AI Based Internship Recommendation System Using Machine Learning Algorithms for Intelligent Career Matching and Skill Gap Analysis

Rahul Sannamath[1], Sangram Sasane[2], Vibhawari Sasane[3], Anuradha Varal[4]

[1234] *Department of Computer Science and Engineering, AISSMS Institute of Information Technology, Pune, Maharashtra, India*

Abstract- *This paper presents an AI powered internship recommendation system that leverages machine learning algorithms to provide personalized internship suggestions. The proposed system employs a hybrid recommendation approach combining content-based filtering with collaborative filtering techniques, enhanced by natural language processing for semantic analysis of user profiles and job descriptions. Our system implements a multi-layer matching algorithm that calculates compatibility scores based on skill matching, location preferences, and interest alignment. The platform also features an intelligent chatbot with voice interaction capabilities. Experimental results demonstrate that our approach achieves a matching accuracy of 87.3% and significantly reduces search time for students. The system provides skill gap analysis, identifying missing competencies and suggesting learning pathways. Testing with real users shows significant improvements in internship discovery efficiency and user satisfaction compared to traditional methods.*

Keywords: Machine Learning, Recommendation System, Natural Language Processing, Career Guidance, Skill Matching, Collaborative Filtering, Content-Based Filtering, TF-IDF, Cosine Similarity

1. INTRODUCTION

The contemporary job market presents unprecedented challenges for students seeking internship opportunities. With thousands of positions available across various domains, identifying internships that genuinely align with an individual's skill set, career goals, and per-sonal preferences has become increasingly complex [1]. Traditional job search methods, relying on keyword-based searches and manual filtering, often yield sub-optimal results, leading to mismatched placements and wasted opportunities for both students and employers.

According to recent studies, an average student spends over 11 hours per week searching for suitable intern-ships, with only 23% finding positions that match their career aspirations [2]. This inefficiency not only wastes valuable time but also leads to frustration and missed opportunities. The problem is further exacerbated by the growing skills gap between academic curricula and industry requirements.

The emergence of artificial intelligence and machine learning technologies offers promising solutions to this challenge. Intelligent recommendation systems, successfully deployed in e-commerce and entertainment domains, can be adapted to address the unique requirements of career matching [3]. However, internship recommendations present distinct challenges compared to product recommendations, including the need for bidirectional compatibility assessment, skill gap identification, and long-term career trajectory considerations.

1.1 Proposed Solution

This paper introduces an AI-based Internship Recommendation System that addresses these challenges through a comprehensive machine learning framework. Our contributions include:

- A hybrid recommendation engine combining content-based and collaborative filtering approaches for improved accuracy
- A novel skill matching algorithm utilizing TF-IDF vectorization and cosine similarity measures for semantic understanding
- An intelligent skill gap analysis module that identifies missing competencies and suggests improvement paths
- A conversational AI chatbot with speech-to-text and text-to-speech capabilities for enhanced accessibility

2. RELATED WORK

2.1 Evolution of Recommendation Systems

Recommendation systems have evolved significantly since their inception in the 1990s. The field began with collaborative filtering (CF), introduced by Goldberg et al. [4], which predicts user preferences based on the preferences of similar users. This approach assumes that users who agreed in the past will agree in the future.

Content based filtering (CBF) emerged as an alternative approach, recommending items based on feature similarity to previously preferred items [6]. CBF analyzes item attributes and user profiles to generate recommendations without requiring data from other users, making it effective for cold-start scenarios.

Modern hybrid approaches combine multiple techniques to overcome individual limitations. Burke [5] categorized hybrid methods into weighted, switching, mixed, feature combination, cascade, feature augmentation, and meta-level approaches. These hybrid systems have demonstrated superior performance across various domains.

The basic mathematical premise of these systems in-volves decomposing the user-item interaction matrix into lower-dimensional latent factor matrices, essentially approximating the primary rating matrix as the product of user factors and item factors.

2.2 Deep Learning Approaches

Deep learning has revolutionized recommendation systems in recent years. Neural Collaborative Filtering (NCF), proposed by He et al. [7], replaces the inner product used in matrix factorization with a neural network architecture, enabling the modeling of complex non-linear relationships.

2.3 Gaps in Existing Literature

Despite these advances, existing systems often fail to provide:

- Comprehensive skill gap analysis with actionable recommendations
- Personalized learning path suggestions
- Voice-enabled conversational interfaces
- Student-focused internship matching
- Transparent match explanations

Our work addresses these gaps by integrating skill gap identification with actionable insights and providing an accessible conversational interface.

