



Research Article

Integrating cognitive intelligence in robotic process Automation: Placement bots in action

Neelam YADAV^{1,*}, Supriya P. PANDA²

¹Department of Computer Science and Engineering, School of Engineering and Technology, Manav Rachna International Institute of Research and Studies, Faridabad, 121004, India; Christ University, Karnataka, 560029, India

²Department of Computer Science and Engineering, School of Engineering and Technology, Manav Rachna International Institute of Research and Studies, Faridabad, 121004, India

ARTICLE INFO

Article history

Received: 10 April 2025

Revised: 01 July 2025

Accepted: 07 November 2025

Keywords:

Cognitive Intelligence;
Placement Bot; RPA-driven
Robotic Process Automation
(RPA); Selenium AI Web
Scraper; UiPath

ABSTRACT

With increasing administrative tasks in Higher Education Institutions(HIEs), automation provides an effective solution to improve efficiency. This research introduces an RPA Cognitive Placement Bot that automates resume screening and job matching using Machine Learning (ML) and Natural Language Processing(NLP). The placement bot automatically capture education, skills, and experience from resumes and assesses them with 159 job descriptions acquired from Indeed and LinkedIn. spaCy's Named Entity Recognition (NER), Selenium, Python-RPA, and Bidirectional Encoder Representations from Transformers (BERT) are employed to expedite the placement process by computing candidate scores through job-resume alignment and sending automated recommendation emails. The bot is trained and tested on 2,006 and 348 resumes respectively with 98.81% training accuracy with BERT, 97.91% with NER, and a net testing accuracy of 99.14%. The proposed virtual bot exhibits outstanding classification efficiency and enhanced performance for HEIs placement automation, when evaluated against current benchmark models —SVM-based (94%), LDA-based (82%), and Decision Tree-based (94%)) proposed by Jagwani et al., (2023) and Prabha et al., (2024).

Cite this article as: Yadav N, Panda SP. Integrating cognitive intelligence in robotic process Automation: Placement bots in action. Sigma J Eng Nat Sci 2026;44(4):2433–2446.

INTRODUCTION

Business organisations increasingly seek innovative solutions to reduce costs, boost efficiency, and reduce manual effort. Rather than digitising all operations, many focus on automating time-consuming tasks [1]. This problem is addressed by a Business Process Automation tool known as Robotic Process Automation (RPA). RPA automates

frequently repetitive tasks, so that the employees can focus on analytical work [2]. RPA bots mimic user interactions within software systems by performing various tasks such as form filling, email handling, web scraping, data processing, and browser automation [3,4].

RPA is a non-intrusive technique that operates on top of existing Information Systems(IS) and Enterprise

*Corresponding author.

*E-mail address: mail2neelam.sanskriti@gmail.com

This paper was recommended for publication in revised form by
Editor-in-Chief Ahmet Selim Dalkilic



Resource Planning (ERP) systems [5], and is extensively deployed through platforms such as Pega, Microsoft Power Automate, UiPath, and WorkFusion [6]. It empowers organisations to automate both back-office and customer-facing processes by facilitating benefits such as consistency, speed, regulatory compliance, and cost efficiency, often resulting in a high Return on Investment (ROI) within a year [3,7,8]. RPA excels in automating structured, rule-based tasks, however it cannot handle complex decision-making and therefore lacks cognitive intelligence. RPA enhances its cognitive capabilities when integrated with Artificial Intelligence (AI) technologies, such as Computer Vision, Deep Learning (DL), Machine Learning (ML), Natural Language Processing (NLP), generative AI and others [9-13]. This technological unification forms Intelligent Process Automation (IPA), enabling bots to make rule-based decisions, self-adjust, handle exception and improve over time [14]. Despite the use of IPA, not all processes are suitable for RPA, as poor selection can lead to bottlenecks [15]. Literature identifies several criteria for task suitability, focusing on processes that

are highly manual, repetitive, rule-based, high-volume, and frequently executed [16-19].

This paper is organised as follows: Section 2 outlines the problem and related work; Section 3 describes the methodology and technologies used; Section 4 presents results and analysis; and Section 5 concludes with findings and recommendations for future work.

Literature Review

Robotic process automation (RPA) is a rising technology for the digital transformations in Higher Education Institutions (HEIs). RPA supports faculty members by improving teaching efficiency through the automation of administrative tasks while also facilitating student development. While existing studies highlight RPA's benefits in improving operations and supporting digital change, its use in education remains underexplored. This research contributes to the field by developing and analysing a Cognitive RPA Placement Bot to automate the placement process in HEIs. Table 1 summarises prior research on RPA's applications and impact across administration, teaching, course delivery and Placements.

Table 1. Literature review on RPA implementations in HEIs

Source	Key Findings	Gaps in the Study
Khan et al.[20]	The study presented a robot, which is trained to test the delivery of a course. These bots can help a teacher improve their lecture delivery.	The study lacks empirical validation of RPA's effectiveness in improving teaching and learning outcomes. Moreover, it does not address the adaptability of RPA systems across different subjects, educational levels, and learning environments.
Turcu et al.[21]	An OCR-enhanced RPA solution was proposed in this work. The resume scanning process was automated by using UiPath for the recruitment process of Small and Medium Enterprises (SMEs).	The study limits the use of RPA technology by primarily depending on OCR. It does not incorporate AI or NLP-based cognitive solutions such as semantic matching, or intelligent resume parsing. Furthermore, it does not explore the scalability and applicability of this solution to other industries.
Somasundaram et al.[22]	An RPA-based report generation tool was proposed to generate advanced results from a programming competency test conducted for students. The tool assisted 540 students and 35 faculty mentors through its successful deployment in the Department of ECE at R.M.K. Engineering College.	Lack of quantitative assessment of the RPA tool's effectiveness in enhancing mentoring efficiency and student performance, as the current system primarily focuses on report generation without cognitive capabilities for in-depth performance analysis.
Vrushil Gajra et al.[23]	RPA's UiPath, MySQL Server, NodeJS server, and Restful API are used to create a ChatBot to automate the procedures of a student management system and integrate it with ERP system.	The study envisions integrating Artificial Intelligence and Machine Learning with RPA to handle unstructured data, predictive analytics, and intelligent decision-making.
D. Boban and J. Nakić [24]	The study presents a step-by-step RPA process design, including web navigation, data retrieval, report generation, and automated communication with users using UiPath. The study demonstrated significant time reduction ($\approx 77.5\%$).	This study automates the course search process on a single platform namely Udemy, and does not extend its functionality to other e-learning systems. It also does not provide an intelligent personalized recommendation system based on user behavior. Moreover, the proposed RPA solution is rule-based and does not integrate NLP or other AI-driven decision-making techniques for intelligent automation.
Nandwani et al.[25]	The study proposed an HTML (Hyper Text Markup Language), CSS (Cascading Style Sheet), Selenium and JavaScript based method for student outcome assessment process and saves 94.44% of the time.	This study solely relies on rule-based RPA automation using Selenium and Java, and lacks the integration of AI or NLP for intelligent decision-making and data interpretation.

