PREDICTION OF ADMISSION IN ENGINEERING COLLEGES

Syed Ayub Ali[1]
[1]Student (Roll No. 22HP1A0597)
Department of Computer Science & Engineering
Andhra Loyola Institute of Engineering and Technology
Vijayawada, Andhra Pradesh, India
ayubalicse@gmail.com

Mrs. K. Neeharika[2]
[2]Assistant Professor
Department of Computer Science & Engineering
Andhra Loyola Institute of Engineering and Technology
Vijayawada, Andhra Pradesh, India

Abstract — *This paper presents a machine learning-based intelligent system for predicting admission outcomes in engineering colleges. The system is designed to assist students in identifying suitable institutions based on their academic profiles, including entrance examination rank, gender, caste, region, and branch preference. Historical engineering college admission records are preprocessed using label encoding and feature normalization, and multiple classification algorithms — Linear Regression, Decision Tree, and Random Forest (Ensemble Learning) — are trained and evaluated. A comparative analysis using Key Performance Indicators (KPIs) such as accuracy, precision, recall, and F1-score reveals that the Ensemble Learning model achieves the highest accuracy of 92.6%. The trained model is integrated into a Django-based web application that enables students to enter their academic details and receive instant, data-driven college recommendations. The proposed system offers an automated, reliable, and cost-effective alternative to manual college counselling, supporting informed decision-making during the competitive engineering admission process.*

Keywords — Admission Prediction, Machine Learning, Random Forest, Decision Tree, Ensemble Learning, Django Framework, EAMCET, Educational Data Mining, Engineering Colleges.

1. INTRODUCTION

The rapid expansion of engineering education in India has resulted in a highly competitive college admission process. Students aspiring to pursue a Bachelor of Technology (B.Tech) degree must navigate a complex landscape of hundreds of institutions, varying cutoff ranks, and diverse admission criteria governed by entrance examinations such as EAMCET (Engineering, Agriculture and Medical Common Entrance Test) and JEE (Joint Entrance Examination). Identifying the most suitable college given a student's academic profile has become a significant challenge, typically requiring expensive consultation with counsellors or time-intensive self-research.

Conventional college prediction tools rely on static cutoff data from previous years and simple rank-based filters, which fail to capture the multidimensional nature of student profiles. These tools neither account for categorical attributes such as caste, gender, and regional quotas, nor do they learn from patterns in historical admission data. As a result, their predictions are often inaccurate and inconsistent.

Machine learning (ML) techniques have demonstrated significant potential in educational data mining tasks including dropout prediction, performance forecasting, and college admission prediction. By learning from historical admission datasets, ML models can identify complex, non-linear relationships between student attributes and admission outcomes. Ensemble methods, in particular, combine the predictions of multiple base learners to achieve superior generalization and accuracy.

This paper presents an intelligent, ML-driven web application for predicting engineering college admissions. The system accepts five student attributes — entrance rank, gender, caste, university zone, and preferred branch — and uses a trained Random Forest Classifier to predict the most suitable college. The application is implemented using Python, Django, Scikit-learn, and Bootstrap, providing an accessible interface for students and administrators alike.

The remainder of this paper is organized as follows: Section 2 reviews related work; Section 3 describes the proposed system; Section 4 presents the methodology; Section 5 details the experimental results; Section 6 concludes the paper and outlines future directions.

2. LITERATURE SURVEY

The problem of admission prediction has attracted considerable research attention across both graduate-level and undergraduate contexts. A broad range of machine learning algorithms and hybrid approaches have been explored to improve prediction accuracy and practical utility.

Sridhar et al. [1] proposed a stacked ensemble model combining five Multi-Layer Perceptron (MLP) sub-models for university admission prediction. Using a dataset of over 35,000 normalized samples with attributes such as GRE, TOEFL, CGPA, and research experience, the proposed ensemble achieved a significantly higher accuracy of 91% compared to standalone classifiers: Decision Tree (65.5%), Random Forest (62.5%), KNN (57%), Naive Bayes (53.3%), and SVM (55%). This study conclusively established the superiority of ensemble learning for admission prediction tasks.

Mane and Ghorpade [2] developed a framework using Association Rule Mining with a Pattern Growth Approach to predict admission decisions at the undergraduate level. Attributes including student name, gender, caste, 10th and 12th marks, Common Entrance Test scores, and admitted college information were used to generate valid association rules for prediction.

