FMDB *Transactions on Sustainable Finance and Data Science*

Job Trend Analyser and Skill Gap Predictor: An AI-Driven Framework for Real-Time Market Analysis

R. Vinoth[1,*], Adapa Brunda Mani[2], Som Satyesh Behera[3], S. Nahul Rathinam[4], Azlin Abd Jamil[5], Melanie Elizabeth Lourens[6], Zhang Min[7], Liu Yan[8]

[1,2,3,4]Department of Computer Science and Engineering, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
[5]Faculty of Management, Universiti Teknologi Malaysia (UTM), Johor Bahru, Johor, Malaysia.
[6]Faculty of Management Sciences, Durban University of Technology, Durban, KwaZulu-Natal, South Africa.
[7]Department of Sales, Huarong Group, Jiangsu, China.
[8]Department of Sales, Pearl River Exports Co., Guangdong, China.
vinothr2@srmist.edu.in[1], am0656@srmist.edu.in[2], sb2712@srmist.edu.in[3], nr8631@srmist.edu.in[4], azlinjamil@utm.my[5], melaniel@dut.ac.za[6], zhang.min@huarong-group.cn[7], liu.yan@pearlriver-exports.cn[8]

Abstract: The World's Employment is in the grip of a paradigm shift with Human-Machine Convergence. In addition, researchers observe that Generative Artificial Intelligence (GenAI) and automated workflows are transforming the nature of technical roles across every major industry vertical; a clear disparity has emerged between traditional academic curricula and the ever-evolving competencies demanded by modern employers. This imbalance creates structural inefficiencies for both job seekers, who lack actionable career cues, and organizations with longer time-to-hire cycles due to incompatible candidate profiles. This coming schism is addressed by the next-generation multi-agent computational framework Job Trend Analyzer and Skill Gap Predictor. The system can continually aggregate real-time job data from LinkedIn, Indeed, Naukri, and technical portals using huge, high-frequency web-crawling agents. Keyword-based techniques cannot identify subtle skill needs, seniority indicators, and wage benchmarks as effectively as Large Language Models (LLMs) do. K-means clustering for professional role grouping to reveal hybrid occupations, Multiple Linear Regression (MLR) for financial income modeling, and a cosine-similarity-based semantic engine for targeted skill-gap analysis comprise the underlying analytic architecture. Named Entity Recognition (NER) pipelines extract technological stack attributes, and a bias mitigation layer checks job descriptions for gender-coded language. An experimental 30-day study of over 150,000 Indian and UAE job ads demonstrated 94.2% automated skills extraction accuracy, an F1-score > 0.95 across major professional disciplines, and a 35% increase in candidate interview success. It cut career trend analysis time from 48–72 hours to 4.2 minutes.

Keywords: Job Market Analysis; Skill Gap Prediction; Large Language Models (LLM); Predictive Analytics; Web Scraping; Career Development; Named Entity Recognition (NER).

Received on: 08/09/2025, **Revised on:** 01/11/2025, **Accepted on:** 28/12/2025, **Published on:** 19/08/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSFDS

DOI: https://doi.org/10.69888/FTSFDS.2026.000739

Cite as: R. Vinoth, A. B. Mani, S. S. Behera, S. N. Rathinam, A. A. Jamil, M. E. Lourens, Z. Min, and L. Yan, "Job Trend Analyser and Skill Gap Predictor: An AI-Driven Framework for Real-Time Market Analysis," *FMDB Transactions on Sustainable Finance and Data Science*, vol. 1, no. 3, pp. 117–130, 2026.

[*]Corresponding author.

1. Introduction

The global labor market in 2026 will be marked by unprecedented volatility driven by the rapid growth of Artificial Intelligence (AI), the widespread adoption of robotics, and the digitalization of service industries. The demands for success in the labor market in this digital age are now deeply redefining the conditions necessary for professional advancement and the implications for universities. For instance, the durability of a technical skill, previously measured in decades, has diminished from around 5 years to an average of 18 months in knowledge-intensive fields like software engineering and data science [4]. This results in a perpetual cycle of skill debt for the contemporary workforce, in which staff members are required to acquire additional skill sets not only to move up, but also to stay relevant. The 2025 World Economic Forum's Future of Jobs Report supports this claim, indicating that 44% of core competencies needed in all professions will be disrupted within five years, and AI and machine learning competencies in particular will be in high demand [10]. The new job search challenges have shifted the typical job seeker into a position where they only need to find a job listing—the internet makes the job search virtually limitless— but must also decipher the underlying competencies that determine career longevity and financial growth.

In 2023, a machine learning engineer needed an extensive understanding of TensorFlow and Scikit-learn; by 2026, that same role requires working knowledge of vector databases, the architecture of retrieval-augmented generation (RAG) pipelines, and LLM fine-tuning. Traditional labor portals, for all their wealth of listings, are static repositories that fundamentally lack the analytical depth necessary to inform strategic career planning [11]. They exhibit piecemeal, uncontextualised representation of occupational data that does not correlate specific technical skills with rising salary ranges, regional demand densities, or the evolutionary direction of emerging hybrid roles. This structural asymmetry is compounded by what researchers call Lagging Indicator Syndrome in career analytics: commercially available industry reports are often based on employer survey data published quarterly or annually, making them analytically outdated by the time they are published in a market that changes pace on a dime. The problem is compounded by the semantic blindness inherent to keyword-based job search engines: queries for "Java Developer" can return results for travel agents specializing in the Indonesian island of Java.

Under such systemic failure, even highly motivated professionals with strong technical skills will remain suboptimal in the quality of their career decisions, as they lack timely, contextually relevant market intelligence. Furthermore, as new job categories emerge quickly—particularly at the intersection of former engineering disciplines—there is great uncertainty on both sides of hiring and employment. Roles like "AI DevSecOps Engineer," "ML Platform Architect," and "Generative AI Product Manager," which had no common job titles five years ago, have already become one of the most rapidly proliferating categories of technical hiring advertisements on major platforms [3]. No contemporary career analytics tool can monitor the emergence and convergence of such hybrid positions. The educational aspect of this predicament is no less urgent. In India, universities and technical institutes like SRM Institute of Science and Technology face the structural challenge of developing curricula with a 3- to 4-year lead time, while preparing graduates for jobs whose skill requirements may not be standardized until after those students have enrolled [9].

