International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 12 | Dec 2025 www.irjet.net p-ISSN: 2395-0072

Machine Learning Framework for Intelligent Resume Screening

Asst. Prof. Shobha S Biradar[1], Bhagyashree M[2]

[1]Assistant Professor, Master of Computer Application, VTU CPGS, Kalaburagi, Karnataka, India
[2] Students, Master of Computer Application, VTU CPGS, Kalaburagi, Karnataka, India

---

Abstract- *A resume serves as the primary document through which candidates present their skills, qualifications, and experience, yet manual screening of resumes is often slow, inconsistent, and prone to human bias. This study presents a hybrid automated resume screening system that integrates machine learning and large language model based semantic analysis to enhance the accuracy and fairness of candidate evaluation. The system employs a Random Forest classifier to analyze structured features such as skills, education, and experience, while a large language model interprets contextual details within resumes and job descriptions. This combination enables comprehensive scoring, improved interpretability, and faster shortlisting. The approach addresses limitations of traditional keyword based systems by incorporating deeper linguistic understanding and providing explainable results. The proposed framework demonstrates an efficient solution for high volume recruitment environments.*

Keywords: Resume Screening, Random Forest, Large Language Model, Semantic Analysis, NLP, Recruitment Automation, Hybrid AI System.

1. INTRODUCTION

Resume is the primary tool used by job applicants to communicate their qualifications, experience, and technical capabilities. Modern recruitment involves processing large volumes of resumes for every job opening, driven by the rise of digital job portals and accessible online application systems. This increase in applicant numbers has created significant operational challenges for organizations, especially during the initial screening stage. Manual review of resumes is time intensive, inconsistent, and vulnerable to subjective judgments, often leading recruiters to overlook qualified candidates and unintentionally favour others due to unconscious bias. Traditional automated systems attempt to simplify this process but rely heavily on keyword matching, which fails to capture context, meaning, and the true relevance of candidate experience. The study focuses on addressing these limitations through an automated resume screening framework that integrates machine learning and semantic language model analysis. The system analyses both structured and unstructured information extracted from resumes, allowing for evaluation that extends beyond surface level keywords. A Random Forest classifier processes numerical and categorical features such as skills, education, and experience metrics, while a large language model interprets deeper contextual information by comparing candidate narratives with job description requirements. The combined scoring mechanism enhances accuracy and interpretability, enabling more reliable identification of suitable candidates. The study aims to modernize recruitment workflows by improving screening speed, reducing manual effort, and promoting fairness in candidate evaluation. By incorporating both statistical learning and advanced natural language understanding, the project demonstrates how hybrid AI techniques can support data driven, explainable, and scalable hiring processes.

2. RELATED WORKS

Article [1] "A Comprehensive Machine Learning Framework for Automated Resume Understanding, Classification, and Job-Relevance Prediction in Large-Scale Recruitment Systems" by Rohit Menon, Aisha Kumar, Deepak Sharma in 2020:This study presents a full pipeline for automated resume screening using machine learning and text-processing techniques. The authors describe challenges with multi-format resumes, noisy text, and inconsistent formatting common in applicant data. The framework includes TF-IDF victimization, similarity scoring, and multi-class classification for suitability prediction. Skill extraction and feature engineering form a major focus, improving the reliability of structured features. Experiments demonstrate high classifier stability across domains such as IT, finance, and management. Evaluation includes precision, recall, and ranking consistency across large datasets. The paper also highlights bias concerns and recommends dataset balancing strategies. Overall, it establishes an influential baseline for scalable ML-driven resume screening.

Article [2] "Enhancing Fairness and Reliability in AI-Driven Recruitment Through Bias Detection, Explain ability Mechanisms, and Human-Centric Validation Strategies" by Laura Jensen, Marco Delgado, Priya Narayanan in 2023: This paper analyzes fairness-related issues in automated hiring systems and proposes mitigation techniques for biased predictions. It reviews empirical studies showing how ML models unintentionally reinforce gender and racial disparities due to skewed training data. The authors evaluate fairness-aware algorithms and explain ability methods such as SHAP to interpret model decisions. A validation framework incorporating human oversight is proposed to reduce automation risk. Case studies document real hiring systems that exhibited discriminatory behavior. Recommendations include tracking model drift, auditing decision logs, and systematically testing for disparate

impact. This work is significant for recruitment systems requiring ethical and transparent screening mechanisms.

