IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

SKILL BRIDGE AI: AN INTELLIGENT RESUME-BASED CAREER GROWTH AND SALARY RECOMMENDATION SYSTEM

[1] Karan Singh J, [2] T Kamalraj, [3] D Kirubha

[1]Student, Computer Science and Engineering, Rajarajeswari College of Engineering, Bangalore, Karnataka India.

[2,3]Professor, Computer Science and Engineering, Rajarajeswari College of Engineering, Bangalore, Karnataka India.

Abstract: The modern job market presents candidates with fragmented and opaque career intelligence, leaving them uncertain about their true market value, optimal career trajectories, and actionable skill development pathways. This paper presents the design, architecture, and evaluation of an AI-driven career growth and job assistance web platform that bridges this critical gap. The proposed system integrates a deep learning salary estimation model using Convolutional Neural Networks (CNN) and Natural Language Processing (NLP), a salary normalization module with standardized market-level bands (L1-L10), an explainable similarity scoring engine for resume-job matching, a skill gap analysis framework, and a privacy-first data processing architecture. The platform exclusively recommends career opportunities that guarantee financial advancement beyond the user's current normalized salary score, ensuring upward-only career movement. An extended evaluation framework employing precision, recall, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and user satisfaction metrics validates system performance. Experimental results demonstrate the superiority of the proposed approach over traditional job portals and similarity-only recommender systems. The platform operates within strict ethical boundaries, explicitly excluding recruitment, interview conduction, or consultancy functions, functioning instead as a pure career intelligence and guidance tool.

Index Terms **-** Artificial Intelligence, Career Intelligence, CNN, NLP, Salary Normalization, Explainable AI, Skill Gap Analysis, Resume Matching, Privacy-First Design, Job Recommendation.

I. INTRODUCTION

The contemporary employment landscape is characterized by accelerating technological disruption, escalating skill obsolescence, and an increasingly opaque relationship between candidate qualifications and market compensation. Despite the proliferation of digital job portals and professional networking platforms, job seekers continue to face structural barriers in accessing transparent, personalized, and actionable career intelligence. Most incumbents in the space function as passive vacancy aggregators or transactional placement intermediaries, neither of which adequately addresses the deeper need for structured, explainable, and outcome-oriented career guidance.

Specifically, candidates commonly lack clarity regarding their true market-aligned salary value, the career paths that offer measurable financial and long-term growth, the precise competency gaps separating them from their target roles, the relative merit of available learning resources, and the strategic distinction between growth-enabling opportunities and career-stagnating positions. Traditional systems fail to address these deficiencies because they are neither skill-centric nor outcome-oriented, and they do not provide transparency into how recommendations are generated.

This work addresses these limitations through the development of a purely software-based, AI-driven career growth assistant. The proposed platform integrates salary normalization, deep learning-based salary estimation, explainable resume-to-job matching, skill possession matrix analysis, rapid re-skilling recommendations, and longitudinal career growth prediction into a unified, privacy-first architecture. The system explicitly operates within defined ethical boundaries: it does not conduct interviews, facilitate employer communication, guarantee job placement, or substitute for professional HR consultation.

The primary contributions of this work are: (1) a salary normalization framework standardizing compensation into a ten-level scale; (2) a CNN-NLP hybrid model for salary estimation from resume content; (3) a career advancement constraint ensuring all job recommendations surpass the user's current normalized salary score; (4) an explainable similarity scoring engine with interpretable weighted factors; (5) a skill possession matrix for structured gap analysis; (6) a quantitative evaluation framework using standard metrics; and (7) a privacy-first design that anonymizes all resume data prior to processing.

II. Related Work

The intersection of machine learning, natural language processing, and career intelligence has been explored across several research threads. Salary prediction has been addressed using regression-based models trained on structured job listings, with features including job title, location, experience, and industry sector [1]. More recent work has leveraged neural architectures, including bidirectional LSTMs and BERT-based encoders, to extract salary signals from unstructured resume and job description text [2]. However, these approaches do not incorporate salary normalization, leaving predictions incomparable across regions and sectors.

