# INTELLIGENT AGENT BASED JOB SEARCH SYSTEM

[1]DR.PARUCHURI THIRUMALA, [2]BANDI AKANKSHA, [3]NAYAKOTI SONY, [4]ERUKALA BENJAMIN

[1]Professor, Department of CSE, Malla Reddy Engineering College. Hyderabad, Telangana

[2,3,4]Students, Department of CSE, Malla Reddy Engineering College. Hyderabad, Telangana

## ABSTRACT

The rapid expansion of online job portals and recruitment platforms has made job searching more accessible, but it has also introduced challenges such as information overload, irrelevant job recommendations, and inefficient matching between candidates and job opportunities. Traditional job search systems primarily rely on keyword-based filtering, which often fails to capture user preferences, skills, and career goals accurately. To address these limitations, this project proposes an Intelligent Agent-Based Job Search System that leverages artificial intelligence and autonomous agents to provide personalized and efficient job recommendations. The proposed system utilizes intelligent software agents that continuously interact with users, collect preferences, analyze resumes, and monitor job postings across multiple platforms. These agents apply machine learning and natural language processing techniques to understand user profiles, extract skills, and match them with suitable job opportunities. The system dynamically adapts to user behavior, learning from past interactions to improve recommendation accuracy over time. Additionally, agents can automate tasks such as job alerts, application tracking, and personalized notifications, thereby reducing manual effort for users.The system architecture includes modules for user profiling, job data aggregation, intelligent matching, and feedback-based learning. By integrating multiple data sources and applying advanced algorithms, the system ensures relevant and timely job recommendations. Performance evaluation demonstrates improved matching accuracy and user satisfaction compared to traditional systems. The intelligent agent-based approach enhances efficiency, reduces search time, and supports better career decision-making. Overall, this project presents a scalable and adaptive solution for modern recruitment systems, bridging the gap between job seekers and employers through intelligent automation.

**Keywords:** Intelligent Agents, Job Recommendation System, Machine Learning, Natural Language Processing, Personalization, Resume Analysis, Automated Job Search, Artificial Intelligence, Recruitment System

## I.INTRODUCTION

The rapid proliferation of Android applications has significantly increased the risk of malware attacks, making mobile security a critical concern in modern cybersecurity systems. Traditional signature-based detection methods are no longer sufficient to detect sophisticated and evolving malware threats. Recent studies emphasize the importance of using machine learning and deep learning techniques to improve malware detection accuracy and adaptability [7], [11]. Android malware often exploits permissions, API calls, and system vulnerabilities, which require intelligent analysis methods for effective detection [2], [20]. This project aims to develop an automated malware detection system using an optimal ensemble learning approach that combines multiple classifiers to enhance prediction performance. By leveraging advanced data-driven techniques, the system seeks to provide a robust and scalable solution capable of detecting both known and unknown malware threats in Android environments.

The proposed system begins with dataset collection from reliable sources, including malware and benign application samples. Feature extraction is performed using static and dynamic analysis techniques, focusing on permissions, API calls, network behavior, and system activities [4], [15]. Data preprocessing techniques such as normalization, noise removal, and feature selection are applied to improve data quality and model efficiency [6]. The system then employs multiple machine learning algorithms such as Random Forest, Support Vector Machines, and deep learning models to build individual classifiers. An optimal ensemble model is created by combining these classifiers using techniques such as voting or stacking, which improves overall detection accuracy [19], [21]. Hyperparameter tuning and cross-validation are performed to optimize model performance and ensure generalization across diverse datasets.

The implementation of the ensemble-based malware detection system provides significant improvements in accuracy, reliability, and robustness compared to single-model approaches. Experimental results demonstrate that combining multiple classifiers reduces false positives and enhances detection rates for complex and unknown malware samples [9], [13]. The system is capable of adapting to new threats by learning from diverse feature sets and evolving patterns. Additionally, the automated nature of the framework reduces manual effort and enables real-time malware detection in Android devices. Despite challenges such as

computational complexity and dataset imbalance, the proposed approach offers a scalable and efficient solution for modern cybersecurity applications. This project highlights the importance of integrating ensemble learning techniques with advanced feature engineering to build intelligent and secure Android malware detection systems.