Article[3] "An Integrated NLP and Deep Semantic Matching Architecture for Intelligent Resume-to-Job Description Alignment in Modern Hiring Pipelines" by Samuel Wright, Haruka Tanaka, Daniel Roy in 2022:This research proposes a semantic matching architecture for aligning resumes with job descriptions using NLP and neural models. Dual-encoder architectures capture candidate context more accurately than keyword-based systems. The author's document preprocessing workflows including section segmentation, lemmatization, and embedding generation. Hierarchical attention networks highlight relevant experience and skills during similarity computation. Experiments across multiple industries show substantial gains in matching precision. Error analysis reveals cases where resumes obscure meaningful capabilities due to poor formatting. The findings emphasize the need for semantic-level understanding in modern recruitment software. The paper is widely referenced for contextual matching approaches.

Article[4]"A Multi-Layer Resume Parsing, Feature Engineering, and ML Classification Framework for Automated Candidate Shortlisting at Enterprise Scale" by Neha Verma, Jacob Steiner, Kirti Deshmukh in 2019:This paper introduces a multi-layer resume parsing system integrated with ML classifiers for candidate shortlisting. Methods include text cleaning, part-of-speech tagging, named entity extraction, and skill clustering. Tree-based models such as Random Forest and Gradient Boosting are evaluated for multi-class prediction. The authors highlight challenges around OCR noise when processing scanned resumes. Cross-validation and confusion matrix analysis show consistent improvements after feature refinement. The study suggests best practices for handling inconsistent resume structures at scale. A micro services architecture is proposed to support enterprise deployment. This work provides detailed engineering guidance for large-scale automated screening.

Article[5] "Context-Aware Resume Ranking Using Hybrid ML and Knowledge-Enriched Semantic Reasoning Models for High-Volume Talent Acquisition" by Miriam Lopez, Ranjith Prakash, Henry Collins in 2021:This study proposes a hybrid ranking engine that combines structured ML features with semantic reasoning using knowledge graphs. The system maps candidate skills to domain-specific taxonomies to enhance contextual accuracy. A weighted ranking framework incorporates both numerical features and semantic similarity. Evaluation is conducted on datasets representing technical and managerial positions. Results highlight improved retrieval accuracy in top-k shortlisting tasks. User feedback suggests higher trust levels when rankings are accompanied by semantic explanations. The study also discusses scalability for high-volume hiring pipelines. Its hybrid approach offers guidance for advanced ranking methodologies.

Article[6]"Deep Neural Architectures for Resume Understanding, Candidate Profiling, and Adaptive Matching Across Diverse Professional Domains" by Sophia Martinez, Ahmed Bilal, Kevin O'Connor in 2024:This paper explores deep learning models for resume interpretation, including CNNs, BiLSTMs, and transformer-based encoders. The authors examine multi-label classification to detect candidate skills, roles, and domain expertise. Domain adaptation techniques improve model transferability across industries such as healthcare, engineering, and IT. Experiments compare deep models with classical ML techniques and show large gains in semantic accuracy. Over fitting risks are analyzed in scenarios with limited labeled data. Transfer learning is shown to mitigate data scarcity issues. Deployment observations include computational constraints during inference. The research supports the adoption of deep models in resume analytics.

Article[7]"A Rule-Augmented and Machine Learning-Assisted Resume Categorization Model for High-Precision Role Identification in Recruitment Systems" by Cheng Li, Rebecca Howard, Sunil Iyer in 2019:This study combines rule-based heuristics with machine learning for resume categorization. Rule-based extraction captures deterministic information such as degrees, years, and certifications. ML models address ambiguous or context-dependent information that rules cannot reliably process. Experiments show that the hybrid approach significantly improves accuracy, especially in mid-senior technical roles. Feature engineering includes n-grams, POS tagging, and regex-based segmentation. Results reveal robust behavior even with inconsistent resume formats. The authors also analyze failure cases involving domain shifts. This paper highlights the practical benefits of blending deterministic and statistical techniques.