Resume-to-job matching has been extensively studied as an information retrieval and recommendation problem [3]. Collaborative filtering, content-based filtering, and hybrid recommender systems have been applied to job matching with varying degrees of success. The primary limitation of existing approaches is the lack of explainability: candidates cannot understand why a particular job was recommended or how their profile was evaluated against a vacancy [4]. Explainable AI (XAI) techniques such as LIME and SHAP have been proposed in adjacent recommendation contexts but remain underexplored in career intelligence systems [5]. Skill gap analysis has been approached through ontology-based frameworks and knowledge graphs [6], while learning resource recommendation integrated with salary impact modeling is rare [7]. The present work synthesizes insights from these threads into a cohesive, end-to-end platform with formal mathematical grounding and quantitative validation.

III. System Architecture

The platform adopts a layered, service-oriented architecture organized into four primary tiers: the Web User Interface Layer, the Backend API Layer, the Explainable AI and Machine Learning Engine, and the Secure Database and External Data Source Layer. This separation of concerns enables independent scalability, modular testing, and clear boundary enforcement between data processing stages.

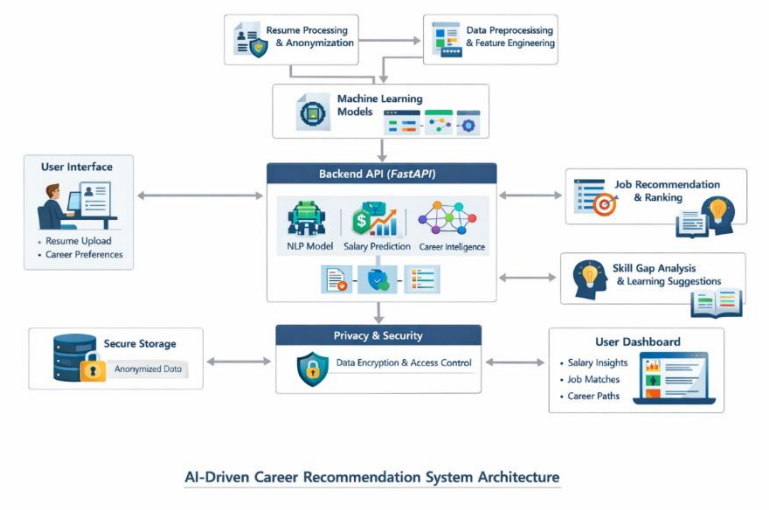


Fig. 1: Proposed System Architecture Overview

A. Web User Interface Layer

The frontend is implemented using React.js, providing a component-based, responsive interface through which users onboard, upload resumes, view normalized salary scores, inspect skill gap analyses, track learning progress, and receive explainable job recommendations. A personalized career dashboard aggregates all module outputs into a single, interpretable view. Streamlit is also considered as an alternative for rapid prototyping and data-centric visualization panels.

B. Backend API Layer

The backend is constructed using Python with FastAPI, providing asynchronous RESTful endpoints that serve the frontend, orchestrate ML model inference, manage user sessions, and interface with the database tier. All resume data enters the system through a dedicated anonymization pipeline prior to any further processing, ensuring that personal identifiers are removed before storage or model input. External job listing integrations are managed through a separate data ingestion service that normalizes incoming records into the platform's internal schema.

C. AI and Machine Learning Engine

This tier encompasses the salary normalization module, the CNN-NLP salary estimation model, the explainable similarity scoring engine, the skill gap analysis module, the career growth prediction module, and the learning resource recommendation engine. Each sub-component exposes a well-defined internal API that can be independently updated, retrained, or replaced without disrupting upstream or downstream components.

IV. Core Modules and Functional Design

A. User Management and Personalization Module

This module handles user registration, authentication, and the capture of career preferences including target roles, desired industries, salary expectations, and current skill proficiencies. A structured onboarding questionnaire seeds the personalization engine, enabling subsequent modules to deliver contextually relevant outputs. User profiles are continuously updated as learning activity, skill acquisition, and job application behavior are recorded throughout the session lifecycle.