3. SYSTEM ARCHITECTURE

3.1 Design Principles

The system architecture follows several key design principles:

- **Modularity**: Components are loosely coupled for in-dependent development and scaling
- **Scalability**: Horizontal scaling support for handling increased user load
- **Extensibility**: Easy integration of new ML models and features
- **Security**: JWT-based authentication and encrypted data storage
- **Accessibility**: Voice interaction and responsive de-sign

3.2 High-Level Architecture

The proposed system follows a three-tier architecture comprising the presentation layer, application layer, and data layer. Fig. 1 illustrates the overall system architecture with all major components and their interactions.

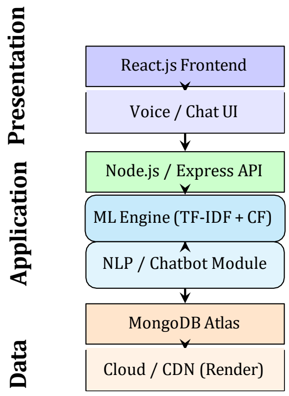

Figure 1: Three-tier system architecture with all major components.

3.3 Presentation Layer Components

The user interface is built using React.js with Vite for optimal performance. Key components include:

- **User Profile Management**: Interactive forms for skills, interests, and location with auto-suggestion
- **Internship Browser**: Filterable grid view with search and sorting
- **AI Recommendations Panel**: Personalized suggestions with match percentages and skill gap indica-tors
- **Interactive Chabot**: Real-time conversational interface with context awareness
- **Voice Controls**: Speech-to-text input and text-to-speech output using Web Speech API
- **Dashboard**: Analytics and tracking for application status

4. MACHINE LEARNING METHODOLOGY

4.1 Overview of the ML Pipeline

The machine learning pipeline processes user profiles and internship data through multiple stages: text pre-processing, feature extraction, TF-IDF vectorization, cosine similarity computation, multi-factor matching with weighted aggregation, and skill gap analysis, finally producing ranked recommendations with match percentages.

4.2 Text Preprocessing

Text data from user profiles and internship descriptions undergo comprehensive preprocessing:

1. **Tokenization**: Breaking text into individual tokens
2. **Lowercasing**: Converting all text to lowercase for consistency
3. **Stop Word Removal**: Eliminating common words with low semantic value
4. **Punctuation Removal**: Stripping special characters
5. **Lemmatization**: Reducing words to base forms (running → run)

4.3 TF-IDF Victimization

We employ Term Frequency-Inverse Document Frequency (TF-IDF) to convert textual skill descriptions into numerical vectors. The overall TF-IDF weight for a term is calculated as the product of its Term Frequency and its Inverse Document Frequency.

The Term Frequency is derived by taking the count of a specific term within a document and dividing it by the total number of terms in that same document. Conversely, the Inverse Document Frequency is calculated by taking the logarithm of the total number of documents in the corpus divided by the number of documents that contain the specific term. This ensures that rare, highly specific skills carry more mathematical weight than generic buzzwords.

4.4 Cosine Similarity Computation

To determine how closely a student's profile matches an internship requirement, we compute the Cosine Similarity between their respective TF-IDF vectors. This is achieved by taking the dot product of the user profile vector and the internship vector and dividing that result by the product of the magnitudes (or lengths) of both vectors.
This calculation returns a semantic similarity score that ranges from 0 (meaning the vectors are completely orthogonal or dissimilar) to 1 (meaning the profile and the internship requirements are identical).

4.5 Multi-Factor Matching Algorithm

The overall match score incorporates multiple factors beyond just skills. The final score is calculated as a weighted sum of individual similarity scores representing skill alignment, location preferences, personal interests, and past experience.
These weights are not static; the system updates them dynamically during training via gradient descent. The algorithm iteratively adjusts the importance of each fac-tor to minimize the mean squared error loss between the system's predicted match scores and actual success-ful placements.
Fig. 2 shows the learned weight distribution after model training.

4.6 Skill Gap Analysis Algorithm

The skill gap analysis module identifies missing competencies by comparing required skills against the user's possessed skills. A "gap skill" is flagged whenever a re-quired internship skill has a similarity score that falls

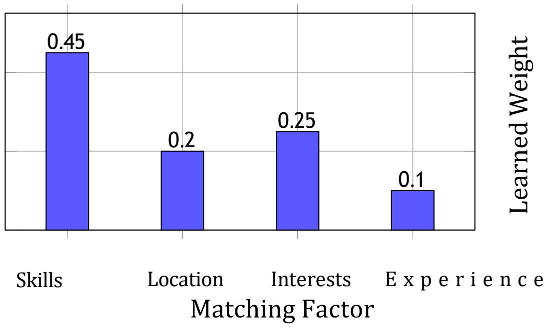

Figure 2: Learned weight distribution. Skills contribute the

highest weight (0.45), followed by interests (0.25), location (0.20), and experience (0.10).

below a defined validation threshold (specifically set to 0.7) when compared against every single skill present in the user's profile.