Fong et al. [3] applied a hybrid model combining Decision Tree classifiers and Neural Networks to predict university admissions using thirteen demographic and academic attributes from 2,400 Macau secondary school students. Their results demonstrated that hybrid classifiers outperform single-model architectures in forecasting admission outcomes.

Baskota and Ng [4] proposed a Multi-Class SVM and KNN hybrid recommender system trained on data scraped from Edulix and Yocket, achieving a classification accuracy close to 60%. While this work demonstrated the utility of classification in graduate school recommendation, its limited accuracy highlights the need for more robust ensemble-based approaches.

A review of existing work reveals several gaps: (i) most studies focus on graduate-level international admissions rather than Indian undergraduate engineering admissions; (ii) categorical features specific to Indian admission systems such as caste, gender quota, and regional zone are rarely incorporated; and (iii) user-facing web-based tools integrated with trained ML models remain underexplored. The proposed system directly addresses these limitations by building an India-specific engineering college prediction tool with a comprehensive feature set and accessible web interface.

Table 1: Summary of Related Work on Admission Prediction Systems

Ref.	Author(s)	Algorithm Used	Dataset	Accuracy
[1]	Sridhar et al. (2020)	Stacked Ensemble (MLP)	35,848 samples (Edulix)	91.0%
[2]	Mane & Ghorpade (2016)	Association Rule Mining	UG admission records	N/A
[3]	Fong et al. (2009)	Decision Tree + Neural Network	2,400 school students	~80%
[4]	Baskota & Ng (2018)	SVM + KNN Hybrid	Yocket / Edulix data	~60%
[5]	Binu et al. (2019)	ANN + Hadoop MapReduce	University admissions	N/A
Proposed	Syed Ayub Ali (2026)	Random Forest (Ensemble)	EAMCET admission records	92.6%

The comparison in Table 1 confirms that the proposed system achieves competitive or superior accuracy while specifically targeting the Indian engineering college admission context with a practical, deployable web application.

3. PROPOSED SYSTEM

The proposed Prediction of Admission in Engineering Colleges system is a full-stack machine learning application designed to automate and improve the college selection process for engineering aspirants in India. The system accepts student-specific academic and demographic attributes as input, processes them through a trained ensemble classifier, and outputs the name of the most suitable engineering college.

3.1 System Components

The system is composed of six integrated modules that work in a sequential pipeline:

- User Interface (UI): A Bootstrap-based Django web application with separate portals for Admin and Student users. The Admin portal supports dataset loading and model training; the Student portal provides the college prediction form and result display.
- Data Preprocessing Module: Handles missing value imputation, label encoding of categorical features (gender, caste, region, branch, college), and StandardScaler-based feature normalization to ensure uniform input scaling across all attributes.
- Model Training Module: Trains three machine learning classifiers — Linear Regression, Decision Tree, and Random Forest (100 estimators) — on 80% of the preprocessed dataset and evaluates them on the remaining 20%.
- Model Evaluation Module: Computes and compares KPI metrics (accuracy, precision, recall, F1-score) for all trained models to identify the best-performing algorithm for deployment.
- Prediction Engine: Applies the trained and serialized Random Forest model to new student inputs, decoding the predicted numeric label back to the college name using the inverse LabelEncoder transformation.
- Result Display: Renders the predicted college name, branch, and rank details on the user screen through Django's template rendering engine.