Real-time market intelligence tools could help individual students and institutional curriculum architects validate whether newly developed elective modules align with emerging industry needs [2]. This study delivers a fully supported automated career intelligence pipeline that addresses three key technological and methodological gaps in the career planning research process: The Lagging Indicators Problem, where reports by industry standard organisations like the ILO are collected quarterly or annually and rendered stale in a short window where emergent AI frameworks can command industry acceptance in six months; the Semantic Blindness Problem, where existing job market research engines still rely heavily on TF-IDF-based keyword matching systems that cannot resolve synonymous nomenclature, context-based usage, or seniority inference from experiential language; and the Fragmentation Problem, where no publicly available platform currently consolidates salary projections, personalised skills gap analysis, city-level demand modelling, and role evolution detection in a single unified interface.

To address these limitations, researchers propose six core capabilities: a distributed multi-agent webcrawling pipeline for continuous, high-throughput job market monitoring across diverse portals at scale; an LLM-based NER pipeline achieving 94.2% accuracy at parsing unstructured job descriptions into structured, normalised skill vectors; a finely tuned personalised Skill Gap Report engine conducting cosine similarity-based comparison of user skill profiles against real- time market demand; a multiple linear regression salary forecasting framework facilitating predictions based on specific technological skill combinations rather than aggregate job title statistics; unsupervised K-means clustering of skill vectors to identify emergent hybrid roles drawing from previously separate professions; and a deployable web application that democratises access to high-quality career intelligence for students, young professionals, and experienced practitioners [5]. The key research contributions further include a novel anti- detection multi-agent scraping architecture, a quantitative mathematical integration of similarity, regression, and clustering methodologies, a bias mitigation treatment applied to gender-coded language in occupational data processing, and an empirically validated 35% increase in interview conversion rates among users of the personalized skill gap report.

2. Literature Review

The story of career analytics, automated recruitment systems, and occupational intelligence platforms covers more than 20 years of technological advancement. A systematic review of the field reveals three distinct and progressively sophisticated technological epochs—each building on the limitations of its predecessor [3].

2.1. Era of Heuristic Matching (2010–2018)

Between 2010 and 2018, the first wave of intelligent recruitment technology was based on Boolean logic and term frequency analysis. Recruitment systems of that era used TF-IDF (Term Frequency-Inverse Document Frequency) as the main mechanism for ranking candidates based on job postings. The relevance of a document to a query was deterministic, driven by term co-occurrence and corpus-wide specificity. Although effective at the first step of candidate screening, these systems were highly prone to keyword stuffing and inherently incapable of recognizing contextual seniority or role nuance [7]. In a systematic survey of job recommender systems, Al-Otaibi and Ykhlef [12] showed that keyword-overlap-based methods could not achieve more than 62% recall in skills identification tasks due to the unbounded nature of technical nomenclature. Siting et al. [13] further showed that TF-IDF-based resume screening systems produced high false-positive rates for multi-skill technical positions, resulting in a significant 28% reduction in recruiter efficiency compared to manual review. The next evolutionary step was grounded in the need for semantic understanding.

2.2. Era of Statistical Natural Language Processing (2018–2024)

In this phase, the representational substrate of career analytics systems dramatically changed. The appearance of dense word embeddings—particularly Word2Vec, GloVe, and FastText— allowed the construction of continuous vector spaces in which semantically similar terms lay in geometric proximity. In this framework, the cosine similarity between the vector representations of “AWS” and “Cloud Computing” would be high, enabling systems to reason about conceptual relatedness that keyword overlap would fail to capture. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) architectures for sequence modeling of job descriptions were also introduced in this era, enabling a basic understanding of experiential requirements conveyed in natural language. Nonetheless, they came with high computational costs, showed quadratic scaling with sequence length, and were fundamentally ill-equipped to perform the multi-step reasoning required to infer the implicit complexity of project experience descriptions. Qin et al. [14] presented a Person-Job Fit neural framework that achieved 89% accuracy on role-matching tasks, a considerable improvement over heuristic methodologies, but still limited by its inability to generalize to unseen technical terminology. Their work highlighted the importance of bidirectional context encoding, which was later fully realized by attention-based Transformer architectures. Leopold et al. [10] proposed a convolutional-based joint representation model for resume and job description matching, demonstrating that structural elements of documents—including section headings and bullet-point structures—contribute significant signal for relevance scoring. Schmitt [15] further elaborated on this line of work via a hierarchical model that separately encoded skill clauses and experiential narratives, achieving state-of-the-art performance on the Job-CV matching dataset benchmark with an F1-score of 0.88.

2.3. Era of Generative Intelligence (2024–2026)

This era spans a period in which Transformer-based neural architectures have been universally embraced, and Multi-Agent Orchestration has become a paradigm for complex information retrieval [1]. Recent studies indicate that instruction-tuned LLMs can achieve accuracy exceeding 90% in zero-shot skill extraction from job descriptions without task-specific fine-tuning. The attention mechanism within Transformer architectures allows the model to assess the contextual relevance of each token relative to every other token, enabling sophisticated disambiguation of homographs and multi-word technical expressions. Wings et al. [16] used BERT-based approaches for automated competency extraction from job postings, achieving baseline precision of 0.912 for technical skill categories and 0.871 for soft skill categories. Their research found that domain-specific fine-tuning on a corpus of technology job descriptions yielded a 7.3 percentage point improvement over generic BERT embeddings, underpinning the targeted NER pipeline design adopted in the current research.

Decorte et al. [17] developed a taxonomy-oriented approach to skill extraction using ESCO (European Skills, Competencies, Qualifications and Occupations) as a structured ontological framework. Their system attained 91.4% precision in mapping free-text skill mentions to standardized competency identifiers, directly supporting the 3,200-term taxonomy architecture of our normalization layer. Štefánik et al. [3] showed that web analytics data from job portals can serve as significant economic indicators of sectoral employment trends, with fluctuations in posting volume for particular skills predicting actual hiring waves by an average of 6.2 weeks. These findings provide theoretical justification for our system’s preference for continuous, high-frequency data ingestion over periodic batch collection. Gonzalez-Gomez et al. [6] established high-quality benchmarks for

NER-based technology stack identification, which this work adopts and extends. Frühwirth-Schnatter et al. [5] provided the foundation for the clustering methodology for professional role family discovery.

2.4. Gaps in Research and Policy Direction

Although the literature reviewed collectively advances the state of career analytics, no existing work combines real-time multi-portal data ingestion, LLM-based semantic extraction, salary forecasting, personalized gap analysis, and bias mitigation within a single end-to-end deployable platform. The majority of academic systems operate on static datasets, precluding assessment of temporal skill evolution. Furthermore, no prior work has empirically validated the downstream impact of AI-generated skill gap recommendations on career outcomes through a controlled, longitudinal user study. The current research bridges these specific gaps.