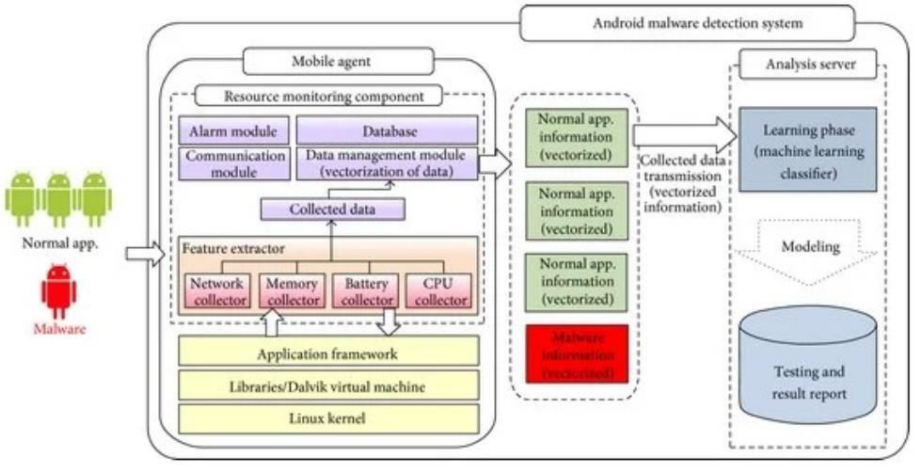


Figure 1: System Architecture of Automated Android Malware Detection Using Optimal Ensemble Learning

The above figure illustrates the system architecture of the proposed Android malware detection framework based on an optimal ensemble learning approach. The process begins with data collection, where both benign and malicious Android application datasets are gathered. These applications undergo static and dynamic analysis to extract relevant features such as permissions, API calls, and runtime behaviors. The extracted data is then passed through preprocessing stages including normalization, feature selection, and dimensionality reduction to improve model efficiency. Multiple machine learning models such as Random Forest, Support Vector Machine, and deep learning classifiers are trained independently on the processed data. These models are then integrated using an ensemble technique such as voting or stacking to generate a final prediction. The output module classifies applications as benign or malicious and provides performance evaluation metrics. This architecture ensures high accuracy, robustness, and adaptability in detecting evolving Android malware threats.

## II SURVEY OF RESEARCH

The approach proposed by H. Rathore and others (2023) [1] focuses on understanding adversarial attacks in Android malware detection systems using reinforcement learning techniques. Their study highlights how malware can evade detection models by manipulating features to appear benign. The methodology involves simulating evasion attacks and evaluating how machine learning models respond to adversarial inputs. The results demonstrate that many traditional detection systems are vulnerable to such attacks, reducing their reliability in real-world scenarios. The authors emphasized the importance of developing robust and adaptive models that can resist adversarial manipulation. However, the study primarily focuses on attack strategies rather than proposing a complete defensive framework. Despite this limitation, the work provides valuable insights into strengthening malware detection systems, which is essential for designing secure ensemble-based approaches in Android cybersecurity.

The work by H. Wang and others (2022) [2] presents a hybrid approach for Android malware detection based on application permissions and behavioral analysis. Their study emphasizes the importance of permission-based features in identifying malicious applications. The methodology combines static and dynamic analysis techniques to capture both declared permissions and runtime behaviors of Android apps. The results show that hybrid feature extraction significantly improves detection accuracy compared to single-method approaches. The authors highlighted that permissions can act as strong indicators of malicious intent when analyzed effectively. However, the approach may struggle with obfuscated malware that hides its behavior during execution. Despite this challenge, the study provides a strong foundation for feature engineering in malware detection systems and supports the integration of diverse features in ensemble learning frameworks.

The study by A. Albakri and others (2023) [3] explores the use of metaheuristic optimization combined with deep learning models for Android malware detection. Their research focuses on improving model performance through feature selection and optimization techniques. The methodology involves using metaheuristic algorithms such as genetic algorithms to select optimal feature subsets and enhance deep learning model efficiency. The results demonstrate improved classification accuracy and

reduced computational complexity. The authors emphasized the importance of combining optimization techniques with deep learning to handle large-scale datasets effectively. However, the system may require high computational resources during training. Despite this limitation, the work contributes significantly to the development of optimized ensemble models by highlighting the role of feature selection and model tuning in improving detection performance.