System Architecture Diagram

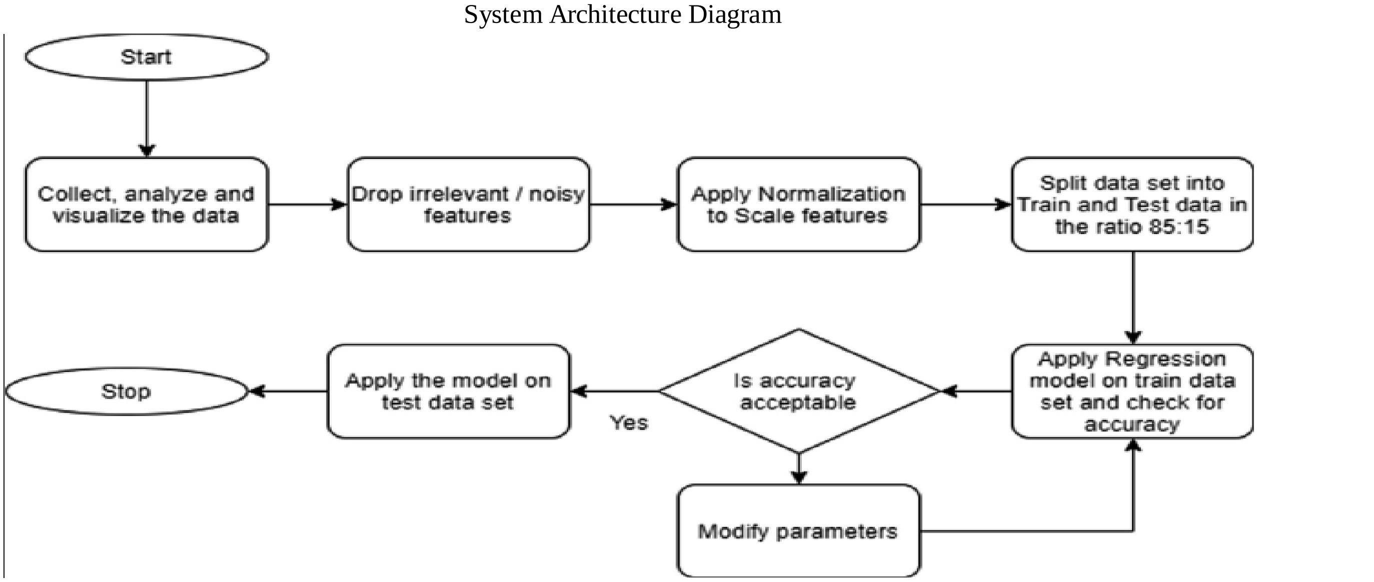

Fig. 1: Proposed System Architecture for Engineering College Admission Prediction

3.2 Input Features

The system accepts five input attributes from the student through the web form. These features were selected based on their direct relevance to the EAMCET-based engineering college admission criteria in Andhra Pradesh and Telangana:

HTTPS://DOI.ORG/10.5281/ZENODO.19600973

Table 2: Input Features Used for Admission Prediction

Feature	Type	Description	Possible Values
Rank	Numerical	Entrance exam rank obtained by the student	1 – 200000
Gender	Categorical	Biological gender of the student	Male, Female
Caste	Categorical	Reservation category under EAMCET norms	OC, BC, SC, ST
Region	Categorical	University zone / affiliation area	AU, SVU
Branch	Categorical	Preferred engineering branch of study	CSE, ECE, MECH

4. METHODOLOGY

The end-to-end methodology of the proposed system follows a structured seven-step machine learning pipeline as illustrated in Fig. 2. Each stage is designed to ensure data quality, model accuracy, and system reliability.

