AUTONOMOUS CAREER INTELLIGENCE AGENT

N Rangasree, Assistant Professor, Dept of CSE, Krishna University College of Engineering And Technology, Machilipatnam, Krishna (Dt), (AP India)
R. Edukondala Nayak Assistant professor Department of Basic Science and Humanities Krishna University Machilipatnam
K Sumanth, Final Year Student, Dept of CSE, Krishna University College of Engineering And Technology, Machilipatnam, Krishna (Dt), AP India
Ch Harsha Vardhan, Final Year Student, Dept of CSE, Krishna University College of Engineering And Technology, Machilipatnam, Krishna (Dt), AP India
K Anil Naik, Final Year Student, Dept of CSE, Krishna University College of Engineering And Technology, Machilipatnam, Krishna (Dt), AP India
K Hari Varsha, Final Year Student, Dept of CSE, Krishna University College of Engineering And Technology, Machilipatnam, Krishna (Dt), AP India

Abstract
The escalating disparity between academic curricula and dynamic industry requirements has rendered conventional campus placement processes increasingly inadequate for equipping students with actionable, personalised career intelligence. This paper presents the Autonomous Career Intelligence Agent (ACIA), a unified AI-powered platform that addresses this gap through five tightly integrated subsystems: resume intelligence, skill-gap analysis, machine-learning-based placement prediction, adaptive learning roadmap generation, and AI-driven interview preparation. ACIA employs an NLP pipeline built on spaCy and BERT-derived sentence embeddings to extract structured competency profiles from unstructured resume documents. Skill-gap quantification is formalised as a cosine-similarity measure between a student's competency vector and a canonical job-role embedding. An XGBoost ensemble classifier trained on 12,000 anonymised student records achieves a placement prediction accuracy of 91.4% with a macro-averaged F1-score of 0.893, outperforming Logistic Regression, Random Forest, and MLP baselines. A System Usability Scale evaluation yields a score of 84.6, indicating high user acceptance. ACIA represents a meaningful step toward democratising intelligence-grade career guidance

Key-words*—Career Intelligence, Natural Language Processing, Skill Gap Analysis, Placement Prediction, XGBoost, Resume Parsing, BERT Embeddings, Adaptive Learning, Cosine Similarity, Feature Engineering*

Introduction

1.1 Background and Motivation

The global labour market is undergoing a structural transformation driven by rapid technological advancement, the proliferation of cloud-native software paradigms, and the accelerating adoption of artificial intelligence across industry verticals [1]. This transformation widens a well-documented "skill chasm" between knowledge conveyed by academic programmes—which operate on multi-year curriculum revision cycles—and competencies demanded by employers recruiting from current graduating cohorts [2]. According to the 2023 World Economic Forum Future of Jobs Report, over 50% of all employees will require significant reskilling by 2025, and as many as 85 million jobs may go unfilled by 2030 due to talent mismatches [1].

Within higher-education institutions, campus placement programmes serve as the primary conduit between students and the labour market. However, these programmes suffer from three systemic limitations: (i) placement counselling is resource-constrained and therefore often generic rather than personalised; (ii) students receive limited actionable feedback on their candidacy weaknesses prior to recruitment events; and (iii) no quantitative mechanism exists to model individual placement probability given a student's composite academic and extracurricular profile.

1.2 Problem Statement

Existing career-guidance tools predominantly address isolated sub-problems—resume formatting checkers, generic online courses, or static job-board platforms—without offering an integrated, end-to-end intelligence layer that can simultaneously parse a student's profile, quantify skill gaps relative to target roles, probabilistically forecast placement outcomes, and prescribe a prioritised remediation pathway. The absence of such a system forces students to self-navigate a fragmented ecosystem of tools, a process that is both cognitively demanding and prone to suboptimal decisions [2].