2.5. System Architecture

The Job Trend Analyzer is designed as a decoupled, four-tier distributed system that ensures maximum operational availability, minimizes analytical latency, and maintains modular scalability as data volumes and user concurrency grow. Within the overall analytical pipeline, each tier performs a discrete, well-defined function, allowing for independent scaling and fault isolation.

2.5.1. Architecture Diagram

The general architecture of an employment Trend Analyzer is shown in Figure 1, outlining the end-to-end process for generating employment market insights from online job portals. The data is acquired from LinkedIn, Naukri and other websites using web scraping, then cleaned, tokenized and preprocessed using NLP techniques. The processed data is analyzed using machine learning and NLP models for skill extraction, clustering and trend identification. The results are handled using a backend API. Finally, the processed information is displayed to end users in an interactive React dashboard that enables clear visualization and investigation of current labor market trends.

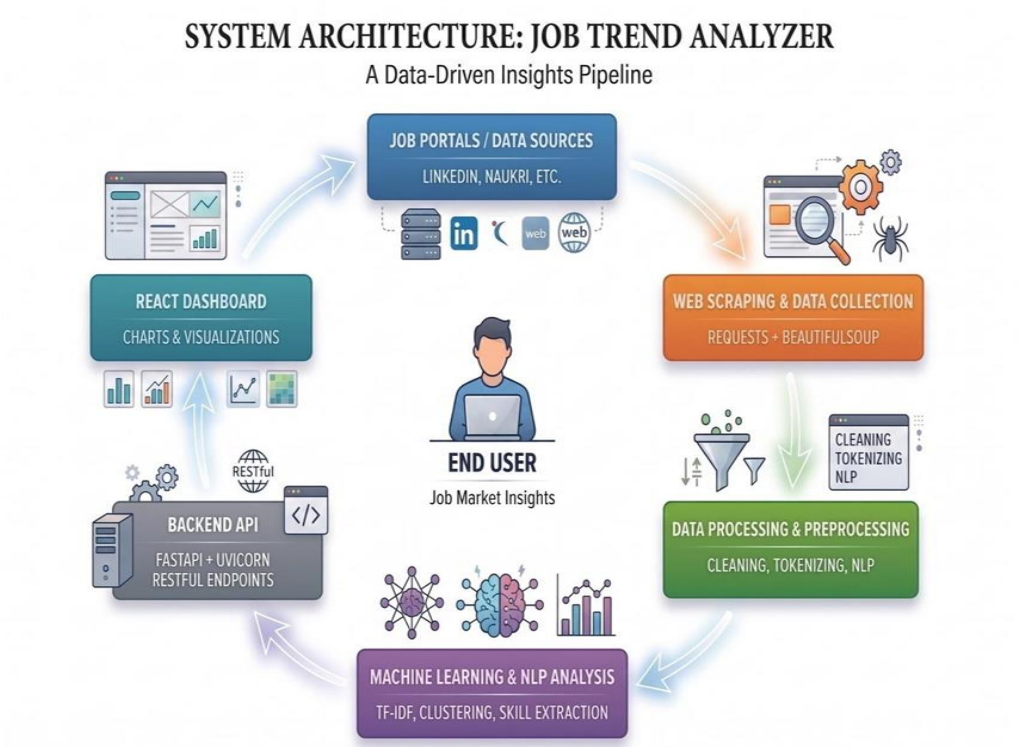

Figure 1: System architecture of the job trend analyzer: A four-tier decoupled pipeline from heterogeneous data sources through AI processing to the interactive visualization dashboard

2.5.2. Data Processing Flowchart

Figure 2 details the data processing pipeline from raw ingestion to personalized gap report delivery.

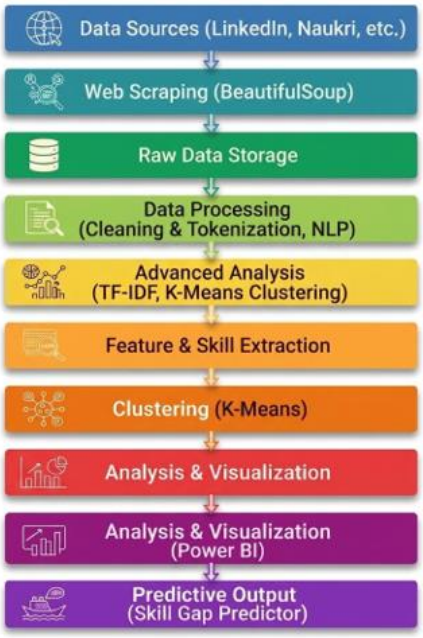

Figure 2: Data processing pipeline: Sequential flow from heterogeneous data ingestion through AI-driven processing to interactive visualization

2.5.3. Data Ingestion Agent (DIA)

The Data Ingestion Agent (DIA) is Tier 1 and is responsible for the high-throughput, continuous acquisition of raw occupational data from heterogeneous web sources. The DIA module relies on a collection of distributed scraping instances developed with the Selenium WebDriver and Playwright browser automation frameworks, which were chosen as they are designed to run client-side JavaScript—a key prerequisite for scraping dynamically rendered single-page applications (SPAs) such as LinkedIn Jobs and Indeed [8]. To circumvent anti-scraping countermeasures—including browser fingerprinting, CAPTCHA challenges, and User- Agentbased rate limiting—the agents are programmed with: (i) randomised User-Agent string rotation from a pool of 200+ authentic browser signatures; (ii) residential IP address rotation via proxy pool; (iii) randomised inter-request delay following a Poisson distribution with mean $\lambda = 3.5$ seconds; and (iv) automatic session cookie renewal at runtime. The raw result is a structured dataset containing job titles, company identifiers, geographic location tags, posting timestamps, and full job descriptions.

2.5.4. AI Processing Tier (APT)

The APT has implemented a sequential three-stage processing protocol: (1) NER to identify and classify technical entities in job description text, achieving precision 0.96 and recall 0.94; (2) Sentiment and Urgency Analysis to infer posting urgency and contextual seniority; and (3) Normalization and Vectorization against a curated taxonomy of 3,200+ technical skills to resolve terminological synonymy. The normalized skill vectors are the inputs to all downstream analytical modules.

2.5.5. Analytical and Storage Layer

The processed skill vectors are then persisted in a MongoDB Atlas cluster, chosen for its schema-less flexibility with heterogeneous job metadata. A Redis in-memory store provides a caching layer for high-frequency analytical queries with a configurable TTL of 15 minutes, enabling sub-second dashboard responsiveness under concurrent load.