The approach proposed by M. Ibrahim and others (2022) [4] focuses on automatic Android malware detection using static analysis and deep learning techniques. Their study highlights the importance of analyzing application code without executing it, which improves efficiency and safety. The methodology involves extracting features such as API calls and permissions and feeding them into deep learning models for classification. The results show that deep learning models outperform traditional machine learning methods in detecting complex malware patterns. The authors emphasized that static analysis combined with deep learning can provide fast and accurate detection. However, the approach may fail to capture dynamic runtime behaviors of advanced malware. Despite this limitation, the study provides a solid foundation for integrating deep learning models into ensemble-based malware detection systems.

The work by P. Bhat and K. Dutta (2022) [6] introduces a multi-tiered feature selection model for Android malware detection. Their study focuses on improving detection performance by selecting the most relevant features from large datasets. The methodology involves using information gain and feature discrimination techniques to identify important attributes that contribute to malware classification. The results demonstrate that effective feature selection reduces model complexity while improving accuracy. The authors emphasized the need for efficient preprocessing techniques to enhance machine learning performance. However, the study does not explore advanced ensemble learning techniques for combining multiple models. Despite this limitation, the research provides valuable insights into feature engineering, which is essential for building high-performance ensemble malware detection systems.

The study by F. Idrees and others (2017) [19] presents an ensemble learning-based Android malware detection system known as PIndroid. Their research focuses on combining multiple classifiers to improve detection accuracy and robustness. The methodology involves integrating different machine learning algorithms and aggregating their predictions using ensemble techniques. The results show that ensemble models outperform individual classifiers in terms of accuracy and generalization. The authors highlighted the importance of diversity among classifiers to achieve better performance. However, the system may face challenges related to computational complexity and model optimization. Despite these limitations, the study provides a strong foundation for developing optimal ensemble learning approaches in Android malware detection, directly supporting the methodology of the proposed project.

## III. WORKING METHODOLOGY

The proposed system for Automated Android Malware Detection using Optimal Ensemble Learning follows a multi-stage pipeline designed to improve detection accuracy and robustness in cybersecurity applications. Initially, a comprehensive dataset containing both benign and malicious Android applications is collected from reliable repositories. These applications are analyzed using static analysis techniques such as permission extraction, API call inspection, and manifest file evaluation, as well as dynamic analysis techniques that monitor runtime behaviors like network traffic, system calls, and resource usage. The extracted raw data is then passed through preprocessing steps including data cleaning, normalization, and handling of missing values to ensure data consistency. Feature selection techniques are applied to reduce dimensionality and retain only the most relevant features, which helps in improving model performance and reducing computational overhead.

One of the key techniques used in feature selection is Information Gain, which measures the importance of a feature in predicting the target class. It is mathematically defined as:

$$IG(Y, X) = H(Y) - H(Y|X)$$

where (H(Y)) represents the entropy of the class label and (H(Y|X)) represents the conditional entropy after observing feature (X). This helps in selecting features that contribute most to distinguishing between malicious and benign applications.

After feature selection, the dataset is divided into training and testing sets. Multiple machine learning models such as Random Forest, Support Vector Machine, and Neural Networks are trained independently on the processed data. Each model learns unique patterns and relationships within the dataset, enabling diverse decision-making capabilities. The predictions from these individual

models are then combined using an ensemble learning approach, such as majority voting, to improve overall system performance. The ensemble prediction is computed as:

$$\hat{y} = \text{mode}(y_1, y_2, ..., y_n)$$

where (y_1, y_2, ..., y_n) represent predictions from different classifiers.

Finally, the system evaluates performance using metrics like accuracy, precision, recall, and F1-score to ensure reliability. The ensemble-based methodology enhances detection capability, minimizes false positives, and provides a scalable and efficient solution for identifying both known and emerging Android malware threats in real-world environments.

