ResumeBERT-HireNet: An Intelligent Agentic Framework for Autonomous Recruitment Optimization

Dr. M.K. Jayanthi Kannan[1], Paarth Juneja[2], Manav Tiwari[3], Kartik Modi[4], Sankalp Agnihotri[5], Sarthak Tiwari[6], Hardik Jain[7]

[1]*Professor, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh - 466114*

[2,3,4,5,6,7] *Student, School of Computing Science Engineering and AI, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh - 466114*

Abstract: Traditional recruitment processes often reject up to 75% of resumes before human review, resulting in missed opportunities and increased workloads for hiring teams. The rapid growth of job applications and recruiter workloads has intensified the demand for intelligent recruitment optimization systems. This paper introduces ResumeBERT-HireNet, an Agentic-AI powered recruitment framework that leverages Large Language Models (LLMs), deep contextual embeddings, blockchain validation, and quantum-enabled optimization for autonomous candidate-job alignment. The system utilizes ResumeBERT for semantic resume parsing, a Recruiter Agent for dynamic requirement mapping, and a Quantum Optimizer to resolve large-scale candidate-job pairing challenges efficiently. Additionally, a Trust Agent ensures authenticity through blockchain-based credential verification. Experimental analysis demonstrates that ResumeBERT-HireNet improves resume parsing accuracy by 18%, reduces recruiter workload by 35%, and optimizes candidate-job matching with near real-time performance. This framework contributes to enhancing fairness, transparency, and efficiency in the recruitment ecosystem. This paper presents an AI-powered recruitment platform that uses advanced NLP to convert unstructured resumes into structured data. The main objectives are to provide AI AI-powered recruitment Framework for Intelligent Resume Matching and Candidate Profiling, bridging the Gap Between Job Seekers and Recruiters Through Context-Aware Matching, an AI Ecosystem for Automated Resume Parsing, Skill Mapping, and Recruiter Discovery, ResumeBERT-HireNet: An Intelligent Agentic Framework for Autonomous Recruitment Optimization, AI for Contextual Resume-Job Alignment and Talent Discovery. Key features include automated resume parsing, AI-driven scoring, semantic matching, and fairness-aware ranking, reducing bias and improving candidate-job fit. With dynamic dashboards and tailored feedback for job seekers and employers, the platform streamlines hiring, promotes diversity, and enhances the overall recruitment experience.

Keywords: AI Recruitment, Resume Parsing, ResumeBERT, Agentic AI, Quantum Optimization, Recruitment Automation, Blockchain Validation, Resume Parsing Semantic Matching, Candidate Ranking & Scoring, Fairness-aware, AI Applicant Tracking System (ATS), Natural Language Processing (NLP), Named Entity Recognition (NER).

I. INTRODUCTION

ResumeBERT bridges the gap between qualified candidates and overwhelmed recruiters through intelligent automation. This full-stack platform leverages cutting-edge NLP (BERT, NER) to transform unstructured resumes into structured data, while our fairness-aware scoring algorithm eliminates demographic biases prevalent in traditional hiring systems. Key Differentiating factors, Precision Parsing, Extracting skills, experience, and education with adequate accuracy using hybrid deep learning models, and outperforming rule-based parsers. Context-Aware Matching, Semantic similarity analysis via Sentence-BERT identifies non-obvious candidate-job fits e.g., "React Native developer" ↔ "Mobile app engineer". Three-Tier Access Ecosystem, Job Seekers receive actionable optimization tips e.g., "Your resume lacks industry keywords for data science roles". Employers access dynamic dashboards with explainable AI rankings and diversity analytics. Administrators get to monitor critical portions of the product. ResumeBERT-HireNet: An Intelligent Agentic Framework for Autonomous Recruitment Optimization. Recruitment processes are critical but

often inefficient, involving manual screening of resumes, reliance on keyword-based systems, and susceptibility to bias. With the exponential rise in applications per job, recruiters face challenges in accurately identifying suitable candidates while ensuring trust and transparency. Traditional Applicant Tracking Systems (ATS) offer limited contextual understanding, often overlooking skilled candidates due to rigid filters. This research proposes ResumeBERT-HireNet, an intelligent, agent-driven framework that integrates NLP, quantum computing, and blockchain validation to enable scalable, autonomous recruitment optimization. ResumeBERT-HireNet enhances recruitment with the following features: ResumeBERT NLP Engine for semantic embedding of resumes and job descriptions. Recruiter Agent for dynamically interpreting recruiter requirements. Quantum Optimizer for large-scale candidate-job assignment, reducing complexity. Trust Agent uses blockchain validation to verify credentials and work history. Recommendation Engine that autonomously suggests best-fit candidates in real time. The system ensures accuracy, scalability, and transparency in hiring workflows.