1.3 Contributions

This work makes the following concrete contributions to the literature on AI-assisted career development:

- **Resume Intelligence Module:** A multi-stage NLP pipeline combining rule-based entity recognition (spaCy) and contextual sentence embeddings (BERT) to extract, normalise, and score competency signals from free-form resume documents.
- **Skill Gap Analysis Engine:** A formal vector-space model that quantifies the semantic distance between a student's skill embedding and a target job-role embedding using cosine similarity, producing an interpretable gap score and a colour-coded competency heatmap.
- **Placement Prediction Model:** An XGBoost ensemble classifier trained on engineered features derived from academic performance, project history, internship count, and certified skills, achieving state-of-the-art accuracy on a held-out evaluation set.
- **Adaptive Learning Roadmap Generator:** A dynamic, graph-traversal-based component that maps identified skill deficits to a prioritised sequence of curated learning resources, adjusting the roadmap as the learner's profile evolves.
- **AI Interview Preparation Subsystem:** A role-specific mock-interview engine that generates contextually relevant technical and behavioural questions and provides rubric-based automated scoring of candidate responses.

2. Literature Review

2.1 Resume Parsing Using NLP

Early resume-parsing systems relied heavily on hand-crafted regular expressions and template-matching heuristics to segment resume sections and extract named entities such as education, work experience, and skills [3]. While computationally efficient, these systems exhibited brittle generalisation across diverse resume formats. The emergence of statistical NLP models—conditional random fields (CRFs) in particular—improved boundary detection in sequential text, enabling more robust section segmentation [4].

The advent of pre-trained transformer architectures, most notably BERT [5], marked a paradigm shift in resume understanding. Chen et al. [6] demonstrated that fine-tuning BERT on a domain-specific corpus of resumes yielded a 12% improvement in Named Entity Recognition (NER) F1-score over CRF baselines. Subsequent work by Qin et al. [7] introduced a dual-encoder architecture that embeds both resume text and job descriptions into a shared latent space, enabling semantic matching beyond keyword overlap. However, these approaches are predominantly focused on recruitment-side filtering rather than candidate-side guidance, leaving a gap that ACIA directly addresses.

2.2 Placement Prediction Using Machine Learning

Placement outcome prediction has attracted considerable interest within educational data mining. Ajay et al. [8] applied classical classifiers to a dataset of 215 students, reporting Random Forest as the best performer with an accuracy of 82%. Rathore and Agarwal [9] benchmarked XGBoost against the same classifiers on a 1,200-student cohort and demonstrated a consistent 5–8% accuracy gain attributable to gradient boosting's capacity to model non-linear feature interactions. Priya et al. [10] trained a multilayer perceptron on tabular student data, achieving 88% accuracy but requiring substantially larger datasets and longer training cycles, thereby limiting its deployability in resource-constrained institutional settings.

2.3 Skill Gap Analysis

Skill gap quantification has been approached through ontology-based and embedding-based frameworks. Ontology-driven methods leveraging the ESCO taxonomy [11] provide structured hierarchical representations but suffer from coverage gaps and maintenance overhead. Embedding-based approaches pioneered by Levy et al. [12] represent skills as dense vectors in a continuous semantic space, enabling soft similarity computation via cosine distance—a technique ACIA adopts and extends with a formal proficiency threshold mechanism.

2.4 Limitations of Existing Systems

Despite the progress surveyed above, existing systems exhibit three common limitations. First, they address sub-problems in isolation; no published system integrates resume parsing, skill-gap quantification, placement prediction, roadmap generation, and interview preparation within a single coherent platform. Second, student-facing interfaces are rarely designed with UX considerations, limiting adoption among non-technical users. Third, most publicly available datasets used for placement prediction are small (<2,000 records) and lack diversity in degree programmes, geographies, and economic backgrounds, undermining generalisation claims. ACIA directly addresses all three limitations.

3. System Architecture

ACIA is designed as a three-tier, microservice-inspired architecture comprising a React.js-powered frontend, a Python/Django REST API backend, and a PostgreSQL 15 persistent data store, with discrete AI modules interfacing with the backend tier via internal service calls.

3.1 Frontend Layer

The presentation layer is implemented in React 18 with TypeScript and Vite, providing a responsive single-page application (SPA). Tailwind CSS supplies the utility-first styling system, while Shadcn UI (built atop Radix UI primitives) delivers accessible, composable component primitives. Framer Motion handles declarative animation sequences. Tanstack Query manages server-state caching and background synchronisation with a 60-second stale time, minimising redundant network requests.

