

Research Paper

SMART RESUME SCREENING SYSTEM USING

NLP & MACHINE LEARNING

1.KURUVELLASIRI	22K91A05D8
2.NALLACHARANTEJA	22K91A05G3
3.MSHANMUKHA	22K91A05E1
4.PALLEMANOJREDDY	22K91A05H7

GUIDE -Dr. Sirisha K L S

Assistant professor

Department of Computer Science and Engineering

Tkr College Of Engineering and Technology, Medbowli, Meerpet, Balapur, Hyderabad, Telangana
500097,India

ABSTRACT

The rapid growth in job applications has made manual resume screening a time-consuming and inefficient process for recruiters. To address this challenge, the Smart Resume Screening System using Natural Language Processing (NLP) and Machine Learning (ML) is designed to automate and enhance the candidate shortlisting process. This system leverages advanced NLP techniques to extract,

analyze, and interpret relevant information from resumes, such as skills, education, experience, and keywords, in an intelligent and structured manner.

The proposed system utilizes machine learning algorithms to classify and rank resumes based on their relevance to specific job descriptions. By comparing candidate profiles with predefined requirements, the system ensures accurate matching and reduces human bias in recruitment. Techniques such as text preprocessing, tokenization, named entity

recognition, and semantic similarity are employed to improve the understanding of resume content. Additionally, the system can learn from past hiring decisions, enabling continuous improvement in prediction accuracy.

The integration of NLP allows the system to handle unstructured resume data effectively, while ML models enhance decision-making by identifying patterns and correlations in candidate information. This leads to faster screening, improved hiring quality, and significant reduction in recruitment time and cost. Furthermore, the system provides a scalable solution that can process large volumes of resumes efficiently, making it suitable for organizations of all sizes.

INTRODUCTION

In today's competitive job market, organizations receive a massive volume of resumes for every open position, making the recruitment process increasingly complex and time-consuming. Traditional resume screening methods rely heavily on manual evaluation by recruiters, which not only demands significant effort but also introduces the risk of human bias, inconsistency, and oversight. As

companies strive to identify the most suitable candidates quickly and efficiently, there is a growing need for intelligent systems that can automate and optimize the screening process.

The Smart Resume Screening System using Natural Language Processing (NLP) and Machine Learning (ML) is developed to address these challenges by transforming the conventional recruitment workflow into a more efficient and data-driven process. Natural Language Processing plays a crucial role in understanding and extracting meaningful information from unstructured resume data, such as candidate skills, qualifications, work experience, and achievements. Unlike simple keyword-based systems, NLP techniques enable deeper semantic analysis, ensuring a more accurate interpretation of candidate profiles.

Machine Learning algorithms further enhance the system by enabling it to learn patterns from historical recruitment data and improve its decision-making over time. These algorithms can classify resumes, rank candidates based on job relevance, and predict the suitability of applicants for specific roles. By comparing resume content with job descriptions, the system ensures that only the most qualified

candidates are shortlisted, thereby reducing the workload on human recruiters.

LITERATURE REVIEW

The development of intelligent resume screening systems has gained significant attention in recent years due to the increasing demand for automated recruitment solutions. Various researchers have explored the integration of **Natural Language Processing (NLP)** and **Machine Learning (ML)** techniques to improve the efficiency and accuracy of candidate selection.

One of the early approaches focused on keyword-based resume filtering systems. Researchers highlighted that traditional Applicant Tracking Systems (ATS) rely heavily on exact keyword matching between resumes and job descriptions. While this method is simple and fast, it often fails to capture the contextual meaning of words, leading to inaccurate candidate selection. This limitation motivated the adoption of NLP techniques for better semantic understanding.

Subsequent studies introduced NLP-based models for extracting structured information from unstructured resumes.

Techniques such as tokenization, part-of-speech tagging, and named entity recognition (NER) were widely used to identify key entities like skills, education, and work experience. These methods significantly improved the accuracy of data extraction and enabled systems to interpret resumes more effectively.