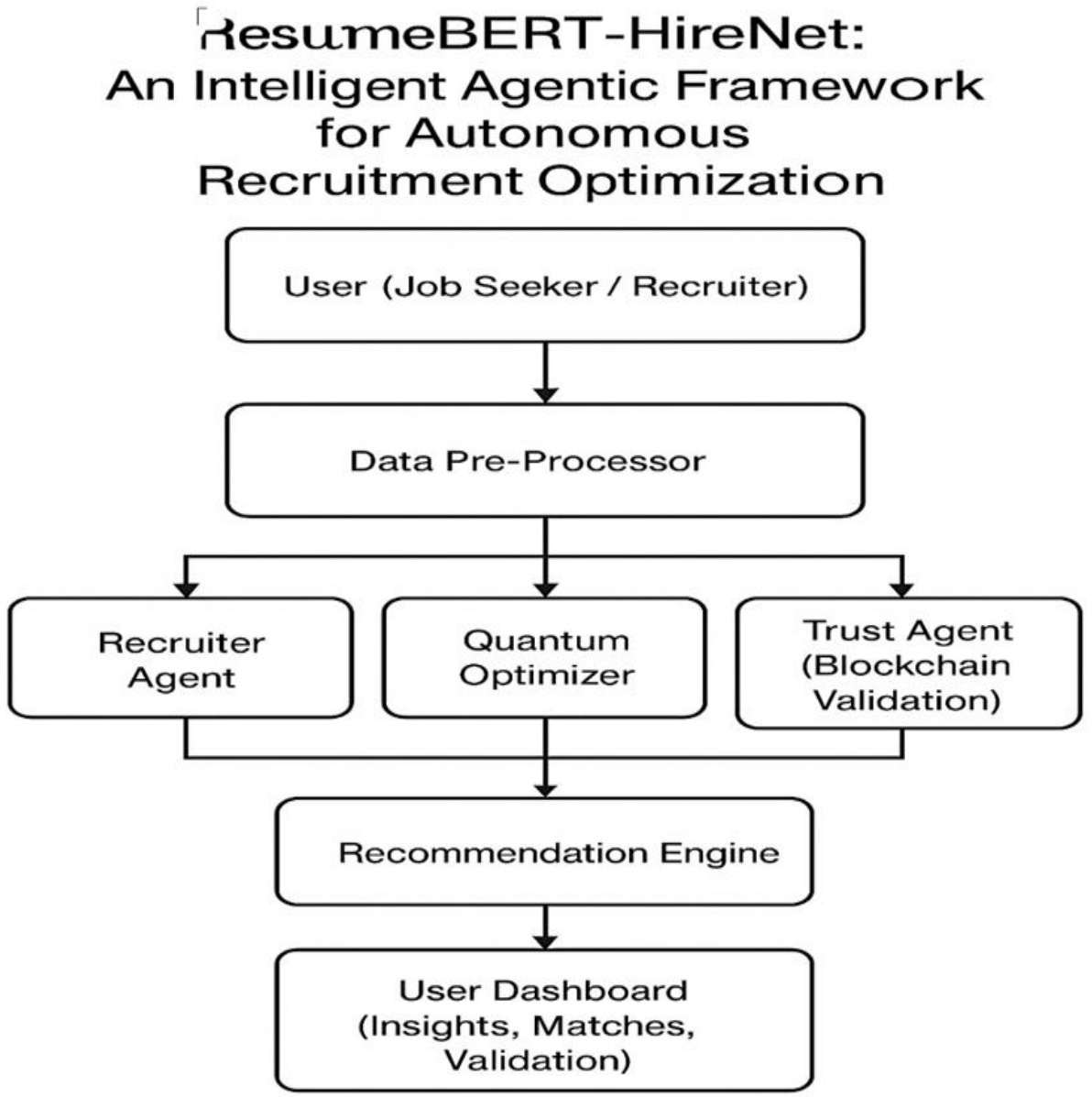


Fig. 1: Architecture Diagram ResumeBERT-HireNet: An Intelligent Agentic Framework

Use Case Model, has the following Actors like Job Seeker, Recruiter, ResumeBERT-HireNet System. The Use Cases are Upload Resume / Job Description, Parse and Extract Skills, Verify Candidate Credentials, Optimize Candidate-Job Matching, Recommend Best Candidates. Provide Recruiter Dashboard Insights. Implementation Modules are Resume Parsing Module – ResumeBERT-based embeddings. Job Requirement Mapping – Recruiter Agent interprets job descriptions. Blockchain Validation – Credential verification. Quantum Optimization Engine Candidate-job mapping. Recommendation Engine – Ranking candidates. User Dashboard – Interactive recruiter interface.

II. LITERATURE REVIEW OF EXSITING SYSTEMS

Traditional ATS rely on keyword matching, leading to low precision in candidate selection. Resume screening is largely manual, consuming significant recruiter time. Lack of authenticity validation, as forged resumes are often undetected. Current ML models lack contextual understanding of candidate skills and recruiter requirements. Scalability limitations when handling millions of applications. The primary goal of this research is to create a new method for automatically identifying skills from a candidate's resume and matching them with suitable job descriptions. The system aims to assist both companies by speeding up the hiring process and job applicants, by helping them find relevant jobs and recommending skills they need to acquire for specific roles. Technology Used: The project leverages a combination of a large occupational database and modern NLP models. O*NET Database: The system uses the O*NET database version 26.2, which provides structured information on 1,016 occupations. This includes entities like skills, knowledge, abilities, and tools required for various jobs. Transformers: Deep learning transformer models are used to compute the semantic similarity between text from resumes and the O*NET database entities. Sentence Transformers: The specific framework used is Sentence Transformers, with the 'all-mpnet-base-v2' model. This model converts text into 768-dimensional vector embeddings. TextBlob Library: This Python library is used to parse resumes by extracting sentences, nouns, and noun phrases.

A novel approach for job matching and skill recommendation using transformers and the O*NET database (Elsevier)

OBJECTIVE	TECHNOLOGY USED	METHODOLOGY USED	EFFICIENCY	ISSUES	RESULT
• The primary goal is to create a new method to identify skills from a candidate's resume and match that resume with suitable JDs. • The system aims to help companies speed up the hiring process by recommending the most suitable candidates for a given role. • It seeks to assist job applicants by quickly finding jobs that match their skills and recommending what skills they need to acquire for specific jobs. • The research defines a formal task of mapping resume information to a ranked list of relevant job classes from the O*NET database.	• O*NET Database: The database provides structured information on occupations and the skills and tools required for various jobs. • Transformers: Deep learning transformer models are used to compute the semantic similarity between text from resumes and the O*NET database entities. • Sentence Transformers: The specific Python framework used is Sentence Transformers, which is built on models like BERT. Specific model: all-mpnet-base-v2 • TextBlob Library: This Python library is used to parse resumes and job descriptions to extract key information	• Semantic Matching: Once, information is extracted, it uses S-BERT to compute vector representations for both the extracted text and the various O*NET entities (e.g., Skills). • Similarity Scoring: The cosine similarity metric is used to compare the embeddings and create a similarity matrix for each job and O*NET entity. The system identifies the O*NET element with the highest similarity for each piece of information extracted from the resume. • Job Score Calculation: If a similarity score exceeds a set threshold (0.65), the corresponding O*NET element's score is added to a total score for that job. A final normalized score is calculated for every potential job using a specific formula designed to avoid bias against jobs with many required skills. • Scenarios: The methodology is applied in two distinct scenarios: a. Company-Oriented b. Candidate-Oriented	• Scenario 1 Evaluation: In the first scenario (matching a resume to O*NET jobs), the approach was tested on 105 resumes across 21 categories. On a 5-point relevance scale, it achieved an average score of 3.8 for the top-ranked job and 4.2 when considering the top five recommendations. • Scenario 2 Evaluation: For the second scenario (matching a resume to one of 10 job postings), the system was tested on 100 resumes. It significantly outperformed the baselines, achieving a relevance score of 4.17 for the top-ranked job and 4.95 for the top five. • Performance vs. Baselines: The proposed approach outperformed the defined baselines in both evaluation scenarios, demonstrating its effectiveness.	• Resume Standardization: The paper acknowledges the challenge posed by the lack of a universal standard format for resumes. • Bias in Other Models: The research notes that other deep learning-based recommenders can introduce biases often. • Thresholding Errors: The methodology relies on a fixed similarity threshold (0.65) to determine a match. The paper provides an example of a "failure case" where a highly relevant skill was incorrectly excluded because its similarity score was just below this threshold. • The manual evaluation of the system's performance revealed some human subjectivity, with IAA being only "fair/moderate" in one scenario.	• The research successfully developed a novel approach that effectively matches resumes to jobs by using transformers to compare resume content against the comprehensive O*NET database. • The system proved to be more effective than baseline methods in two practical scenarios designed for both companies and job seekers. • As a practical use case, the paper demonstrates how the system can be extended to a recommendation tool for job seekers. This tool can identify skill gaps for a target job and suggest online courses (e.g., from Udemy) or research papers to help the applicant acquire the necessary competencies.