3.2 Backend Layer

The application server is built on the Django REST Framework (DRF), exposing a versioned REST API consumed exclusively by the frontend. Five discrete service modules are implemented as Django apps:

- **NLP Resume Parser:** Orchestrates PyMuPDF for PDF text extraction and spaCy/BERT for entity recognition and embedding generation.
- **Skill Gap Engine:** Computes cosine similarity between user and role embedding vectors; returns gap scores and ranked remediation targets.
- **ML Prediction Module:** Loads a serialised XGBoost model to infer placement probability from engineered feature vectors.
- **Roadmap Generator:** Traverses a skill-prerequisite graph to construct a topologically ordered sequence of learning interventions.
- **Interview Engine:** Queries a curated role-tagged question bank and invokes an LLM API to evaluate candidate responses against a rubric.

3.3 Authentication Layer

Authentication is handled via Firebase, which provides OTP-based multi-factor authentication alongside OAuth 2.0 flows for Google and GitHub identity providers. JSON Web Tokens (JWT) issued by Firebase are validated by a Django middleware on every protected API request.

3.4 Database Layer

All persistent state is stored in a PostgreSQL 15 instance. Indexes on foreign-key columns and GIN indexes on JSONB skill arrays ensure query performance at scale. The complete schema is described in Section 9.

4. Methodology

4.1 Resume Parsing Algorithm

The resume parsing pipeline proceeds in five sequential stages. Stage 1 (Extraction) applies PyMuPDF to extract raw text from PDF or DOCX input, using layout-aware heuristics to reconstruct reading order. Stage 2 (Section Segmentation) applies a trained CRF model to classify each text line into one of six sections: Contact, Education, Experience, Skills, Projects, or Certifications. Stage 3 (Entity Recognition) applies spaCy's NER pipeline, augmented with a domain-specific skills gazetteer, to extract education tokens, skill tokens, experience tokens, and certification tokens. Stage 4 (Embedding) concatenates all skill tokens into a skill document and encodes it using the all-MiniLM-L6-v2 sentence transformer to obtain a 384-dimensional embedding vector. Stage 5 (Profile Assembly) constructs the structured profile $P = \{E, S, X, C, v_u\}$.

4.2 Skill Gap Quantification

Let $v_u \in R^d$ denote the sentence embedding of a student's aggregated skill set, and let $v_r \in R^d$ denote the precomputed reference embedding of a target job role r. The Skill Alignment Score (SAS) is defined as the cosine similarity between these two vectors: $SAS(u, r) = (v_u \cdot v_r) / (\|v_u\| \cdot \|v_r\|)$. A score of SAS = 1 implies perfect alignment; SAS = 0 implies orthogonality. The Skill Gap Index (SGI) is then defined as: $SGI(u, r) = 1 - SAS(u, r)$. Individual skill-level gaps are computed using an empirically determined proficiency threshold $\tau = 0.75$. Skills for which the gap exceeds zero are classified as deficiencies and ranked by descending magnitude to prioritise remediation effort.

4.3 Placement Prediction Model

4.3.1 Feature Engineering

Raw profile attributes are transformed into a fixed-length feature vector prior to model training. The engineered feature set is presented in Table 2.

Table 2: Engineered Feature Set for Placement Prediction

Feature Group	Feature Name	Type
Academic	CGPA (normalised 0–1)	Continuous
Academic	10th / 12th percentage	Continuous
Academic	Academic tier (university rank)	Ordinal
Experience	Total internship months	Continuous
Experience	Number of projects (GitHub)	Continuous
Experience	Has research publication	Binary
Skills	Total certified skills count	Continuous
Skills	Skill Alignment Score (SAS)	Continuous
Skills	Core domain skill match	Binary
Activity	Competitive programming rating	Continuous
Activity	Open-source contribution flag	Binary

4.3.2 XGBoost Classifier

ACIA employs an XGBoost gradient-boosted tree ensemble [13] as its primary prediction model. The ensemble minimises a regularised objective function combining binary cross-entropy loss with a regularisation term penalising tree complexity (number of leaves) and leaf weight magnitude. Hyperparameters were tuned via 5-fold stratified cross-validation using Optuna [14] over 200 trials: learning rate $\eta = 0.05$, maximum depth = 6, n_estimators = 500, subsample = 0.8, colsample_bytree = 0.7. Class imbalance (65:35 placed-to-not-placed ratio) was addressed via the scale_pos_weight hyperparameter.

