IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

An AI-Driven Placement Assessment and Performance Evaluation Framework

[1]Akshatha A M, [2]Dr.Nirmala C R, [3]Dr.Sreenivasa B R

[1]M.Tech Student, [2]Professor and Head, [3]Professor and PG Coordinator

[1]Artificial Intelligence and Data Science,

[1]Bapuji Institute of Engineering and Technology, Davangere, India

Abstract: Success in campus recruitment usually calls for a good mix of aptitude, technical, and the ability to communicate well. Most students, however, struggle to judge how prepared they actually are and to pinpoint exactly which areas need more work before they sit for placement drives. To address this gap, the present work proposes an AI-based placement evaluation system that brings aptitude testing, technical review, and HR interview assessment together inside one web platform, all while offering each student guidance tailored to their own performance. Multiple-choice and objective-type questions are graded automatically, whereas responses given during the technical and HR rounds are examined through rule-based Natural Language Processing (NLP) methods. Scores gathered across all three rounds are then fed into a Decision Tree classifier, which labels a student as either Placement Ready or Needs Improvement. Building on this outcome, the system flags the student's weaker areas and puts together a tailored improvement plan. To judge how well the model performs, Accuracy, Precision, Recall, F1-score, and ROC-AUC were used as evaluation metrics, and the results point to consistent, dependable classification. Built on Python, Django, HTML, CSS, JavaScript, and SQLite, the resulting application lets students keep track of their progress and sharpen their placement preparation over time.

Keywords: Placement Intelligence, PrepEdge AI, Decision Tree, Machine Learning, Rule-Based Natural Language Processing (NLP), Placement Readiness Prediction, Skill Gap Analysis, Personalized Learning Roadmap, Django.

I. Introduction

Stepping from college into a job is what campus placement is really about, and it's a stage that tests more than academics. Recruiters weigh candidates on several fronts together aptitude, technical depth, how well they communicate, how they solve problems, and how they hold up in an interview. Coursework gives students theory, but plenty of them walk into placement season not knowing where they truly stand or what to fix first. That uncertainty is exactly why smart, data-backed platforms that assess performance and point students toward the right improvements are in growing demand [1], [2].

AI and Machine Learning have pushed educational tools toward far more data-backed evaluation and prediction than before. These methods now regularly help track how students are doing, catch patterns in the way they learn, and forecast how placements might turn out giving institutions and learners alike a clearer read on strengths and gaps that still need attention [3], [4].

Personalization has become a big part of this picture too. Rather than herding every student through the same fixed syllabus, adaptive systems point each one toward resources that match their own strengths, weaknesses, and pace, so effort goes where it's actually needed. Even so, most tools out there handle just one job prediction or recommendation instead of covering the full placement-prep journey end to end [5], [6].

NLP and AI are also stepping into interview evaluation, scoring technical and HR answers and returning feedback almost hands-free. The catch is that many of these setups lean on heavy compute, which isn't always something smaller institutions can spare [7], [8].

This project tries to close that gap with a single AI-driven web app that walks students through placement prep from start to finish aptitude checks, technical and HR interview practice, readiness prediction, and learning support, all under one roof. Technical and HR answers go through rule-based NLP, and a Decision Tree classifier, whether more prep is needed. The system then maps out where a student is weak and builds a roadmap around it. Running on Python, Django, HTML, CSS, JavaScript, and SQLite, it gives students a live, interactive way to track progress and get placement-ready.

II. LITERATURE REVIEW

AI and ML now show up across educational systems for jobs like predicting placement outcomes, scoring interviews, and adjusting learning paths on the fly. Research in this space has looked at student performance from angles like academic record, technical skill, communication, internships, and interview results. What follows is a rundown of that work and where it falls short of what this framework tries to do.

Dake and colleagues [1] built a tree-based ML pipeline to gauge how employable IT graduates are, running it through Decision Tree, Random Forest, Extra Trees, and XGBoost. The tree-based models came out ahead when the data was clean and well-structured — though that dependence on data quality remained a limiting factor.

Saidani and colleagues [2] combined academic records with internship history in a model built to forecast employability, and internship experience turned out to carry real weight in the prediction. Naturally, that only holds if the internship data itself is solid separately.