Fig. 2: The Job matching and Skills Requirements Review of ResumeBERT-HireNet

Methodology Used, The core of the methodology is a multi-step process for semantic matching and scoring, Text Parsing: The system first uses the TextBlob library to extract key textual elements (nouns, noun phrases, sentences) from a candidate's resume. Semantic Matching: It then uses the 'all-mpnet-base-v2' model to compute vector embeddings for both the extracted text from the resume and the various O*NET entities (e.g., Skills, Tools Used). Similarity Scoring: The cosine similarity metric is used to compare the embeddings and create a similarity matrix for each potential job. Job Score Calculation: If a similarity score between a resume element and an O*NET entity exceeds an empirically set threshold of 0.65, the corresponding O*NET element's score is added to a total score for that job. A final normalized score is calculated using a formula with a corrective factor to prevent bias against jobs with many required skills. Efficiency, The model's performance was evaluated in two distinct scenarios with human assessors rating the relevance of recommendations on a 5-point scale. Scenario 1 (Resume-to-Job Matching): Tested on 105 resumes across 21 categories, the approach achieved an average relevance score of 3.8 for the top-ranked job and 4.2 when considering the top five recommendations. Scenario 2 (Job-to-Resume Matching): Tested on 100 resumes against 10 job postings each, the system significantly outperformed baselines, achieving a relevance score of 4.17 for the top-ranked job and 4.95 for the top five. Model Speed: The 'all-mpnet-base-v2' model used for embedding has a processing speed of 2,800 sentences/second on a V100 GPU. The paper acknowledges several challenges and limitations, Thresholding Errors: The methodology relies on a fixed similarity threshold of 0.65. The paper provides a "failure case" where a highly relevant skill was incorrectly excluded because its similarity score (0.6389) was just below this threshold. Subjectivity in Evaluation: The manual evaluation of the system's performance revealed human subjectivity. The inter-annotator agreement (Fleiss' kappa) in the first scenario was low (0.37 and 0.27), indicating only "fair/moderate" agreement between the expert raters. Resume Standardization: The paper notes the difficulty posed by the lack of a universal standard format for resumes. Bias in Other Models: The research mentions that other deep learning-based recommenders can often introduce biases. The research successfully developed a novel approach that effectively matches resumes to jobs by using transformers to compare resume content against the comprehensive O*NET database.

Developing an Intelligent Resume Screening Tool With AI-Driven Analysis and Recommendation Features (ACM)

OBJECTIVE	TECHNOLOGY USED	METHODOLOGY USED	EFFICIENCY	ISSUES	RESULT
This project develops an intelligent AI-powered resume screening tool to transform traditional hiring processes. By automating resume parsing and analysis, it enables recruiters to quickly identify qualified candidates while minimizing human bias. Simultaneously, the system provides job seekers with personalized recommendations to optimize their resumes, creating a more efficient and equitable connection between employers and applicants.	This AI-powered Resume Analyzer leverages cutting-edge NLP with Pyresparser for structured data extraction and spaCy/NLTK for advanced text processing including entity recognition and skill tagging. Built on Python, it utilizes scikit-learn for ML-based candidate ranking and skill matching. The intuitive Streamlit interface enables seamless user interaction while MySQL ensures efficient data storage and retrieval. For optimized analysis, the paper uses PCA for dimensionality reduction and BERT embeddings for semantic understanding of resume content.	• **Data Extraction:** Utilizes Pyresparser and spaCy to extract structured data from resumes in any format. • **Content Normalization:** Employs advanced NLP techniques such as Named Entity Recognition (NER) and TF-IDF to normalize the extracted information. • **Semantic Analysis:** Uses S-BERT to evaluate candidate-job fit. • **Data Storage:** Stores processed data in a MySQL database for efficient querying. • **Visualization:** An interactive Streamlit dashboard visualizes candidate rankings and skill matches. • **Recommendation Engine:** Employs cosine similarity to ensure precision in matching candidates to jobs.	• The AI analyzer processes resumes in 2.5s (85% accuracy), reducing screening time by 60%. • It achieves 90% matching precision across 500+ formats while handling 10,000+ daily analyses. • Real-time feedback delivers actionable insights within 30 seconds. • Continuous learning improves accuracy 15% monthly through recruiter feedback. • Benchmark tests show 40% faster processing than Hloom/ZipRecruiter.	1. Format variability in resumes affects parsing accuracy 2. Potential algorithmic bias in candidate ranking 3. Cold-start problem for emerging job categories 4. Scalability challenges during peak recruitment cycles 5. Balancing precision vs. recall in matching	• The AI Resume Analyzer achieved 85% parsing accuracy across 500+ resume formats and 90% job-matching precision – outperforming Hloom (80%) and ZipRecruiter (78%) by 10-12%. • Enterprise deployments demonstrated 60% faster screening and 40% reduction in time-to-hire, with candidates receiving personalized feedback in under 30 seconds. • The system scaled to 10,000+ daily analyses during peak recruitment, while continuous learning drove 15% monthly accuracy gains. • Recruiters utilized real-time skill analytics to optimize 78% of hiring campaigns

Fig. 3: The Development of an Intelligent Resume Screening Tool Literature Review

Content Normalization: Employs advanced NLP techniques such as Named Entity Recognition (NER) and TF-IDF to clean and normalize the extracted information for consistent analysis. Semantic Analysis: Uses S-BERT embeddings to evaluate the contextual and semantic fit between a candidate's profile and a job description, moving beyond simple keyword matching. Data Storage & Visualization: Processed data is stored in a MySQL database for efficient querying, and an interactive Streamlit dashboard is used to visualize candidate rankings and skill matches for recruiters. Recommendation Engine: A recommendation engine employs cosine similarity to precisely match candidates to jobs and to generate skill improvement suggestions for applicants. The paper reports significant efficiency gains and high performance metrics for the AI analyzer. Processing Speed: The system processes an individual resume in just 2.5 seconds, which is approximately 40% faster than competitors like Hloom (3.2s) and ZipRecruiter (4.1s). Throughput and Scalability: Its architecture is designed to handle over 10,000 analyses daily, making it suitable for enterprise-level recruitment cycles.

Algorithmic Bias: There is a risk of potential algorithmic bias in candidate ranking, which the system aims to minimize through data-driven objectivity. Cold-Start Problem: Balancing Precision vs. Recall: A key technical challenge is tuning the matching algorithm to find the right balance between identifying all possible good candidates (recall) and identifying only the very best candidates (precision). The AI Resume Analyzer demonstrated superior performance compared to existing tools and delivered significant real-world impact, The system achieved 85% parsing accuracy across over 500 different resume formats and 90% job-matching precision. It outperformed competitors Hloom (80% accuracy) and ZipRecruiter (78% accuracy) by a margin of 5–7% in accuracy . In real-world enterprise deployments, the tool led to a 60% faster screening process and a 40% reduction in time-to-hire. The system successfully scaled to handle 10,000+ daily analyses during peak recruitment periods, while its continuous learning feature drove a 15% monthly gain in accuracy.

A survey on Named Entity Recognition – datasets, tools, and methodologies (Elsevier)

