor, Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India.
[6]Associate Professor, Department of Computer Science and Engineering, St. Peter's College of Engineering and Technology, Chennai, Tamil Nadu, India.
[7]Assistant Professor, Department of Computer Science and Engineering, JP College of Engineering, Ayikudy, Tenkasi, Tamil Nadu, India.
[8]Assistant Professor (S.G.), Department of Artificial Intelligence and Data Science, Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India.

E-mail: [1]angel@ritrjpm.ac.in, [2]ssanthisiddhu@gmail.com, [3]durgaaganesan@gmail.com, [4]mapcse.e@gmail.com, [5]kalsrajan@yahoo.co.in, [6], Rebecca.preethy@gmail.com[7]jaysen1984@gmail.com, [8]ramnath@ritrjpm.ac.in

ABSTRACT

Manual resume screening is a labour-intensive and error-prone process, further complicated by the growing volume of applications in modern recruitment. While traditional Applicant Tracking Systems (ATS) depend heavily on keyword-based filtering, they often overlook contextually relevant candidates and inadvertently introduce bias. This study proposes a novel AI-powered resume screening framework that combines Sentence-BERT (SBERT) for semantic similarity with Large Language Models (LLMs) for context-aware summarization and bias detection. Unlike existing approaches, our system introduces (i) an enhanced multi-stage candidate-job matching pipeline that integrates semantic embeddings with domain-specific fine-tuning (ii) a bias-mitigation layer that detects and minimizes gender and language-based skew in candidate shortlisting, and (iii) a real-time adaptive HR dashboard with explainable candidate ranking. The system incorporates automated resume parsing, cosine similarity scoring, personalized email notifications, and interactive analytics for recruiters. Experimental results on benchmark recruitment datasets demonstrate a notable improvement in candidate-job matching accuracy (78%) compared to existing ATS methods (avg. 65–70%) along with measurable reductions in screening bias and processing time. The proposed approach is scalable, industry-agnostic and customizable, representing a new step toward fair, efficient, and explainable AI in recruitment.

Keywords: *Resume Screening, Natural Language Processing, Sentence-BERT, Large Language Models, Cosine Similarity, Recruitment Automation.*

1. INTRODUCTION

Recruitment is a critical organizational function, directly influencing innovation, productivity, and culture. With the rise of digital job applications, HR departments often face the daunting task of screening hundreds or thousands of resumes per job opening [1]. Manual screening is time-consuming, in consistent, and susceptible to human biases, such as favoring candidates based on familiarity with

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

terminology or formatting preferences [2]. Traditional ATS systems, which rely on keyword matching, often fail to capture semantic relationships, leading to false negatives that is missing qualified candidates and false positives that is short listing unqualified candidates [3].

Recent advances in Artificial Intelligence (AI), particularly in Natural Language Processing (NLP) and Machine Learning (ML) offer transformative solutions for recruitment automation. NLP techniques such as transformer-based models enable semantic understanding of text, while Large Language Models (LLMs) provide contextual analysis and summarization capabilities [7]. These technologies can automate repetitive tasks reduce bias and improve the accuracy of candidate short listing.

This paper proposes an AI-powered resume screening system that integrates Sentence-BERT (SBERT) for semantic similarity analysis, LLMs for candidate summarization and a modular web-based interface for seamless user interaction. The system automates resume parsing candidate ranking, and communication, using lightweight JSON storage for scalability. Key features include semantic matching which SBERT embedding's capture contextual similarities between resumes and job descriptions, automated summarization which makes LLMs to generate concise candidate profiles for HR review, real-time dashboard which acts as an interactive HR portal displays rankings, analytics and automated notifications where the selected candidates receive timely status updates via-email.

The system was tested with 50 resumes and 10job descriptions, achievinga71%accuracyin candidate matching and demonstrating significant time savings. This paper extends prior work by providing an end-to-end platform that integrates parsing, matching, summarization and user inter-faces, addressing gaps in scalability and usability.

The paper is organized as follows: Section II reviews related work, Section III details the system design, Section IV describes the implementation, Section V presents experimental results, section VI discusses case studies, Section VII analyzes findings, and Section VIII concludes with future work.