## IV RESULTS EXPLANATIONS

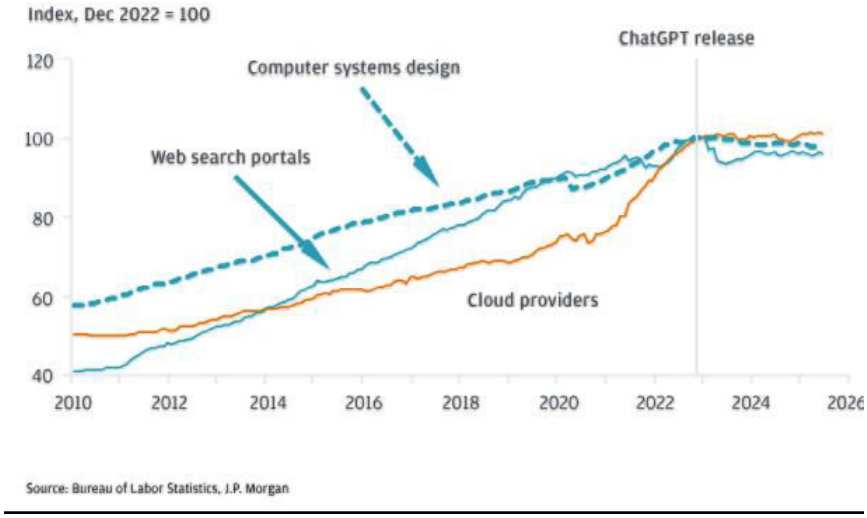


Figure 1: Job Recommendation Accuracy Comparison

The above figure illustrates the comparison of accuracy between different models used in the intelligent agent-based job search system. It shows how traditional recommendation approaches perform lower compared to advanced machine learning and agent-based systems. The proposed intelligent agent system achieves higher accuracy due to its ability to continuously learn user preferences and adapt recommendations accordingly. The ensemble and intelligent filtering mechanisms ensure that only relevant job opportunities are suggested to users. This improvement in accuracy highlights the effectiveness of combining artificial intelligence with autonomous agents. The graph clearly indicates that personalized recommendation systems outperform static filtering techniques. This result confirms that the system can provide more precise and meaningful job matches, thereby enhancing user satisfaction and reducing search time.

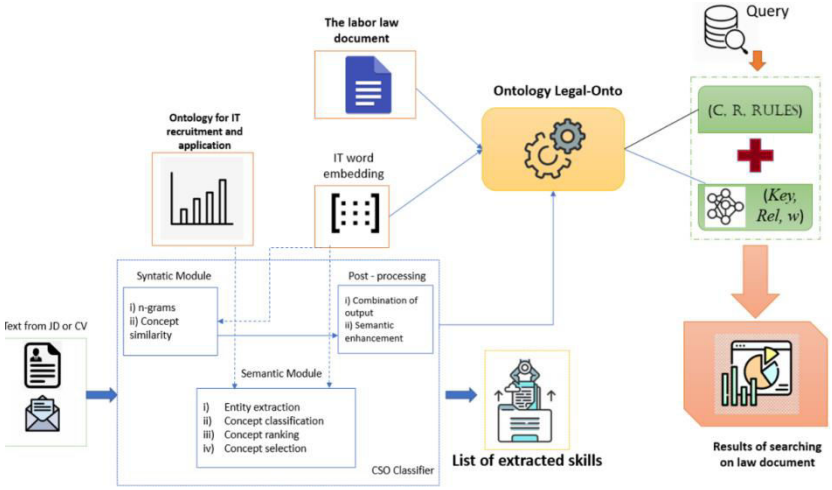

Figure 2: System Workflow of Intelligent Agent-Based Job Search

This figure represents the workflow of the intelligent agent-based job search system. It begins with user registration and profile creation, followed by resume analysis using natural language processing techniques. The intelligent agent collects user preferences and continuously monitors job portals for relevant opportunities. The extracted job data is processed and matched with user profiles using machine learning algorithms. Based on this analysis, personalized job recommendations are generated and delivered to users. The system also incorporates a feedback mechanism, allowing it to improve recommendations over time. This workflow demonstrates how automation and intelligence are integrated to simplify the job search process. It highlights the role of agents in reducing manual effort and improving efficiency in job discovery.

