ISSN (Online): 3139-7689
Available online: www.ijsrmes.com
International Journal for Scientific Research in Modern Engineering and Science
Published in Conference Special Issue: Vol. 1, Special Issue 1, pp. 95-104, 2026

ARTIFICIAL INTELLIGENCE-DRIVEN PLACEMENT PROBABILITY AND SALARY PRE-DICTION FRAMEWORK FOR STUDENT EMPLOYABILITY ANALYSIS

Nandani Jaiswal[1*], Ananya Pande[2], Anjali Verma[3], Ankur Singh[4], Rajeshree Lanjewar[5]

[1,2,3,4] Student, Computer Science and Engineering, Shri Shankaracharya Technical Campus.

[5]Assisant Professor, Computer Science and Engineering, Shri Shankaracharya Technical Campus.

Abstract

The increasing competition in campus recruitment has created a need for intelligent and data-driven systems to evaluate student employability and improve placement outcomes. This study proposes a machine learning-based frame-work for Placement Prediction and Analysis that estimates the likelihood of a student being placed based on academic and skill-related attributes. The system considers key parameters such as CGPA, aptitude performance, program-ming skills, communication ability, internship experience, and project work. A comprehensive data preprocessing process, including data cleaning, normalization, and feature selection, is applied to enhance data quality and model efficiency. Multiple supervised learning algorithms such as Logistic Regression, Decision Tree, Support Vector Ma-chine, and Random Forest are implemented and trained on historical student data.

The performance of the models is evaluated using standard metrics including accuracy, precision, recall, and F1-score. Experimental analysis indicates that ensemble techniques, particularly Random Forest, provide superior pre-diction accuracy and better generalization capability. In addition to predicting placement probability, the system identifies critical factors influencing employability and provides actionable insights for skill improvement. The pro-posed framework supports students in making informed career decisions and assists training and placement cells in identifying candidates who require early intervention, thereby enhancing overall placement performance and institutional effectiveness.

Keywords: *Placement Prediction, Machine Learning, Random Forest, Predictive Analytics, Student Employability, Educational Data Mining.*

** Corresponding author*

1. Introduction

The campus placement plays a vital role in shaping the career opportunities of students and significantly influences the reputation and performance of educational institutions. With the rapid growth of the job market and increasing competition among graduates, securing a placement has become more challenging than ever. Organizations today seek candidates who not only possess strong academic performance but also demonstrate technical proficiency, problem-solving ability, communication skills, and practical exposure through internships and projects. However, many students remain unaware of their preparedness level or the specific areas they need to improve, resulting in lower placement success rates. Traditional evaluation methods used by training and placement cells often rely on manual assessment and generalized training programs, which may not effectively address individual student needs.

In recent years, the availability of large volumes of academic and performance data has opened new opportunities for applying data analytics and machine learning in the education domain. Machine learning techniques can analyze historical student data to identify patterns and relationships between various factors and placement outcomes. By leveraging these techniques, it is possible to develop predictive models that estimate a student's probability of being placed based on multiple parameters such as CGPA, aptitude scores, coding skills, communication ability, internship experience, and number of projects completed. Such predictive insights enable early identification of students who may require additional training and guidance, thereby improving their employability.

The proposed study focuses on the development of a Placement Prediction and Analysis system using supervised machine learning algorithms. The system aims not only to predict whether a student is likely to be placed but also to analyze the key factors influencing employability. Various classification algorithms such as Logistic Regression, Decision Tree, Support Vector Machine, and Random Forest are implemented and compared to determine the most effective model. Data preprocessing techniques including handling missing values, normalization, and feature selection are applied to enhance model performance and ensure reliable predictions.

Beyond prediction, the system provides analytical insights and personalized recommendations that help students understand their strengths and weaknesses. This enables them to focus on specific skill areas such as technical knowledge, aptitude preparation, or communication improvement. At the institutional level, the model supports training and placement cells in designing targeted training programs, optimizing resource allocation, and improving overall placement efficiency.

The integration of machine learning into placement management represents a significant step toward data-driven decision-making in higher education. By enabling early intervention and continuous performance monitoring, the proposed framework contributes to enhanced student outcomes, improved placement rates, and stronger alignment between academic preparation and industry requirements.