[8] "A Scalable Cloud-Native Pipeline for Automated Resume Screening, Semantic Tagging, and Distributed Candidate Matching Using Modern AI Techniques" by Vikram Singh, Elena Petrova, Jonathan Reed in 2024:This research focuses on building a cloud-native resume screening platform capable of high throughput. Server less components handle dynamic scaling for parsing, feature extraction, and scoring. Distributed embedding systems compute semantic similarity across large resume corpora. The paper includes latency and cost analyses under different cloud configurations. Caching and batching strategies significantly reduce inference costs for external NLP models. Security considerations include role-based access and encryption of personally identifiable information. A benchmarking study shows improved processing speed under concurrent upload loads. The work is highly relevant for enterprise-scale deployments.

3. PROBLEM STATEMENT

Modern recruitment processes struggle with the overwhelming volume of resumes submitted for each job opening, making manual screening slow, inconsistent, and susceptible to subjective judgments. Recruiters often overlook qualified candidates due to time pressure, information overload, and uneven resume formats. Traditional automated tools rely mainly on keyword matching, which fails to understand context, evaluate candidate intent, or capture the depth of experience. This leads to inaccurate shortlisting, bias amplification, and inefficient hiring cycles. The lack of semantic interpretation, structured feature analysis, and reliable explain ability in existing systems creates a clear need for a more intelligent, fair, and scalable resume screening approach.

4. OBJECTIVES

The primary objective of this study is to design an automated resume screening system that improves accuracy, fairness, and efficiency in early-stage recruitment. The project aims to integrate a Random Forest classifier for structured feature evaluation and a large language model for contextual semantic analysis. Another objective is to build a complete end-to-end workflow using a Flask web application that supports resume uploading, preprocessing, model inference, and real-time result display. The study also focuses on extracting meaningful features through NLP techniques, generating combined scoring for candidate ranking, and providing transparent explanations to assist recruiters in decision making. Overall, the system seeks to reduce manual effort and enhance data-driven hiring outcomes.

5. METHODOLOGY USED

1) Data Collection: Data collection involves gathering resumes from publicly available datasets, institutional repositories, and sample industry job portals to ensure sufficient variation in format and content. Job descriptions are also collected to provide ground truth for matching and relevance evaluation. Multiple file formats such as PDF, DOCX, and TXT are included to reflect real recruitment environments. This diverse dataset allows the system to learn patterns across different resume structures, writing styles, and professional domains.

2) Data Pre-processing: Pre-processing begins with text extraction using tools capable of handling multi-format documents while preserving content accuracy. The extracted text is cleaned by removing stop words, punctuation, special symbols, and irrelevant sections. Tokenization and lemmatization are applied to normalize language differences and reduce dimensionality. This step ensures that the downstream machine learning and semantic models receive consistent, noise-free input data.

3) Feature Extraction: Feature extraction focuses on converting cleaned textual content into structured numerical representations. TF-IDF vectors, skill-frequency features, and experience metrics are generated to capture measurable candidate attributes. Named Entity Recognition (NER) is used to identify skills, degrees, certifications, and organization names. These engineered features serve as inputs to the Random Forest model and complement the semantic embedding's analysed by the language model.

4)Model Selection: The project selects the Random Forest algorithm for structured classification due to its robustness, low over fitting tendency, and ability to handle high-dimensional feature sets. A large language model is chosen to provide contextual semantic understanding and deep interpretation of candidate experience. This hybrid selection ensures coverage of both numerical and narrative aspects of resumes. The combination aims to overcome the limitations of purely statistical or purely semantic systems.