Table 1. Literature review on RPA implementations in HEIs (continued)

Source	Key Findings	Gaps in the Study
Ding et al.[26]	The study employs a BiLSTM network and CNN to extract unstructured text features for Job-Resume matching tasks. This study proposes a novel Dual-Branch Mutual Learning (DBML) framework for job–resume matching in HR management.	This study lacks contextual understanding, as it uses fixed 100-dimensional skip-gram vectors. The semantic representations could be enhanced by incorporating newer contextualized embeddings such as BERT. Furthermore, the framework does not employ dynamic techniques such as BeautifulSoup or Selenium to scrape evolving job markets or candidate profiles; rather, it operates on static datasets.
Hawari1 et al. [27]	A job-recommendation system is proposed that uses content-based filtering mechanism to list the vacancies and to recommend Jobs.	The dataset is relatively small, and the integration of AI and NLP techniques could further enhance the mechanism.
Mydyti et al. [28]	A job internship matching strategy based on online scraping is proposed. This work uses an intelligent web scraping technique and Machine Learning in an Internship Management System (IMS).	This study provides only a theoretical approach and does not represent any practical evaluation. The proposed framework has not been validated in real-world settings. Furthermore, this study is based on Machine Learning and NLP techniques hence wouldn't be relevant for advanced technological models such as BERT. Therefore, it lacks contextual understanding and fails to give any feedback or recommendation for improvement for the skills that aren't inherited by it.
Suman et al. [29]	This study utilizes web scraping techniques to computerize tasks such as collection of job descriptions and resumes from LinkedIn. Fuzzy matching techniques were used for Job-Resume matching. A Decision Tree classifier was used to classify resumes.	The web scraping was only limited to LinkedIn and no other job portals were considered for scraping job descriptions and resumes. Furthermore, only the Decision Tree algorithm was used for classification and no other classifiers were considered to assess or improve accuracy. Moreover, other NLP models such as BERT were not considered, although they could enhance contextual and semantic understanding.
Rojas-Galeano et al. [30]	This work presents the research landscape analysis of job–résumé matching. This study identifies leading institutions, key trends and influential authors contributing to the field.	The study discusses literature-based finding without empirical validation of job–résumé matching models. It highlights the paradigm shift toward transformer-based architectures but does not discuss the advantages and disadvantages of each AI technique used in recruitment systems.
Lin et al. [31]	A Machine Learning based solution is proposed for the Job-Resume matching problem. Random Forest and CNN are used to improve 'Job-Resume matching' for online applicants. They help solve the problems of traditional rule-driven and keyword-based search engines.	The framework is developed using predefined datasets. It does not leverage real-time job market data. Deep learning and ensemble models are employed to model the training dataset. Contextual NLP approaches such as BERT or GPT-based embeddings are not considered. Additionally, the system doesn't offer personalization features, including adaptive recommendations and user preference learning.
Lunn et al. [32]	This study proposes an application to examine current job trends in the market for computer science(CS) students. A web scraping technique is used to collect job listings from the web with an accuracy of 98.5%. NLP techniques are used to extract information from textual data, while Python's BeautifulSoup library is used for scraping websites.	This study uses Python's BeautifulSoup, which cannot process dynamic content, thereby limiting data extraction. Selenium WebDriver should instead be used for web scraping. The scraping process was restricted to Computer Science–related data, which further limits its applicability to other domains. The primary focus is to identify job trends in the market; however, the work could be extended to job–resume matching and user feedback mechanisms for assessing students' current skill levels.
Jirjees et al.[33]	A BERT-based approach was used in the recruitment process to provide accurate and context-aware job–resume matching. This study highlights the superiority of BERT over other models in keyword matching. Several ML algorithm such as Support Vector Machine (SVM), Long Short-Term Memory (LSTM), and Multi-Layer Perceptron (MLP) models were used to test the dataset. The SVM-based model provide 94% accuracy, whereas the LSTM and MLP models were 92% and 90% accurate, respectively.	Sophisticated machine learning approaches such as clustering and hybrid sampling can reveal complex patterns. This study uses only a Kaggle dataset for the Computer Science field, which may not be fully representing other domains, and thus, results may vary. The model could be extended to other industries.

Table 1. Literature review on RPA implementations in HEIs (continued)

Source	Key Findings	Gaps in the Study
Jagwani et al.[34]	This study uses an Object Identification Technique to propose a Resume Evaluation Framework using spaCy and the Latent Dirichlet Allocation (LDA) model, achieving 82% accuracy.	This study was tested on a very limited dataset consisting of only 20 resumes, which is too limited to generalize the findings across all Higher Education Systems. Furthermore, the model does not provide recommendations or feedback regarding students' skills and competency gaps. In addition, this research has not been evaluated using BERT-based techniques, which could potentially improve the overall accuracy. The performance of the system could be further enhanced by incorporating ensemble learning techniques.
Prabha et al.[35]	The study was conducted to classify resumes for selecting or rejecting candidates. Various classification algorithms, such as Logistic Regression (LR), Random Forest (RF), and K-Nearest Neighbour (KNN) were tested for their performance in the study. Logistic Regression provide an accuracy of 95%, whereas Random Forest and KNN were 82% and 88% accurate respectively.	This model was tested on a very limited dataset consisting of only 10 resumes. Such a limited dataset does not provide a broader perspective on job-resume matching and restricts the generalizability of this study across all the HEIs.

As stated in the literature, traditional job-resume matching approaches primarily rely on exact keyword matching, old web-scraping techniques for acquiring resumes and job descriptions, and limited datasets. These approaches often lack contextual understanding and do not provide recommendations and feedback mechanisms for assessing candidates' skill sets despite the integration of advanced artificial intelligence and machine learning techniques. Furthermore, existing studies do not leverage Named Entity Recognition (NER) within BERT-based frameworks and often overlook critical components such as resume ranking and personalized feedback generation. This model proposes an intelligent placement bot that replicates all the tasks of a placement officer, ranging from resume acquisition to job-resume matching and sending skill recommendation emails to candidate. First, the automated system retrieves resumes through Gmail integration using OAuth 2.0 authentication. It then employs API-based web scraping techniques using Selenium to collect job descriptions from platforms such as LinkedIn and Indeed. After collecting job description and resumes, the placement bot automates various tasks, including resume parsing, candidate-job matching, ranking, and automated email communication. The proposed bot integrates machine learning techniques including Logistic Regression with TF-IDF features [36], fuzzy matching, spaCy-based Named Entity Recognition (NER) [37], BERT-based techniques and automation frameworks to deliver accurate, adaptive, and intelligent placement recommendations. Additionally, the system sends personalized selection and rejection emails based on candidate rankings. These recommendation emails help students understand their skill gaps relative to a specific job, replicating the role of human recruiters in an efficient manner.

This virtual bot is adaptive to the dynamically changing real-time job market and addresses the shortcomings of conventional job-resume systems. It provides personalized feedback by identifying skill gaps and rank resumes based on a weighted scoring framework that considers education, skills, and work experience. Overall, the proposed software solution provides a more flexible and responsive approach to align student competencies with internship opportunities while enhancing job-resume matching accuracy and optimizing the recruitment process. The bot significantly outperforms traditional methods in terms of accuracy, scalability, and automation.

MATERIALS AND METHODS

The approach begins with the Manual Process of Job-Resume Matching as illustrated in Figure 1, which serves as a baseline for comparison.

Manual Tasks of Job-Resume Matching Process

The routine manual tasks performed by a placement officer in an HEI during the placement process are typically organized step by step to guarantee students have the best job opportunities. This involves a series of repeated administrative tasks that must be completed each time a new Job Description (JD) is issued by the company as depicted in Figure 1. RPA is best suited to automate such repetitive, time-consuming processes making it ideal for RPA automation. The selection of this process aligns with the suitability of the criteria discussed in the literature. For the purpose of comprehension and visualisation, Figure 2 lists every step involved in automating the Placement process.