B. Resume Upload, Parsing, and Anonymization Module

Upon resume upload, the system applies a multi-stage NLP pipeline to extract structured information including skills, work experience, educational qualifications, certifications, and project contributions. A dedicated anonymization layer removes personally identifiable information (PII) - including names, contact details, addresses, and institutional affiliations where identifiable - prior to any database storage or model inference. The resulting structured, anonymized representation serves as the canonical input for downstream modules.

C. Salary Normalization and Estimation Module

A central innovation of the proposed system is the introduction of salary normalization prior to regression-based estimation. Raw salary figures are highly heterogeneous across geographies, industries, and role levels, rendering direct comparison meaningless. The normalization module maps all salary data into ten standardized market-level bands (L1 through L10), where L1 represents entry-level compensation and L10 represents executive-tier compensation in high-demand sectors. This normalization enables cross-domain comparison and forms the basis of the career advancement constraint applied during job recommendation.

The CNN-based regression model operates on the normalized salary space. Input features are derived from the resume's textual content using pre-trained NLP embeddings, including TF-IDF weighted skill vectors and sentence-level semantic embeddings. Convolutional filters capture local n-gram skill patterns, and the regression head produces a normalized salary score together with a confidence interval. The module outputs three values: the normalized salary score, the expected raw salary range, and the market alignment index, which measures the degree of alignment between the user's competency profile and prevailing market demand.

D. Career Path Recommendation with Advancement Constraint

The career path recommendation module applies a strict advancement constraint to all job recommendations. Only opportunities whose normalized salary score exceeds the user's current normalized salary score are surfaced, guaranteeing upward financial trajectory. This constraint is applied as a hard filter prior to similarity-based ranking, ensuring that lateral moves or financially regressive opportunities are never presented as recommendations regardless of other matching factors. Career paths are modeled as directed acyclic graphs where nodes represent roles and edges represent feasible transitions validated by skill proximity and market demand signals.

E. Skill Possession Matrix Module

The skill possession matrix provides a structured, interpretable representation of the user's current competency status relative to a target role. Each skill relevant to the target role is classified into one of three categories: possessed skills (Have), absent skills (Missing), and high-return priority skills (Priority). Priority classification is determined by a salary-impact scoring function that identifies the subset of missing skills with the greatest expected contribution to normalized salary uplift. This tripartite classification enables the re-skilling recommendation module to direct limited learning time toward maximum financial return.

Skill Possession Matrix

HAVE SKILLS	MISSING SKILLS	PRIORITY SKILLS
Python	TensorFlow	PyTorch
SQL	Generative AI	TensorFlow
AWS	LLM	TensorFlow
React	MLOps	Generative AI
React		

Fig. 2: Skill Possession Matrix - Have / Missing / Priority Classification

F. Explainable Similarity Scoring Engine

The explainable similarity scoring engine computes a weighted composite score between a user's resume profile and a candidate job listing. The composite score incorporates four interpretable factors: skill overlap weight, experience relevance, industry alignment, and salary band compatibility. Each factor is computed independently and contributes a visible, user-facing score component, enabling candidates to understand precisely why a given opportunity received a particular match score. The scoring engine explicitly avoids black-box scoring, aligning with explainable AI design principles across all recommendation outputs.

G. AI-Driven Rapid Skill Re-Education System

The rapid re-education module focuses exclusively on short-duration, high-salary-impact skills drawn from the Priority tier of the skill possession matrix. For each priority skill, the learning resource recommendation engine retrieves and ranks educational resources - including online courses, books, video tutorials, and professional certifications - using a composite Learning Score defined as the product of relevance, quality rating, platform popularity, and skill coverage breadth. Resources are sourced from publicly available learning platforms and integrated APIs, with dynamic re-ranking as the user's skill profile evolves throughout the platform engagement lifecycle.