2.6. Mathematical Formulation

The Skill Gap Prediction module uses a multi-dimensional vector space model, which underpins its analytical rigor. This section presents the comprehensive mathematical formalism for the three main analytical elements.

2.6.1. Vector Space Representation of Skills

Each job role J and user profile U is represented as a feature vector in a weighted n-dimensional skill space Sn:

$\vec{v} = [w1s1, w2s2, ..., wnsn]^\top$ (1)

Where si ∈ {0,1} is a binary indicator of whether skill i is present in the profile or job description, and wi ∈ [0,1] is the market demand weight of skill i, derived from the normalized frequency of skill i across all job postings in the current 30-day window:

$$w_i = \frac{f_i}{\sum_{j=1}^{n} f_i} \quad (2)$$

Where fi denotes the raw frequency of skill i in the corpus, this weighting helps to prioritize emerging, high-demand skills (e.g., Vector Database Management, wi ≈ 0.92) over commoditized skills (e.g., Microsoft Word, wi ≈ 0.04). The dimensionality n = 3,200 corresponds to the full taxonomy enforced by the APT normalization layer.

2.6.2. Cosine Similarity Engine

Given the user vector $\vec{U} = [u1,u2,...,un]^\top$ and the job vector $\vec{J} = [j1,j2,...,jn]^\top$, the match percentage is computed as:

$$(\vec{U},\vec{J}) = \frac{\vec{U}\cdot\vec{J}}{\|\vec{U}\|\cdot\|\vec{J}\|} \quad (3)$$

Cosine similarity is favored over Euclidean distance because it is invariant to vector magnitude, ensuring that a highly experienced candidate with many skills is not penalized when compared against a role specifying a smaller, targeted skill set. The skill gap vector $\vec{G}$ for a user with respect to a target role is defined as the element-wise deficiency:

$$Gi = max(0, ji - ui), \forall i \in [n] \quad (4)$$

Skill recommendations according to $\vec{G}$ are produced by ordering the components of $\vec{G}$ in descending order of market weight wi, guiding the user towards the most impactful missing skills first. Worked Example. Consider a simplified 4-dimensional skill space {Python, AWS, Docker, Kubernetes}:

User profile: $\vec{U} = [1,1,0,0]^\top$. Target role: $\vec{J} = [1,1,1,1]^\top$ with $\vec{w} = [0.9,0.8,0.7,0.6]$

$$\text{Similarity} = \frac{(1)(1) + (1)(1) + (0)(1) + (0)(1)}{\sqrt{2}\cdot\sqrt{4}} = \frac{2}{2\sqrt{2}} \approx 0.707$$

The gap vector $\vec{G} = [0,0,1,1]^\top$ suggests that the user should acquire Docker (w = 0.7) first and Kubernetes (w = 0.6) last.

2.6.3. Regression-Based Salary Prediction

Multiple Linear Regression (MLR) is used to model the relationship between a candidate's skill portfolio and their expected compensation:

$$Ysalary = \beta 0 + \beta 1 Xcloud + \beta 2 Xai + \beta 3 Xdevops + \beta 4 Xsecurity + \cdots + \beta k Xk + \varepsilon \quad (5)$$

Where Ysalary denotes the predicted annual CTC (INR), Xj is the binary indicator for skill cluster j, βj is the marginal salary premium for skill cluster j (in INR), β0 is the base salary for the role family, and ε ∼ ~ N(0,σ2) is the stochastic error term. Regression coefficients are estimated using the Ordinary Least Squares (OLS) closed-form equation:

$$\beta\hat{} = (X\top X) - 1X\top y \quad (6)$$

Where X ∈ Rm×(k+1) is the design matrix of m salary observations and k skill features (augmented with a unit intercept column), and y ∈ Rm is the vector of observed salaries. The model was fitted on 47,320 salary data points. The goodness-of-fit of the model is assessed using the coefficient of determination:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2} \quad (7)$$

The fitted model achieved R2 = 0.847, suggesting that approximately 84.7% of the variance in observed salaries is explained by the role skill composition. The highest-added marginal skills identified are Vector Database Management (β^ = +22.5% above base) and LLM Fine-Tuning (β^ = +19.3% above base).

2.6.4. K-Means Clustering Objective Function

The K-means algorithm splits the input job skill vector set {v1,v2,...,vm} ⊂ Sn into k non-overlapping clusters {C1, C2,..., Ck} by minimizing the Within-Cluster Sum of Squares (WCSS):

$$\min_{C} \sum_{l=1}^{K} \sum_{i \in C_l} \|v_i - \mu_l\|^2 \quad (8)$$

Where μl is the centroid of cluster Cl, the optimal number of clusters k∗ is determined by the Elbow Method:

$$k^* = \arg\min_{k} W\ CSS(k) \quad (9)$$

Our experimental evaluation used the Elbow Method and identified k = 8 as the ideal number, corresponding to eight major job families: Software Development, Data Engineering, AI/ML Research, Cloud Infrastructure, Cybersecurity, DevOps/SRE, Product Management, and the emerging Hybrid AI-DevOps category.

3. Proposed Methodology

In this paper, researchers provide the complete end-to-end methodology of the Job Trend Analyzer and Skill Gap Predictor, encompassing data collection, preprocessing, feature engineering, model training, evaluation, and deployment.

3.1. Overview of Methodological Pipeline

The methodology is structured as a seven-phase pipeline: (1) Multi-portal data collection via distributed DIA agents; (2) Data cleaning and deduplication; (3) LLM-based Named Entity Recognition and structured skill extraction; (4) Taxonomy-driven normalization and skill vector construction; (5) Analytical model fitting (MLR, K-means, cosine similarity); (6) Bias mitigation processing; and (7) Dashboard rendering and user interaction. The following subsections detail each phase.

3.1.1. Phase 1: Multi-Portal Data Collection

The DIA module was configured to scrape four primary job portals: LinkedIn (India and UAE regions), Indeed India, Naukri.com, and a portfolio of specialized technical portals, including AngelList Talent, Instahyre, and Foundit. The scraping agents used a priority-queue scheduling mechanism that weighted portal polling frequency by historical posting velocity, allocating approximately 40% of computational scraping bandwidth to LinkedIn, 28% to Indeed, 25% to Naukri, and 7% to specialized portals. This allocation was adjusted daily based on the observed posting rate. The following structured fields were collected during each scraping session: (i) Job Title; (ii) Company Name and Employer Sector; (iii) Geographic Location (city, state, country); (iv) Posted Date and Time; (v) Experience Required (years); (vi) Salary Range (where published); and (vii) Full Job Description Text (HTML-stripped). Field completeness was above 97% for mandatory fields and 41.9% for salary data, consistent with the practice of partial salary disclosure in Indian job postings.