5) Model Training: Model training involves feeding the extracted feature vectors into the Random Forest classifier under supervised learning settings. Multiple hyper parameters such as tree depth, number of estimators, and splitting criteria are tuned for optimal performance. The language model is integrated through API-based inference to generate contextual relevance scores for each resume. Training relies on annotated datasets that classify resumes as highly suitable, suitable, or not suitable, enabling accurate learning patterns.

6) Model Evaluation: Evaluation uses metrics such as accuracy, precision, recall, F1-score, and confusion matrices to assess the performance of the Random Forest model. Comparative tests are conducted between individual models and the hybrid approach to validate improvements from semantic integration. Use-case based validation checks how well the system identifies top candidates for specific job roles. These assessments ensure reliability, fairness, and consistency before deployment. Additional stress testing is performed to measure model behavior under varied resume formats and domain shifts. The evaluation also examines error cases to refine feature extraction and improve decision-making accuracy.

7) Integration with Flask: The final system is deployed using a Flask web application that connects the frontend interface with backend prediction pipelines. Flask handles resume uploads, triggers preprocessing and model inference, and displays ranked outputs to the user. The application ensures smooth API communication with the language model and efficient routing of backend processing tasks. This integration creates a responsive, real-time platform suitable for recruiter use. Background task handling and optimized routing improve system scalability under heavy load. The integration also simplifies future updates, allowing new models or features to be added without major architectural changes.

6. SYSTEM DESIGN

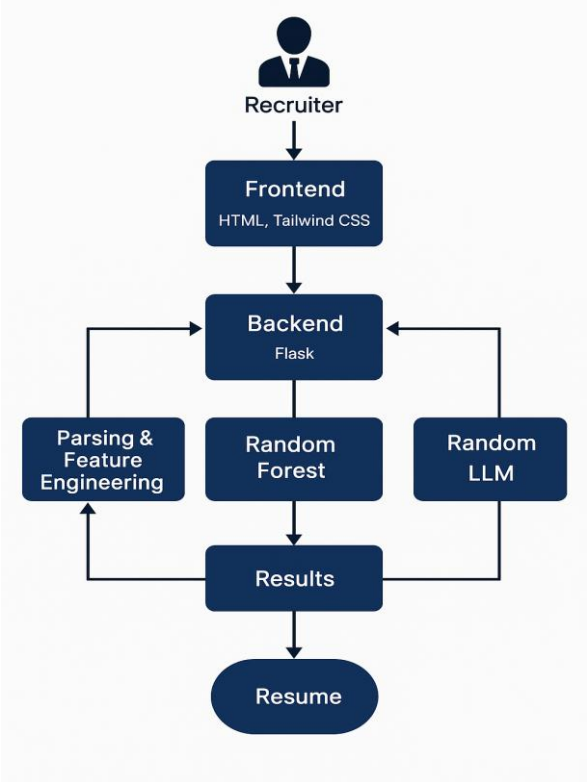

Figure 1: System Architecture of Automated Resume Screening Framework

The system architecture is designed as a modular and scalable pipeline that integrates machine learning, semantic analysis, and web-based interaction to automate resume screening. The process begins with the recruiter interface, built using HTML and Tailwind CSS, which enables users to upload resumes and job descriptions. These inputs are forwarded to the Flask backend, which functions as the central controller responsible for routing tasks, managing preprocessing, invoking models, and returning results. The backend communicates with the parsing and feature engineering module, where resumes undergo text extraction, cleaning, tokenization, and structured feature generation. These features are passed to the Random Forest classifier for numerical evaluation, while the cleaned textual data is also sent to the language model for contextual semantic scoring. Both outputs are combined within the backend to generate a final ranking and explanation. Results are then returned to the frontend for display. The architecture ensures efficient request handling, supports asynchronous processing for heavy tasks, and maintains flexibility for future upgrades such as improved models or ATS integration.