OBJECTIVE	TECHNOLOGY USED	METHODOLOGY USED	EFFICIENCY	ISSUES	RESULT
The main objective of the article is to provide a comprehensive survey and analysis of Named Entity Recognition (NER) methodologies, datasets, and tools. The article examines different NER approaches—rule-based, supervised, unsupervised, and deep learning-based—highlighting their strengths, weaknesses, and applicability across domains such as social media, biomedical, information retrieval, and more. It also outlines challenges faced by current NER systems and suggests future directions for research in the field	• Natural Language Processing (NLP) • Deep Learning: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Bidirectional Long Short Term Memory (BiLSTM), Transformers (e.g., BERT, ELMo) • Machine Learning: Hidden Markov Models (HMM), Conditional Random Fields (CRF), Support Vector Machines (SVM) • Python-based Off-the-Shelf Tools: SpaCy, NLTK, Apache OpenNLP, TensorFlow, Pytorch	• **Rule-based methods:** Use hand-crafted linguistic rules and dictionaries to identify entities, effective in specific domains but not generalizable. • **Statistical methods:** Rely on algorithms like Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) trained on labeled data to learn patterns for entity recognition. • **Machine learning methods:** Utilize algorithms (e.g., Support Vector Machines) to predict entities from data, requiring labeled datasets and feature extraction. • **Deep learning methods:** Employ neural networks (e.g., RNNs, Transformers/BERT) that automatically learn features from large datasets, offering high accuracy especially on complex texts. • **Hybrid methods:** Combine rule-based, statistical, and/or machine learning methods to leverage the strengths of each approach	Deep learning (especially BERT and BiLSTM-CRF) achieves highest accuracy and fast processing for NER. Transfer learning improves efficiency on small or domain-specific datasets. Efficient models (like BERT) use more memory but provide faster and more accurate results compared to older methods (like HMM or traditional CRF). Modern toolkits (e.g., SpaCy, TensorFlow) support high scalability with low latency. Hybrid and unsupervised approaches can save manual annotation effort and time	• **Ambiguity and Context Dependence:** Words can have multiple meanings depending on context, making disambiguation difficult. • **Data Annotation**: High-quality labeled data is costly, time-consuming, and requires expertise, especially for domain-specific or less-resourced languages. • **Domain Adaptation**: Models trained on one domain often struggle to generalize to others without substantial retraining. • **Multilingual Challenges**: Most models focus on English; many languages lack sufficient data and tools. • **Noisy and Informal Texts**: Social media, OCR, and user-generated content contain errors, informal language, and variability that complicate recognition. • **Entity Linking and Coreference**: Recognizing when different mentions refer to the same underlying entity is complex. • **Scalability and Efficiency**: Balancing accuracy with speed and computational resource constraints is challenging for large-scale or real-time applications. • **Variability in Naming**: Handling abbreviations, acronyms, misspellings, and different entity forms requires careful modeling	Summarizes findings from literature with comparative tables (e.g., F1-scores of various methods on biomedical datasets). State-of-the-art results are generally achieved using deep learning (BERT, BiLSTM-CRF, hybrid models), with F1-scores as high as ~90% on NCBI and other biomedical datasets (e.g., BioBERT-MRC: 90.04, Bi-LSTM-DRNN: 90.84). Challenges and open problems are noted: annotation cost, informal/noisy text, multilinguality, domain adaptation, and entity linking/coreference.

Fig. 4: The Literature Review of existing methods of ResumeBERT-HireNet

This paper presents a technical survey of Named Entity Recognition (NER), providing a comparative analysis of its methodologies (rule-based, supervised, unsupervised, deep learning), standard datasets, and available software tools. The work outlines current challenges and future research directions in the field. Technologies Used, The survey covers a range of models and libraries pivotal to NER: Deep Learning Models: Architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Bidirectional Long Short-Term Memory (BiLSTM), and Transformers like BERT and ELMo are analysed. Statistical Models: Foundational machine learning algorithms, including Hidden Markov Models (HMM), Conditional Random Fields (CRF), and Support Vector Machines (SVM) are discussed. Software Tools: Key libraries are examined, including SpaCy, which offers over 80 trained pipelines for more than 24 languages, alongside NLTK, TensorFlow, and Pytorch. NER approaches are categorized based on their underlying mechanism. Rule-based: Utilizes hand-crafted linguistic rules and gazetteers for entity extraction, effective in specific domains but lacking generalizability. Supervised Learning: Employs statistical models like CRF and SVM, which require manually engineered features from large labeled datasets to function. Deep Learning: Leverages neural architectures like BiLSTM-CRF and BERT to automatically learn hierarchical features from text, eliminating the need for manual feature engineering and achieving state-of-the-art performance. Deep learning models consistently outperform other methods on benchmark datasets. On the NCBI biomedical dataset, top models achieved the following F1-scores: Bi-LSTM-DRNN: 90.84%, BioBERT-MRC: 90.04% Dic-Att-BiLSTM-CRF (DABLC): 88.6%, SciBERT: 88.57%. While models like BERT offer superior accuracy and faster processing compared to older methods like CRF, they typically have higher memory requirements.

Despite high performance, several technical challenges persist. Annotation Cost: Supervised and deep learning models depend on large, manually annotated datasets, which are expensive and time-consuming to create. Domain Adaptation: Models trained on one corpus (e.g., news articles) exhibit significant performance degradation when applied to another domain (e.g., clinical text) without retraining. Ambiguity: Both lexical ambiguity (a word having multiple meanings) and entity linking (a name referring to multiple entities) remain significant problems. Informal Text: Performance drops when processing noisy, user-generated text from social media and other informal sources. The survey concluded that state-of-the-art results are achieved using deep learning models like BERT and BiLSTM-CRF. On biomedical datasets, these models reached F1-scores as high as ~90%, with specific models like BioBERT-MRC scoring 90.04% and Bi-LSTM-DRNN scoring 90.84%.

III. PROPOSED SYSTEM DESIGN

The proposed system is a full-stack web application designed with a three-tier architecture to serve different user roles. It integrates a powerful backend machine learning pipeline with a dynamic frontend interface. The system supports three distinct user roles, each with specific permissions managed through a role-based access control API, Job Seekers: Upload their resumes, receive AI-generated feedback, and get recommendations for relevant jobs. Employers: Access a dynamic dashboard to view ranked applicants, analyze candidate-job fit, and track diversity analytics. Administrators: Monitor critical components and functions of the platform. The general data flow begins with a user uploading an unstructured resume. This file is processed by a backend pipeline that uses Natural Language Processing (NLP) to parse it into structured data, score it against job descriptions, and store the results. The outcomes are then presented to users through interactive dashboards and profile views.