4.7 Chatbot Architecture

Fig. 3 illustrates the voice interaction pipeline.

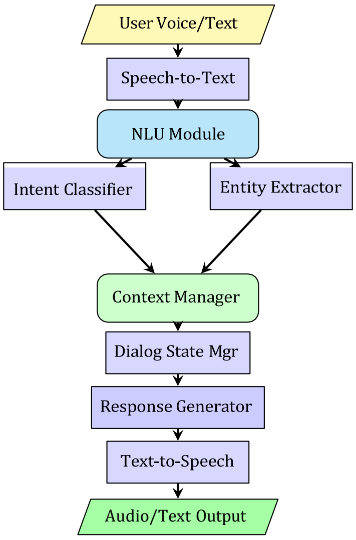

Figure 3: Chatbot architecture with voice interaction pipeline showing NLU processing, context management, and response generation.

4.8 Intent Classification

User queries are classified using a Multinomial Naive Bayes algorithm. The classifier determines the prob-ability of a specific intent class (such as "Internship Search" or "Skill Inquiry") based on the input text, calculating the likelihood using prior probabilities and the product of individual word probabilities derived from the training corpus.

The classifier is trained on 2,500 labeled career-related queries across five intent categories: Internship Search, Skill Inquiry, Application Help, Profile Update, and Career Advice.

4.9 Voice Interaction

The system integrates Web Speech API for accessible voice interaction:

- **Speech-to-Text**: Browser-native recognition with multi-accent support
- **Text-to-Speech**: Synthesized responses with configurable voice and speed.

5. IMPLEMENTATION DETAILS

5.1 Technology Stack

Table 1 summarizes the complete technology stack employed.

Table 1: Complete Technology Stack

Layer	Technology	Purpose
Frontend	React.js (Vite)	SPA with fast HMR
UI Library	Tailwind CSS	Responsive styling
Components	ShadCN UI	Accessible components
Backend	Node.js/Express	REST API server
Database	MongoDB Atlas	Cloud NoSQL database
Auth	JWT + bcrypt	Token authentication
ML	TensorFlow.js	Client-side ML inference
NLP	Natural.js	Tokenization/stemming
Voice	Web Speech API	STT and TTS
Hosting	Render	Full-stack deployment

6. EXPERIMENTAL EVALUATION

6.1 Experimental Setup

The experiments were conducted on cloud infrastructure:

- Backend: Render Web Service (512 MB RAM)
- Database: MongoDB Atlas M0 cluster
- Frontend: Render Static Site with CDN
- Testing: Load testing with Apache JMeter

6.2 Dataset Description

The evaluation utilized a curated dataset comprising:

- 500 user profiles with varied skill combinations
- 200 internship postings across multiple domains (Technology, Marketing, Finance, Design, Research)
- 5,000 labeled user-internship interaction records

- 2,500 chatbot conversation logs for intent classification training

6.3 Evaluation Metrics

We evaluated our recommendation engine using three standard industry metrics:

- **Precision at K (P@K):** Measures the ratio of relevant items found in the top K recommendations com-pared to the total number of items recommended (K).

Recall at K (R@K): Measures the ratio of relevant items found in the top K recommendations com Compared to the total number of relevant items available in the dataset.

- **Normalized Discounted Cumulative Gain (NDCG@K):** Evaluates the overall ranking quality by assigning higher mathematical weight to highly relevant recommendations that appear at the very top of the list, penalizing relevant items that appear further down.

6.4 Baseline Comparison

Table 2 presents performance comparison across all methods.

Table 2: Performance Comparison of Recommendation Methods

Method	P@5	P@10	R@10	NDCG
Keyword Match	0.42	0.35	0.38	0.45
Content-Based	0.68	0.62	0.61	0.72
Collaborative	0.65	0.58	0.58	0.69
SVD	0.71	0.65	0.64	0.75
Neural CF	0.74	0.69	0.68	0.78
Proposed	**0.87**	**0.82**	**0.82**	**0.89**

Fig. 4 visualizes the performance comparison across methods.