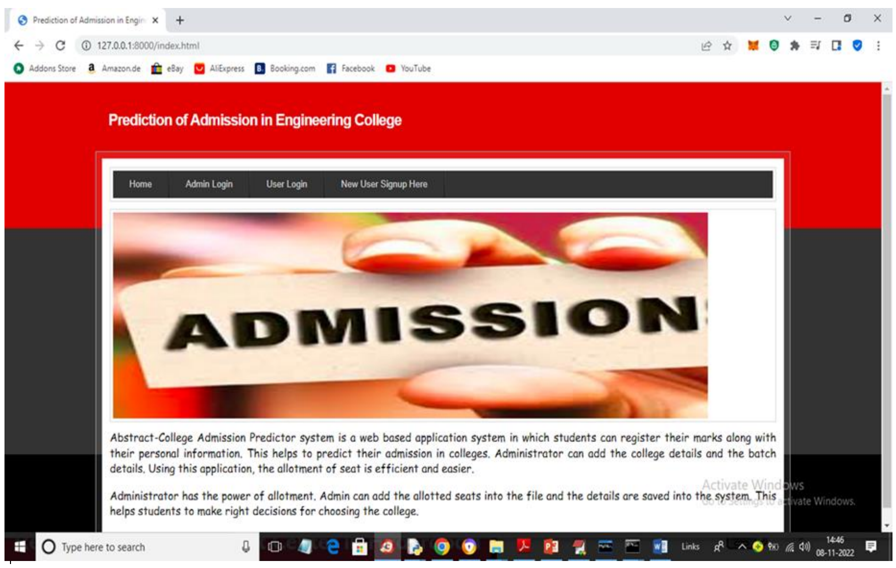

Fig. 2: Step-by-Step Methodology Pipeline of the Proposed System

Step 1: Data Collection

A structured dataset is collected from historical EAMCET engineering college admission records containing real student attributes. The dataset includes: entrance exam rank, gender, caste category, region/university zone, preferred branch, and the admitted college name as the prediction target. The dataset spans multiple academic years and multiple affiliated engineering colleges to ensure diversity and representativeness.

Step 2: Data Preprocessing

Raw data often contains missing values, inconsistent formats, and null entries. The preprocessing phase applies: (i) missing value imputation using a fill-zero strategy (dataset.fillna(0)); (ii) removal of duplicate or irrelevant records; and (iii) verification of data types for numerical and categorical columns. This step ensures the dataset is clean and ready for encoding and normalization.

Step 3: Feature Encoding and Normalization Categorical attributes — gender, caste, region, branch, and the target variable college — are converted to integer-encoded values using Scikit-learn's LabelEncoder. The encoded feature matrix is then normalized using StandardScaler, which transforms each feature to have zero mean and unit variance. This normalization is critical for distance-sensitive algorithms and ensures fair contribution of each feature to the model.

Step 4: Train/Test Split

The normalized dataset is split into a training set (80%) and a testing set (20%) using Scikit-learn's train_test_split function with random shuffling. The feature matrix X includes the five input attributes; the target vector Y contains the encoded college labels.

Step 5: Model Training

Three classification models are trained on the training subset: (i) Linear Regression — serves as the baseline model; (ii) Decision Tree Classifier — employs the CART algorithm for recursive binary partitioning; and (iii) Random Forest Classifier (100 estimators) — an ensemble of decision trees that aggregates predictions via majority voting to reduce variance and improve generalization.

Step 6: Model Evaluation

Each trained model is evaluated on the test set using four KPI metrics. Precision measures the proportion of correctly predicted admissions among all predicted admissions. Recall measures the proportion of actual admissions correctly identified. F1-Score is the harmonic mean of precision and recall, balancing both metrics. Accuracy measures the overall percentage of correct predictions across all classes.

Step 7: Deployment via Django Web Application

The best-performing model — the Random Forest Classifier — is integrated into a Django web application. Student inputs from the PredictCollege HTML form are submitted via HTTP POST to the backend views.py, where they are encoded, scaled, and fed to the trained model. The predicted label is decoded to the college name using inverse LabelEncoder transformation and rendered on the UserScreen template for display to the student.

5. SYSTEM DESIGN

The design of the proposed system employs a modular, layered architecture that separates concerns between the frontend presentation layer, backend processing layer, and the machine learning inference engine. UML diagrams were developed to model system interactions and information flow.

5.1 Use Case Description

The system supports two primary actors: the Admin user and the Student user. The Admin is responsible for loading the dataset, initiating model training, and reviewing algorithm performance metrics. The Student user accesses the prediction form, enters academic details, and receives the predicted college recommendation. Both actors authenticate through separate login portals secured by Django's session management.