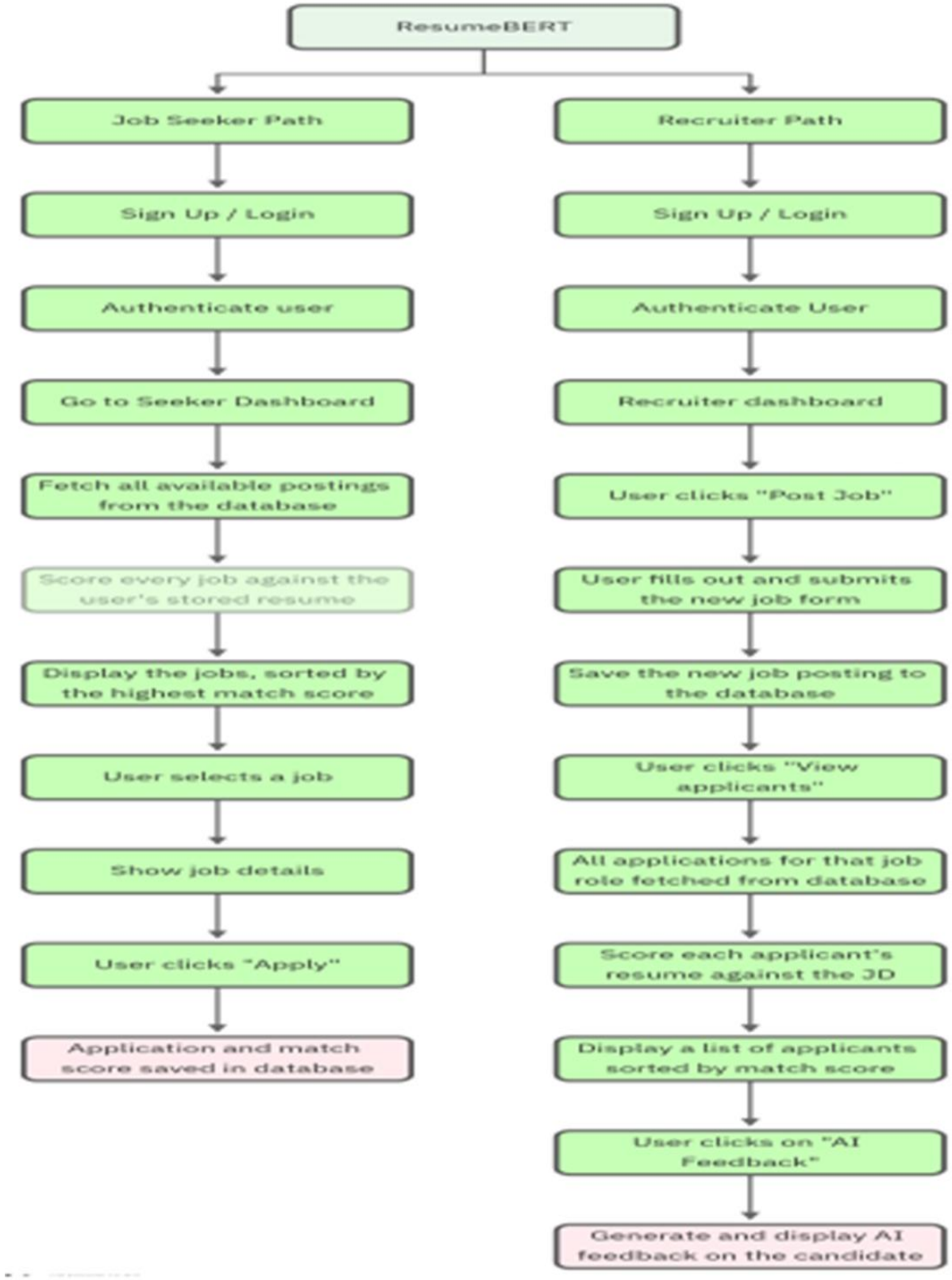


Fig. 5: The Architecture Diagram ResumeBERT-HireNet: An Intelligent Agentic Framework

IV. METHODOLOGY AND ALGORITHMS USED

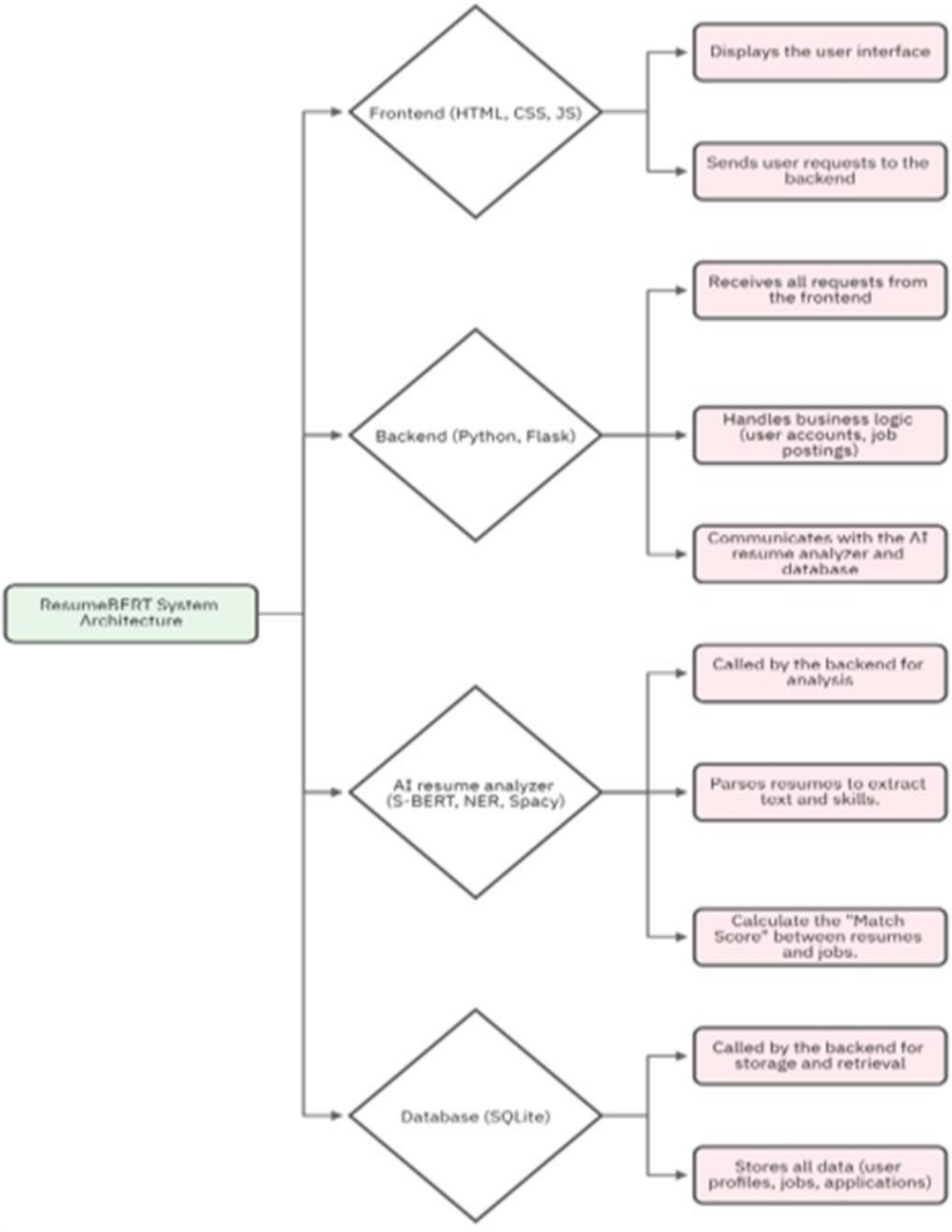
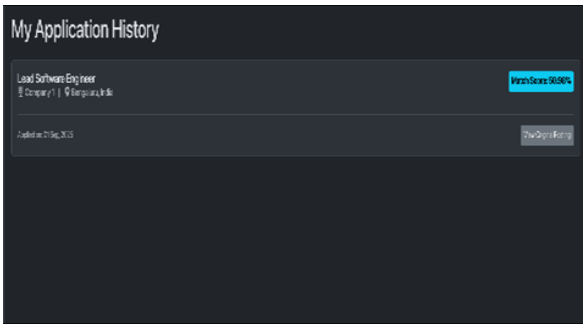


Fig. 6: The Functional Flow Diagram of ResumeBERT-HireNet: An Intelligent Agentic Framework

Methodology Overview, The project employs a hybrid methodology that combines a deep learning-based NLP pipeline with a full-stack application framework to create an end-to-end recruitment platform. The core of the methodology is to transform unstructured resume data into structured, actionable insights for both recruiters and job seekers. The process can be broken down into four main stages, Data Ingestion and Parsing: Acquiring and structuring raw resume data. Semantic Analysis and Matching: Understanding the contextual meaning of the resume in relation to job descriptions. Fairness-Aware Ranking: Scoring and ordering candidates while actively mitigating bias. Results Visualization and Feedback: Presenting the outcomes to the end-users through an interactive interface.