Figure 1. Manual Job-Resume Matching Process Performed by the Placement Officer.

Automation steps of “RPA Cognitive Placement Bot”

The automated RPA Cognitive Placement Bot as depicted in Figure 2 comprises five steps:

Step 1: Resume and job description extraction

Resume extraction

Resumes are extracted from the placement officer’s Gmail Id. The resumes and job descriptions used in this study were provided by the placement cell of a higher education institution (ABES Engineering College, Ghaziabad, India) and represent real-world data from final-year students currently undergoing the placement process. All data was used with the institution’s approval, and strict measures were taken to maintain candidate privacy. The data was used exclusively for academic intentions and according to the institutional research ethics. The bot uses OAuth 2.0 to authenticate and connects to Gmail API. OAuth 2.0 is an authorization system that facilitates third-party apps to utilize a user’s data without revealing their credentials (such as

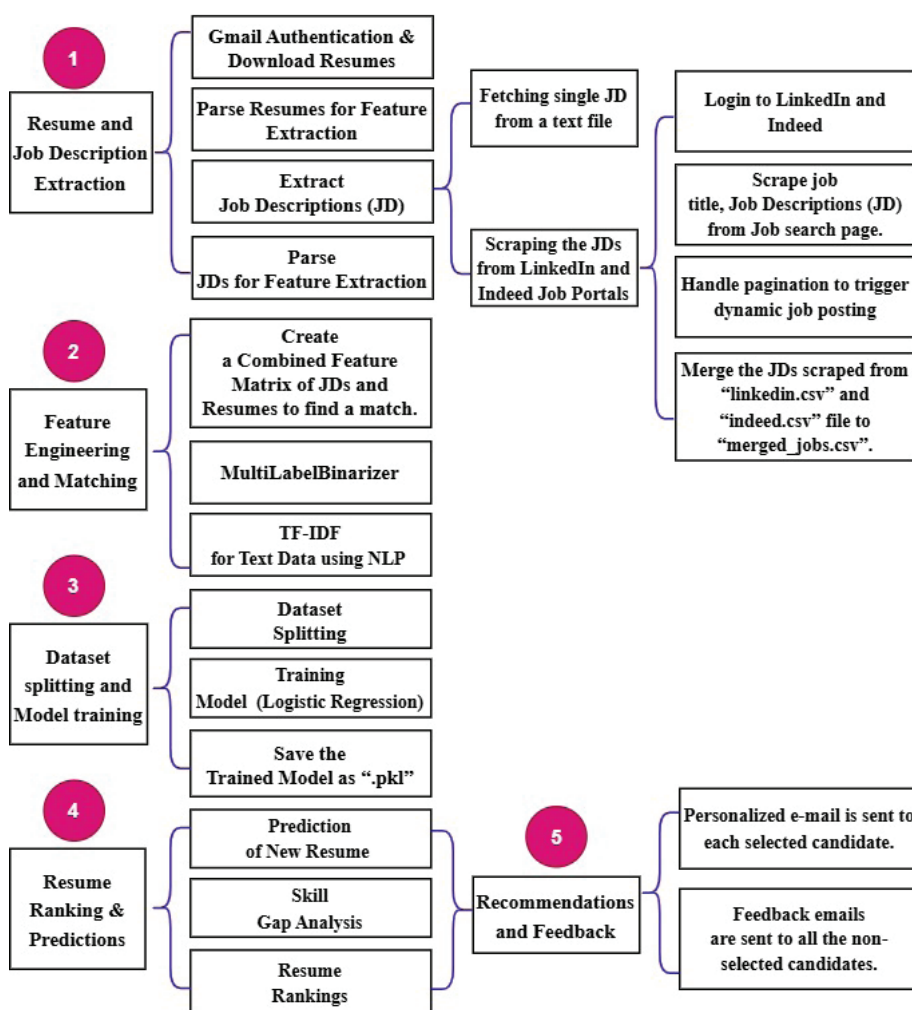


Figure 2. Automation steps of “RPA cognitive placement bot”.

resume_name	skills	experience	education	email
Sujeet Kumar.pdf	c, java, python, react, sql	e, n, t, r, y, -, l, e, v, e, l	Not specified	ksujeet0707@gmail.com
Gaurang Gaur.pdf	c, java, javascript, node.js, python, react, sql	e, n, t, r, y, -, l, e, v, e, l	bachelor	pegauranggaurl0@gmail.com
aaayush.pdf	c, django, leadership, node.js, node.js, r, react, sql	e, n, t, r, y, -, l, e, v, e, l	bachelor	yush91620@gmail.com
Aadesh Kumar.pdf	java, leadership	e, n, t, r, y, -, l, e, v, e, l	mca	aadeshrajput250609@gmail.com
Aaditya Bhardwaj.pdf	c, django, java, javascript, python	e, n, t, r, y, -, l, e, v, e, l	bachelor	aadibhardwajdec98@gmail.com
Aditya Kumar.pdf	aws, c, java, machine learning, python, sql	e, n, t, r, y, -, l, e, v, e, l	Not specified	0906@gmail.com
aakash jayesh.pdf	not specified	e, n, t, r, y, -, l, e, v, e, l	Not specified	aakash1234shah@gmail.com
Aakash Kumar.pdf	c, java, javascript, python, sql	e, n, t, r, y, -, l, e, v, e, l	master	aakimrt619@gmail.com
Aakash Kumar.pdf	not specified	m, i, d, -, l, e, v, e, l	bachelor	aakashkumar171555@gmail.com
Aakash Sharma.pdf	c, java, javascript, sql	e, n, t, r, y, -, l, e, v, e, l	b.tech	aakashg837799@gmail.com
Aalok mishra.pdf	c	e, n, t, r, y, -, l, e, v, e, l	Not specified	alok.mishra5192@gmail.com

Figure 3. Extracted Resume Features stored in ‘resume_extracted.csv’ file.

passwords). The resumes are filtered according to specific subject lines and contents of an email.

Parsing resumes for feature extraction

A CSV file of the name- “resumes_extracted.csv” is created, as depicted in Figure 3, to store the extracted information from the collected resumes such as candidate names, skills, educational qualifications, certifications, contact information, and work experience.

Experience extraction

Experience related information such as ‘5 years’ or ‘since 2020’, are extracted from resumes using ‘Named Entity Recognition (NER) and spaCy’s ‘en_core_web_sm’ model. The bot identifies the experience levels by identifying different phrasing such as ‘5+ years’ or ‘over 5 years’ and classifies candidates into three groups: Entry-Level (less than two years of experience), Mid-Level (2-5 years of experience), and Senior-Level (more than 5 years of experience).

Education extraction

A hybrid approach (regex+NLP) is used for identifying degree names (e.g., “B.Tech”, “Master’s”). It uses Sentence-Bidirectional Encoder Representations from Transformers (SBERT) “all-MiniLM-L6-v2” model and spaCy’s NER for semantic similarity matching and to fetch full educational phrases (e.g., “Master’s in Data Science”).

Skills and email extraction

The bot scans resumes for predefined skills such as ‘Python’, ‘ML’, and ‘SQL’, and uses regex to extract email addresses. It also stores unique skills for efficient processing.