Aruna and colleagues [3] built a placement-prediction tool aimed at flagging students needing extra prep; it classified well but stopped there, without any feedback loop or learning suggestions attached.

Wu and colleagues [4] built a system that recommends learning activities based on a student's own strengths, weaknesses, and pace, and saw better learning outcomes come out of it. It wasn't aimed at placement prep specifically, though general education was the target.

Kadu and colleagues [5] folded CGPA, aptitude, internships, certifications, projects, and soft skills into one placement-prediction and skill-recommendation engine. Pairing prediction with recommendation gave it an edge, but like most such systems, results still tracked the quality of the data going in.

Uppalapati and colleagues [6] leaned on Generative Language Models to build a mock-interview grader that scores answers and hands back detailed feedback. Interview prep got noticeably better as a result, though the compute cost made it a tough sell for institutions without deep resources.

Bershika and colleagues [7] paired NLP with emotion recognition to build a virtual interviewer that reads communication skill during a mock session. That emotional layer made the experience feel more genuine, at the cost of extra computational load and hardware demand.

Matzavela and Alepis [8] built a prediction model for mobile learning environments that turns academic performance into a personalized study plan. It handled continuous learning well, though running it at scale meant keeping the data pipeline constantly fed and monitored.

ElSharkawy and colleagues [9] pitted SVM against Decision Tree and Random Forest for employability prediction, and the tree-based pair came out on top for accuracy. One caveat: the employability labels came from surveys, which leaves some room to question how reliable the ground truth really was.

Mezhoudi and colleagues [10] took stock of the ML techniques used across employability prediction and flagged ongoing struggles around feature selection and data quality. The review made the case for broader solutions but didn't go as far as proposing one unified system that ties assessment, prediction, feedback, and skill-building together.

Put side by side, this body of work mostly tackles one job at a time prediction, interview scoring, or personalized learning, rarely all three. What's proposed here goes a different way, folding assessment, prediction, skill-gap analysis, and learning recommendations into one web-based system that stays with the student through the whole placement journey.

III. PROPOSED METHODOLOGY

This framework ties aptitude checks, technical evaluation, HR interview simulation, placement prediction, and personalised guidance into one platform. A Decision Tree classifier decides Placement Ready versus Needs Improvement, and rule-based NLP works through technical and HR answers to build feedback that actually means something. From there, the weaker spots get flagged and a learning roadmap gets built around them. Figure 1 lays out how it all fits together.