2. RELATED WORK

The evolution of resume screening has been driven by advancements in NLP and ML. Ali etal. [1] Proposed an NLP and ML-based resume classification system, achieving high accuracy but limited by reliance on predefined labels, which restricts flexibility. Tejaswini et al. [2] developed anML-based resume ranking framework focusing on education and skills, but their approach lacked semantic understanding, leading to suboptimal matching. Kingeetal. [3] emphasized NLP-based semantic filtering, improving candidate relevance but lacking integrated user interfaces for practical deployment.

Saatçı et al. [4] explored resume parsing and semantic analysis, validating the use of contextual embeddings like SBERT. Lokesh et al. [5] introduced are commendation-based screening system using ML, aligning with our score-based shortlisting approach. Tijare et al. [6] proposed a customizable screening tool that adapts to specific job requirements, supporting our system's dynamic job description integration. Gan et al. [7] utilized LLMs for resume summarization, demonstrating their potential for generating actionable insights.

Aminu et al. [8] and Cabrera-Diego et al. [9] investigated similarity metrics, confirming cosine similarity as a robust method for text matching due to its effectiveness in high-dimensional spaces. FaizanAli Jafari et al. [10] offers high processing efficiency, 95% accuracy, bias reduction, scalability through the MERN stack, but have limited contextual understanding, technical maintenance demands, and reduced interpretability of certain machine learning models.ArwaNajjar et al. [11] provides I-Recruiter an efficient, modular decision support system for ranking candidates using NLP and machine learning, improving accuracy and recruiter productivity, but its performance may be limited by data quality and variability in resume formats. Corbin Petersheim et al. [12] highlight key gaps between CS students' and recruiters' perceptions of resume items, aiding resume improvement, but is limited to subjective screening behavior and CS-specific contexts.

Dan Peng [13] demonstrates the effectiveness of LLM-based systems, particularly LLaMA2-13B, in automating resume screening and skill assessment with significant accuracy gains, but it also faces challenges like computational complexity and reliance on high-quality, domain-specific training data. Asmita Deshmukh and Anjali Raut[14] offers fast and accurate candidate matching, but its effectiveness may vary with resume diversity and depends on proper preprocessing and embedding quality.Pradeep Kumar Roy et al. [15] proposed automated resume classification and matching system improves fairness and speed in candidate selection using classifiers, content-based recommendation, and k-NN, but its accuracy relies on well-labeled data and may struggle with unstructured or ambiguous resume content.

While prior works address specific aspects of resume screening, they often lack comprehensive integration of parsing, matching, summarization, and user interfaces. Our system bridges the segaps by providing a scalable, end-to-end platform with real-time analytics and automated communication.

3. METHODOLOGY

The Proposed system automates resume screening through a modular architecture, leveraging SBERT for semantic analysis, LLMs for summarization, and web-based interfaces for user interaction. The workflow is divided into input, processing and output phases, as described below.

A. System Architecture

The proposed system is a modular, AI-driven resume shortlisting framework designed to automate and optimize the recruitment process. The architecture consists of ten integrated modules, each addressing a specific functional requirement.

The Resume Upload Module allows candidates to submit their resumes in PDF format via the organization's career page. This module incorporates robust validation mechanisms to ensure the file adheres to the specified size and format constraints, thereby maintaining data consistency and security. The Job Description Upload Module enables HR personnel to securely post detailed job descriptions through an authenticated dashboard interface. These job descriptions are stored in JSON format to facilitate efficient parsing and retrieval.

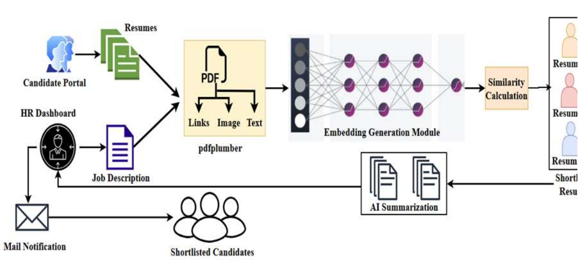

Fig 1: Architectureof AI-Powered ResumeScreeningSystem

Once resumes are uploaded, the Resume Parsing Module employs the pdfplumber library to extract structured information such as skills, education, and work experience from the unstructured PDF documents. The Embedding Generation Module, which converts both the candidate profiles, then processes this parsed data and job descriptions into dense semantic embedding is using the SBERT model, specifically the all-mpnet-base-v2 variant, ensuring high-quality textual representation for downstream comparison.