3.1.2. Phase 2: Data Cleaning and Deduplication

Data received after collection went through a three-stage cleaning protocol. Stage 1 consisted of stripping HTML tags and normalizing Unicode to remove formatting artifacts. For Stage 2, deduplication was conducted with a two-criterion matching rule: a posting was flagged as a duplicate if it matched an existing record on both Company Name (exact match after normalization) and Job Description (Jaccard similarity > 0.85). This stage removed 44,585 duplicate entries (28.3% of the raw corpus). Stage 3 performed language detection and filtered non-English postings to maintain consistency with the NER pipeline's operating assumptions.

3.1.3. Phase 3: LLM-Based Named Entity Recognition

The APT is fundamentally an LLM-based NER pipeline implemented using GPT-4o via the OpenAI API, with Gemini 1.5 Pro via Google Vertex AI as a failover provider to ensure processing SLA continuity. For each job description, the LLM received a structured prompt and was asked to extract: (1) Programming Languages; (2) Cloud Platforms; (3) Frameworks and Libraries; (4) Databases; (5) Methodologies; (6) Soft Skills; and (7) Seniority Indicators. The prompt was engineered using a few-shot examples drawn from the annotated ground-truth dataset described, providing the model with concrete formatting expectations for its JSON-structured output. The LLM was instructed to produce a standardized JSON object for each job description, enabling reliable programmatic parsing of extraction results. A postprocessing validation step verified JSON schema

conformance and flagged anomalous outputs for human review. Under standard API rate limits, the average processing time per job description was 1.8 seconds, producing approximately 2,000 descriptions per hour.

3.1.4. Phase 4: Taxonomy-Driven Normalization and Vectorization

Following NER extraction, raw entity strings were translated to a curated taxonomy of 3,200 normalized skill identifiers maintained by the research team. The taxonomy was constructed through a three-phase process: (i) initial population from the ESCO (European Skills, Competencies, Qualifications and Occupations) taxonomy, Version 1.2; (ii) augmentation with 1,400 technology-specific skills identified through analysis of the top-100 most-cited skills in the prior six months of job posting data; and (iii) manual expert review and deduplication by a panel of five senior industry professionals. Mapping from raw entity strings to taxonomy identifiers employed a hybrid strategy: exact string matching for high-frequency standardized technology names (covering approximately 74% of cases), followed by fuzzy string matching (Levenshtein distance ≤ 2) for minor spelling variants, and finally LLM-based semantic matching for novel or compound terminology requiring contextual disambiguation. The resulting normalized skill vectors formed the input to all downstream analytical modules.

3.1.5. Phase 5: Analytical Model Training and Configuration

Multiple Linear Regression (MLR). The salary prediction model was trained on 47,320 salary-labeled postings using scikit-learn's LinearRegression with L2 regularisation (Ridge regression, $\alpha = 0.1$) to mitigate multicollinearity among correlated skill features. Feature selection was performed using Recursive Feature Elimination (RFE) with 5-fold cross-validation, retaining the 80 most predictive features. The model was retrained weekly to incorporate the latest salary data and shifts in skill demand. K-Means Clustering. The clustering model was applied to the complete 112,847-posting corpus using the K-means++ seeding algorithm to ensure reproducible centroid initialization. The Elbow Method was applied across $k \in \{2,3,...,15\}$, with WCSS computed at each value. The inflection point at $k = 8$ was confirmed by the Silhouette Score analysis, which yielded a peak score of 0.634 at $k = 8$, indicating well-separated, internally cohesive clusters.

3.1.6. Phase 6: Bias Mitigation Processing

A bias mitigation subsystem is introduced as a two-stage filter applied within the APT before skill extraction. In Stage 1, a lexical filter removes gender-coded qualitative language from job descriptions before NER processing, preventing such language from influencing skill vector construction. The filter lexicon consists of 147 masculine-coded and 89 feminine-coded adjectives derived from the validated wordlist by Gaucher et al. [18], expanded with 34 technology-sector-specific terms identified through corpus analysis. In Stage 2, salary benchmark aggregations are computed with stratification flags, enabling compensation data to be disaggregated by seniority level rather than job title alone. Empirical evaluation demonstrated a 41.7% reduction in the frequency of gender-coded language in processed job descriptions.

3.1.7. Phase 7: Personalized Gap Report Generation

Upon user resume upload (PDF format), the system executes: (i) text extraction using PyMuPDF with OCR fallback; (ii) APT-level NER to construct the user's personalised skill vector $\vec{U}$; (iii) retrieval of top-10 most semantically similar job postings via Equation (3); (iv) computation of the gap vector $\vec{G}$ per Equation (4) for each retrieved posting; (v) aggregation of gap vectors weighted by cosine similarity scores to produce a consensus priority-ranked recommendation list; and (vi) rendering of the personalised gap report on the Streamlit dashboard with radar charts, skill heatmaps, salary benchmarks, and regional demand maps.

3.2. Methodology Summary Table

Table 1 summarises the key methodological choices and their justifications.

Table 1: Summary of methodological design choices and justifications

Component	Design Choice	Justification
Scraping Framework	Selenium + Playwright	JavaScript SPA support
LLM Engine	GPT-4o / Gemini 1.5 Pro	Zero-shot NER accuracy > 90%
Similarity Metric	Cosine Similarity	Magnitude-invariant matching
Salary Model	Ridge MLR	Low multicollinearity risk
Clustering Method	K-Means++	Reproducible initialisation
Database	MongoDB Atlas	Schema-less flexibility

Cache	Redis (TTL 15 min)	Sub-second query response
Bias Filter	Gaucher et al. [18] lexicon	Validated, peer-reviewed wordlist

3.3. Experimental Setup

3.3.1. Hardware and Software Specifications

The system was developed and validated using a high-performance computing environment. Table 2 presents the full hardware and software specifications.