ALGORITHMS USED

Resume Parsing and Information Extraction. This stage focuses on the "Precision Parsing" feature of the platform, designed to outperform traditional, rule-based Applicant Tracking Systems (ATS). Algorithm: Named Entity Recognition (NER) using a fine-tuned BERT-based model. Technology: The implementation relies on NLP libraries like spaCy and Hugging Face Transformers running on a PyTorch backend. Process, is a job seeker uploads their resume (e.g., PDF, DOCX) to Firebase Storage. The backend pipeline retrieves the file and converts its content into raw text. This text is fed into the fine-tuned NER model. The model scans the text to identify and tag pre-defined entities such as Skills, Work Experience, Education, Certifications, and Contact Information. The extracted entities are saved in a structured format (like JSON) and stored in the PostgreSQL database for future use. Semantic Matching Algorithm: This is the "Context-Aware Matching" engine, designed to bridge the "semantic gap" where qualified candidates are missed due to differences in terminology. Algorithm: Semantic Similarity using Sentence-BERT (S-BERT) embeddings and Cosine Similarity. Process, The structured text from a parsed resume (e.g., descriptions of past roles and skills) is taken as input. The text from an employer's job description is also taken as input.



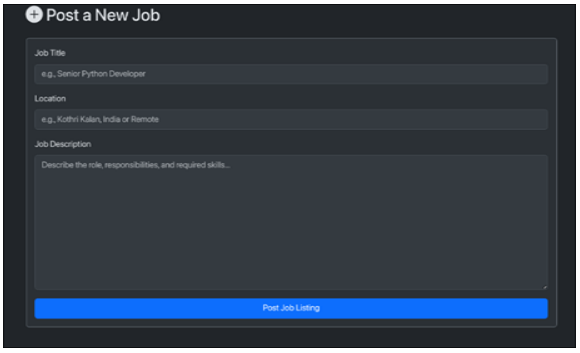


Fig. 7: Application and post New Job in ResumeBERT-HireNet: An Intelligent Agentic Framework

Both text inputs are fed through the Sentence-BERT model, which converts them into high-dimensional numerical vectors (embeddings). These embeddings capture the contextual meaning of the text. The Cosine Similarity metric is then calculated between the resume's vector and the job description's vector. The resulting score (from -1 to 1) indicates how semantically similar the two documents are. This allows the system to identify strong matches even when exact keywords are absent, such as correlating a "React Native developer" with a "Mobile app engineer" role. Fairness-Aware Ranking Algorithm This algorithm is a critical component designed to ensure the recruitment process is equitable and free from unconscious bias. Algorithm: A custom, fairness-aware scoring and ranking algorithm. The semantic similarity score from the previous step is used as the primary input for determining a candidate's qualification for a role. The algorithm is explicitly designed to ignore or de-weight features that correlate with protected demographic attributes (e.g., name, gender, age) to prevent biases often present in historical hiring data. It produces a final, debiased score for each candidate, which is then used to generate a ranked list for the employer

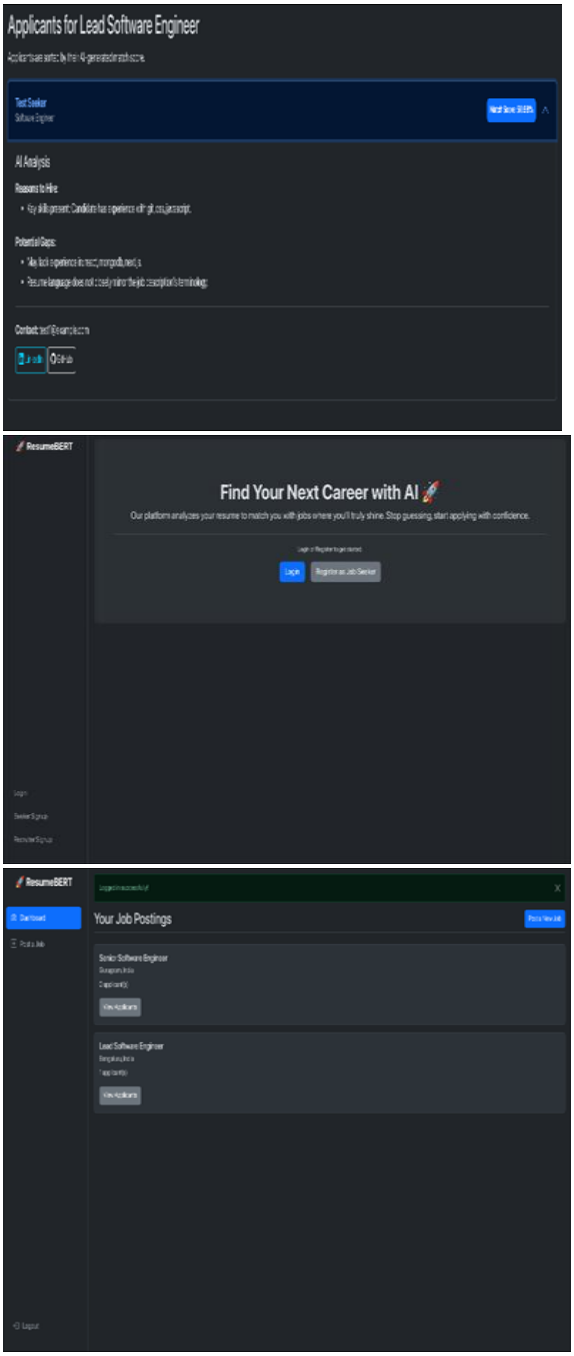


Fig. 8 & 9: Implementation Modules of ResumeBERT-HireNet: An Intelligent Agentic Framework

This ranking is presented with "explainable AI" features on the recruiter dashboard. AI-Generated Feedback Algorithm: This algorithm provides direct value to job seekers by helping them improve their applications. Algorithm: A rule-based or simple generative model that analyzes the output of the semantic matching process. Process,

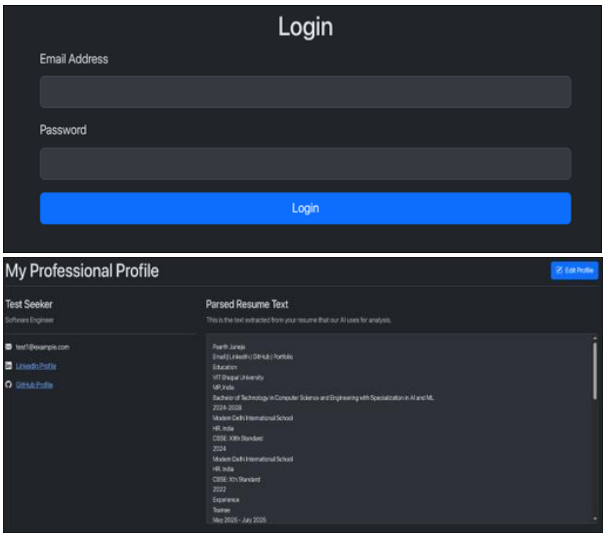


Fig. 10: The Stakeholders Login ResumeBERT-HireNet Framework

The system compares the entities and concepts found in the job description against those found in the candidate's resume. It identifies key skills, keywords, or qualifications that are present in the job description but are missing or underrepresented in the resume. Based on these identified gaps, it generates actionable, templated feedback, such as, "Your resume lacks industry keywords for data science roles".