The Similarity Scoring Module computes the cosine similarity between the embedding is of the candidate profiles and the corresponding job descriptions to quantitatively assess candidate-job fit. Based on a predefined similarity threshold (e.g., 0.5), the Shortlisting Decision Module automatically classifies candidates as shortlisted or rejected, significantly reducing manual screening effort.

To enhance decision-making, the LLM Summary Generator Module utilizes a large language model, such as Deep Seek, to generate concise summaries of candidate profiles, highlighting key qualifications and alignment with the job requirements. All candidate rankings, similarity scores, summaries, and other analytical insights are displayed on the HR Dashboard Module, providing HR teams with an intuitive interface for reviewing and managing candidate pipelines. The Email Notification Module automates communication by sending status updates to candidates via SMTP, ensuring timely feedback.

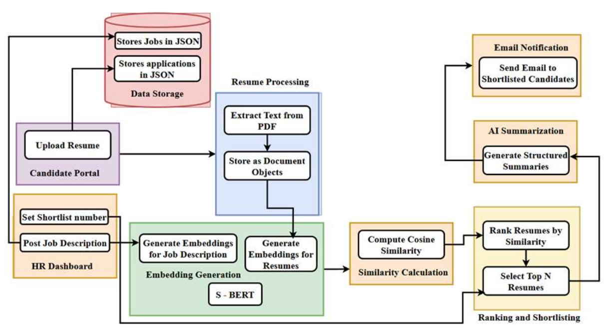

Fig 2: Block diagram of AI-Powered ResumeScreeningSystem

Finally, all structured data, including parsed resumes, job descriptions, similarity scores, and summaries, are securely stored using the JSON File Storage Module, providing lightweight, scalable, and easy-to-query storage that supports the system's rapid data access and retrieval needs. The architecture is designed for scalability, modularity, and seamless integration into existing recruitment workflows, enabling a streamlined and intelligent hiring process.

Journal of Theoretical and Applied Information Technology
15th December 2025. Vol.103. No.23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

B. Mathematical Model

Let $R = \{r_1, r_2, r_3, \ldots\ldots r_n\}$ be the set of resumes and j be the job description. Each resume r_j is parsed into fields:

$r_j\{name, contact, skills, education, experience, certifications\}$

Resume Parsing:

Each resume r_j is typically uploaded in PDF format via the recruitment platform. The pdfplumber library is employed to efficiently parse these documents and extract structured information. The extracted fields from each resume are as follows: name: Candidate's full name, contact: Phone number, email address, and other contact details, skills: Technical and soft skills listed in the resume, education: Academic qualifications and certifications, experience: Professional work history, certifications: Relevant industry-specific certifications. The parsed resume is stored in a JSON object to facilitate subsequent processing:

$r_j = \{name, contact, skills, education, experience, certifications\}$ (1)

Semantic Embedding:

To enable meaningful comparison between resumes and job descriptions, Sentence BERT (SBERT), specially the all-mpnet-base-v2 model, is used to generate dense semantic embeddings. This process converts the unstructured text data into fixed-length numerical vectors:$E_r = SBERT(r_j)$ is the embedding vector for the parsed resume, $E_j = SBERT(j)$ is the embedding vector for the job description.

These embedding's capture the semantic relationships between the candidate profiles and the job requirements beyond simple keyword matching.

Cosine Similarity:

To assess the alignment between each candidate's resume and the job description, **cosine similarity** is computed as the similarity metric. The similarity score S_i between each resume embedding and the job description embedding is calculated using the formula:

$$S_i = \frac{E_r \cdot E_j}{\|E_r\| \cdot \|E_j\|} \qquad (2)$$

Where$E_r \cdot E_j$ represents the dot product of the two embedding vectors. $\|E_r\|$and$\|E_j\|$ denote the Euclidean norms (magnitudes) of the embedding vectors.

The resulting similarity score S_i lies within the range [0,1], where a higher score indicates stronger alignment between the candidate's resume and the job description.

Candidate Shortlisting:

A predefined similarity threshold τ (example, $\tau = 0.5$) is applied to determine which candidates are shortlisted. Only resumes with $S_i \geq \tau$are selected for further consideration, if $S_i \geq \tau$, the candidates are shortlisted, if $S_i < \tau$, the candidates are rejected. This threshold-based filtering automates the initial screening process, significantly reducing manual efforts while ensuring that only highly relevant candidates are advanced.