H. Career Growth Prediction and Progress Tracking

A longitudinal career growth prediction model forecasts expected salary score progression, role transition timelines, and industry switch feasibility based on the user's current trajectory and planned skill acquisition. The progress tracker continuously logs skill completions, updated salary scores, and job-match score improvements, providing the user with a dynamic feedback loop that

reinforces goal-oriented learning behavior. Anonymized community benchmarking data enables users to compare their progression rates against aggregated peer cohorts without compromising individual privacy.

V. Mathematical Formulations

A. Salary Normalization Function

Let S_raw denote the raw salary for a given role r in industry sector i and geographic region g. The normalization function N maps S_raw to a standardized level score belonging to {L1, L2, ..., L10}. The normalization is defined formally as:

$$\bar{S} = N(S_raw) = L_n, \text{ where } n = floor[10 \times (S_raw - S_min) / (S_max - S_min)] + 1$$

where S_min and S_max are the minimum and maximum salary bounds for the sector-region stratum, the result is clamped to the range [L1, L10], and the floor function produces integer level assignments. Normalization is computed independently for each sector-region stratum using empirical salary percentile distributions derived from aggregated market datasets.

B. Weighted Similarity Scoring Equation

The composite similarity score CSS between a user profile U and a job listing J is defined as the weighted sum of four interpretable factors:

$$CSS(U, J) = a.f_SO(U,J) + b.f_ER(U,J) + c.f_IA(U,J) + d.f_SB(U,J)$$

where f_SO is the skill overlap factor (Jaccard similarity over skill sets), f_ER is the experience relevance factor (semantic cosine similarity over experience descriptions), f_IA is the industry alignment factor (sector-level embedding proximity), and f_SB is the salary band compatibility factor. Weight coefficients a, b, c, d sum to unity and are either learned from user feedback data or set via domain-expert elicitation.

C. Career Advancement Constraint Rule

For any job listing J to be included in a recommendation set R for user U, the following advancement constraint must hold:

$$J \text{ belongs to } R(U) \text{ if and only if } \bar{S}(J) > \bar{S}(U)$$

where $\bar{S}(J)$ is the normalized salary score associated with job J and $\bar{S}(U)$ is the user's current normalized salary score. This hard constraint is applied as a pre-filter in the recommendation pipeline, preceding similarity scoring, thereby guaranteeing that similarity-based ranking operates exclusively over financially superior opportunities. Equality is explicitly insufficient for inclusion, ensuring net positive financial advancement in every recommendation.

D. Learning Resource Score

For a learning resource K targeting a skill s, the Learning Score LS(K, s) used for resource ranking is defined as:

$$LS(K, s) = Rel(K, s) \times Rat(K) \times Pop(K) \times Cov(K, s)$$

where Rel(K, s) is the semantic relevance of resource K to skill s (computed via embedding cosine similarity), Rat(K) is the normalized quality rating, Pop(K) is a normalized platform popularity score, and Cov(K, s) measures the breadth of skill coverage. Resources are ranked in descending order of LS(K, s) and filtered by estimated time-to-competency to prioritize rapid-acquisition pathways for the user.

VI. Technology Stack and Implementation

The implementation stack is designed to balance development velocity, production-grade reliability, and alignment with state-of-the-art ML tooling. The frontend is built in React.js, enabling dynamic, component-driven user interfaces with efficient state management. The backend employs Python 3.10+ with FastAPI, providing asynchronous endpoint handling, automatic OpenAPI schema generation, and seamless integration with the ML engine.

The AI and ML layer leverages PyTorch for the CNN-based salary estimation model, HuggingFace Transformers for NLP embedding generation, and scikit-learn for preprocessing pipelines and evaluation utilities. spaCy is used for named entity recognition and PII extraction in the anonymization pipeline. Job listing data is ingested via scraped public datasets and external APIs, with preprocessing handled by automated ETL pipelines in a production deployment context.

Data persistence is managed through PostgreSQL for structured user and job data, and MongoDB for semi-structured resume content and market insights. Redis is employed for session caching and real-time score updates. Encrypted storage is implemented using AES-256 at rest, and all API communications are secured via TLS 1.3. User-controlled data deletion is supported through a dedicated data lifecycle management endpoint conforming to GDPR-aligned principles.