Scrape job descriptions

From LinkedIn web portal

Python’s Selenium Web Driver is used to dynamically log in, identify input fields, and perform the authentication. It queries LinkedIn for specific job titles, locations, and experience levels. It then fetches job titles, company names, and descriptions using CSS selectors, storing the data in

JavaScript object Notation (JSON) and Comma Separated Values (CSV) formats. Job Title and Job Description are scraped from LinkedIn and stored in the ‘job_data.csv’ and ‘job_data.json’ files.

From Indeed web portal

Selenium-based web browser automation is used to dynamically interact and scrape Indeed’s job listings. The web driver waits for up to 40 seconds for the job listings to appear to avoid scraping an empty page.

From text file

The Job Descriptions (JDs) provided by the companies are simple text files containing sections like job title, responsibilities, required skills, qualifications, and experience. In this study a “job_description.txt” file as shown in Figure 4 is used to match student’s resumes with the company’s JD.

Parsing job descriptions for feature extraction

To generate a feature matrix, Job Descriptions (JDs) taken from text files, Indeed, and LinkedIn need to be refined. The data sourced from LinkedIn and Indeed presents several challenges, including irregular newline formatting, unclear experience format, and unclear skills in the free text. Two different CSV’s are generated as linkedin_jobs.csv and indeed_jobs.csv. Both CSV are merged into one csv file “merged_jobs.csv” as depicted in Figure 5.

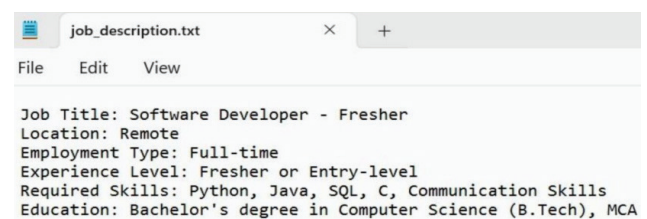


Figure 4. Snapshot of a “job_description.txt” file used for job–resume matching.

job_title	skills	experience	education
Jr Software Engineer	['agile', 'angular', 'aws', 'azure', 'communication skills', 'java', 'javascript', 'leadership', 'react', 'sql', 'teamwork']	[]	['bachelor's'], ['bachelor']
Engineer, II Software	['agile', 'c', 'java', 'javascript', 'python']	['3']	['bachelor's']
Software Engineer - Java	['aws', 'azure', 'java', 'python', 'teamwork']	['3']	['master's', 'mca'], ['m.tech']
Software Engineer	['communication skills', 'java', 'sql']	['5', '10', '7', '2', '15']	['bachelor's']
Software Engineer II	['agile', 'angular', 'aws', 'azure', 'c', 'gcp', 'sql']	['2']	['bachelor's'], ['bachelor']
Software Development Engineer II	['communication skills', 'java', 'python', 'sql', 'teamwork']	['2']	['not specified']
Software Engineer I	['agile', 'aws', 'java', 'react', 'sql']	['2', '1']	['not specified']
Software Engineer	['azure', 'c', 'leadership', 'project management', 'sql']	[]	['bachelor's'], ['bachelor']
Software Engineer	['agile', 'angular', 'aws', 'java', 'javascript', 'node.js', 'python', 'react', 'restful apis', 'scrum']	['5', '2']	['bachelor's'], ['bachelor']
Senior Software Development Engineer I	['aws', 'azure', 'gcp', 'java', 'python']	['5', '2']	['phd'], ['bachelor's'], ['bachelor', 'master']
Software Developer	['java', 'python', 'sql']	['4']	['bachelor's']
Software Engineer	['angular', 'java', 'leadership', 'python', 'sql']	[]	['master's'], ['bachelor', 'master']

Figure 5. Merged job dataset (“merged_jobs.csv”) compiled from LinkedIn and Indeed.

Step 2: Feature engineering and matching

Flow of combining feature matrix

The bot uses two modules, `exact_match()` for direct matches and `fuzzy_match()` techniques to match resumes with job descriptions, scraped from LinkedIn and Indeed. `fuzzy_match()` employs an AI-based Levenshtein distance to suggest approximate matches (e.g., “ML” vs. “Machine Learning”). It evaluates and differentiates between candidate’s education, skills and experience with job requirements and returns true if a match is found.

MultiLabelBinarizer

MultiLabelBinarizer (MLB) is employed to transform skills, education, and experience levels by converting multiple fields into binary matrices. MultiLabelBinarizer is used when a single column contains multiple values (e.g., a candidate having multiple skills). Each unique skills (we are accessing the resumes for 33 skills) becomes a separate column with 1 or 0 (1 = candidate has the skill, 0 = candidate doesn’t have the skill). It converts experience levels

(entry-level, mid-level, senior-level) into a binary matrix (0, 1). The same process applies to encode education as depicted in Figure 6.

Term frequency (TF) - Inverse document frequency (TF-IDF)

TF measures how frequently a term appears in a document. For ex: If “python” appears 5 times in a document containing 100 words. Then the TF will be $TF = 5/100 = 0.5$. IDF measures how unique a term is across all documents. For ex: If “python” appears in 10 out of 1000 times in a document containing 100 words. Then the $IDF = \log(1000 / 10) = \log(100) \approx 2$.

Step 3: Dataset splitting and model training

Dataset splitting

To split the dataset into training and testing sets, the bot used Stratified K-Fold cross-validation. It ensures that each fold has the same proportion of class labels as the original dataset [39]. Stratified K-Fold cross validation splits the

filename	skills	experience_level	education	match	agile	angular	aws	azure	c	c#
aaayush.pdf	['c', 'django', 'leadership', 'node.js', 'node.js', 'r', 'react', 'sql']	['entry-level']	['bachelor']	0	0	0	0	0	1	0
Aadesh Kumar.pdf	['java', 'leadership']	['entry-level']	['mca']	1	0	0	0	0	0	0
Aaditya Bhardwaj.pdf	['c', 'django', 'java', 'javascript', 'python']	['entry-level']	['bachelor']	0	0	0	0	0	1	0
Aditya Kumar.pdf	['aws', 'c', 'java', 'machine learning', 'python', 'sql']	['entry-level']	['Not specified']	0	0	0	1	0	1	0
Aakanksha.pdf	['communication skills']	['entry-level']	['Not specified']	0	0	0	0	0	0	0
aakash jayesh.pdf	['not specified']	['entry-level']	['Not specified']	0	0	0	0	0	0	0
Aakash Kumar bsc.pdf	['c', 'java', 'javascript', 'python', 'sql']	['entry-level']	['master']	0	0	0	0	0	1	0
Aakash Kumar.pdf	['not specified']	['mid-level']	['bachelor']	0	0	0	0	0	0	0
Aakash Sharma.pdf	['c', 'java', 'javascript', 'sql']	['entry-level']	['b.tech']	1	0	0	0	0	1	0

Figure 6. Encoding of experience, skills, and education attributes using MultiLabelBinarizer.

dataset into K equal folds, but preserves the original distribution of classes in each fold. The 2,006 resumes used in this study are split into 5 equal folds and each fold has the same class distribution. Logistic Regression (LR) is a supervised machine learning algorithm is perfect for classes with binary (0 or 1) values. The output of the trained LR classifier's and TF-IDF vectorizer's are saved in files named "logistic_regression.pkl" and "vectorizer.pkl" respectively for future predictions.