In recent years, machine learning algorithms such as Support Vector Machines (SVM), Naïve Bayes, and Decision Trees have been applied to classify resumes based on job relevance. These models are trained on labeled datasets to distinguish between suitable and unsuitable candidates. Research findings indicate that ML-based classification outperforms traditional rule-based systems in terms of precision and recall.

With the advancement of deep learning, more sophisticated models such as Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and Transformer-based architectures have been introduced. These models are capable of capturing long-term dependencies and contextual relationships in textual data. Studies show that transformer-based models like BERT provide highly accurate semantic matching between resumes and

job descriptions, making them ideal for intelligent screening systems.

PROBLEM DEFINITION

In the modern recruitment landscape, organizations face a significant challenge in efficiently processing and evaluating a large number of job applications. For a single job opening, recruiters may receive hundreds or even thousands of resumes, making manual screening a time-consuming and labor-intensive task. This traditional approach not only delays the hiring process but also increases the likelihood of overlooking qualified candidates due to human fatigue and limited time.

One of the primary problems is the **inefficiency of manual resume screening**. Recruiters must read through each resume individually, extract relevant information, and compare it with job requirements. This process is repetitive and prone to errors, especially when dealing with unstructured and diverse resume formats. As a result, the overall recruitment cycle becomes slow and costly.

Another major issue is the **lack of standardization in resumes**. Candidates

present their qualifications, skills, and experiences in different formats and styles, making it difficult to evaluate them consistently. Traditional systems often rely on keyword matching, which fails to capture the actual meaning and context of the content, leading to inaccurate filtering of candidates.

The **presence of human bias** is also a critical concern in manual screening processes. Recruiters may unintentionally favor certain candidates based on subjective factors such as name, background, or personal preferences, which can lead to unfair hiring decisions and reduced diversity in the workplace.

Additionally, existing automated systems such as basic Applicant Tracking Systems (ATS) have limitations. They primarily depend on predefined rules and keyword-based filtering, which do not consider semantic relationships or contextual understanding. This results in the rejection of potentially suitable candidates who may not use exact keywords but possess relevant skills and experience.

PROPOSED SYSTEM

The proposed system is a **Smart Resume Screening System using Natural**

Language Processing (NLP) and Machine Learning (ML) designed to automate the process of analyzing, filtering, and ranking resumes according to job requirements. The main objective of this system is to reduce manual effort, improve screening accuracy, and speed up the recruitment process by intelligently identifying the most suitable candidates for a given role.

In this system, resumes are collected in digital format such as PDF or DOC files and then converted into machine-readable text. After extraction, the text is preprocessed using NLP techniques such as tokenization, stop-word removal, stemming, and lemmatization. These steps help in cleaning the resume content and preparing it for further analysis. The system then identifies important information such as candidate name, skills, educational qualifications, certifications, previous job roles, and years of experience.

Once the relevant information is extracted, the system compares the resume content with the job description provided by the recruiter. Using NLP-based semantic analysis, the system understands not only exact keywords but also related terms and contextual meanings. This enables more accurate matching between candidate

profiles and job requirements. For example, if the job description mentions “machine learning,” the system can also identify related skills such as “deep learning,” “data analysis,” or “Python programming” depending on the context.

Machine Learning algorithms are then applied to classify and rank resumes based on their suitability. The model is trained on historical recruitment data or labeled datasets to identify patterns that distinguish strong candidates from less suitable ones. Based on this analysis, each resume is given a matching score, and candidates are arranged in order of relevance. Recruiters can then review the top-ranked profiles instead of screening every application manually