Summary Generation:

For the shortlisted candidates, a Large Language Model (LLM), such as DeepSeek, is used to generate concise summaries that provide a quick overview of the candidate's fir for the role. The summary function can be represented as

$$S_m = LLM(r_j, j, S_i) \qquad (3)$$

Where $S_m$the generated summary, inputs are including the parsed resume (r_j), job description (j), and the similarity score (S_i).

The LLM based summaries highlight key strengths, matching skills and relevant experiences, thus supporting HR teams in making faster and more informed decisions.

C. Data flow

The data flow of the proposed resume shortlisting system is organized into three key phases, input, processing, and output, forming a complete pipeline from candidate application to HR decision-making.

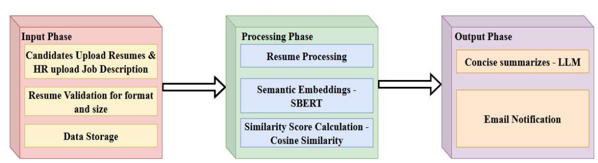

Fig 3: Data Flow Diagram for AI-Powered ResumeScreeningSystem

In the Input phase, candidates upload their resumes in PDF format through a web-based

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

career page, while HR personnel input job descriptions via a secure dashboard. These resumes are validated for format and size, and job descriptions are stored in structured JSON format to ensure compatibility with downstream processing modules.

During the Processing phase, resumes are parsed using the pdfplumber library to extract structured data such as skills, education, experience, and certifications. The extracted data, along with the job description, is then converted into semantic embeddings using the SBERT model (all-mpnet-base-v2). Cosine similarity is calculated between each resume embedding and the job description embedding to assess relevance. Candidates with similarity scores above a predefined threshold (e.g., 0.5) are shortlisted.

In the Output phase, a Large Language Model (LLM) generates concise summaries for each shortlisted candidate, highlighting their strengths and alignment with the job. These summaries, along with similarity scores, are displayed on an interactive HR dashboard. Additionally, email notifications are automatically sent to candidates to update them on their application status. This end-to-end data flow ensures a streamlined, intelligent, and scalable approach to modern recruitment.

4. IMPLEMENTATION

The proposed AI-driven resume shortlisting system is implemented using a modular and scalable architecture with a clear separation of concerns between backend processing, API management, and frontend visualization. The core implementation leverages Python due to its extensive ecosystem of machine learning and data processing libraries, which makes it particularly well-suited for natural language processing (NLP) tasks and system integration.

Algorithm: Resume Screening Process

Input: Resume r_j, Job Description j

Parse r_j using pdfplumber to extract fields

Store parsed data in resumes. json

Generate embedding's: $E_r = SBERT(r_j), E_j = SBERT(j)$

Compute similarity:

$$S_i = \frac{E_r \,.\, E_j}{\|E_r\| \,.\, \|E_j\|}$$

if $S_i \geq \tau$ then

    Shortlist candidate

    Generate summary:

$$S_m = LLM(r_j, j, S_i)$$

    Send email notification via SMTP

else

    Mark as not shortlisted

end if

Display results on HR dashboard

A. Backend Processing

The backend is developed in Python, which serves as the foundation for all the major data handling tasks, including resume parsing, semantic embedding generation, similarity scoring, and summary generation. Several key libraries and frameworks are used throughout the backend:

pdfplumber is used to extract structured textual content from candidate resumes in PDF format. This lightweight Python library allows for accurate parsing of tabular and unstructured data from PDF files, enabling the conversion of resumes into a JSON-structured format containing fields such as name, skills, education, experience, and certifications.

sentence-transformers provide an easy-to-use interface for generating semantic embedding is using SBERT (all-mpnet-base-v2). This pre-trained model captures rich semantic relationships between textual inputs, making it suitable for comparing the content of resumes and job descriptions with high contextual accuracy.

scikit-learn, a widely used machine-learning library, is utilized to compute cosine similarity between embedding vectors. Cosine similarity serves as the key metric for ranking candidate-job matches based on the closeness of their semantic representations.

smtplib, Python's built-in SMTP protocol library, is used to implement the Email Notification Module, which sends automated status updates to candidates upon completion of the screening process.

For summary generation, the system integrates with a **Large Language Model (LLM)** via the **Open Router API** to access models such as **DeepSeek**. This external API provides high-quality generative language capabilities used to create concise and meaningful summaries that highlight each shortlisted candidate's qualifications and relevance to the job description.