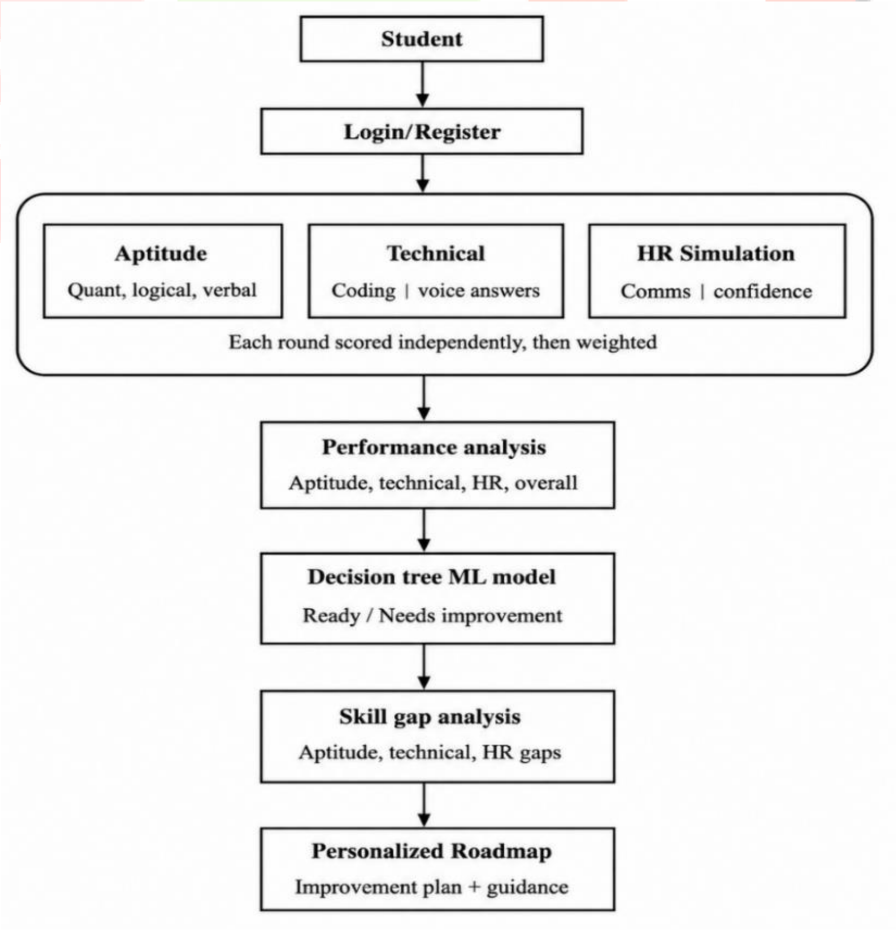

Figure 1: System-Architecture of the-Proposed Placement Intelligence with PrepEdge AI Framework

3.1 Student Assessment Module

The student assessment module runs a student through three separate checks Aptitude, Technical, and HR Interview Simulation. Aptitude questions grade themselves instantly, while technical and HR answers get run through rule-based NLP using TF-IDF cosine similarity alongside keyword checks. Every score from all three lands in storage, ready for the performance-analysis and prediction stages down the line.

3.1.1 Aptitude Assessment

A round of objective questions makes up the aptitude check, covering quantitative reasoning, logic, and verbal skill. Each answer gets matched against the correct one and the score comes out automatically and that number carries real weight later, feeding straight into the overall performance read and the readiness call.

3.1.2 Technical Assessment

Technical evaluation covers both theory and hands-on coding sense. Students speak their answers, which speech recognition turns into text, and rule-based NLP then measures how close that text lands to the expected answer using TF-IDF cosine similarity plus keyword checks. Coding questions sit alongside this to test actual programming chops.

3.1.3 HR Interview Simulation

HR interview simulation puts students through commonly-asked questions in a setting built to feel real. Just like the technical round, spoken answers get captured and converted to text, and rule-based NLP scans them for communication-relevant keywords before returning feedback on quality practice that, over time, builds both skill and confidence for the real thing.

3.2 Performance Analysis

Performance analysis takes the three score sets aptitude, technical, HR and merges them into one full profile, calling out what's strong and what still needs work, and giving a bird's-eye view of readiness. That combined picture then flows straight into both prediction and the skill-gap step.

3.3 Placement Prediction

Placement prediction runs on a Decision Tree trained across student assessment data. Feeding in the aptitude, technical, and HR numbers, it sorts each student into Ready or Needs Improvement giving them a straight answer on where they stand before placements actually begin.

3.4 Skill Gap Analysis

Skill-gap analysis works by comparing performance across aptitude, technical, and HR scores to spot where a student is falling short. It turns those scores into targeted feedback, pointing out exactly what needs more practice, and helps students put their effort where it counts instead of spreading it thin.

3.5 Personalized Learning Roadmap

Combining the skill gaps found earlier with the Decision Tree's readiness call, the platform puts together a learning roadmap from a set of predefined rules resources, exercises, and strategies matched to how the student actually performed, aimed at lifting the weak spots and boosting overall readiness.

IV. RESULTS AND DISCUSSION

4.1 Model Performance

Checking the Decision Tree's output meant running it against the standard classification metrics — Accuracy, Precision, Recall, F1-score, ROC-AUC picked specifically to see how well it separates Ready students from those who Need Improvement, working off aptitude, technical, HR, and CGPA data. Across the board, the numbers stayed consistent. Accuracy shows overall correctness; Precision and Recall speak to how trustworthy and complete the calls were; F1-score balances the two; and ROC-AUC shows how cleanly the model splits the two outcome groups. Table 1 lists all of it out.

Table 1. Model Performance

Metric	Formula	Decision Tree (%)
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	92.00
Precision	$\frac{TP}{TP+FP}$	93.48
Recall	$\frac{TP}{TP+FN}$	89.58
F1	$\frac{2\times(Precision\times Recall)}{Precision+Recall}$	91.49
ROC-AUC	$\int_0^1 TPR(FPR)d(FPR)$	94.56

4.2 Decision Tree Visualization

Technical Score, HR Score, Aptitude Score, and CGPA all feed the Decision Tree that decides Ready versus Needs Improvement. Technical Score landed as the root node — it carried the most information gain during training — and everything else gets filtered through the learned rules from there to reach a final call. Because every step of that logic is visible, reading the model isn't hard. Figure 2 shows the tree as trained.