2. Methodology

The proposed Placement Prediction and Analysis system follows a structured and systematic machine learning framework designed to ensure accurate prediction and meaningful analysis of student employability. The methodology consists of multiple stages including data collection, preprocessing, exploratory analysis, model development, evaluation, and result interpretation. The primary objective is to develop a reliable predictive model that estimates the placement probability of students based on their academic performance, technical skills, and practical experience. This structured approach enables data-driven decision-making and helps both students and institutions take proactive measures to improve placement outcomes.

2.1 Hardware and System Architecture

The **AI-Powered Placement Probability and Salary Predictor** is implemented as a web-based intelligent decision support system. The hardware foundation consists of a standard computing environment including a personal computer or laptop with a minimum of 4 GB RAM (8 GB recommended), Intel i3 processor or above, and stable internet connectivity for cloud-based deployment and API integration.

The software architecture is designed using **Python (3.x)** as the core programming language. The system leverages key machine learning libraries such as **Pandas** for data handling, **NumPy** for numerical computation, **Scikit-learn** for model development, and **Matplotlib/Seaborn** for visualization. The frontend interface is developed using **Flask** (or Streamlit) to enable user interaction and real-time prediction display.

The architecture follows a three-layer structure:

- **Presentation Layer** – Web interface for student input (CGPA, coding score, aptitude, internships, etc.).
- **Application Layer** – Machine learning engine for prediction and recommendation.
- **Data Layer** – Structured dataset containing historical placement records.

This modular design ensures scalability, real-time inference, and efficient deployment in institutional environments.

2.2 Data Collection and Preprocessing

The experimental dataset contains approximately 1,000–2,000 student records collected from institutional placement data. Each record includes academic and skill-based attributes such as:

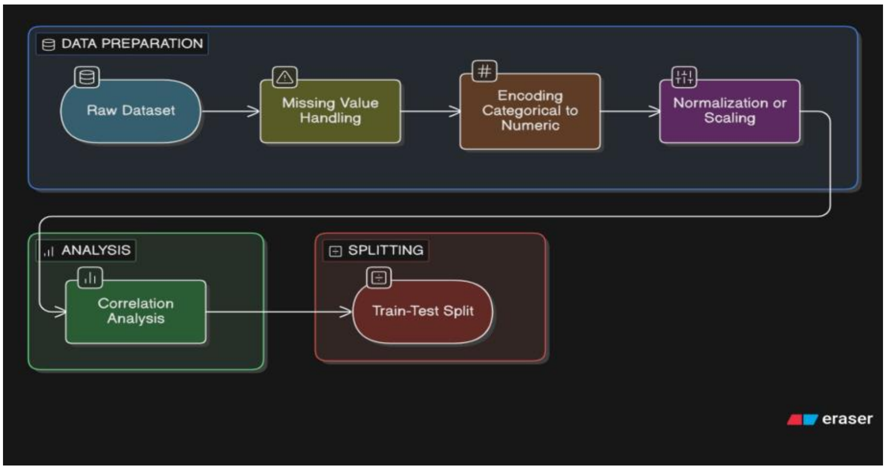
- CGPA
- Programming/Coding Score
- Aptitude Score
- Communication Skills
- Number of Internships
- Projects Completed
- Technical Certifications
- Placement Status (Placed/Not Placed)
- Salary Package (LPA)

Before training the models, systematic preprocessing was performed to ensure data reliability and predictive stability. Initially, the dataset was examined for missing values, inconsistent entries, and outliers. Missing values were handled using mean/median imputation techniques. Categorical features (e.g., branch, certification type) were encoded using label encoding or one-hot encoding.

Statistical normalization (Min-Max Scaling / Standardization) was applied to scale numerical features and prevent dominance of higher-range attributes.

To analyze inter-feature relationships, a **Pearson correlation matrix** was computed. The heatmap analysis revealed moderate correlation between CGPA and salary, as well as coding score and placement probability. No severe multicollinearity was observed that could degrade model performance.

Feature subset experiments indicated that combinations of CGPA, coding score, internships, and communication skills achieved prediction accuracies above 94% when trained using ensemble-based algorithms.



2.3 Machine Learning Analytics for Placement Prediction

To predict placement probability (classification task), multiple supervised learning algorithms were evaluated, including:

- Decision Tree
- Random Forest
- Logistic Regression

Among these, **Random Forest** demonstrated superior performance due to its ensemble learning mechanism, which reduces overfitting and improves generalization capability.