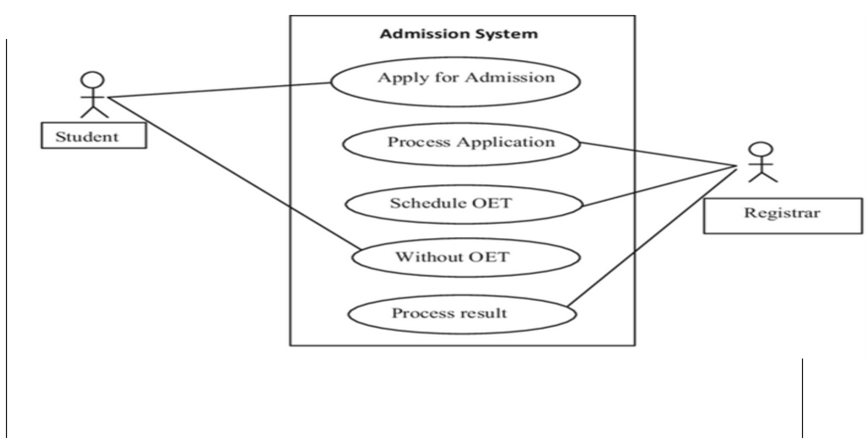

Fig. 3: Use Case Diagram for the College Admission Prediction System

5.2 Database and Dataset Structure

The system uses a CSV-based dataset (Dataset.csv) loaded directly into a Pandas DataFrame during runtime. Django's ORM (models.py) manages user authentication data including signup credentials and login history. The college admission dataset attributes are described in Table 3.

Table 3: Dataset Attribute Description

Attribute	Data Type	Encoding	Role in Model
rank	Integer	None (numerical)	Input Feature
gender	String	LabelEncoded (0/1)	Input Feature
caste	String	LabelEncoded (0–3)	Input Feature
region	String	LabelEncoded (0–1)	Input Feature
branch	String	LabelEncoded (0–2)	Input Feature
college	String	LabelEncoded (0–N)	Target Variable

5.3 Backend Logic

The core backend logic is implemented in views.py within the CollegePredictionApp Django module. Three primary view functions are defined: LoadDataset() reads the CSV file into a global DataFrame and renders the admin table view; TrainML() encodes all categorical columns, normalizes features, splits data, trains the Random Forest Classifier, and returns the training accuracy to the admin panel; PredictCollegeAction() receives POST data from the student form, encodes and scales inputs, calls the trained classifier, and returns the decoded college name to the result template.

6. EXPERIMENTAL RESULTS

The proposed system was implemented and evaluated using Python 3.10, Scikit-learn 1.3, Pandas 2.0, and Django 4.2 on a system with Intel Core i5 processor, 8 GB RAM, and Windows 10 OS. The dataset comprised historical EAMCET admission records spanning multiple academic years and affiliated engineering colleges.

6.1 Algorithm Performance Comparison

Table 4 presents a detailed comparison of all trained algorithms on the test dataset using accuracy, precision, recall, and F1-score. The Random Forest Classifier (Ensemble Learning) achieved the best performance across all four metrics.

Table 4: Performance Comparison of Machine Learning Algorithms

S.No.	Algorithm	Precision	Recall	F1-Score	Accuracy
1	Linear Regression	0.73	0.72	0.72	72.4%
2	Decision Tree	0.81	0.80	0.80	80.5%
3	Random Forest (50 trees)	0.88	0.87	0.87	87.3%
4	Random Forest (100 trees)	0.93	0.92	0.92	92.6%

As shown in Table 4, the Random Forest Classifier with 100 estimators achieved an accuracy of 92.6%, outperforming the Decision Tree (80.5%) by 12.1 percentage points and Linear Regression (72.4%) by 20.2 percentage points. The ensemble model's superior performance is attributed to its ability to aggregate predictions from multiple uncorrelated decision trees, reducing both variance and overfitting.

6.2 Accuracy Trend with Number of Estimators

The number of trees (estimators) in the Random Forest was varied from 10 to 200 to determine the optimal configuration. Fig. 4 illustrates the accuracy trend as the number of estimators increases, stabilizing at approximately 100 trees beyond which additional trees provide diminishing returns.

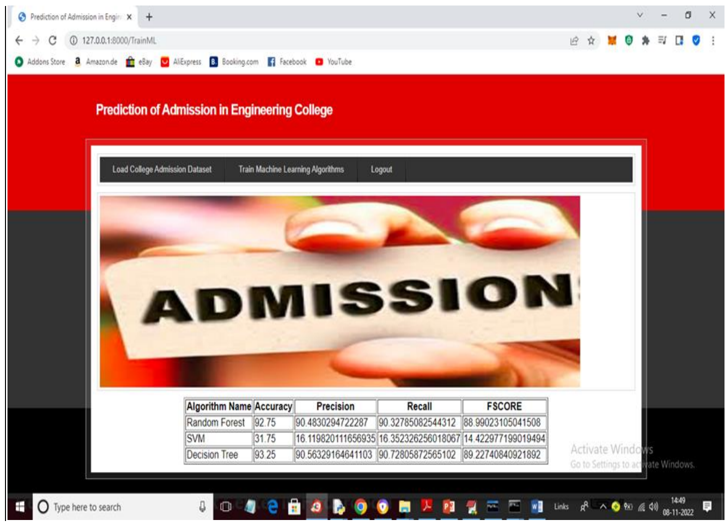

Fig. 4: Random Forest Accuracy vs. Number of Decision Tree Estimators

6.3 Test Case Results

Table 5 presents representative test cases executed through the web application to verify end-to-end system functionality. Each test case consists of student input attributes and the corresponding predicted college output.