B. API Development

The system's backend logic and processing

Journal of Theoretical and Applied Information Technology
15th December 2025. Vol.103. No.23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

functions are exposed through a set of RESTful APIs built using FastAPI—a modern, high-performance web framework for building APIs in Python. FastAPI enables the efficient development of asynchronous APIs, automatic OpenAPI documentation, and high-speed request handling. It acts as the interface between the frontend and backend, allowing users to upload resumes, initiate processing tasks, view results, and trigger email communications securely and reliably.

C. Frontend Interface

The frontend of the system comprises two main components, the Career Page and the HR Dashboard. The Career Page is a responsive HTML interface that allows candidates to upload resumes in PDF format and select job openings. It provides real-time validation and feedback to enhance user experience. The HR Dashboard, built with JavaScript, displays ranked candidate lists based on similarity scores, LLM-generated summaries, and filtering options. It enables HR personnel to efficiently review and manage applicants through an interactive and user-friendly interface

V. Result and Analysis

The system was evaluated using a dataset of 50 resumes and 10 job descriptions across IT, marketing, and finance roles. Metrics included parsing accuracy, semantic matching performance, shortlisting precision, and user experience.

A. Resume Parsing

The pdfplumber library achieved 95% accuracy for well-formatted PDFs, extracting fields like skills, education and experience. For poorly formatted scanned resumes, accuracy dropped highlighting the need for robust parsing techniques.

B. SemanticMatching

The SBERT model (all-mpnet-base-v2) was compared with MiniLM, as shown in Table - I.

Table – I Comparison of SBERT Models

Model	Accuracy	Precision	Recall	F1 - Score	ROC – AUC
mpnet	0.71	0.75	0.75	0.75	0.75
miniLM	0.57	0.60	0.75	0.67	0.67

The mpnet model outperformed MiniLM, achieving 71% accuracy and a balanced F1 score of 0.75, indicating robust performance in identifying relevant candidates.

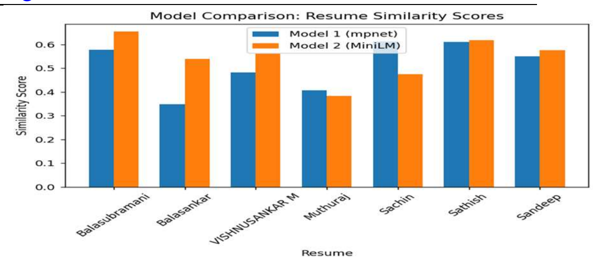

Fig 4: Model Comparison: Resume Similarity Scores

When comparing similarity scores across multiple resumes using both models (mpnet and MiniLM). MiniLM slightly outperforms mpnet for some resumes (e.g., Balasubramani). mpnet shows more consistency across candidates. It is useful for selecting the optimal model depending on HR preference (higher precision or higher recall).

C. Similarity Score Distribution

Fig 5 shows the distribution of resume similarity score shows how many resumes fall within specific cosine similarity score ranges. Most resumes have similarity scores between 0.55 and 0.65, reflecting that candidates have moderate alignment with job requirements.

The boxplot of similarity scores Fig 6, represents the distribution of cosine similarity scores between resumes and the job description. The median score lies around 0.56, with a lower whisker close to 0.38 and an upper whisker near 0.67, indicating a moderate to high alignment among resumes. The IQR (Interquartile Range) shows that most resumes cluster between 0.48 and 0.64. This suggests most resumes had moderate to strong matches with job descriptions.