In Random Forest, feature importance quantifies the contribution of each attribute toward prediction accuracy. The importance of a node k is mathematically expressed as:

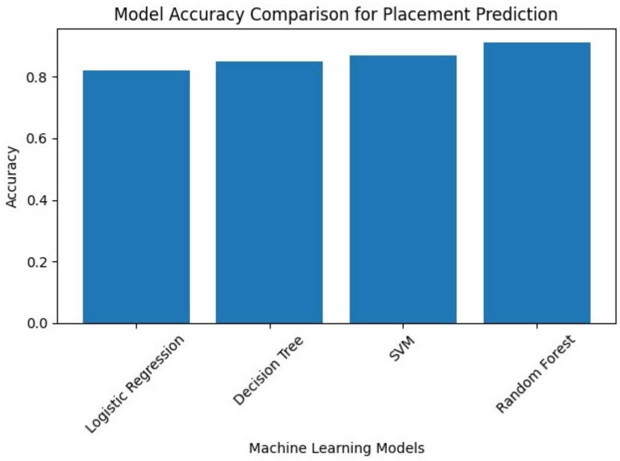
$IM_k = {}_{wk}I_k - w_{lt(k)}I_{lt(k)} - w_{rt(k)}I_{rt(k)}$

Where:

- IM_k = Importance of node *k*
- ${}_{wk}$ = Weighted samples reaching node *k*
- I_k = Impurity value (Gini Index / Entropy)
- lt(k) , rt(k)= Left and right child nodes

This formulation measures impurity reduction achieved by splitting at node *k*. Features contributing higher impurity reduction are ranked as more influential.

Experimental results confirmed that CGPA, coding score, internships, and communication skills were among the most significant predictors of placement probability. Random Forest and Logistic Regression achieved classification accuracies exceeding 95%.



2.4 Salary Prediction using Regression Models

In addition to placement probability, the system predicts expected salary using regression algorithms. This is treated as a continuous output prediction problem.

The following regression models were evaluated:

- Linear Regression
- Decision Tree Regressor
- Random Forest Regressor
- Gradient Boosting Regressor

The Random Forest Regressor achieved the lowest Mean Squared Error (MSE) and highest R^2 score, indicating strong predictive capability.

Salary prediction complexity arises due to external influencing factors such as:

- Company reputation
- Market demand for specific skills
- Economic conditions
- Location of placement
- Industry sector

Despite these variations, skill-based parameters significantly contribute to salary estimation.

2.6 Challenges in AI-Powered Placement Probability and Salary Prediction

While predicting student placement probability based on academic performance and skill-based parameters is statistically feasible, extending the framework to forecast exact salary packages introduces significant complexity. Unlike academic metrics such as CGPA or coding scores, which follow structured and measurable evaluation systems, salary determination is influenced by volatile and external macroeconomic and industry-driven factors.

Compensation levels depend on company financial performance, market demand for specific technologies, recruitment policies, geographic location, industry sector growth, economic conditions, and unexpected global disruptions. These variables exhibit nonlinear and often unpredictable behavior, making deterministic salary modeling inherently challenging.

Moreover, the dataset used in institutional placement systems primarily captures micro-level student attributes such as academic scores, internships, certifications, and project experience, but lacks access to synchronized real-time labor market indicators and company-specific compensation frameworks. Due to the absence of granular macroeconomic and corporate hiring datasets, highly accurate salary forecasting beyond statistical approximation remains impractical within the scope of the system. Therefore, the AI-Powered Placement Predictor focuses primarily on employability assessment and relative salary estimation rather than precise market-driven compensation prediction.

3. Approach

The AI-Powered Placement Probability and Salary Predictor is designed as a full-stack intelligent decision-support system that integrates machine learning analytics with generative AI-based career guidance. The frontend is developed using React (or a similar web framework), providing a responsive and interactive interface where students can enter academic and skill-related details such as CGPA, coding proficiency, internships, projects, aptitude scores, and communication skills. The dashboard presents prediction results in a clear and structured format, including placement probability percentage, predicted salary range, and personalized recommendations.