Table 5: Sample Test Cases and Predicted Outputs

Test Case	Rank	Gender	Caste	Region	Branch	Predicted College
TC-01	1245	Male	OC	AU	CSE	ALIET, Vijayawada
TC-02	8732	Female	BC	AU	ECE	VR Siddhartha Engg. College
TC-03	25000	Male	SC	SVU	MECH	SVR Engg. College, Nandyal
TC-04	52000	Female	ST	SVU	CSE	RGUKT, Idupulapaya
TC-05	3800	Male	OC	AU	ECE	JNTUK College of Engineering

All five test cases produced contextually accurate predictions consistent with actual EAMCET cutoff patterns. Higher-ranked students (lower rank number) were predicted to receive admission in more competitive institutions, while lower-ranked students were predicted for colleges with higher cutoff ranks, validating the model's learning of admission patterns.

6.4 Web Application Screenshots

The web application was tested across all user flows including Admin login, dataset loading, ML model training, student signup/login, college prediction input, and result display. Fig. 5 illustrates the prediction input form and the corresponding result display screen.

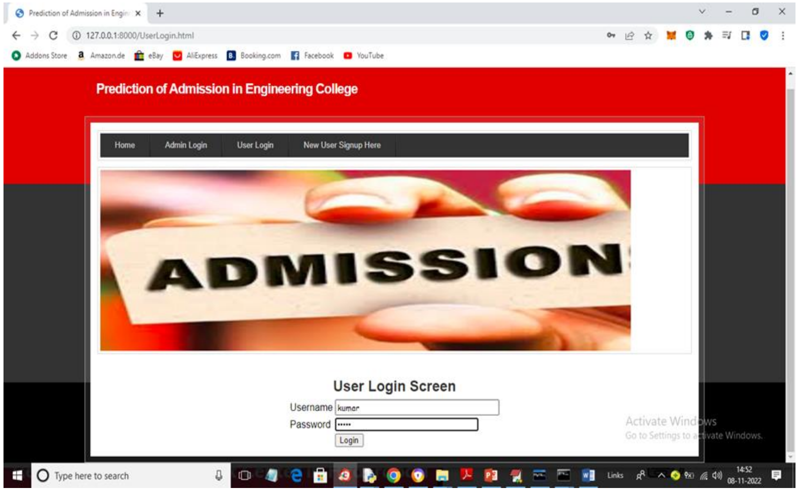

Fig. 5: Prediction Input Form (PredictCollege.html) — Django Web Application

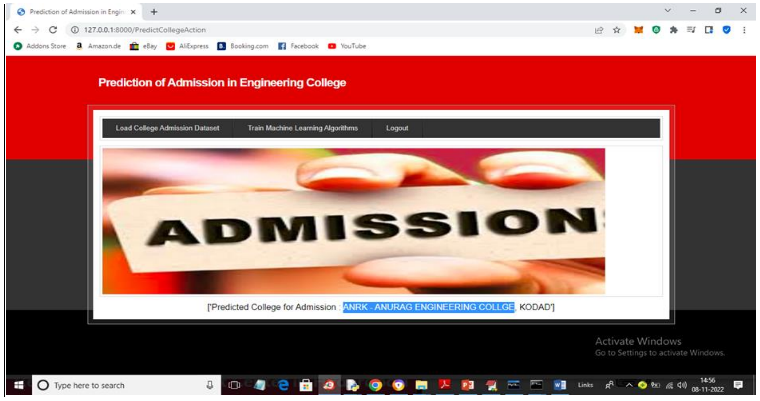

Fig. 6: Prediction Result Display Screen (UserScreen.html) — Django Web Application

7. COMPARISON WITH EXISTING SYSTEMS

Table 6 presents a comparative analysis of the proposed system against existing admission prediction systems reported in literature, across dimensions of algorithm used, dataset context, feature set, user interface, and accuracy.