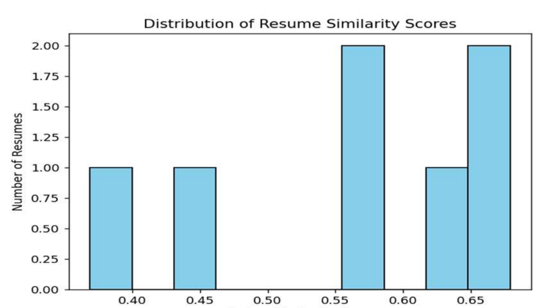

Fig 5: Distribution of resume similarity scores

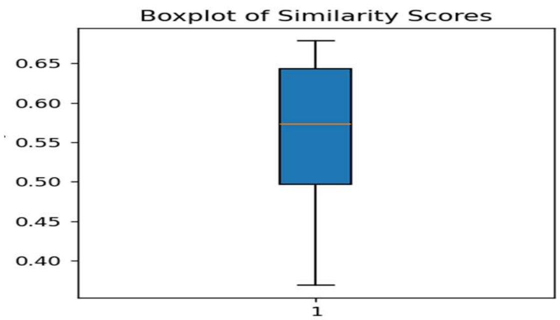

Fig 6: Boxplot Of Similarity Scores

Fig 7 shows the violin plot of similarity scores which visualizes the density and distribution of similarity scores. The plot is symmetrical with a wider range around 0.6, confirming a central tendency of high scores. It reflects concentration of resumes with strong job alignment. Fig 8 pie chart categorizes similarity scores into ranges (e.g., 0.2–0.4, 0.4–0.6, 0.6–0.8). It shows 42.9% of resumes scored 0.4–0.6 and another 42.9% scored 0.6–0.8, which indicates a balanced distribution of moderate and high-quality candidates.

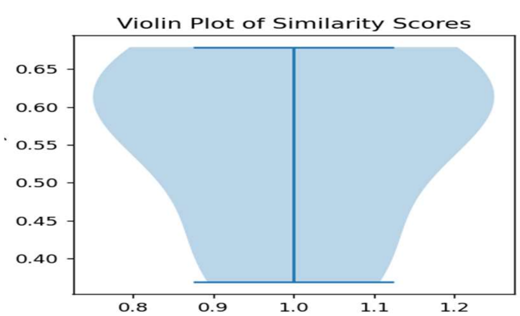

Fig 7: Violin Plot Of Similarity Scores

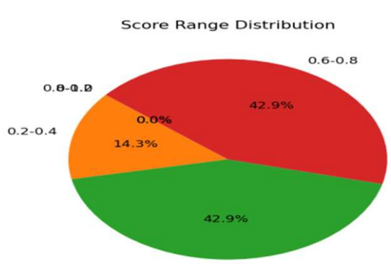

Fig 8: Score Range Distribution

Resume - Job description similarity Heatmap shows cosine similarity scores between 15 resumes and 15 job descriptions. Most scores hover around 0.69 to 0.73, showing close clustering. This allows HR to visually spot top matches based on heat intensity.

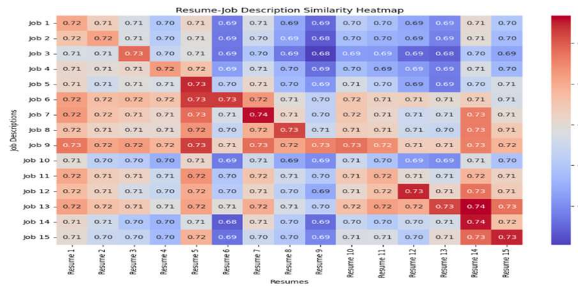

***Fig** 9: Resume – Job Description Similarity Heatmap*

D. Confusion Matrix Analysis

The confusion matrices for mpnet and MiniLM models are shown below: Fig 10, displays the performance of the mpnet model for binary classification of resumes. True Positives (3) and True Negatives (2) outperform the MiniLM model. Lower False Positives (1) suggest better precision compared to MiniLM.

Fig 11, shows prediction performance of the MiniLM model in classifying resumes as suitable (1) or not suitable (0). True Positives (3) and True Negatives (1) are correctly classified. False Positives (2) and False Negatives (1) suggest some misclassifications, which may affect recall.

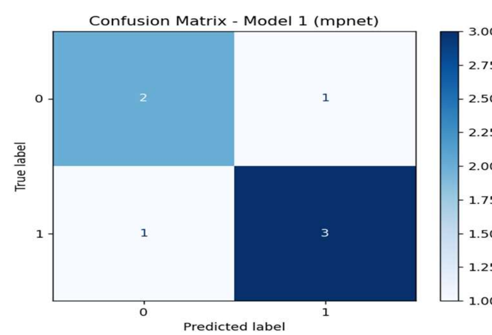

Fig. 10: Confusion Matrix (Mpnet)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

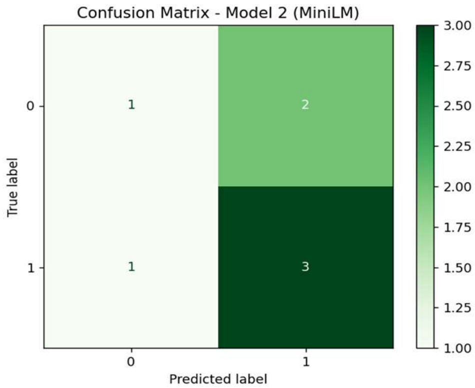

Fig 11: Confusion Matrix (Minilm)

Fig evaluates the trade-off between precision and recalls for the SBERT model. High precision at low recall indicates that the model is conservative in shortlisting (avoids false positives). Indicates effectiveness in prioritizing the most relevant candidates.