Step 4: Resume ranking & predictions

Prediction on new resumes

The trained model is used for prediction on new resumes. The pickle files "logistic_regression.pkl" and "vectorizer.pkl" are loaded. The CSV files containing parsed skills, experience level, education, and match labels (true labels) of 348 students is also loaded by the bot. After this, the pre-trained LR model predicts each resume for a match (1) or no match (0).

Skill gap analysis

This module determines missing and matching skills by assessing an applicant's abilities with a job description. The candidate's skills and the required job skills are represented by sets. Missing skills are calculated using the set difference (-). The set intersection (&) is used to find the common skills between the candidate and the job.

Ranking resumes

The calculate_scores() module evaluates each resume by computing a composite matching score derived from three primary metrics, skills, experience, and education. For skills, a 60% similarity threshold is applied, and only those matching or exceeding this threshold contribute positively to the score. The candidate's experience score is determined by calculating the ratio between the required job experience and candidate's experience, which is subsequently normalized to 1. A value of 1 is assigned to the education score, if the candidate's qualification exactly matches the job requirement; otherwise a value of 0 is assigned. A weighted scoring mechanism is used to rank resumes by assigning weights of

30% to experience, 50% to skills, and 20% to education. The resulting sum produces a ranking score, where 1 indicates a perfect match across all criteria. Finally, the scored results, along with missing and matched skills, are stored in the file "ranked_resumes.csv," as shown in Figure 7a.

Step 5: Selection, recommendations and feedbacks

The bot sends email notifications to candidates with a score of 1. The authenticate_gmail() module is used to obtain credentials for accessing the Gmail service. Further the bot fetches email Ids of candidates from "ranked_resumes.csv" and sends personalized selection emails via send_emails_to_selected_resumes() module. Filtered candidates with scores <1 in the "ranked_resumes.csv" file are fetched to give feedback on missing skills. The send_email_gmail() authenticates using OAuth2.0 using authenticate_gmail() .

RESULTS AND DISCUSSION

This "RPA Cognitive Placement Bot" automates traditional placement processes, making them faster, more objective, and more efficient. This Placement Bot completely automates the whole process of obtaining emails, processing resumes and job descriptions, screening each resume to find a selected candidate, and then sending emails to selected and rejected prospects, without human intervention. This AI-driven approach ensures that placement processes remain up-to-date, efficient, and equitable; promising a new alternative to traditional job placement strategies in higher education institutions.

Placement Bot's Training Model Evaluation

This Placement Bot is trained on 2006 resumes with 159 Job Descriptions. The LR classifier achieved 97.91% accuracy and the average precision, Recal and F1-Score is 0.98. The confusion matrix depicted in Figure 8 shows that True Positive (TP) =471, which means that 471 resumes were correctly classified as selected. True Negative (TN) =1493, means that 1493 resumes are correctly classified as rejected. False Positive (FP) =26 means 26 resumes were mistakenly classified as selected when they should have been rejected.

candidate_id	missing_skills	matching_skills	experience_match	education_match	email	score
ANIMESH KUMAR DUBEY.pdf	{'sql', 'communication skills'}	{'java', 'c', 'python'}	TRUE	TRUE	animeshkdji@gmail.com	1
Akeel Ahmad.pdf	{'communication skills', 'python'}	{'sql', 'java', 'c'}	TRUE	TRUE	akeel.21m0141196@abes.ac.in	1
Ashruti Srivastava.pdf	{'communication skills'}	{'sql', 'java', 'c', 'python'}	TRUE	TRUE	ashrutisrivastava20@gmail.com	1
Divyanshu Varshney.pdf	{'communication skills'}	{'sql', 'java', 'c', 'python'}	TRUE	TRUE	dbyan8859@gmail.com	1
Vikash Singh.pdf	{'communication skills'}	{'sql', 'java', 'c', 'python'}	TRUE	TRUE	sahilsingh11012002@gmail.com	1
Harshit kumar.pdf	{'communication skills'}	{'sql', 'java', 'c', 'python'}	TRUE	TRUE	harshitlucky11@gmail.com	1
Saurabh Saroha.pdf	{'java', 'communication skills'}	{'sql', 'c', 'python'}	TRUE	TRUE	saurabh.22m0141056@abes.ac.in	1
Akshay Kumar Resume.pdf	{'communication skills'}	{'sql', 'java', 'c', 'python'}	TRUE	TRUE	akshaykumar.uk18@gmail.com	1

Figure 7. Output of the resume ranking module stored as "ranked_resumes.csv" file.

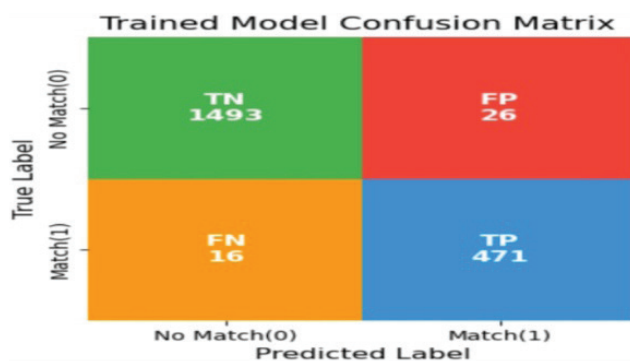


Figure 8. Confusion matrix showing the performance of the trained logistic regression classifier.

False Negative (FN) = 16 means 16 resumes were mistakenly classified as rejected when they should have been selected.

A comprehensive comparison is conducted among ML classifiers using Bidirectional Encoder Representations from Transformers (BERT) embeddings to identify the most effective model for Job-Resume matching. BERT substantially extends matching accuracy by acquiring contextual semantics, understanding how education, experience, and skills fit together within context. Specifically, the semantic embeddings for resume and job description texts are extracted using the 'bert-base-nli-mean-tokens' model from the SBERT family. Each resume and job description is represented through a structured integration of skills,

experience, and education. The preprocessed text is subsequently passed through the SBERT model, which generates a 768-dimensional dense vector representations, denoted by 'x_bert', for each resume.

The variable x_bert is used as the input feature set for Machine Learning Models such as Logistic Regression and Support Vector Machine (SVM). In the resume classification process, match = 1 indicates that the resume is predicted as selected, whereas a match = 0 indicates that the resumes is predicted to be rejected. When a new resume is provided, the model predicts candidate selection or rejection based on its representation in the BERT embedding space.

The SBERT-derived x_bert embeddings are assessed using various Machine Learning models, including Logistic Regression (LR), Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Gradient Boosting (XGBoost), and Naïve Bayes (NB). Among these models, LR achieved the highest F1 score of 0.9881, as shown in Table 2 and Figure 9. The tree-based Random Forest, distance-based KNN and NB models achieved comparatively lower F1-Scores. Whereas SVM and XGBoost achieved F1 scores above 0.95. These results indicate that **linear classifiers** achieve higher accuracy, when integrated with BERT-based semantic features for job-resume matching systems.

Figure 9 presents a comparison of evaluation metrics such as Accuracy, Precision, Recall, and F1-Score across multiple ML classifiers applied to the job resume matching problem using SBERT embeddings.