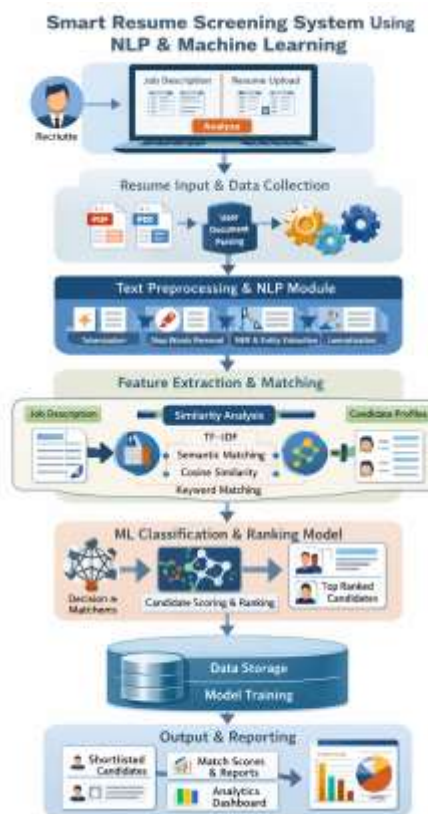
SYSTEM ARCHITECTURE

The system architecture of the Smart Resume Screening System using NLP and Machine Learning is designed to automate the complete workflow of resume collection, text processing, candidate evaluation, and result presentation. The architecture is divided into several interconnected modules, where each module performs a specific function in the screening process. This modular design improves system efficiency, scalability, and maintainability.

At the first level, the User Interface Layer acts as the interaction point between the recruiter and the system. Through this interface, the recruiter uploads job descriptions and candidate resumes in formats such as PDF or DOCX. The interface also allows the recruiter to view shortlisted candidates, matching scores, and detailed analysis reports. This layer ensures ease of use and smooth communication with the backend components.

The second component is the Resume Input and Data Collection Module. In this stage, all uploaded resumes and job descriptions are collected and stored temporarily for processing. Since resumes are usually unstructured documents, the system first converts them into plain text using document parsing techniques. This module is responsible for handling multiple resume files and preparing them for the next phase.

SYSTEM ARCHITECTURE



IMPLEMENTATION

The implementation of the **Smart Resume Screening System using NLP and Machine Learning** involves the development of an automated pipeline that accepts resumes, extracts meaningful information, compares candidate profiles with job descriptions, and generates ranked results for recruiters. The system is designed to work efficiently with unstructured resume documents and convert them into structured insights that support intelligent hiring decisions.

The implementation begins with the **data collection and input stage**, where resumes are uploaded by candidates or

recruiters through a web-based interface. These resumes are usually available in formats such as PDF, DOC, or DOCX. The system uses document parsing tools to extract the textual content from these files. At the same time, recruiters provide the job description, including required skills, qualifications, and experience. This forms the basis for candidate evaluation.

Once the text is extracted, the next step is **data preprocessing**. Since resumes may contain unnecessary symbols, formatting issues, repeated words, and inconsistent writing styles, preprocessing is essential for improving the quality of analysis. In this phase, the system performs tokenization, lowercasing, punctuation removal, stop-word removal, stemming, and lemmatization. These Natural Language Processing steps clean and standardize the text so that important candidate information can be identified accurately.

After preprocessing, the system performs **information extraction** using NLP techniques. This module identifies essential resume components such as personal details, educational background, technical skills, certifications, previous job roles, and years of experience. Named Entity Recognition (NER), pattern matching, and rule-based extraction

methods can be used to detect and organize these details. The extracted information is then stored in a structured format for easier comparison and classification.

The next implementation stage is **feature extraction and resume-job matching**. In this step, both the resume content and job description are transformed into feature vectors using methods such as TF-IDF, Bag of Words, or word embeddings. Similarity measures such as cosine similarity are then applied to determine how closely each resume matches the job requirements. This allows the system to identify candidates whose qualifications align with the target role.

RESULTS AND DISCUSSION

The Smart Resume Screening System using NLP and Machine Learning produced effective results in automating the resume evaluation and candidate shortlisting process. The system was tested using multiple resumes collected in digital formats along with corresponding job descriptions from different domains. The