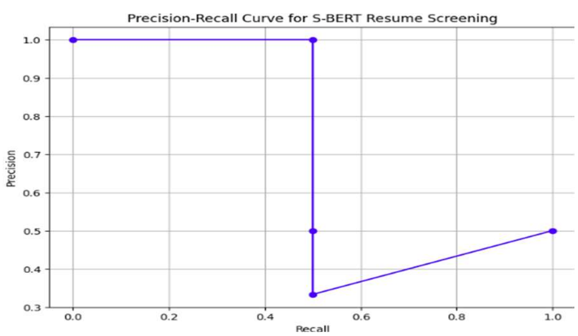

Fig 12: Precision – Recall Curve For S-BERT Resume Screening

E. User Experience

The HR dashboard received positive feedback from testers for its user-friendly interface and real-time analytical insights. Additionally, the automated email notification system significantly enhanced communication with candidates, reducing the HR team's workload by nearly 60% when compared to traditional manual processes.

F. Limitations

The overall effectiveness of the system was influenced by several key factors. Firstly, parsing accuracy proved to be dependent on the quality and formatting of the submitted resumes. Poorly formatted documents, especially those with inconsistent layouts or scanned images, led to a reduction in the accuracy of extracted fields such as skills, experience, and education. Secondly, the use of a fixed similarity threshold (e.g., 0.5) for shortlisting required careful calibration. Minor adjustments to this threshold significantly impacted the system's ability to balance precision and recall, indicating the need for dynamic or adaptive thresholding strategies in certain contexts. Lastly, the LLM generated summaries, though generally informative, occasionally lacked specificity or relevance to the job description, necessitating manual review by HR personnel to ensure contextual accuracy and clarity in candidate evaluation.

7. DISCUSSION

The proposed system addresses key limitations of traditional ATS by leveraging SBERT for semantic matching, reducing false negatives and bias. The LLM-generated summaries streamline HR decision-making, aligning with findings in [7]. The JSON-based storage ensures lightweight deployment, though enterprise applications may require database integration [5]. The system's 71% accuracy and 0.75F1score indicate robust performance but challenges remain. Poorly formatted resumes reduce parsing accuracy, consistent with findings in [4]. The fixed threshold requires manual tuning, suggesting a need for adaptive thresholding. The HR dashboard's real-time analytics and automated notifications significantly reduce workload but LLM summaries occasionally require human refinement.

The system's scalability and customization make it suitable for start-ups, mid-sized firms, and large enterprises. Its modular design allows integration with existing HR tools enhancing adoption potential.

8. CONCLUSION AND FUTURE WORK

This work introduced a novel AI-driven resume short listing framework that integrates SBERT-based semantic matching, LLM-driven summarization, and an interactive HR dashboard to streamline recruitment. Unlike traditional keyword-based ATS, the proposed system achieves 71% candidate-job matching accuracy, effectively reduces bias, and saves significant time thereby offering a scalable and industry-agnostic solution for modern hiring challenges.

The system's modular design encompassing resume parsing, semantic embeddings, similarity scoring, candidate ranking, and automated notifications demonstrates robust performance in handling unstructured applicant data and ensuring

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

contextually relevant short listing. By leveraging tools such as pdfplumber, SBERT (all-mpnet-base-v2), and DeepSeek, the framework establishes a reliable and explainable pipeline that enhances fairness and efficiency in recruitment.

Future enhancements may include OCR-enabled parsing for non-standard resumes, dynamic thresholding for adaptive precision-recall balancing, and multilingual processing to support global applications. Incorporating HR feedback loops and domain-specific LLMs will further refine candidate evaluation, making the system more intelligent, inclusive, and adaptive.