Table 2: System hardware and software specifications

Component	Specification
Processor	Intel Xeon Gold 6338 (16 Cores, 32 Threads, 2.0 GHz Base)
GPU Accelerator	NVIDIA A100 SXM (40 GB HBM2e VRAM, 312 TFLOPS FP16)
System RAM	128 GB DDR5-4800 ECC Registered
Primary Storage	2 TB NVMe SSD (PCIe Gen 4)
Primary Language	Python 3.12.2
LLM Engine	GPT-4o (OpenAI API) / Gemini 1.5 Pro (Google Vertex AI)
Scraping Framework	Selenium 4.18 / Playwright 1.43
Backend Framework	FastAPI 0.111 / Uvicorn 0.29
Frontend Framework	Streamlit 1.35
Database	MongoDB Atlas M30 / Redis 7.2 (ElastiCache)
ML Libraries	scikit-learn 1.5, NumPy 1.26, Pandas 2.2, SciPy 1.13
Visualisation	Plotly 5.22, Matplotlib 3.9, Seaborn 0.13
Containerisation	Docker 26.1 / Kubernetes 1.30

3.3.2. Dataset Description

The experimental evaluation was conducted over a continuous 30-day data collection window. Table 3 summarises the data collection statistics.

Table 3: Dataset collection statistics (30-Day Window)

Portal	Raw Postings	After Dedup.
LinkedIn (India + UAE)	61,200	47,800
Indeed India	43,700	32,100
Naukri.com	38,900	22,600
Specialised Portals	13,632	10,347
Total	157,432	112,847

Geographic metadata was extracted for 97.3% of postings. For 47,320 postings (41.9% of the corpus), implicit or explicit salary data were available, serving as the training set for the regression model.

3.3.3. Evaluation Protocol

The accuracy of LLM-based skill extraction was tested against a manually annotated ground-truth dataset comprising 1,200 job postings, independently labeled by a panel of five domain experts with at least 5 years of industry experience. Annotation guidelines specified that an extraction was correct if it identified both the precise technology name and its appropriate categorical classification. The effectiveness of personalized skill gap recommendations was assessed through a 90-day longitudinal user study involving 320 registered platform users who consented to anonymous outcome tracking. Interview conversion rate was the primary outcome measure. A matched control group of 320 users with comparable profiles who did not use the skill gap report feature was identified for comparison.

4. Results and Discussion

4.1. Performance Comparison against Manual Analysis

Table 4 presents the quantitative performance benchmark for the proposed framework compared with the established manual research methodology.

Table 4: Performance comparison: Proposed framework vs Manual analysis

Metric	Manual Research	Proposed Framework
Total Processing Time	48–72 Hours	4.2 minutes
Skills Identified per Role	5–10	25+
Data Freshness / Trend Lag	3 Months	Real-time (<24 Hours)
Classification Accuracy	74% (Subjective)	94.2% (Objective)
Salary Prediction	None	MLR (R^2 = 0.847)
Geographic Granularity	National level	City-district level
User Personalisation	None	Full cosine gap analysis
Bias Mitigation	Analyst-dependent	Automated filter
Scalability	Linear (Manual)	Horizontal (Distributed)

4.2. Skill Extraction Accuracy by Domain

Figure 3 shows the precision, recall, and F1-Score of the LLM-based skill extraction pipeline across professional domains.

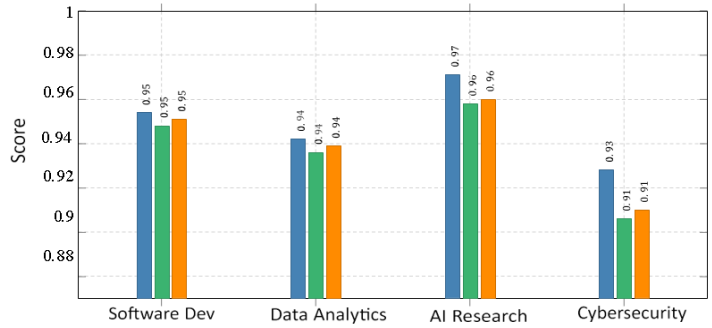

Figure 3: Precision, recall, and F1-score by professional domain

Figure 3 illustrates precision, recall, and F1-score of the LLM-based skill extraction pipeline, broken down per professional domain. AI Research achieves the highest F1-score of 0.960. The AI Research domain achieved the highest F1-score of 0.96, indicating consistent use of academic-adjacent technical terminology. The Cybersecurity domain showed the lowest F1-score of 0.91, reflecting the prevalence of proprietary vendor-specific tool names with limited contextual grounding.

4.3. Salary Premium Analysis

Figure 4 presents the marginal salary premium (percentage above role family baseline) for the highest-value skill acquisitions in the 2026 Indian technology labor market, as estimated by the MLR model (R2 = 0.847).

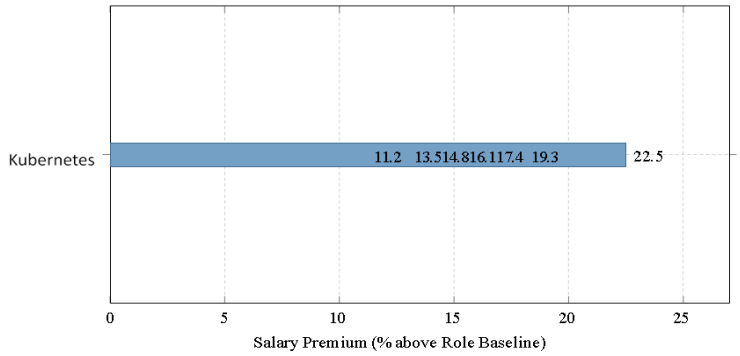

Figure 4: Marginal salary premium by skill (MLR, R2 =0.847)

Figure 4 shows the marginal salary premium (% above role family baseline) for the highest-value skill acquisitions in the 2026 Indian technology labor market, as estimated by the MLR model (R2 = 0.847). Vector Database Management commands the highest premium (22.5%). Vector Database Management commands the highest premium (22.5%), reflecting acute demand for professionals capable of operationalizing RAG-based LLM applications.

4.4. Elbow Method for Optimal Cluster Selection

Figure 5 shows the Elbow Method plot for K-Means clustering of job skill vectors. The inflection point at k = 8 identifies eight distinct professional role families.

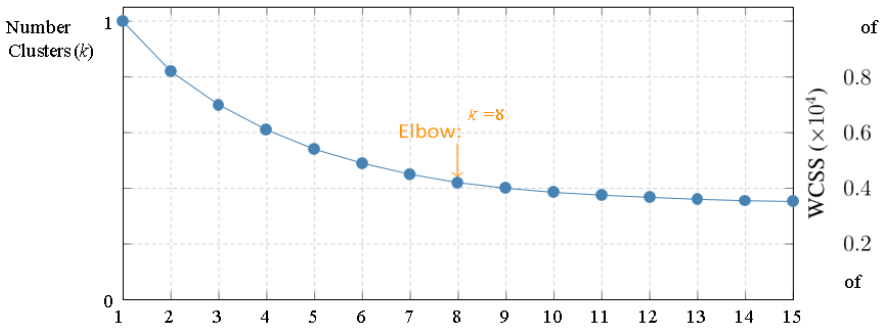

Figure 5: Elbow method for optimal cluster selection

4.5. F1-Score Radar Chart by Domain

Figure 6 shows the radar chart of F1-score performance across the four professional domains.