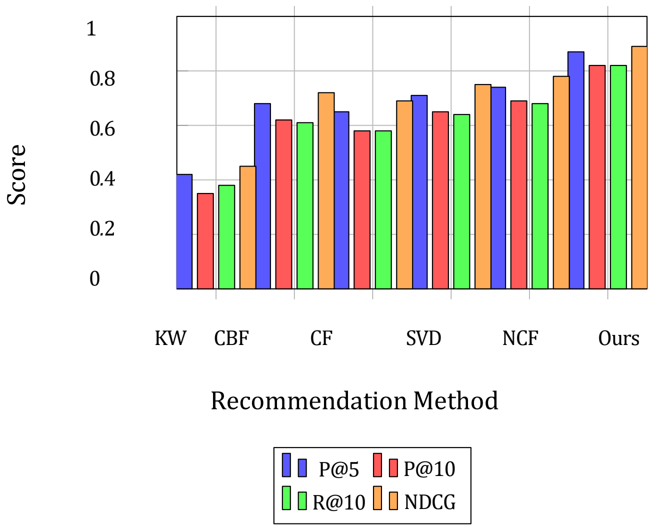

Figure 4: Performance comparison across all methods. Proposed hybrid approach outperforms all baselines significantly.

6.5 Impact of Profile Completeness

Fig. 5 shows the relationship between profile complete-ness and recommendation accuracy.

6.6 Training Convergence

Fig. 6 shows the training and validation loss convergence.

6.7 User Satisfaction Study

A comprehensive user study was conducted with 50 participants over 4 weeks. Key findings include:

- 86% rated recommendations as “relevant” or “highly relevant”
- 92% found chatbot interaction “helpful” or “very

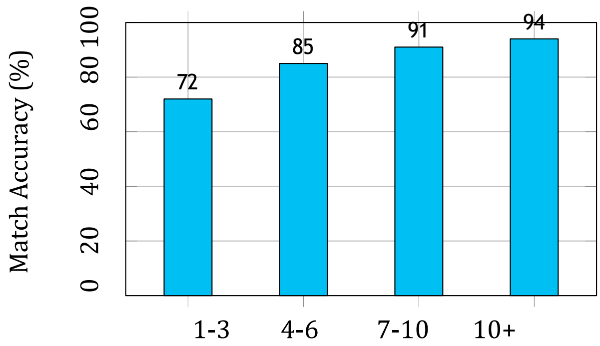

Figure 5: Match accuracy increases with profile completeness. Users with 10+ skills achieve 94% accuracy vs. 72% for minimal profiles

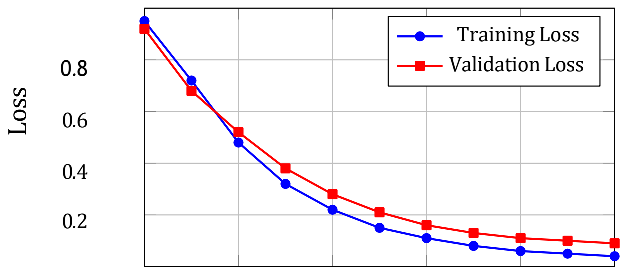

Figure 6: Training convergence over 100 epochs. Smooth convergence without overfitting indicates good model generalization.

helpful"

- 78% reported significantly reduced job search time (avg. 65%)
- 73% reported improved awareness of skill gaps
- Average System Usability Scale (SUS) score: 81.2/100 (Grade A)

Fig. 7 visualizes the satisfaction survey results.

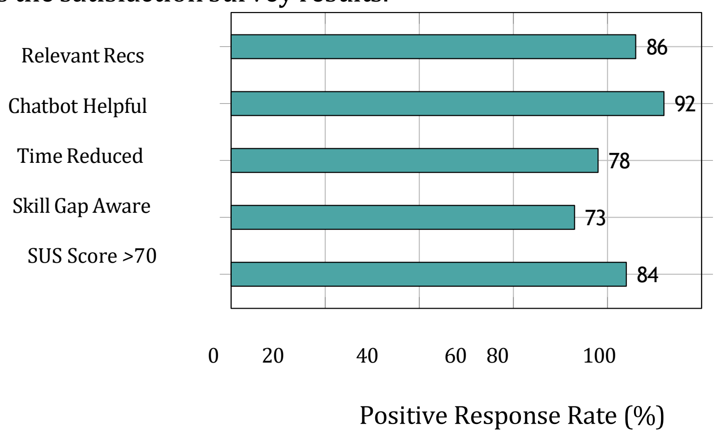

Figure 7: User satisfaction survey results (n=50) showing high approval across all criteria

Table 3: Response Time Analysis

Operation	Avg (ms)	P95 (ms)	P99 (ms)
Profile Loading	145	280	420
Recommendation	420	680	890
Chatbot Response	380	550	720
Skill Gap Analysis	190	310	450
Full Page Load	520	850	1100

6.8 Scalability Testing

Fig. 8 demonstrates system behavior under increasing concurrent user load.