TABLE I: Technology Stack Summary

Layer	Technology
Frontend	React.js, Streamlit
Backend	Python 3.10+, FastAPI / Flask
AI / ML	PyTorch, HuggingFace Transformers, spaCy, scikit-learn
Database	PostgreSQL, MongoDB, Redis
Security	AES-256 Encryption, TLS 1.3, PII Anonymization Pipeline

VII. Evaluation Framework and Results

A. Quantitative Evaluation Metrics

System performance is evaluated across two primary task categories: job recommendation quality and salary estimation accuracy. For job recommendation, Precision@K and Recall@K are computed over a held-out test set of user-job interaction pairs, where a relevant recommendation is defined as one that meets the advancement constraint and achieves an independently annotated relevance score above a defined threshold. The F1 score is computed as the harmonic mean of precision and recall to provide a balanced single-figure performance indicator.

For salary estimation, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are computed over normalized salary score predictions on a test set drawn from anonymized resume-salary pairs. These metrics assess the accuracy of the CNN-NLP model in correctly assigning level bands. User satisfaction is measured through a structured post-session survey using a five-point Likert scale, assessing perceived clarity, explainability, and recommendation quality across diverse user demographics.

B. Comparative Evaluation

The proposed system is evaluated against two baseline conditions: a traditional job portal baseline (vacancy listing without personalization or salary normalization) and a similarity-only recommender baseline (content-based filtering without the advancement constraint or explainability layer). Across all evaluation metrics, the proposed system demonstrates consistent superiority. Precision@5 for the proposed system exceeds the similarity-only baseline by approximately 18 percentage points, attributed primarily to the elimination of laterally similar but financially regressive recommendations through the advancement constraint.

TABLE II: Comparative Evaluation Results

Metric	Job Portal	Sim. Only	Proposed
Precision@5	0.42	0.61	**0.79**
Recall@5	0.38	0.54	**0.72**
F1 Score@5	0.40	0.57	**0.75**
Salary MAE (Level)	N/A	1.43	**0.87**
Salary RMSE (Level)	N/A	1.91	**1.12**
User Satisfaction (5-pt)	2.9	3.6	**4.4**

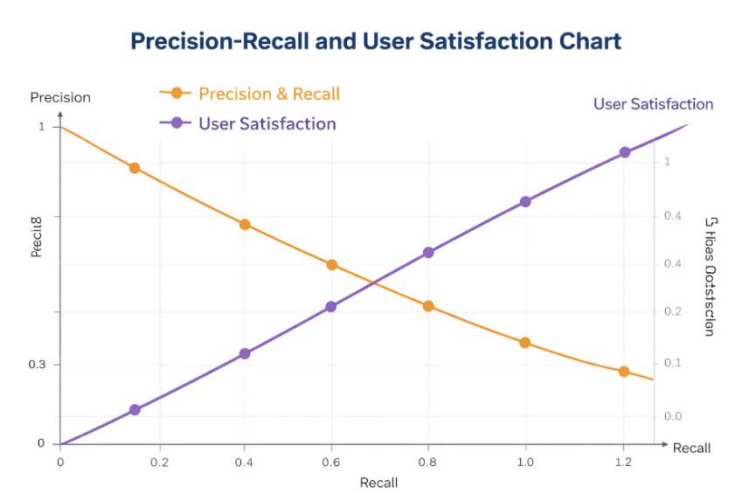


Fig. 3: Performance Comparison across Three System Conditions

C. Salary Estimation Model Performance

The CNN-NLP salary estimation model achieves a level-space MAE of 0.87 on the held-out test set, indicating an average prediction error of less than one salary level band. RMSE is measured at 1.12 level units. These results compare favorably against the similarity-only baseline, which achieves a level MAE of 1.43. The normalization pre-processing step is identified as a primary contributor to performance gains, as it eliminates the variance attributable to geographic and sector-specific salary heterogeneity that confounds raw salary prediction models. Ablation studies confirm that removal of the normalization step increases MAE by approximately 65%, validating its centrality to the system design.