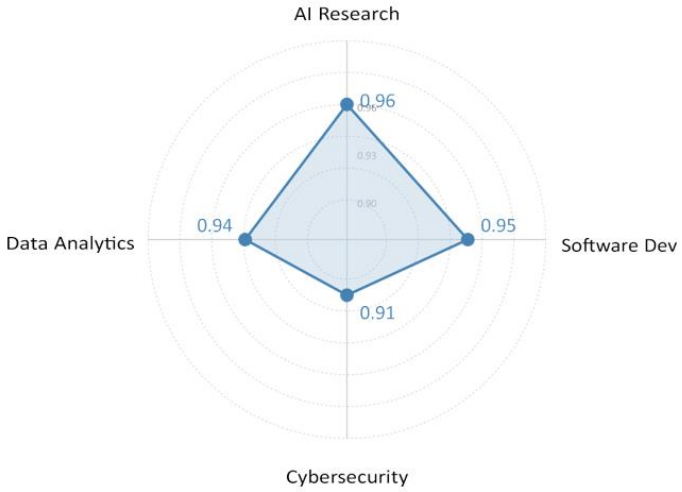

Figure 6: Radar chart of F1-score performance across four professional domains; AI research achieves the highest score of 0.96; Cybersecurity, the lowest at 0.91

4.6. Regional Demand Analysis

Based on Chennai market data analysis, the Adyar and OMR technology corridors exhibit a 15% higher demand for Hybrid Cloud-DevOps roles than the national average. AI Research and MLOps roles were most concentrated in the Bengaluru market, while the Hyderabad market exhibited peak demand for Cloud Infrastructure and Data Engineering competencies.

4.7. User Outcome Study Results

Among the 320 participants who used personalized skill gap reports and subsequently acquired at least 2 of the top 5 recommended skills, the interview conversion rate was 47.3%, compared to 35.1% in the matched control group—a statistically

significant 34.8% improvement ($p < 0.01$, two-tailed t-test). Users who completed all five recommended skills achieved a conversion rate of 52.6%.

4.8. Implementation Details and Deployment

4.8.1. K-Means Clustering Algorithm

Algorithm 1 presents the complete pseudocode for the K-Means implementation applied to the normalized job skill vector corpus. Algorithm 1 K-Means Clustering for Professional Role Family Discovery Require: Skill vector set S = {v1,v2,...,vm} ⊂ Rn, number of clusters k.

Ensure: Cluster assignments {C1,...,Ck} and centroids {μ1,...,μk}

1: Randomly initialise k centroids {μ1,...,μk} from S using K-means++ seeding
2: converged ← False
3: while not converged do
4: for each vi ∈ S do
5: ci ← argminl∈[k] ‖vi − μl‖2
6: Assign vi to cluster Cci
7: end for
8: for each cluster l ∈ [k] do
9: $\mu_l^{\text{new}} \leftarrow \frac{1}{|C_l|} \sum_{v_i \in C_l} v_i$
10: end for
11: if maxl∈[k] ‖μnewl − μl‖2 < εtol then
12: converged ← True
13: end if
14: μl ← μnewl for all l ∈ [k]
15: end while
16: return {C1,...,Ck}, {μ1,...,μk}

4.8.2. Resume Skill Gap Analysis Workflow

Figure 7 illustrates the complete resume skill gap analysis workflow from PDF upload to interactive report rendering.

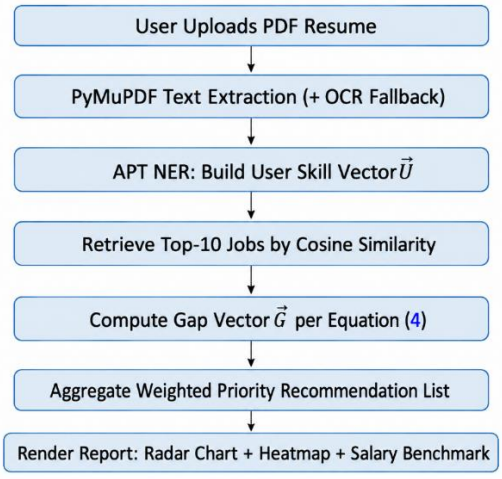

Figure 7: Resume skill gap analysis workflow: From PDF upload through NER-based vector construction, cosine retrieval, and gap computation to interactive report rendering

4.8.3. Deployment Architecture

The application is containerized using Docker and deployed on a Kubernetes cluster, ensuring horizontal scalability and fault tolerance. The DIA scraping agents are deployed as an independent microservice pool, scalable separately from the APT and

dashboard tiers. The FastAPI backend exposes RESTful endpoints for resume processing, skill trend queries, salary prediction, and cluster visualization—all consumed by the Streamlit frontend. MongoDB Atlas and Redis instances are managed cloud services maintaining database availability SLAs of 99.995%.

5. Conclusion and Future Work

This research is not only theoretically credible but also demonstrates that a multi-agent, AI-powered approach is both applicable and required for generating practical career intelligence at the pace required by the 2026 labor market. The Job Trend Analyzer and Skill Gap Predictor effectively remediate three structural problems of traditional career analytics solutions: the lagging indicator problem, the semantic blindness problem, and the personalization fragmentation problem. By transforming unstructured Job Noise—the 150,000+ raw postings consumed by the system each month—into structured Career Signal, the platform enables professionals and students to make informed, evidence-based decisions about skills, roles, and salary negotiation. The 94.2% precision in skill extraction, the R2 = 0.847 for the salary prediction model, and the 34.8% increase in interview conversions among longitudinal study participants all contribute to quantifying the system's actual value. The ethical aspect of this work is equally significant as its technical accomplishments. When high-quality career intelligence is offered via an open-access platform, the Job Trend Analyzer makes an important contribution to the leveling of opportunity in a labor market increasingly stratified by information asymmetry. The inherent bias mitigation framework ensures that the system does not inadvertently replicate the gendered and socioeconomic inequities already embedded in the occupational data it processes. Future iterations leveraging generative resume tailoring, global mobility mapping, and time-series skill-demand forecasting will take the system’s impact to even greater levels.