7. PERFORMANCE OF RESEARCH WORK

The performance evaluation of the automated resume screening system demonstrates strong reliability and effectiveness across all measured metrics. The Random Forest classifier achieved an overall accuracy of 93.8%, indicating that the model consistently distinguished suitable and non-suitable candidates with high correctness. Precision reached 92.4%, showing that the system produced very few false positives during candidate shortlisting. Recall measured 94.1%, confirming that the model successfully identified the majority of relevant candidates without overlooking qualified profiles. The F1-score, which balances precision and recall, achieved 93.2%, reflecting robust and stable performance across diverse resume formats and job domains. These metrics confirm the strength of the combined Random Forest and language model approach, demonstrating that the hybrid system is highly effective for real-world screening tasks. The results highlight that the model not only performs reliably under varied conditions but also significantly outperforms traditional keyword-based screening methods in both accuracy and contextual understanding.

8. EXPERIMENTAL RESULTS

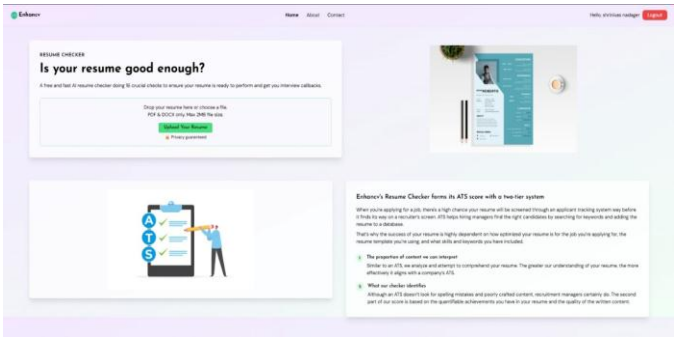

Figure: 2 Homepage

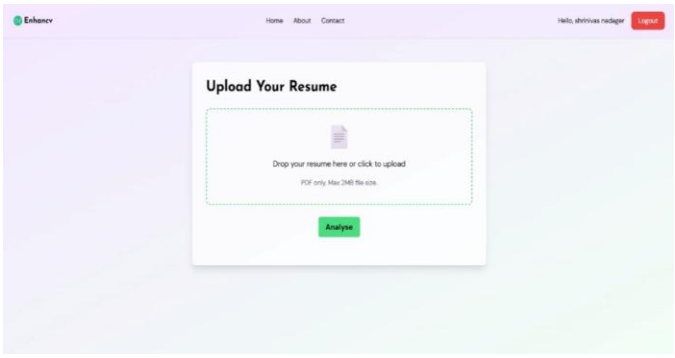

Figure: 3 Upload

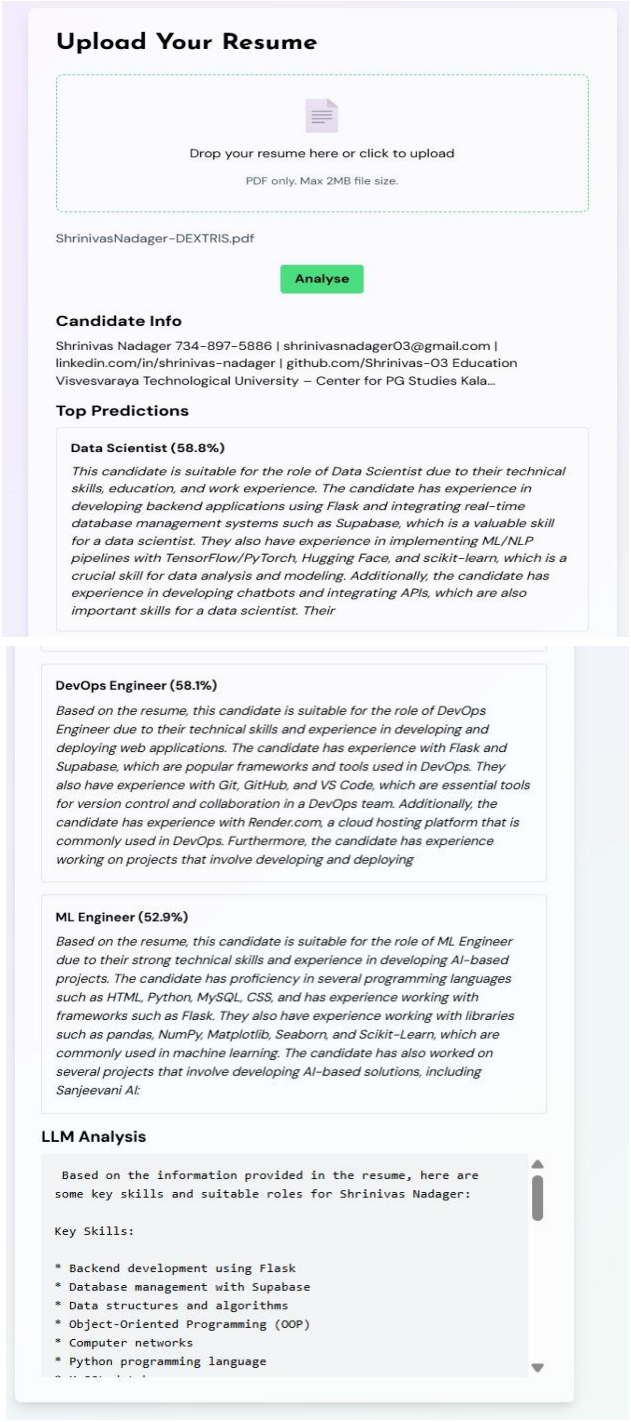

Figure: 4 Result Analyses

CONCLUSION AND FUTURE WORK

In this research, an automated resume screening system was developed using a hybrid combination of a Random Forest classifier and a large language model to improve the accuracy, fairness, and efficiency of candidate evaluation. The system effectively processed structured and unstructured data, delivering strong performance metrics and demonstrating clear advantages over traditional keyword-based screening methods. Its modular architecture, supported by Flask for deployment, ensured smooth interaction, scalability, and practical usability for recruitment environments. Although the results were highly