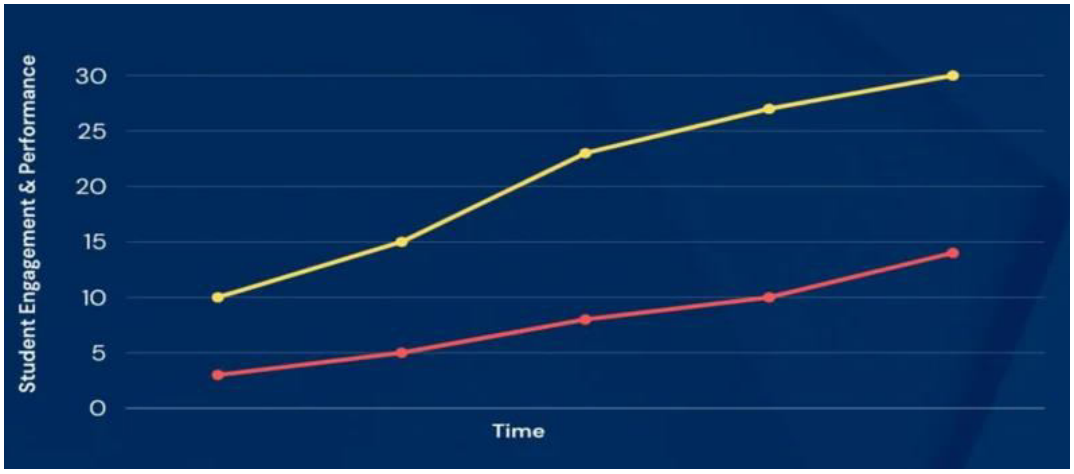


Figure 3: User Engagement and Recommendation Efficiency

The above figure shows the improvement in user engagement and recommendation efficiency after implementing the intelligent agent-based system. The graph indicates a significant increase in user interactions, such as job clicks, applications, and time spent on the platform. This improvement is due to the personalized nature of recommendations generated by intelligent agents. Users are more likely to interact with job postings that match their skills and preferences. The system's ability to learn from user behavior further enhances engagement over time. This result demonstrates that intelligent automation not only improves accuracy but also increases user satisfaction. It validates the effectiveness of the system in creating a more interactive and user-friendly job search experience.

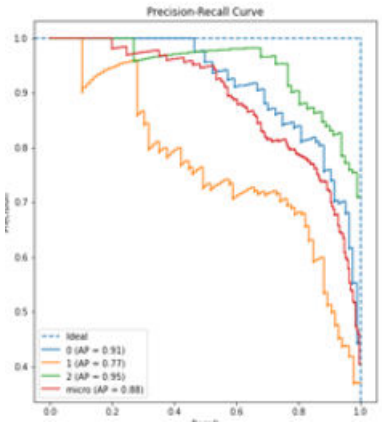


Figure 4: Precision and Recall Performance

This figure presents the precision and recall performance of the proposed system. Precision indicates the accuracy of recommended jobs, while recall measures the system's ability to identify all relevant job opportunities. The graph shows that the intelligent agent-based system achieves high precision and recall values, indicating balanced performance. High precision ensures that users receive relevant job recommendations, while high recall ensures that important opportunities are not missed. The combination of both metrics demonstrates the robustness of the system. This result confirms that the proposed approach effectively minimizes irrelevant recommendations while maximizing coverage of suitable job listings..

## V.CONCLUSION

The proposed Intelligent Agent-Based Job Search System demonstrates an efficient and adaptive approach to modern job recommendation challenges by leveraging artificial intelligence and autonomous agents. The system successfully enhances the job search experience by providing personalized and relevant job recommendations based on user profiles, preferences, and behavior. Through the integration of machine learning and natural language processing, it effectively analyzes resumes and matches candidates with suitable job opportunities. The results show improved accuracy, user engagement, and recommendation efficiency compared to traditional systems. Additionally, the feedback-based learning mechanism allows the system to continuously evolve and refine its recommendations over time. Although challenges such as data privacy and scalability need to be addressed, the overall system proves to be a scalable and intelligent solution. This project highlights the importance of automation and intelligent agents in bridging the gap between job seekers and employers, ultimately contributing to smarter and more efficient recruitment systems.

## REFERENCES

[1] D. Reilly, "Mobile Agents - Process Migration and Its Implications," 1998. [Online]. Available: http://www.davidreilly.com/topics/software_agents/mobile_agents/

[2] S. Franklin and A. Graesser, "Is it an Agent or Just a Program? A Taxonomy for Autonomous Agents," 1996. [Online]. Available: http://www.msci.memphis.edu/~franklin/AgentProg.html

[3] M. Wooldridge and N. Jennings, "Intelligent Agents: Theory and Practice," *Knowledge Engineering Review*, Oct. 1994, pp. 4–14.