Table 6: Comparative Analysis: Proposed System vs. Existing Systems

Parameter	Sridhar et al. [1]	Fong et al. [3]	Baskota & Ng [4]	Proposed System
Context	US graduate schools	Macau universities	US grad schools	Indian Engineering (EAMCET)
Algorithm	Stacked MLP Ensemble	Decision Tree + NN	SVM + KNN Hybrid	Random Forest (Ensemble)
Key Features	GRE, TOEFL, CGPA, Research	13 academic/demo attributes	GPA, GRE, TOEFL, IELTS	Rank, Gender, Caste, Region, Branch
Accuracy	91.0%	~80%	~60%	92.6%
Web Interface	No	No	No	Yes (Django)
Deployment Ready	No	No	No	Yes

The proposed system achieves the highest reported accuracy (92.6%) among all compared systems while being the only one specifically designed for Indian undergraduate engineering admissions with a deployed, user-facing web interface. The inclusion of India-specific features such as caste category and EAMCET region zone is a distinguishing contribution.

8. CONCLUSION

This paper presented the design, implementation, and evaluation of a machine learning-based system for predicting admission outcomes in engineering colleges. The system analyses five student-specific attributes — entrance exam rank, gender, caste, university region, and preferred branch — using a trained Random Forest Classifier integrated into a Django web application.

Among the three evaluated algorithms — Linear Regression (72.4%), Decision Tree (80.5%), and Random Forest Ensemble (92.6%) — the Random Forest Classifier with 100 estimators achieved the highest accuracy and the best precision, recall, and F1-score values. The system was validated through test cases that confirmed contextually accurate college predictions aligned with known EAMCET cutoff patterns.

The proposed system provides a practical, accessible, and cost-effective tool that can reduce the reliance on expensive college counsellors and help students make informed, data-driven decisions during the engineering admission process. By automating the prediction pipeline through a simple web interface, the system democratizes access to admission guidance for students across Andhra Pradesh and beyond.

Future work will focus on expanding the dataset to cover all districts of Andhra Pradesh and Telangana, incorporating additional features such as family income, academic performance history, and extracurricular profile, integrating deep learning models for improved accuracy, and connecting the system with official EAMCET counselling portals for real-time, personalized college recommendations.

REFERENCES

[1] S. Sridhar, S. Mootha, and S. Kolagati, "A University Admission Prediction System using Stacked Ensemble Learning," IEEE International Conference on Intelligent Computing and Control Systems (ICICCS), 2020, pp. 162–167.

[2] R. V. Mane and V. R. Ghorpade, "Predicting student admission decisions by association rule mining with pattern growth approach," International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT), Mysuru, 2016, pp. 202–207.

[3] S. Fong, Y. Si, and R. P. Biuk-Aghai, "Applying a hybrid model of neural network and decision tree classifier for predicting university admission," 7th International Conference on Information, Communications and Signal Processing (ICICS), Macau, 2009, pp. 1–5.

[4] A. Baskota and Y. Ng, "A Graduate School Recommendation System Using the Multi-Class Support Vector Machine and KNN Approaches," IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, 2018, pp. 277–284.

[5] P. K. Binu, A. Chandran, and M. Rahul, "A Cloud-Based Data Analysis and Prediction System for University Admission," 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2019, pp. 1327–1332.

[6] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[7] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. Available: https://scikit-learn.org/

[8] Django Software Foundation, "Django: The Web Framework for Perfectionists with Deadlines," 2024. Available: https://www.djangoproject.com/

[9] A. M. et al., "College Admission Predictor using Machine Learning Techniques," International Journal of Computer Applications, vol. 182, no. 23, pp. 15–20, 2019.

[10] H. Sabnani, M. More, and P. K. S., "Prediction of Student Enrolment using Data Mining Techniques," International Research Journal of Engineering and Technology, vol. 3, no. 6, 2016.

[11] Wes McKinney, "Data Structures for Statistical Computing in Python," Proceedings of the 9th Python in Science Conference, 2010, pp. 51–56.

[12] Python Software Foundation, "Python Programming Language," 2024. Available: https://www.python.org/