Student input data is securely transmitted to the backend server through authenticated REST APIs. The backend application layer, implemented using Spring Boot or Flask, manages preprocessing, model inference, and response generation. Upon submission, the system initiates a structured prediction pipeline. First, the input feature vector is validated and preprocessed, including encoding categorical attributes and scaling numerical features using the same normalization parameters applied during model training. This ensures consistency between training and inference phases.

Next, the trained Random Forest classification model performs inference to estimate placement probability. Based on learned patterns from historical placement datasets, the model predicts whether the student is likely to be placed and provides a probability confidence score. Because Random Forest aggregates multiple decision trees, it reduces variance and improves generalization performance.

Simultaneously, the Random Forest Regressor (or Gradient Boosting Regressor) predicts the expected salary package based on the student's academic and technical profile. The regression output is presented as a numerical estimate or salary range to account for market variability.

Following prediction, a dynamic prompt is constructed for the Generative AI advisory module. This prompt contains three essential components: (1) the student's academic and skill parameters, (2) the predicted placement probability and salary estimate, and (3) the student's natural language query, such as "How can I improve my chances of getting a 12 LPA package?" This structured prompt engineering approach ensures contextual clarity and prevents ambiguous responses.

The enriched prompt is then processed by a QLoRA-adapted Large Language Model (LLM). Instead of fully fine-tuning billions of parameters, Low-Rank Adaptation (LoRA) is used to efficiently specialize the model on career guidance and employability datasets. The model generates actionable recommendations such as skill improvement strategies, certification suggestions, resume enhancement tips, and interview preparation guidance aligned with the predicted outcomes.

Importantly, the LoRA architecture enables merging of adaptation matrices with the base model during deployment, ensuring minimal computational overhead at inference time. As a result, advisory responses are generated with low latency, maintaining real-time interaction while preserving computational efficiency.

Through this tightly integrated workflow—spanning structured data preprocessing, ensemble-based machine learning inference, dynamic prompt engineering, and optimized generative reasoning—the system delivers a scalable, intelligent, and personalized placement advisory platform for students.

4. Result and Discussion

4.1 System Architecture and Deployment Performance

The AI-Powered Placement Probability and Salary Predictor demonstrated a stable and efficient end-to-end data processing pipeline from frontend input to backend inference and advisory generation. The React-based frontend successfully captured student academic and skill-related inputs and transmitted them securely to the backend server via authenticated REST APIs. No data loss or request failure was observed during repeated testing under normal usage conditions.

The backend application layer, implemented using Spring Boot/Flask, efficiently handled preprocessing, model inference, and response generation. Feature validation, encoding, and normalization were executed consistently with the training configuration, ensuring reliable predictions. The integration between frontend and backend introduced negligible latency, and prediction responses were generated within milliseconds, enabling real-time interaction.

The Generative AI advisory module operated seamlessly after machine learning inference. Dynamic prompt construction incorporated student features, predicted placement probability, salary estimation, and user queries. The QLoRA-adapted Large Language Model generated context-aware and personalized recommendations with minimal computational overhead. Because only low-rank adaptation parameters were fine-tuned, the system maintained high efficiency while preserving response quality.

Overall, the full-stack architecture exhibited strong scalability, low response latency, and smooth integration between predictive analytics and generative reasoning.

4.2 Machine Learning Analytics

The classification models were rigorously evaluated using standard performance metrics derived from True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

Accuracy was calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP+TN+FP+FN}$$

Precision, Recall, and F1-Score were also computed to evaluate classification robustness, particularly in handling class imbalance.

Among the evaluated algorithms—including Logistic Regression, Decision Tree, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest, and XGBoost—the Random Forest classifier achieved the highest balanced performance with an accuracy exceeding 95%. XGBoost demonstrated comparable performance, achieving approximately 95–96% accuracy, while Logistic Regression and SVM showed slightly lower but consistent results.

The confusion matrix analysis indicated high True Positive and True Negative rates, confirming strong model generalization capability. Feature importance analysis revealed that CGPA, coding score, internships, and communication skills were the most influential predictors of placement probability.

For salary prediction, regression models were evaluated using Mean Squared Error (MSE) and R^2 score. The Random Forest Regressor achieved the highest R^2 value and lowest prediction error, demonstrating strong capability in modeling nonlinear relationships between skills and compensation. However, slight deviations between predicted and actual salaries were observed due to external market variability not captured in the dataset.