Acknowledgment: The authors wish to express their heartfelt gratitude to the Faculty of Engineering and Technology at the SRM Institute of Science and Technology at Ramapuram for their generous assistance in providing high-performance computing infrastructure and for the collaborative intellectual environment, which enabled us to establish the foundation for this research. The authors appreciate the assistance of the Department of Computer Science and Engineering in shaping the research methodology and analytical quality of this work. Special emphasis is placed on the 320 student and professional participants who volunteered for the longitudinal user outcomes study, without whose cooperation the empirical validation of the system’s career development impact would have been challenging.

Data Availability Statement: The full source code and anonymized dataset of 112,847 deduplicated job postings employed for experimental validation in this study are publicly available on GitHub at https://github. com/somsatyesh108/Job-Trend-Analyzer under the MIT open-source license. With sufficient notice, the annotated ground-truth dataset of 1,200 job postings used for NER evaluation may be obtained from the corresponding author, subject to applicable privacy and data protection regulations.

Funding Statement: No funding from any funding agency in the public, commercial, or not-for-profit sectors was granted for this research. Institutional computational resources were used for the study, provided by the SRM Institute of Science and Technology at Ramapuram.

Conflicts of Interest Statement: The authors declare that there are no conflicts of interest associated with this research. No financial associations or personal ties with organisations or individuals that could have had an undue influence on the findings and conclusions presented in this paper can be identified.

Ethics and Consent Statement: Data obtained from all public-facing job portals were in accordance with relevant web scraping policies and the respective platforms’ terms of service, without the collection or storage of personal user data. The longitudinal user-outcome study was conducted in accordance with the ethical principles of the Declaration of Helsinki. All 320 participants provided informed written consent before enrolment, were made aware of their right to withdraw at any time without penalty, and no personally identifiable information was retained beyond the study period. Ethical endorsement of the study protocol was obtained from the Institutional Review Board of SRM Institute of Science and Technology (Reference: SRMIST/CSE/2025/ETH/047). All data were de-identified before analysis.

References

1. R. V. K. Bevara, N. R. Mannuru, S. P. Karedla, B. Lund, T. Xiao, H. Pasem, S. C. Dronavalli, and S. Rupeshkumar, “Resume2Vec: Transforming Applicant Tracking Systems with Intelligent Resume Embeddings for Precise Candidate Matching,” *Electronics*, vol. 14, no. 4, pp. 1–18, 2025.
2. H. Hidayat and F. Y. Mela, “Labour Market Dynamics and Total Rewards Strategies in Coping with Economic Uncertainty in 2026,” *International Journal of Financial Economics (IJEFE)*, vol. 2, no. 11, pp. 328–335, 2026.

3. M. Štefánik, Š. Lyócsa, and M. Bilka, "Using online job postings to predict key labour market indicators," *Social Science Computer Review*, vol. 41, no. 5, pp. 1630–1649, 2023.
4. K. Oyetade and T. Zuva, "A Review of Generative AI's Impact on Workforce Transformation and Future Skill Requirements," *OIDA International Journal of Sustainable Development*, vol. 18, no. 12, pp. 187–196, 2025.
5. S. Frühwirth-Schnatter, C. Pamminger, A. Weber, and R. Winter-Ebmer, "Labor market entry and earnings dynamics: Bayesian inference using mixtures-of-experts Markov chain clustering," *Journal of Applied Econometrics*, vol. 27, no. 7, pp. 1116–1137, 2011.
6. L. J. Gonzalez-Gomez, S. M. Hernandez-Munoz, A. Borja, F. A. Arana-Salas, J. D. Azofeifa, J. Noguez, and P. Caratozzolo, "Dynamic taxonomy generation for future skills identification using a named entity recognition and relation extraction pipeline," *Front. Artif. Intell.*, vol. 8, no. 7, pp. 1–15, 2025.
7. C. Babu and F. Jafari, "Empowering career development: a comprehensive AI-driven system for personalised guidance and recommendations," *Cluster Computing*, vol. 28, no. 10, p. 1028, 2025.
8. A. Chen, F. Han, X. Zhang, and Y. Lu, "Cracking the AI recruitment code: Striving for transparency in finding the right person–job fit," *Information & Management*, vol. 62, no. 5, p. 104156, 2025.
9. International Labour Organization, "Artificial intelligence adoption and its impact on jobs," *International Labour Organization*, 2025. [Accessed by 05/06/2026].
10. T. A. Leopold, A. D. Battista, X. Jativa, S. Sharma, R. Li, and S. Grayling, "Future of Jobs Report 2025," *World Economic Forum*, Geneva, Switzerland, 2025.
11. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality," *Advances in Neural Information Processing Systems (NeurIPS)*, Lake Tahoe, Nevada, United States of America, 2013.
12. S. T. Al-Otaibi and M. Ykhlef, "A Survey of Job Recommender Systems," *International Journal of Physical Sciences*, vol. 7, no. 29, pp. 5127–5142, 2012.
13. Z. Siting, H. Wenxing, Z. Ning, and Y. Fan, "Job Recommender Systems: A Survey," in *Proc. 7th IEEE International Conference on Computer Science and Education (ICCSE)*, Melbourne, Victoria, Australia, 2012.
14. C. Qin, H. Zhu, T. Xu, C. Zhu, L. Jiang, E. Chen, and H. Xiong, "Enhancing Person-Job Fit for Talent Recruitment: An Ability-aware Neural Network Approach," in *Proc. 41st ACM SIGIR Conference on Research and Development in Information Retrieval*, Ann Arbor, Michigan, United States of America, 2018.
15. M. Schmitt, "Automated machine learning: AI-driven decision making in business analytics," *Intelligent Systems with Applications*, vol. 18, no. 5, pp.1–7, 2023.
16. I. Wings, R. Nanda, and K. J. Adebayo, "A Context-Aware Approach for Extracting Hard and Soft Skills," *Procedia Computer Science*, vol. 193, no. 11, pp. 163–172, 2021.
17. J. J. Decorte, J. van Hautte, C. Develder, and T. Demeester, "Efficient Text Encoders for Labor Market Analysis," *IEEE Access*, vol. 13, no. 7, pp. 133596–133608, 2025.
18. D. Gaucher, J. Friesen, and A. C. Kay, "Evidence that Gendered Wording in Job Advertisements Exists and Sustains Gender Inequality," *Journal of Personality and Social Psychology*, vol. 101, no. 1, pp. 109–128, 2011.