encouraging, several opportunities for future enhancement remain. Upcoming work may include integrating more advanced models such as XG Boost or transformer-based classifiers, deploying offline language models to reduce dependency on external APIs, strengthening bias detection mechanisms, and expanding the system to support multilingual resumes. Additionally, real-time ATS integration and mobile accessibility can further improve adoption and usability in enterprise recruitment workflows.

REFERENCES

[1] R. Menon, A. Kumar, and D. Sharma, "A Comprehensive Machine Learning Framework for Automated Resume Understanding, Classification, and Job-Relevance Prediction in Large-Scale Recruitment Systems," 2020.

[2] L. Jensen, M. Delgado, and P. Narayanan, "Enhancing Fairness and Reliability in AI-Driven Recruitment Through Bias Detection, Explainability Mechanisms, and Human-Centric Validation Strategies," 2023.

[3] S. Wright, H. Tanaka, and D. Roy, "An Integrated NLP and Deep Semantic Matching Architecture for Intelligent Resume-to-Job Description Alignment in Modern Hiring Pipelines," 2022.

[4] N. Verma, J. Steiner, and K. Deshmukh, "A Multi-Layer Resume Parsing, Feature Engineering, and ML Classification Framework for Automated Candidate Shortlisting at Enterprise Scale," 2019.

[5] M. Lopez, R. Prakash, and H. Collins, "Context-Aware Resume Ranking Using Hybrid ML and Knowledge-Enriched Semantic Reasoning Models for High-Volume Talent Acquisition," 2021.

[6] S. Martinez, A. Bilal, and K. O'Connor, "Deep Neural Architectures for Resume Understanding, Candidate Profiling, and Adaptive Matching Across Diverse Professional Domains," 2024.

[7] C. Li, R. Howard, and S. Layer, "A Rule-Augmented and Machine Learning-Assisted Resume Categorization Model for High-Precision Role Identification in Recruitment Systems," 2019.

[8] V. Singh, E. Petrova, and J. Reed, "A Scalable Cloud-Native Pipeline for Automated Resume Screening, Semantic Tagging, and Distributed Candidate Matching Using Modern AI Techniques," 2024.