Table 2. Performance comparison of ML Models using SBERT

Model Configuration	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.9880	0.9884	0.9880	0.9881
Support Vector Machine (SVM)	0.9566	0.9624	0.9566	0.9577
XGBoost	0.9551	0.9555	0.9551	0.9552
Random Forest	0.9382	0.9388	0.9382	0.9363
KNN	0.8579	0.8551	0.8579	0.8556
Naïve Bayes	0.7333	0.8499	0.7333	0.7521

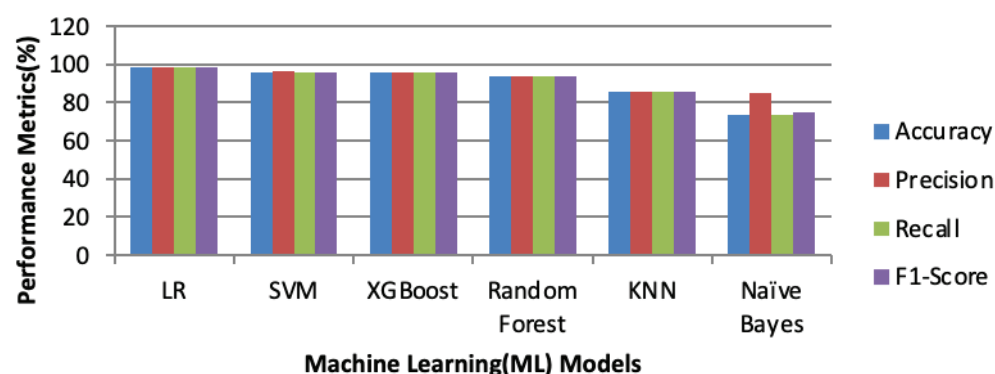


Figure 9. Comparison of evaluation metrics for ML models using SBERT.

Placement Bot's Testing Model Evaluation

The Placement Bot was tested on 348 resumes using a Logistic Regression classifier, achieving high classification performance. Table 3 summarizes the values of all evaluation metrics.

The confusion matrix in Figure 10 (a) illustrates the classification outcomes in detail. The precision-recall trade-off shown in Figure 6(b) highlights the balance between false positives (FP) and false negatives (FN), with an accuracy of 0.99, recall of 0.97, and F1-score of 0.98. Figure 10 (b) presents the confusion matrix results, where True Positive (TP) = 66, indicate that 66 resumes were correctly classified as selected, while True Negative (TN) = 279 shows that 279 resumes were properly predicted as rejected. False Positive (FP) = 1 represents one resume incorrectly predicted as selected instead of rejected, whereas False Negative (FN) = 2 indicate two resumes that were incorrectly rejected despite being relevant.

Table 3. Evaluation metrics

Metrics	Values
Accuracy	99.14%
Recall	0.97
Precision	0.99
F1-Score	0.98

Recommendations and Feedback

The Placement Bot not only automates the resume-matching process, but also sends personalised feedback letters to rejected candidates that highlight their skill gaps. Selected candidates will be sent automated congratulatory emails. Automated emails (rather than handwritten comments) ensure that all applicants receive a timely response and thereby reducing ambiguity and enhancing the overall candidate experience. The bot helps candidates to identify missing skills and competencies so that they can improve their resume profiles and be better prepared for

future resume submissions and job applications. Figure 11 depicts the selection emails sent to the candidates, who achieved a matching score of 1. Figure 12 shows the rejection and recommendation emails sent to the candidates who were not selected due to obtaining a matching score of below 1.

The automated Placement Bot changes the way manual processes performed by placement officers in Higher Education Institutions (HEIs). The proposed Placement Bot mimics human-like intelligence and exhibits cognitive capabilities such as strategic thinking, pattern recognition, learning, evaluation, and communication, thereby enhancing the way conventional automation tools work. The current study offers precision and objectivity by bridging the gap between automated and human-like placement processes. To understand the practical relevance of the automated placement bot, a dataset comprising final-year students of ABES Engineering College, Ghaziabad, India was used. Although the dataset did not include resumes of all the final year students, currently seeking placement assistance, the preliminary study verified the reliable performance of the bot under current HEI conditions and extends its viability in automated placement assistance. Consent to use their personal information such as name, email, experience and skill set was obtained from all participating students during the study. The prototype system retrieved such personal information solely for illustration and testing purposes. However, if the bot is deployed in a real-world environment, the system must acquire proper explicit user consent for resume processing and communication activities. Throughout the study, personal information was handled ethically and strict measures were implemented to ensure data safeguarding.

Furthermore, no information apart from students' personal details, such as phone numbers, email, skills, experience, education and gender was used. Resumes in non-conventional formats, such as screenshots, scanned PDFs, and PDFs converted to Word documents, introduce inconsistencies during NLP processing and are comparatively more challenging for the automated processing.



Figure 10 (a). Confusion matrix of the prediction model.

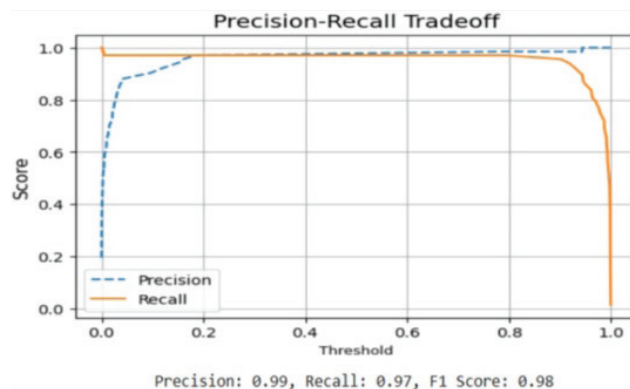


Figure 10 (b). Precision-recall trade-off.

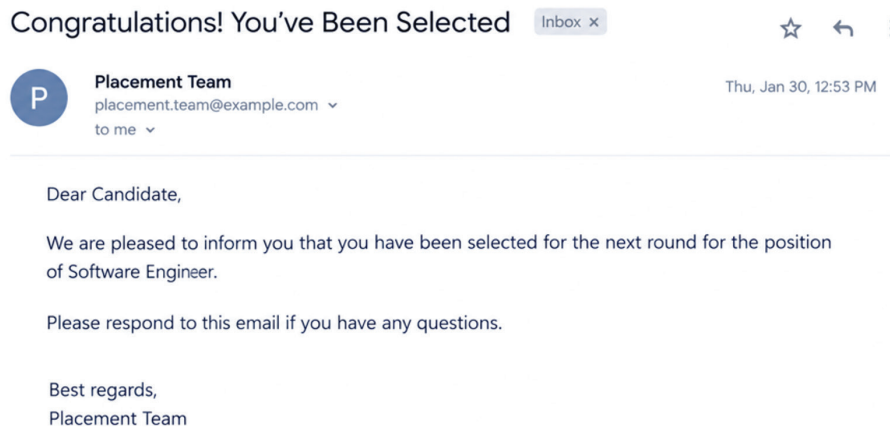


Figure 11. Selection email sent to Khadija Razvi.

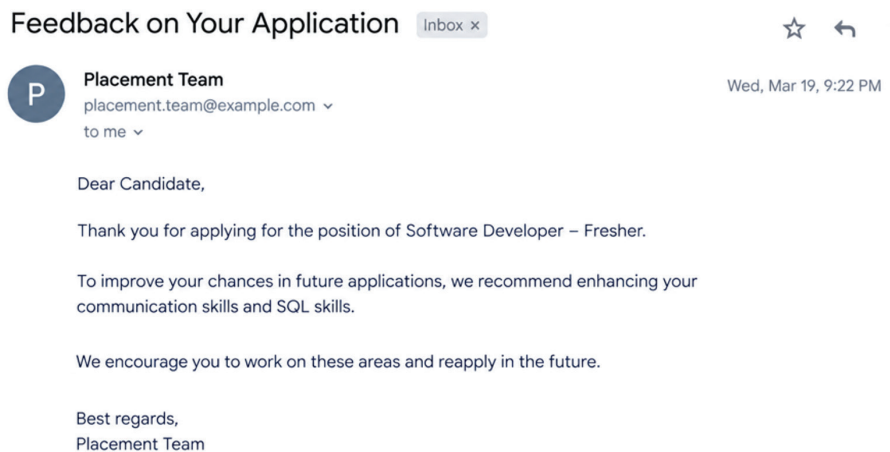


Figure 12. Feedback email sent to Sahil Kapil with missing skills.