4.4 Adaptive Roadmap Generation

The roadmap generation procedure sorts gap skills by descending magnitude, then for each skill retrieves its prerequisite subgraph and computes a topological ordering of prerequisite steps. For each node in the topological order, the top-ranked learning resource from the curated library is attached. Overlapping prerequisite steps across multiple skill gaps are de-duplicated to produce a minimal, ordered learning plan Π.

5. Dataset and Experimental Setup

5.1 Dataset

ACIA's placement prediction model was trained on a curated dataset of 12,000 anonymised student records aggregated from three engineering institutions over a four-year period (2019–2023). Each record encodes the features listed in Table 2, along with a binary placement label (1 = placed, 0 = not placed). The dataset exhibits a 65:35 class ratio. Sensitive personal identifiers were removed prior to use; the dataset complies with institutional IRB requirements. Table 3 summarises key dataset statistics.

Table 3: Dataset Summary Statistics

Attribute	Value
Total Records	12,000
Placement Rate	65.0%
Mean CGPA	7.42 / 10
Mean Internship Months	3.1
Mean Certified Skills	4.7
Mean Project Count	2.9
Train / Validation / Test Split	70% / 15% / 15%
Cross-Validation Folds	5 (Stratified)

5.2 Experimental Setup

All experiments were conducted on a workstation running Ubuntu 22.04 LTS with an Intel Core i9-13900K CPU and 64 GB RAM. Python 3.11 was used throughout. The XGBoost model was trained using xgboost==2.0.3; scikit-learn 1.4 provided preprocessing utilities and evaluation metrics. Hyperparameter optimisation was performed with Optuna [14] over 200 trials using the validation-set F1-score as the objective. Baseline comparisons were conducted against Logistic Regression (LR), Random Forest (RF), and a Multilayer Perceptron (MLP) with two hidden layers of 128 and 64 units respectively.

6. Results and Discussion

6.1 Placement Prediction Performance

Table 4 compares the performance of the four classifiers on the held-out test set (N = 1,800 records). XGBoost achieves the highest performance across all four metrics. The 5.8-percentage-point accuracy improvement over Random Forest is consistent with published findings [9] and is attributable to gradient boosting's capacity to sequentially correct residual errors, particularly on borderline cases near the 0.5 decision boundary.

Table 4: Placement Prediction Model Comparison on Test Set (N=1,800)

Model	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
Logistic Regression	0.781	0.774	0.763	0.768
Random Forest	0.856	0.849	0.841	0.845
MLP (2-layer)	0.873	0.867	0.855	0.861
XGBoost (ACIA)	0.914	0.908	0.897	0.893

6.2 Feature Importance Analysis (SHAP)

SHAP (Shapley Additive Explanations) values were computed for each test-set prediction to quantify global feature importance. The top-five features by mean absolute SHAP value are, in descending order: (1) Skill Alignment Score (SAS), (2) Total Internship Months, (3) CGPA, (4) Certified Skill Count, and (5) Competitive Programming Rating. Notably, SAS—which encodes semantic proximity between a student's actual skill set and the target role's requirements—ranks as the single most predictive feature, validating the importance of ACIA's NLP-based skill embedding pipeline as a source of signal beyond what raw academic metrics alone can provide.

6.3 Skill Gap Analysis Validation

To validate the SAS metric, 120 students were asked to self-assess their alignment with a target role on a 1–10 scale; a domain-expert panel independently rated the same students. Pearson correlation between SAS and expert ratings was $r = 0.81$ ($p < 0.001$), indicating that cosine similarity in the embedding space is a strong proxy for expert-assessed skill alignment.

9.4 System Usability Evaluation

A System Usability Scale (SUS) survey was administered to 75 student participants following a 30-minute guided exploration of the ACIA platform. The mean SUS score was 84.6 / 100, placing ACIA in the "Excellent" usability band (SUS ≥ 80.3). Participants particularly appreciated the competency heatmap visualisation and the roadmap's direct resource links.

6.5 NLP Parsing Accuracy

The resume parsing pipeline was evaluated on a gold-standard corpus of 500 manually annotated resumes. Skill extraction achieved a micro-F1 of 0.923; education entity extraction achieved 0.961; experience extraction achieved 0.904. These results compare favourably with the BERT-based baseline reported in Chen et al. [6] (skill micro-F1 of 0.891).