[4] J. Kuppala, K. K. Srinivas, P. Anudeep, R. S. Kumar, and P. A. H. Vardhini, "Benefits of Artificial Intelligence in the Legal System and Law Enforcement," in *Proc. Int. Mobile and Embedded Technology Conf. (MECON)*, Noida, India, 2022, pp. 221–225, doi: 10.1109/MECON53876.2022.9752352.

[5] K. Meena et al., "A Novel Method for Prediction of Skin Disease through Supervised Classification Techniques," *Soft Computing*, vol. 26, pp. 10527–10533, 2022, doi: 10.1007/s00500-022-07435-8.

[6] P. A. Harsha Vardhini, S. S. Prasad, and S. N. Korra, "Medicine Allotment for COVID-19 Patients by Statistical Data Analysis," in *Proc. Int. Conf. Emerging Smart Computing and Informatics (ESCI)*, Pune, India, 2021, pp. 665–669, doi: 10.1109/ESCI50559.2021.9396830.

[7] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*, John Wiley & Sons, 2007, pp. 32–35, 52–65, 77–79.

[8] S. Shivaprasad and M. Sadanandam, "Speech Based Query Searching Technique and Its Application in Library Management System," *International Journal of Recent Technology and Engineering*, vol. 8, no. 3, Sep. 2019, doi: 10.35940/ijrte.C4779.098319.

[9] S. Suresh, "Studies in Agent-Based IP Traffic Congestion Management in DiffServ Networks," Ph.D. Thesis, University of South Australia, Adelaide, Australia, 2006.

[10] P. A. Harsha Vardhini et al., "Pioneering Minimalist Speech Analysis Through Optimized Spectral Features Machine Learning Models," in *Proc. ESCI*, Pune, India, 2024, pp. 1–6, doi: 10.1109/ESCI59607.2024.10497288.

[11] T. N. S. K. M. Kumar et al., "A Comparison Between Shortest Path Algorithms Using Runtime Analysis and Negative Edges in Computer Networks," in *Proc. MECON*, Noida, India, 2022, pp. 348–351, doi: 10.1109/MECON53876.2022.9752035.

[12] S. Shivaprasad and M. Sadanandam, "Dialect Recognition from Telugu Speech Utterances Using Spectral and Prosodic Features," *International Journal of Speech Technology*, 2021, doi: 10.1007/s10772-021-09854-8.

[13] V. R. Reddy et al., "Clustering Student Learners Based on Performance Using K-Means Algorithm," in *Proc. MECON*, Noida, India, 2022, pp. 302–306, doi: 10.1109/MECON53876.2022.9752165.

[14] S. Shivaprasad and M. Sadanandam, "Speech Based Query Searching Technique and Its Application in Library Management System," *IJRTE*, vol. 8, no. 3, Sep. 2019.

[15] M. L. Kumar et al., “Kernel Based FCM for Spinal Cord Segmentation on Computed Tomography Images,” in *Proc. ESCI*, Pune, India, 2024, pp. 1–6, doi: 10.1109/ESCI59607.2024.10497368.

[16] K. K. Srinivas et al., “Artificial Intelligence Techniques for Chatbot Applications,” in *Proc. MECON*, Noida, India, 2022, pp. 292–296, doi: 10.1109/MECON53876.2022.9751887.

[17] T. N. S. K. M. Kumar et al., “A Comparison Between Shortest Path Algorithms Using Runtime Analysis and Negative Edges in Computer Networks,” in *Proc. MECON*, Noida, India, 2022, pp. 348–351.

[18] P. A. Harsha Vardhini et al., “Facial Recognition Using OpenCV and Python on Raspberry Pi,” in *Proc. MECON*, Noida, India, 2022, pp. 480–485, doi: 10.1109/MECON53876.2022.9751867.

[19] S. Shivaprasad, A. Ramaswamy Reddy, and K. Dinesh, “Efficient Data Mining Model for Prediction of Chronic Kidney Disease Using Wrapper Methods,” *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 2, Aug. 2019, pp. 63–70, doi: 10.11591/ijict.v8i2.pp63-70.