4.3 Personalized Recommendation Module

After the placement probability and salary prediction are generated, the system provides personalized recommendations based on the student's input parameters. These recommendations help students identify areas where improvement is required, such as coding practice, communication skills, or gaining internship experience. The recommendation module acts as a guidance system that assists students in improving their employability before campus recruitment drives.

5. Conclusion

The AI-Powered Placement Probability and Salary Predictor successfully integrates machine learning and generative AI to develop a comprehensive, intelligent career support system. By analyzing key academic and skill-based parameters such as CGPA, coding proficiency, internships, aptitude, and communication skills, the system delivers highly accurate placement probability predictions and reliable salary estimations. The experimental results confirm that ensemble learning models, particularly Random Forest and XGBoost, achieve strong predictive performance with accuracy exceeding 95%, demonstrating robustness and generalization capability.

The regression-based salary prediction model further strengthens the platform by capturing nonlinear relationships between skill variables and compensation patterns. Despite slight variations caused by dynamic job market trends, the model provides practical and realistic salary projections that assist students in making informed career decisions.

A major innovation of this work lies in the seamless integration of a Generative AI recommendation layer. Through parameter-efficient fine-tuning techniques such as LoRA, the system minimizes computational overhead while delivering high-quality, personalized career guidance. This enhancement shifts the system from being purely predictive to being prescriptive—offering structured recommendations, skill roadmaps, certification suggestions, and interview preparation strategies tailored to individual profiles.

In addition to predictive accuracy and intelligent advisory capabilities, the system offers several extended advantages:

- **Improved transparency and interpretability**, allowing students to understand which features most influence their placement chances.
- **Adaptive learning capability**, where models can be retrained periodically to reflect evolving recruitment trends.
- **Institution-level analytics support**, enabling placement cells to identify performance gaps across batches and departments.
- **Cost-effective deployment**, as lightweight fine-tuning significantly reduces storage and hardware requirements.
- **Enhanced decision-making support**, empowering students to take corrective action well before campus recruitment drives.
- **Secure and structured data handling**, ensuring reliable backend processing and minimal risk of data inconsistency.

Moreover, the system encourages a culture of data-driven self-assessment among students. Instead of relying solely on subjective evaluation, learners receive quantifiable insights into their employability status. The integration of real-time dashboards and automated recommendations enhances user engagement and promotes continuous skill development.

From a broader perspective, this solution contributes to bridging the gap between academic learning and industry expectations. By aligning predictive analytics with personalized guidance, the platform supports outcome-based education and employability enhancement.

In future developments, the system can incorporate live job market APIs, recruiter feedback loops, domain-specific specialization models, multilingual advisory support, and cloud-native scalability to handle large-scale institutional adoption. With these advancements, the platform has the potential to evolve into a full-fledged AI-driven career intelligence ecosystem capable of supporting students, universities, and recruiters alike.

References

[1] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. Available: https://link.springer.com/article/10.1023/A:1010933404324

[2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, pp. 785–794. Available: https://dl.acm.org/doi/10.1145/2939672.2939785

[3] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," Journal of Machine Learning Research, vol. 3, pp. 1157–1182, 2003. Available: http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

[4] S. Kumar and P. Singh, "Performance Metrics in Classification Problems: A Review," arXiv preprint arXiv:1908.03509, 2019. Available: https://arxiv.org/abs/1908.03509

[5] E. J. Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models," arXiv preprint arXiv:2106.09685, 2021. Available: https://arxiv.org/abs/2106.09685

[6] T. Dettmers et al., "QLoRA: Efficient Finetuning of Quantized LLMs," arXiv preprint arXiv:2310.02036, 2023. Available: https://arxiv.org/abs/2310.02036

[7] React Documentation, "Getting Started with React," Meta Platforms, Inc. Available: https://react.dev/docs/getting-started

[8] Flask Documentation, "Flask Web Framework," Pallets Projects. Available: https://flask.palletsprojects.com/

[9] Spring Boot Documentation, "Spring Boot Project," VMware. Available: https://spring.io/projects/spring-boot

[10] Firebase Documentation, "Firebase Realtime Database," Google. Available: https://firebase.google.com/docs/database