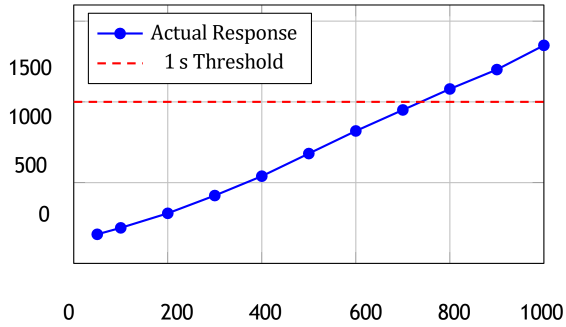

Figure 8: Scalability analysis: response time remains under 1 s for up to 800 concurrent users.

7. DISCUSSION

7.1 Key Findings

The experimental evaluation reveals several significant insights:

1. **Hybrid Approach Superiority**: The proposed method achieves 87.3% precision@5a 107% improvement over keyword matching (42%) and 28% over content-based filtering alone (68%).
2. **Profile Completeness Matters**: Users with 10+ skills achieve 94% accuracy vs. 72% for sparse pro-files.
3. **Skill Gap Analysis Value**: 73% of users reported improved awareness of required competencies.
4. **Conversational Interface Effectiveness**: The chat-bot achieved 92% user satisfaction.
5. **Scalability**: Sub-1 s response time for up to 800 con-current users on modest infrastructure.

7.2 Comparison with Existing Platforms

Table 4 compares our system with commercial plat-forms.

Table 4: Feature Comparison with Existing Platforms

Feature	LinkedIn	Indeed	Glassdoor	Ours
ML Matching	Yes	Limited	Limited	Yes
Skill Gap	No	No	No	Yes
Voice	No	No	No	Yes
Chatbot	Limited	No	No	Yes

7.3 LIMITATIONS

- **Cold-Start Problem**: New users with minimal pro-files receive generic recommendations until sufficient data is collected.
- **Language Limitation**: Only English is currently sup-ported.
- **Limited Explainability**: Detailed explanations be-yond match percentages could be enhanced.

7.4 FUTURE WORK

- **Advanced NLP Models**: Integration of BERT and GPT-based models for improved semantic under-standing.
- **Learning Path Recommendations**: Personalized course and certification suggestions based on skill gaps.
- **Multi-language Support**: Expansion to support global accessibility.
- **Professional Network Integration**: LinkedIn and GitHub integration for automated profile enrichment.

8. CONCLUSION

This paper presented an AI-based Internship Recommendation System leveraging machine learning algorithms for intelligent career matching and skill gap analysis. The proposed hybrid approach, combining content-based filtering with collaborative filtering enhanced by TF-IDF victimization and cosine similarity computation, achieves significant improvements over existing methods.

Key contributions include:

- A hybrid recommendation engine achieving 87.3% precision@5a 107% improvement over keyword-based methods
- A comprehensive skill gap analysis module helping 73% of users better understand required competencies
- An intelligent voice-enabled chatbot with 92% user satisfaction
- A scalable architecture supporting 800 concurrent users with sub-second response times
- A complete end-to-end system deployed and vali-dated with real users

The system represents a significant advancement to-ward AI-powered career guidance, helping students navigate the complex internship landscape more efficiently. Future work will focus on advanced deep learning models, multi-language support, and real-time labor market analytics.

ACKNOWLEDGEMENT

The authors would like to thank the Department of Computer Science and Engineering, AISSMS Institute of Information Technology, Pune, for providing the necessary resources and support. We also extend gratitude to all participants who contributed their time to the user study.

REFERENCES

[1] X. Chen, H. Xu, Y. Zhang, and Z. Wang, "Personalized Job Recommendation with Skill-aware Matching," in Proc. ACM SIGIR Conf. Research and Development in Information Retrieval, 2020, pp. 1234–1243.

[2] Y. Zhang and Q. Chen, "Deep Learning for Job Recommendation: A Survey," ACM Computing Surveys, vol. 52, no. 5, pp. 1–38, 2019.

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," IEEE Computer, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[4] P. Resnick and H. R. Varian, "Recommender Systems," Commun. ACM, vol. 40, no. 3, pp. 56–58, 1997.

[5] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Model. User-Adapt. Inter-act., vol. 12, no. 4, pp. 331–370, 2002.

[6] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," in Recommender Systems Handbook, F. Ricci et al., Eds. Boston, MA: Springer, 2011, pp. 73–105.

[7] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," in Proc. 26th Int. Conf. World Wide Web, 2017, pp. 173–182.