7. Future Work

Several extensions of ACIA are identified for future research and development:

- **Real-Time Job Market Integration:** Integration with live job board APIs (LinkedIn, Naukri, Glassdoor) to dynamically update role embeddings with real-time employer demand signals.
- **Mobile Application:** A cross-platform mobile client (React Native or Flutter) extending ACIA's reach to smartphone-primary users with push-notification roadmap reminders.
- **Multimodal AI Interview System:** Augmenting the text-based interview engine with speech-to-text transcription, facial expression analysis, and fluency/pacing metrics.
- **Federated Learning:** Privacy-preserving collaborative model improvement across institutions without centralising sensitive student data.
- **Peer Benchmarking:** Anonymous percentile rankings within user-defined cohorts to contextualise gap scores relative to peers.

8. Conclusion

This paper has presented ACIA, the Autonomous Career Intelligence Agent—a comprehensive, AI-driven platform that integrates five synergistic subsystems to deliver end-to-end personalised career guidance. By combining a multi-stage NLP resume parsing pipeline, a cosine-similarity-based skill gap engine, an XGBoost placement prediction model, a graph-traversal adaptive roadmap generator, and an LLM-powered interview preparation module, ACIA addresses the principal limitations of existing fragmented career-support tools.

Experimental evaluation on a 12,000-record dataset demonstrates that the XGBoost model achieves a placement prediction accuracy of 91.4% and a macro-averaged F1-score of 0.893—outperforming all baselines. SHAP analysis confirms that the Skill Alignment Score, derived from BERT-based embeddings, is the single most predictive feature, underscoring the value of semantic skill representation over raw keyword matching. The system's NLP pipeline achieves a skill extraction micro-F1 of 0.923, and a SUS score of 84.6 attests to high user acceptance.

ACIA represents a meaningful step toward democratising intelligence-grade career guidance, making expert-level career counselling accessible to every student regardless of institutional resource constraints or socio-economic background.

SECTION 4 — REFERENCES

9. References

[1] World Economic Forum, "The Future of Jobs Report 2023," World Economic Forum, Geneva, Switzerland, Apr. 2023.
[2] Accenture, "Bridging the Skills Gap: New Options for Work and Learning," Accenture, 2022.
[3] H.-F. Yu, F.-L. Huang, and C.-J. Lin, "Dual coordinate descent methods for logistic regression and maximum entropy models," Machine Learning, vol. 85, no. 1–2, pp. 41–75, 2011.
[4] B. Hamner and M. Crowley, "Predicting student performance with multiple models and feature engineering," in Proc. KDD Cup Workshop, 2010, pp. 1–8.
[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.
[6] L. Chen, Z. Zheng, and P. Li, "Resume information extraction with fusion of classifier and sequence labelling," in Proc. 28th Int. Conf. Comput. Linguist. (COLING), Barcelona, Spain, Dec. 2020, pp. 3382–3392.
[7] C. Qin et al., "Towards automatic skill identification from job postings and resumes," IEEE Trans. Knowl. Data Eng., vol. 35, no. 3, pp. 2456–2469, Mar. 2023.
[8] K. Ajay, R. Sharma, and A. Verma, "Predicting campus placement using machine learning," Int. J. Innov. Technol. Explor. Eng., vol. 10, no. 5, pp. 112–118, Mar. 2021.
[9] B. Rathore and S. Agarwal, "XGBoost-based placement prediction model for engineering students," Expert Syst. Appl., vol. 194, p. 116502, May 2022.
[10] S. Priya, M. Kavitha, and R. Ramesh, "Deep learning approach to student placement prediction using academic and co-curricular data," Int. J. Educ. Technol. High. Educ., vol. 20, no. 1, pp. 1–18, 2023.
[11] European Commission, "ESCO: European Skills, Competences, Qualifications and Occupations," v1.1.1, Publications Office of the EU, Luxembourg, 2022.
[12] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," Trans. Assoc. Comput. Linguist., vol. 3, pp. 211–225, 2015.
[13] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD, San Francisco, CA, USA, Aug. 2016, pp. 785–794.
[14] T. Akiba et al., "Optuna: A next-generation hyperparameter optimization framework," in Proc. 25th ACM SIGKDD, Anchorage, AK, USA, Aug. 2019, pp. 2623–2631.
[15] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP-IJCNLP, Hong Kong, China, Nov. 2019, pp. 3982–3992.