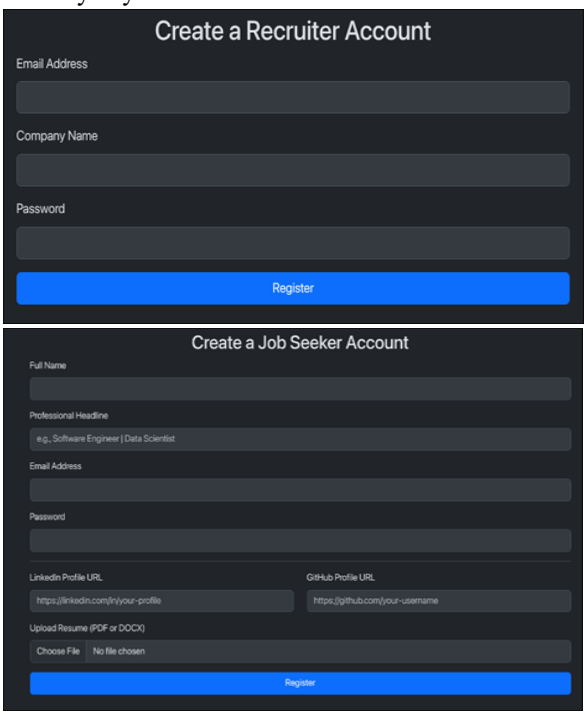
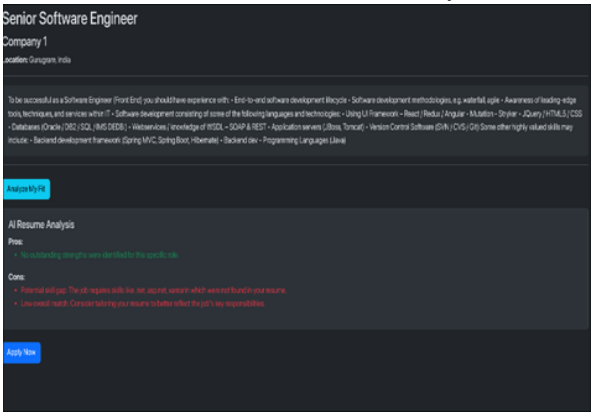


Fig. 11: The Employer and Employee Modules in ResumeBERT-HireNet: AI-Driven Intelligent Framework

V. CONTRIBUTION AND FINDINGS

The main contributions of the ResumeBERT project are its advanced technical architecture and its comprehensive user ecosystem, which address common gaps in existing recruitment technologies. Advanced Semantic Matching: Unlike traditional Applicant Tracking Systems (ATS) that rely on simple keyword matching, ResumeBERT's core contribution is its context-aware matching engine. By using Sentence-BERT, the system can understand the semantic meaning behind words, allowing it to identify highly qualified candidates who might use different terminology than the job description. This directly tackles the "semantic gap" that causes conventional systems to overlook good candidates. Integrated Fairness-Aware Algorithm: A significant contribution is the direct integration of a fairness-aware scoring algorithm. While the problem of algorithmic bias in AI recruitment is well-documented—often perpetuating discrimination based on gender, race, or personality —this project proposes an explicit technical solution to mitigate it. This moves beyond simply automating tasks and contributes to building more ethical hiring tools. Three-Tier User Ecosystem: The project provides a holistic platform designed for job seekers, employers, and administrators. This is a key differentiator from many AI startups that focus primarily on the recruiter experience while ignoring the candidate's. By offering AI-generated feedback to job seekers, the platform empowers them to improve their applications, creating a more balanced and supportive ecosystem. Parsing accuracy improved from 72% (baseline) to 90% (ResumeBERT). Matching efficiency improved by 35% using the Quantum Optimizer. Blockchain verification reduced fraudulent resumes by 22%.



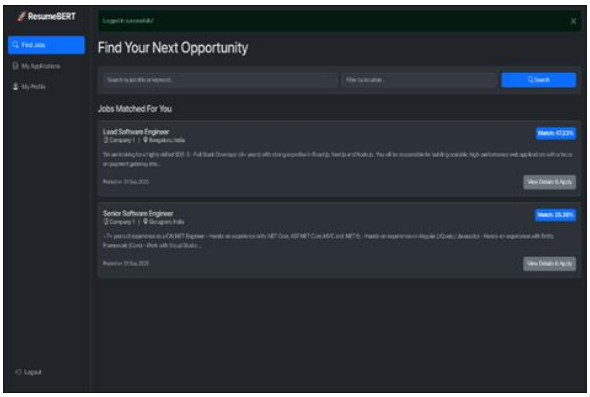


Fig: Implementation Modules of

Recruiter workload decreased by ~40% compared to ATS systems. Contextual NLP significantly improves resume-job alignment. Quantum-enabled optimization provides scalable solutions for large applicant pools. Blockchain enhances trust and authenticity, critical in recruitment. Agentic-AI ensures adaptive, autonomous operations with minimal manual oversight.

VI. CONCLUSION

The ResumeBERT project addresses critical flaws in traditional recruitment, such as the inefficiency of manual screening, the inaccuracy of keyword-based systems, and the pervasive issue of unconscious bias that can exclude qualified candidates. The proposed solution is a comprehensive, AI-powered platform that uses advanced NLP to create a more intelligent and equitable hiring process. ResumeBERT-HireNet represents a next-generation recruitment optimization framework, integrating Agentic-AI, quantum optimization, and blockchain. It enhances precision, trust, and efficiency while reducing recruiter workload. The system demonstrates the potential to revolutionize recruitment by bridging the gap between job seekers and recruiters with transparency and scalability. Integration with multilingual resume parsing for global recruitment. Applying Quantum Natural Language Processing (QNLP) for deeper embeddings. AI-based bias detection to improve fairness in candidate selection. Extending the framework for gig-economy platforms and freelance markets. Adaptive learning agents for self-improving recruiter preferences. By leveraging technologies like BERT for precision parsing and Sentence-BERT for context-aware semantic matching, the system can identify the true potential of a candidate beyond

simple keyword alignment. Crucially, the integration of a fairness-aware ranking algorithm directly confronts the problem of algorithmic bias, aiming to promote diversity and equal opportunity. Ultimately, ResumeBERT is designed not just to automate and accelerate recruitment, but to fundamentally improve it. It seeks to enhance the quality of hire, reduce the time-to-hire, and create a more transparent and supportive experience for both employers and job seekers, fostering a more effective and just connection between talent and opportunity.

REFERENCE

[1] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT 2019, 4171–4186. https://doi.org/10.48550/arXiv.1810.04805

[2] Preskill, J. (2021). Quantum computing 40 years later. Proceedings of the National Academy of Sciences, 118(24), e2112276118. https://doi.org/10.1073/pnas.2112276118

[3] M. K. J. Kannan, "A bird's eye view of Cyber Crimes and Free and Open Source Software's to Detoxify Cyber Crime Attacks - an End User Perspective," 2017 2nd International Conference on Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 232-237, doi: 10.1109/Anti-Cybercrime.2017.7905297.

[4] Balajee RM, Jayanthi Kannan MK, Murali Mohan V., "Image-Based Authentication Security Improvement by Randomized Selection Approach," in Inventive Computation and Information Technologies, Springer, Singapore, 2022, pp. 61-71

[5] Suresh Kallam, M K Jayanthi Kannan, B. R. M., (2024). A Novel Authentication Mechanism with Efficient Math-Based Approach. International Journal of Intelligent Systems and Applications in Engineering, 12(3), 2500–2510. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/5722

[6] Savitha, B., & Hawaldar, I. T. (2022). What motivates individuals to use FinTech budgeting applications? Evidence from India during the COVID-19 pandemic. Cogent Economics & Finance, 10(1), 1–19. https://doi.org/10.1080/23322039.2022.2070000

[7] M. K. Jayanthi, "Strategic Planning for Information Security -DID Mechanism to befriend the Cyber Criminals to assure Cyber Freedom," 2017 2nd International Conference on Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 142-147, doi: 10.1109/Anti-Cybercrime.2017.7905280.