Text-based resumes allow smooth processing, whereas resumes containing graphical elements such as icons, images, signature, tables often hinder NLP parsing, potentially leading to inaccurate information extraction. In such cases, there is a chance of inaccurate extraction of skills, education, or experience, which further affects ML model training and testing. Additionally, the use of Optical Character Recognition (OCR) for unstructured resumes (resumes with screenshots and scanned PDFs or graphical elements) can further reduce parsing accuracy, as it introduces extra noise into the extracted text. These limitations can cause the rejection of candidates who do not follow the standard resume format. Apart from the test dataset consisting of only 348 resumes, the performance metrics obtained from the ML models, particularly the BERT-based models are encouraging. However, the limited sample size may not represent real-world resume patterns. Therefore, further validations of the trained model using larger and more heterogeneous dataset is required to ensure the

generalizability of the proposed system. On average, the bot requires approximately 0.5–0.7 seconds to for end-to-end automation, whereas BERT-based models require approximately 1.2–1.5 seconds per resume.

Cognitive aspect of RPA placement bot

The Placement Bot uses intelligent RPA and emulates human-like intelligence in classification, pattern recognition, strategic thinking and decision-making. These features provide the bot with an edge over other conventional manual approaches. This intelligent bot enables the system to learn, reason, adapt, and provide meaningful insights to candidates and recruiters.

- a. The bot mimics the end-to-end placement activities typically performed by human recruiters, including assessing resumes and providing job opportunities and sending career-related recommendations. It employs a weighted scoring mechanism that allocates 50% weight to skills, 30% to experience, and 20% to education. Based on these

scores the bot assesses resumes, makes decisions and rank candidates according to their weighted scores.

- b. The bot sends recommendation emails based on candidate rankings, explaining why a particular candidate has been selected or rejected. Based on the candidates' scores, it sends personalized selection or rejection emails. It mimics a human recruiter by providing feedback in a logical and efficient manner. Moreover it informs candidates about their skill gaps, which can support their future career development and enhance their employability.
- c. The bot acts dynamically by retrieving current jobs opportunities through the continuous updating of job descriptions and resume datasets. It also supports recruiters by utilizing information from real-time job market platforms such as LinkedIn and Indeed, to provide more accurate and relevant recommendations.

CONCLUSION

This study proposes an RPA-based intelligent Placement Bot designed to automate placement-related activities traditionally carried out by placement officers in Higher Education Institutions (HEIs). The system integrates technologies such as Selenium-based AI web scraping, spaCy's Named Entity Recognition (NER), Python RPA frameworks, and machine learning models to construct an automated placement pipeline. The dataset comprises 159 job descriptions scraped from LinkedIn and Indeed, with 2,006 resumes used for training and 348 samples reserved for testing. The developed bot attained an overall accuracy of 99.14%, with precision and recall values of 0.99 and 0.97, respectively. Its performance outperformed the previous reported approaches based on LDA (82%), SVM (94%) and decision tree (94%). The results indicate that the proposed system has the potential to standardize the placement process and improve the accuracy of job-resume matching in Higher Education Institutions (HEIs).

The proposed RPA-based virtual bot has some limitations that may affect its overall performance and applicability. Variations in resume structure, formatting and quality affect the accuracy of information extraction. Infographic style resumes and image-based PDF documents are particularly challenging to conventional text-based extraction approaches. In addition, resumes that are sent as scanned images or screenshots have to be processed through Optical Character Recognition (OCR) that can cause recognition errors because of low image resolution, inconsistent font styles and complex document layouts. The dependencies on OCR can also lead to inconsistencies in parsing, particularly with non-standard fonts or degraded input quality. The current study is evaluated on a dataset of 2,006 resumes with 348 test samples. However, large scale deployment in Higher Education Institution (HEI) environments would require further considerations such as infrastructure optimization, distributed processing architectures and

batch-parallel processing strategies to ensure efficiency, scalability, and robustness.

Ethical considerations were strictly followed throughout the study. Explicit consent was obtained from all participating students before collecting their resumes for this institutional pilot study. No personal information was shared or disseminated across any platforms, ensuring privacy and confidentiality of all participants. In addition, the system fetches job descriptions from third-party platforms such as LinkedIn and Indeed. Consequently, the bot's workflow may be disrupted by changes in the user interfaces, APIs or underlying website structures of these platforms. Such dynamic changes require continuous maintenance and regular software updates to ensure system reliability. Although the proposed placement bot does not directly employ an Explainable Artificial Intelligence (XAI) framework, its architecture includes inherently interpretable components such as skill-experience-education-based scoring and rule-based feature extraction mechanisms. Future work may involve the integration of XAI techniques such as SHAP and LIME to enhance interpretability, transparency, explainability.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE

Artificial intelligence was not used in the preparation of the article.

REFERENCES

- [1] Andrade D. Challenges of automated software testing with robotic process automation RPA - a comparative analysis of UiPath and Automation Anywhere. *Int J Intell Comput Res* 2020;11:1066–1072. [CrossRef]