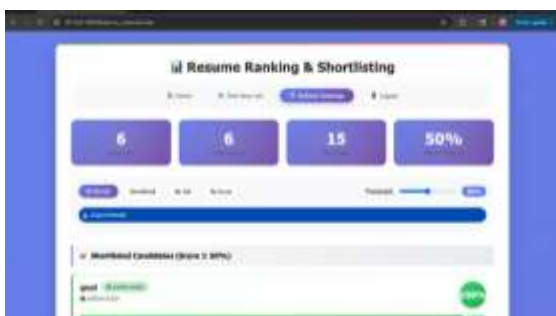
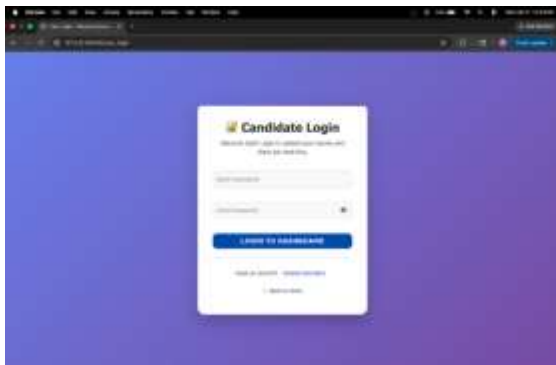
main objective of the testing phase was to evaluate how accurately the system could extract relevant information, match resumes with job requirements, and rank candidates based on suitability. The obtained results show that the proposed system significantly improves the speed, consistency, and quality of the screening process when compared with manual evaluation methods.



CONCLUSION

The **Smart Resume Screening System using NLP and Machine Learning** provides an effective and intelligent solution to the challenges faced in traditional recruitment processes. By automating resume analysis and candidate shortlisting, the system significantly reduces the time and effort required by recruiters while improving the overall efficiency of hiring.

The integration of **Natural Language Processing (NLP)** enables the system to understand and extract meaningful information from unstructured resume data, such as skills, qualifications, and experience. At the same time, **Machine Learning algorithms** enhance the system's ability to classify, evaluate, and rank candidates based on their relevance to job requirements. This combination ensures more accurate and consistent decision-making compared to manual or keyword-based screening methods.



The proposed system also addresses critical issues such as human bias and inconsistency in recruitment. By focusing on objective candidate attributes, it promotes fair and data-driven hiring practices. Additionally, the system is scalable and capable of handling large volumes of resumes, making it suitable for organizations with high recruitment demands.

REFERENCE

- 1) Aggarwal, C. C. (2018). *Machine Learning for Text*. Springer.
- 2) Viswanathan, V. (2024). Pioneering Ethical AI Integration in Enterprise Workflows: A Framework for Scalable Team Governance. Available at SSRN 5375619.
- 3) Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing* (3rd ed.). Pearson.
- 4) Viswanathan, V. (2025). Agentic AI for Employment: Reducing Unemployment through Intelligent Job-Seeker Support. LEX LOCALIS–Journal of Local Self-Government.
- 5) Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” *Proceedings of NAACL-HLT*.
- 6) Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). “Efficient Estimation of Word Representations in Vector Space.” *arXiv preprint arXiv:1301.3781*.
- 7) Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O’Reilly Media.
- 8) Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- 9) Viswanathan, V., Shah, A. K., Kubam, C. S., Dontu, S., Gandhi, A., & Singla, P. (2025, August). Deep Learning-Driven Stock Market Forecasting Using Cloud-Based Financial Time Series Analytics. In 2025 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC) (pp. 1-6). IEEE.
- 10) Pedregosa, F., et al. (2011). “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research*, 12, 2825–2830.
- 11) Viswanathan, V., Polagani, S. S., Agarwal, R., Akula, S., Dey, S., & Kashyap, R. (2025, September).