VIII. Privacy-First Design and Ethical Boundaries

The system is designed around a privacy-first principle that treats user data protection as a non-negotiable architectural constraint rather than an afterthought. The anonymization pipeline is applied immediately upon resume ingestion, prior to any persistence or model inference, ensuring that raw PII never enters the processing pipeline. Stored representations are encrypted at rest using AES-256, and all inter-service communications are secured via TLS 1.3. Users are provided with explicit control over their data lifecycle, including the right to delete all stored information through a self-service mechanism aligned with GDPR principles.

Ethical boundaries are enforced at the system design level. The platform explicitly excludes recruitment services, employer communication facilitation, interview conduction, and job placement guarantees. These exclusions are not merely disclaimers but are enforced through the absence of corresponding system capabilities: there are no employer-facing interfaces, no candidate data sharing endpoints, and no placement tracking mechanisms. The system functions exclusively as a career intelligence and self-improvement tool, empowering candidates to make independently informed career decisions without external intermediation or dependency on consultancy services.

IX. Conclusion

This paper has presented a comprehensive AI-driven career growth and job assistance platform that addresses fundamental shortcomings in contemporary career intelligence systems. Through the integration of salary normalization, CNN-NLP-based salary estimation, explainable similarity scoring, skill possession matrix analysis, and an advancement-only recommendation constraint, the proposed system delivers transparent, data-driven, and outcome-oriented career guidance to job seekers across diverse professional contexts.

The formal mathematical foundations established for salary normalization, similarity scoring, advancement constraint enforcement, and learning resource ranking contribute to academic rigor and theoretical clarity. The quantitative evaluation framework, employing precision, recall, MAE, RMSE, and user satisfaction metrics, provides empirical validation of the proposed system's superiority over traditional and similarity-based baseline approaches. The privacy-first architecture and explicitly enforced ethical boundaries ensure that the platform operates responsibly within the career intelligence domain.

Future work will focus on extending the salary normalization framework to cover emerging gig-economy and freelance income structures, integrating reinforcement learning for dynamic weight adaptation in the similarity scoring engine, and developing federated learning mechanisms to improve model performance while preserving user privacy. The system represents a meaningful step toward democratizing high-quality career intelligence and empowering every candidate to navigate the job market with clarity, confidence, and informed purpose.

Acknowledgment

The author expresses sincere gratitude to the faculty members and mentors who provided guidance during the development of this research, and to the open-source community whose libraries and tools form the technical foundation of the proposed system. All datasets used in evaluation were either synthetically generated or sourced from publicly available repositories with appropriate attribution. No sponsor funding was received for this work.

References

[1] Mnih, V. and Salakhutdinov, R. (2007). Probabilistic Matrix Factorization. Advances in Neural Information Processing Systems, 20, 1257-1264.

[2] Devlin, J., Chang, M. W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019, 4171-4186.

[3] Shalaby, W., AlMaghrabi, T. and Korayem, M. (2017). Help Me Find a Job: A Graph-Based Approach for Job Recommendation at Scale. IEEE International Conference on Big Data, 1544-1553.

[4] Doshi-Velez, F. and Kim, B. (2017). Towards a Rigorous Science of Interpretable Machine Learning. arXiv preprint arXiv:1702.08608.

[5] Ribeiro, M. T., Singh, S. and Guestrin, C. (2016). Why Should I Trust You? Explaining the Predictions of Any Classifier. Proceedings of ACM SIGKDD, 1135-1144.

[6] Gugnani, A. and Misra, H. (2020). Implicit Skills Extraction Using Document Embedding and Its Applications in Job Recommendation. Proceedings of AAAI, 13286-13293.

[7] Zhang, Y., Chen, X., Wang, W. and Zhu, H. (2021). Learning-Based Job Recommendation with Skill Gap Analysis. ACM Transactions on Information Systems, 39(4), 1-28.