- [2] Syed R, Suriadi S, Adams M, Bandara W, Leemans SJJ, Hofstede AHM, et al. Robotic process automation: contemporary themes and challenges. *Comput Ind* 2020;115:103162. [\[CrossRef\]](#)
- [3] Siderska J, Aini SNBM, Kedziora D. Complementing robotic process automation with generative artificial intelligence (ChatGPT), case of Robocorp. In: Arai K, editor. *Advances in Information and Communication*. Cham: Springer Nature Switzerland; 2024. p. 37–53. [\[CrossRef\]](#)
- [4] Khan S, Tailor RK, Uygun H, Gujrati R. Application of robotic process automation (RPA) for supply chain management, smart transportation and logistics. *IJHS*. 2022. p. 11051–11063. [\[CrossRef\]](#)
- [5] van der Aalst WMP, Bichler M, Heinzl A. Robotic process automation. *Bus Inf Syst Eng* 2018;60:269–272. [\[CrossRef\]](#)
- [6] Khan S. Comparative analysis of RPA tools-Uipath, Automation Anywhere and BluePrism. *Int J Comput Sci Mobile Appl* 2020;8:1–6. [\[CrossRef\]](#)
- [7] Patrício L, et al. Literature review of decision models for the sustainable implementation of robotic process automation. *Procedia Comput Sci* 2023;219:870–878. [\[CrossRef\]](#)
- [8] Desai D, Jain A, Naik D, Panchal N, Sawant D. Invoice processing using RPA & AI. *SSRN J Preprint*. 2021. Doi: 10.2139/ssrn.3852575. [\[CrossRef\]](#)
- [9] Exploring heart diseases survival patterns: a multifaceted approach using survival analysis techniques on time to event data. *Sigma J Eng Nat Sci* 2025;43:1307–1320. [\[CrossRef\]](#)
- [10] Nagrale P. Indoor-outdoor scene recognition: a multi-feature framework using CNN for complex environment. *Sigma J Eng Nat Sci* 2025;43:1355–1365. [\[CrossRef\]](#)
- [11] Otundo Richard M. Robotic process automation (RPA) and AI: an empirical analysis. *SSRN Electron J* 2024. Preprint. doi: 10.2139/ssrn.4908263. [\[CrossRef\]](#)
- [12] Chakraborti T, et al. From robotic process automation to intelligent process automation: emerging trends. In: Asatiani A, García JM, Helander N, Jiménez-Ramírez A, Koschmider A, Mendling J, Meroni G, Reijers HA, editors. *Business Process Management: Blockchain and Robotic Process Automation Forum*. Cham: Springer International Publishing; 2020. p. 215–228.
- [13] Dolui C, Roy D. Advancement of bridge health monitoring using magnetostrictive sensor with machine learning techniques. *Nondestruct Test Eval* 2025;40:4219–4240. [\[CrossRef\]](#)
- [14] Kholiya PS, Kapoor A, Rana M, Bhushan M. Intelligent process automation: the future of digital transformation. In: 2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART); 2021 Dec; Moradabad, India. *IEEE*; 2021. p. 185–190. [\[CrossRef\]](#)
- [15] Jalonon H. Assessing robotic process automation potential (master's thesis). Tampere: Tampere University of Technology; 2017.
- [16] Jeeva Padmini KV, Perera GIUS, Bandara HMND, Silva RKO. A decision support tool to select candidate business processes in robotic process automation (RPA): an empirical study. In: Smys S, Balas VE, Kamel KA, Lafata P, editors. *Inventive Computation and Information Technologies*. Singapore: Springer Nature Singapore; 2021. p. 567–582. [\[CrossRef\]](#)
- [17] Santos F, Pereira R, Vasconcelos JB. Toward robotic process automation implementation: an end-to-end perspective. *Business Process Manag J* 2019;26:405–420. [\[CrossRef\]](#)
- [18] Wanner J, Hofmann A, Fischer M, Imgrund F, Janiesch C, Geyer-Klingenberg J. Process selection in RPA projects - towards a quantifiable method of decision making. In: 40th International Conference on Information Systems (ICIS); 2019 Nov; Munich, Germany.
- [19] Agaton B, Swedberg G. Evaluating and developing methods to assess business process suitability for robotic process automation - a design research approach [master's thesis]. Gothenburg: Chalmers University of Technology, Department of Computer Science and Engineering; 2019.
- [20] Khan S, Tailor RK, Pareek R, Gujrati R, Uygun H. Application of robotic process automation in education sector. *J Inf Optim Sci* 2022;43:1815–1834. [\[CrossRef\]](#)
- [21] Turcu CE, Turcu CO. Digital transformation of human resource processes in small and medium sized enterprises using robotic process automation. *Int J Adv Comput Sci Appl* 2021;12:10. [\[CrossRef\]](#)
- [22] Somasundaram M, Sumitra V, Vardan RSPV, Sakthipriya B, Pavithra K, Nivedha MK. Monitoring and facilitating students programming skill development using robotic process automation (RPA) and artificial intelligence (AI)-a case study. In: 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT); 2022. p. 1–6. [\[CrossRef\]](#)
- [23] Gajra V, Lakdawala K, Bhanushali R, Patil S. Automating student management system using ChatBot and RPA technology. *SSRN Electron J* 2020. [\[CrossRef\]](#)
- [24] Boban D, Nakić J. Enhancing informal education with robotic process automation: a method for automating course search on e-learning platforms. *J Commun Softw Syst* 2024;20:215–225. [\[CrossRef\]](#)
- [25] Nandwani T, Sharma M, Verma T. Robotic process automation - automation of data entry for student information in university portal. *SSRN J* 2021. Preprint. doi: 10.48175/IJARSC-4004. [\[CrossRef\]](#)
- [26] Ding T, Liu Y, Wang S. Dual-branch mutual learning framework for job-resume matching. *Int J Pattern Recogn Artif Intell* 2025;40:1. [\[CrossRef\]](#)

- [27] Hawari AR, Munawir M, Zulfan Z, Satria DSD, Susmanto S. Job recommendation system using the content-based filtering method. *Proceeding* 2022;5:138–140. [\[CrossRef\]](#)
- [28] Mydyti H, Ware A. Integrating intelligent web scraping techniques in internship management systems: enhancing internship matching. *Ann Emerg Technol Comput* 2025;9. [\[CrossRef\]](#)
- [29] Suman H, Tamkiya H, Kushwa AS. Candidate background verification using machine learning and fuzzy matching. *Int J Emerg Technol Innov Res* 2020;7:1643–1647.
- [30] Rojas-Galeano S, Posada J, Ordoñez E. A bibliometric perspective on AI research for job-résumé matching. *ScientificWorldJournal* 2022;2022:1–15. [\[CrossRef\]](#)
- [31] Lin Y, Lei H, Addo PC, Li X. Machine learned resume-job matching solution. *arXiv*. 2016. Preprint. doi: 10.48550/arXiv.1607.07657.
- [32] Lunn S, Zhu J, Ross M. Utilizing web scraping and natural language processing to better inform pedagogical practice. In: 2020 IEEE Frontiers in Education Conference (FIE); 2020 Oct; Uppsala, Sweden. IEEE; 2020. p. 1–9. [\[CrossRef\]](#)
- [33] Jirjees AK, Ahmed AM, Abdullah AA, Lu J, Noori EM, Kareem R, et al. Machine learning for recruitment: analyzing job-matching algorithms. *TechRxiv* 2025. Preprint. doi: 10.36227/techrxiv.174240031.12397312/v1. [\[CrossRef\]](#)
- [34] Jagwani V, Meghani S, Pai K, Dhage S. Resume evaluation through latent Dirichlet allocation and natural language processing for effective candidate selection. *arXiv*. 2023. Preprint. doi: 10.48550/arXiv.2307.15752.
- [35] Mariappan P, Krishna K, Sahoo J, Preetham S. Smart resume screening and job represents recommendation system. *SSRN J* 2025. Preprint. doi: 10.2139/ssrn.5141402. [\[CrossRef\]](#)
- [36] Dolui C, Devi SF, Dasadhikari S, Roy D. Magnetostrictive sensor based bridge health assessment using machine learning classifiers. In: 2023 IEEE Pune Section International Conference (PuneCon); 2023 Dec; Pune, India. IEEE; 2023. p. 1–6. [\[CrossRef\]](#)
- [37] Chawrasia SK, Hembram D, Bose D, Chanda CK. Design of solar battery swapping station for EV using LSTM-assisted solar power forecasting. *Microsyst Technol* 2024;30:1087–1098. [\[CrossRef\]](#)
- [38] Thakur S, Jha R, Dalawat M, Jha P, Khandare A. Scraping data from Google Maps: comparisons and experimentations. In: Ghinea G, Semwal VB, Khandare A, editors. *Intelligent Computing and Networking*. Singapore: Springer Nature Singapore; 2025. p. 237–259. [\[CrossRef\]](#)
- [39] Ünal S, Günay O, Akkurt I, Gunoglu K, Tekin HO. A comparative study on breast cancer classification with stratified shuffle split and K-fold cross validation via ensembled machine learning. *J Radiat Res Appl Sci* 2024;17:101080. [\[CrossRef\]](#)