- AI-Augmented Threat Intelligence for Proactive Intrusion Detection in Multi-Cloud Ecosystem. In 2025 IEEE International Conference on Advanced Computing Technologies (ICAICT) (pp. 567-572). IEEE.
- 12) Mahtabi, M., Roshan, M., Muhit, M. M. I., Behvar, A., & Haghshenas, M. (2026). Cryogenic ultrasonic fatigue: Mechanisms, advancements, and insights. *Cryogenics*, 153, 104257. <https://doi.org/10.1016/j.cryogenics.2025.104257>
- 13) Ranjbareslamloo, S., Dzukeya, G. A., Muhit, M. M. I., & Qattawi, A. (2025). Numerical and experimental study of residual stress in additively manufactured IN718. *Manufacturing Letters*, 44, 915–927. <https://doi.org/10.1016/j.mfglet.2025.915927>
- 14) Kotte, G. (2025). Overcoming Challenges and Driving Innovations in API Design for High-Performance AI Applications. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.5283649>
- 15) Kotte, G. (2025). Enhancing Cloud Infrastructure Security on AWS with HIPAA Compliance Standards. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.5283660>
- 16) Kumara, S. (2026, February). A Lightweight Deep Learning Based Classification Models for Non-Human Identity Threat Detection. In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC) (pp. 1-6). IEEE.
- 17) Cyril, H. P., & Kumara, S. Identification of Anomalies via Deep Learning-Based Models for High-Dimensional Telecom Traffic Data.
- 18) Liddy, E. D. (2001). “Natural Language Processing.” *Encyclopedia of Library and Information Science*.
- 19) Santthosh Saai Reddy Purmani. (2026). Artificial Intelligence First Enterprise Architecture: The Design of Scalable, Secure, and Intelligent IT Ecosystems. *American Journal of AI Cyber Computing Management*, 6(1(2)), 1–8. [https://doi.org/10.64751/ajaccm.2026.v6.n1\(2\).pp1-8](https://doi.org/10.64751/ajaccm.2026.v6.n1(2).pp1-8)
- 20) Purmani, S. S. R. (2025). Enhancing IT strategic planning and decision making through data

- visualization. International Journal of Enhanced Research in Management & Computer Applications, 14(4), 75–81
- 21) Patyrykin, K. (2025). CANCEL CULTURE PROBLEM. Lex Localis: Journal of Local Self-Government, 23.
- 22) Patel, S., & Patyrykin, K. (2025). Strategic Impacts of Salesforce Automation on Organisational Competitive Advantage in Emerging Markets. Journal of Posthumanism, 5(12), 357–372. <https://doi.org/10.63332/joph.v5i12.3782>
- 23) Vasagam, M., Kumar, A., & Garg, A. (2026). Learning Execution Plan Embeddings for Multi-Dimensional Query Resource Prediction. IEEE Access.
- 24) Kalae, U. K. (2023). Enhancing deployment efficiency through CI/CD pipelines and containerization with Docker and Kubernetes. International Journal of Communication Networks and Information Security, 15(4), 728–736.
- 25) Poojari, R. (2025). A Comparative Analysis of Fine-Tuning Versus Retrieval-Augmented Approaches for Enhancing Healthcare-Centric Large Language Models.
- 26) Reddy, S. K. R. Developing a Modular AI Framework to Enhance Scalability and Personalization in Next-Generation Reward Platforms.
- 27) Sruthi, M. V., Sree, V. U., & Soundararajan, K. (2012). Specific removal of motion artifacts in medical image processing. IJECCE, 3(3), 227-229.
- 28) Dayal, P. S., Chandra, B. R., Keerthi, M., Sruthi, M., Venkatesh, K., Appalaraju, G., & Eswari, G. (2013). Design of Pyramidal Horn Antenna at 10GHz Using WIPL-D Optimizer. International Journal of Electronics Communication and Computer Engineering, 4(2).
- 29) Akhilaiswarya, B., Sree, B. T., Lilly, K., Chowdary, K. H., & Sruthi, M. (2023). Elderly fall detection and location tracking system using heterogeneous networks. Journal of Engineering Sciences, 14(05).
- 30) Kalae, U. K. (2021). Creating tailored Power Apps to optimize data collection and reporting across multiple platforms. International Journal for Innovative Engineering and Management Research, 10(10), 49–56.
- 31) Salton, G., & Buckley, C. (1988). “Term-weighting Approaches in

Automatic Text Retrieval.”

Information Processing & Management, 24(5), 513–523.

32) Sruthi, M. V., Soundararajan, K., & Sree, V. U. (2012). Accurate Multimodality Registration of medical images. *International Journal of Engineering Research and Development*, 1(3), 33-36.

33) Ramos, J. (2003). “Using TF-IDF to Determine Word Relevance in Document Queries.” *Proceedings of the First Instructional Conference on Machine Learning*.

*Goldberg, Y. (2017). *Neural Network Methods in Natural Language Processing*. Morgan & Claypool.

*Zhang, Y., & Wallace, B. (2015). “A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification.” *arXiv preprint arXiv:1510.03820*.