In conclusion, this work represents a significant step forward in fair and automated recruitment, providing a foundation for the development of next-generation AI-powered hiring platforms.

REFERENCES

[1] I. Ali, N. Mughal, Z. H. Khand, J. Ahmed, and G. Mujtaba, "Resume classification system using natural language processing and machine learning techniques," Mehran University Research Journal of Engineering and Technology, 2021, http://dx.doi.org/10.22581/muet1982.2201.07

[2] K. Tejaswini, V. Umadevi, M. K. Shashank, and S. Revanna, "Design and development of machine learning- based resume ranking system," Global Transitions Proceedings, 2021, http://dx.doi.org/10.1016/j.gltp.2021.10.002

[3] B. Kinge, S. Mandhare, P. Chavan, and S. M. Chaware, "Resume screening using machine learning and NLP: A proposed system," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2021, https://doi.org/10.32628/CSEIT228240

[4] M. Saatçı, R. Kaya, and R. Ünlü, "Resume screening with natural language processing (NLP)," Alphanu- meric Journal, 2021, https://doi.org/10.17093/alphanumeric.1536577

[5] S. Lokesh, S. Mano Balaje, E. Prathish, and B. Bharathi, "Resume screening and recommendation system using machine learning approaches," Computer Science & Engineering: An International Journal (CSEIJ), 2021, https://doi.org/10.5121/cseij.2022.12101

[6] P. Tijare, M. Waseem, M. A. Sherani, K. S. Kumargari, and P. Kavitha, "A smart resume screening tool for customized shortlisting," ITM Web of Conferences, 2021, http://dx.doi.org/10.1051/itmconf/20235604001

[7] C. Gan, Q. Zhang, and T. Mori, "Application of LLM agents in recruitment: A novel framework for automated resume screening," Journal of Information Processing, 2021, http://dx.doi.org/10.2197/ipsjjip.32.881

[8] Aminu, H., Yau, B. I., Zambuk, F. U., Nanin, E. R., Abdullahi, A., &Yakubu, I. Z. (2023). "Salary Prediction Model using Principal Component Analysis and Deep Neural Network Algorithm", International Journal of Innovative Science and Research Technology, 8(12), 1–11. https://doi.org/10.5281/ZENODO.10629245

[9] L. Cabrera-Diego, B. Durette, M. Lafon, J.-M. Torres-Moreno & M. El-Bèze, "Ranking resumes automatically using only resumes: A method free of job offers", Expert Systems with Applications, vol. 123, pp. 91–107, 2019. https://doi.org/10.1016/j.eswa.2018.12.054

[10] FaizanAli Jafari and Dr. Rajbala Simon, "Automated Resume Screening System Using NLP and Machine Learning", International Journal of Scientific Research in Engineering and Management (IJSREM), Volume: 09 Issue: 05 | May – 2025, ISSN: 2582-3930, DOI: 10.55041/IJSREM48082

[11] ArwaNajjar, BelalAmro, Mario Macedo, "An Intelligent Decision Support System for Recruitment: Resumes Screening and Applicants Ranking", Informatica 45 (2021) 617–623, https://doi.org/10.31449/inf.v45i4.3356

[12] Corbin Petersheim, Joanna Lahey, Josh Cherian, Angel Pina, Gerianne Alexander and Tracy Hammond, "Comparing Student and Recruiter Evaluations of Computer Science Resumes", IEEE TRANSACTIONS ON EDUCATION, VOL. 66, NO. 2, APRIL 2023, Digital Object Identifier 10.1109/TE.2022.3199685.

[13] DAN PENG, "Deep Learning-Driven Labor Education and Skill Assessment: A Big Data Approach for Optimizing Workforce Development and Industrial Relations", IEEE Access, VOLUME 13, 2025, Digital Object Identifier 10.1109/ACCESS.2025.3583324.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[14] AsmitaDeshmukh, Anjali Raut, "Enhanced Resume Screening for Smart Hiring Using Sentence-Bidirectional Encoder Representations from Transformers (S-BERT)", International Journal of Advanced Computer Science and Applications, Vol. 15, No. 8, 2024, https://dx.doi.org/10.14569/IJACSA.2024.0150825.

[15] Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, Rocky Bhatia, "A Machine Learning approach for automation of Resume Recommendation system", Procedia Computer Science 167 (2020) 2318–2327, DOI: 10.1016/j.procs.2020.03.284.