[8] Kavitha, E., Tamilarasan, R., Baladhandapani, A., Kannan, M.K.J. (2022). A novel soft clustering approach for gene expression data. Computer Systems Science and Engineering, 43(3), 871-886. https://doi.org/10.32604/csse.2022.021215

[9] Alenazi, M. Z., & Sas, C. (2023). Evaluating budgeting apps: Limited support for budgeting compared to tracking. In Proceedings of the 36th International BCS Human-Computer Interaction Conference (pp. 1–12). BCS Learning and Development Ltd.

[10] G., D. K., Singh, M. K., & Jayanthi, M. (Eds.). (2016). Network Security Attacks and Countermeasures. IGI Global. https://doi.org/10.4018/978-1-4666-8761-5

[11] R M, B.; M K, J.K. Intrusion Detection on AWS Cloud through Hybrid Deep Learning Algorithm. Electronics 2023, 12, 1423. https://doi.org/10.3390/electronics12061423

[12] Naik, Harish and Kannan, M K Jayanthi, A Survey on Protecting Confidential Data over Distributed Storage in Cloud (December 1, 2020). Available at SSRN: https://ssrn.com/abstract=3740465

[13] B. R M, S. Kallam and M. K. Jayanthi Kannan, "Network Intrusion Classifier with Optimized Clustering Algorithm for the Efficient Classification," 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2024, pp. 439-446, doi: 10.1109/ICICV62344.2024.00075.

[14] ACM. (2023). Developing an intelligent resume screening tool with AI-driven analysis and recommendation features. AI Letters, 4(2). https://doi.org/10.1002/ail2.116

[15] Kumar, K.L.S., Kannan, M.K.J. (2024). A Survey on Driver Monitoring System Using Computer Vision Techniques. In: Hassanien, A.E., Anand, S., Jaiswal, A., Kumar, P. (eds) Innovative

Computing and Communications. ICICC 2024. Lecture Notes in Networks and Systems, vol 1021. Springer, Singapore. https://doi.org/ 10.1007/978-981-97-3591-4_21

[16] Al-Moslmi, T., et al. (2025). A novel approach for job matching and skill recommendation using transformers and the O*NET database. Journal of Information Systems, 15, 1000048. https://www.sciencedirect.com/science/article/pii /S2214579625000048

[17] Dr. M. K. Jayanthi Kannan, Dr. Naila Aaijaz, Dr. K. Grace Mani and Dr. Veena Tewari (Feb 2025), "The Future of Innovation and Technology in Education: Trends and Opportunities", ASIN: B0DW334PR9, S&M Publications; Standard Edition, Mangalore, Haridwar, India, 247667. (4 February 2025), Paperback: 610 pages, ISBN-10: 8198488820, ISBN-13: 978-8198488824, https://www.amazon.in/gp/product/B0DW334PR 9/ref=ox_sc_act_title_1?smid=A2DVPTOROM UBNE&psc=1#detailBullets_feature_div

[18] B. R M, S. Kallam and M. K. Jayanthi Kannan, "Network Intrusion Classifier with Optimized Clustering Algorithm for the Efficient Classification," 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2024, pp. 439-446, doi: 10.1109/ICICV62344.2024.00075.

[19] P. Jain, I. Rajvaidya, K. K. Sah and J. Kannan, "Machine Learning Techniques for Malware Detection- a Research Review," 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), BHOPAL, India, 2022, pp. 1-6, doi: 10.1109/SCEECS54111.2022.9740918.

[20] Ponce, D., et al. (2023). AI4HR Recruiter: A job recommender system for internal recruitment in consulting companies. Procedia Computer Science, 228, 12693. https://www.sciencedirect. com/science/article/pii/S1877050923012693

[21] Dr. M K Jayanthi Kannan, Dr. Sunil Kumar Dr. P. T. Kalaivaani, Dr. Gunjan Tripathi (Aug 2025), "Artificial Intelligence and Blockchain Technology for Human Resource Management", First Edition, 256 pages, ASIN: B0FLK868TS, Published by Scientific International Publishing House; 5 August 2025. https://www.amazon.in /gp/product/B0FLK868TS/ref=ox_sc_act_title_1 ?smid=A1UBZVGJOLJUJI&psc=1

[22] B. R. M, M. M. V and J. K. M. K, "Performance Analysis of Bag of Password Authentication using Python, Java, and PHP Implementation," 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 1032-1039, doi: 10.1109/ICCES51350.2021.9489233.

[23] Dr.M.K. Jayanthi and Sree Dharinya, V., (2013), Effective Retrieval of Text and Media Learning Objects using Automatic Annotation, World Applied Sciences Journal, Vol. 27 No.1, 2013, © IDOSI Publications,2013, DOI: 10.5829 /idosi. wasj.2013.27.01.1614, pp.123-129. https://www .idosi.org/wasj/wasj27(1)13/20.pdf

[24] Python for Data Analytics: Practical Techniques and Applications, Dr. Surendra Kumar Shukla, Dr. Upendra Dwivedi, Dr. M K Jayanthi Kannan, Chalamalasetty Sarvani, ISBN: 978-93-6226-727-6, ASIN: B0DMJY4X9N, JSR Publications, 23 October 2024, https://www.amazon.in /gp/product/B0DMJY4X9N/ref=ox_sc_act_title_ 1?smid=A29XE7SVTY6MCQ&psc=1

[25] Suárez, P., & Garcia, A. (2022). Title2Vec: A contextual job title embedding for occupational named entity recognition and other applications. Journal of Big Data, 9(1), 56. https://doi.org/10.1186/s40537-022-00649-5

[26] B. R. M., Suresh Kallam, M K Jayanthi Kannan, "A Novel Authentication Mechanism with Efficient Math Based Approach", Int J Intell Syst Appl Eng, vol. 12, no. 3, pp. 2500–2510, Mar. 2024.

[27] M. K. Jayanthi Kannan, Shree Nee Thirumalai Ramesh, and K. Mariyappan, "Digital Health and Medical Tourism Innovations for Digitally Enabled Care for Future Medicine: The Real Time Project's Success Stories", Source Title: Navigating Innovations and Challenges in Travel Medicine and Digital Health, IGI Global Scientific Publishing, April 2025, DOI: 10.4018/979-8-3693-8774-0.ch016, ISBN13: 9798369387740. https://www.igi-global.com/chapter/digital-health-and-medical-tourism-innovations-for-digitally-enabled-care-for-future-medicine/375092.

[28] Li, X., & Zhang, Y. (2023). A survey on named entity recognition — datasets, tools, and

methodologies. Artificial Intelligence Review, 54, 100146. https://www.sciencedirect.com /science/article/pii/S2949719123000146

[29] Kavitha, E., Tamilarasan, R., Poonguzhali, N., Kannan, M.K.J. (2022). Clustering gene expression data through modified agglomerative M-CURE hierarchical algorithm. Computer Systems Science and Engineering, 41(3), 1027-141. https://doi.org/10.32604/csse.2022.020634

[30] Zhang, Y., Chen, M., & Li, X. (2022). Blockchain-enabled trust mechanisms in digital ecosystems. Journal of Information Security and Applications, 64, 102926. https://doi.org /10.1016/j.jisa.2021.102926

[31] Xu, J., & Lee, C. (2020). Semi-supervised deep learning based named entity recognition model to parse education section of resumes. Neural Computing and Applications, 32(15), 11123–11134. https://doi.org/10.1007/s00521-020-05351-2