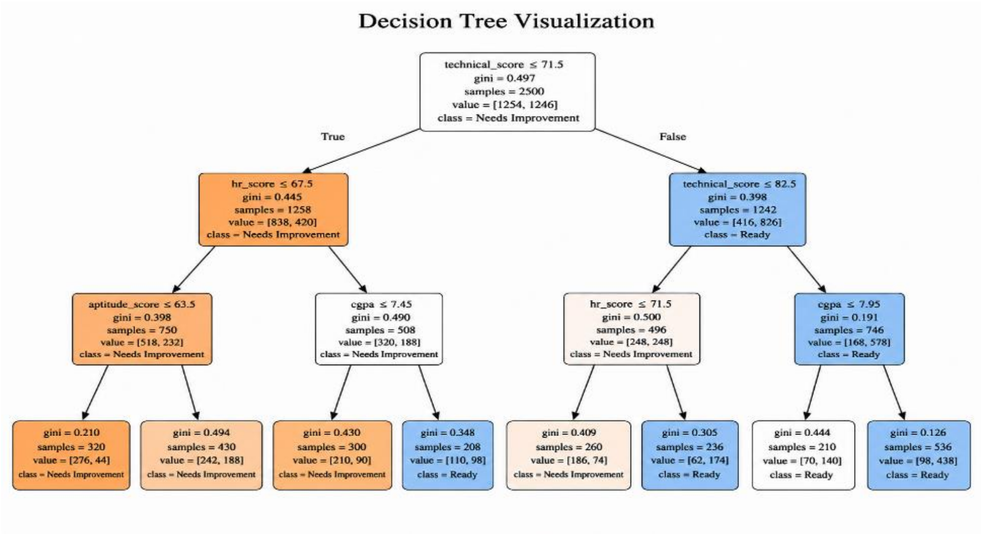

Figure 2: Decision Tree Visualization for Placement Readiness Prediction

4.3 Confusion Matrix

Figure 3 breaks down where the Decision Tree's predictions landed versus each student's real outcome. Out of the whole test set, 245 Needs Improvement cases and 215 Placement Ready cases were caught correctly, leaving just 15 false positives and 25 false negatives on the board. Put together, that comes to a 92.00% accuracy rate, and neither class was predicted noticeably worse than the other.

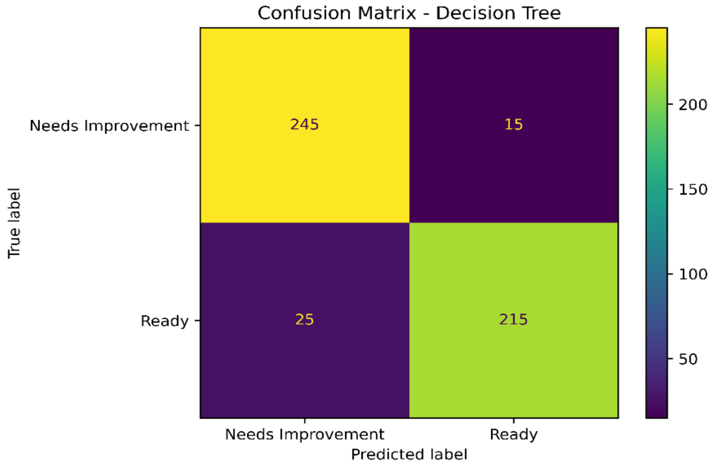

Figure 3: Confusion Matrix of the Proposed-Decision Tree Model

4.4 ROC Curve Analysis

To see how the model held up across different thresholds, a Receiver Operating Characteristic (ROC) curve was plotted, tracking the trade-off between True Positive Rate and False Positive Rate as those thresholds shifted. The ROC-AUC number sums up how well it separates the two classes overall, and Figure 4 shows that value landing high — a sign the model reliably tells Ready students apart from those who Need Improvement, and further proof the whole approach holds up.

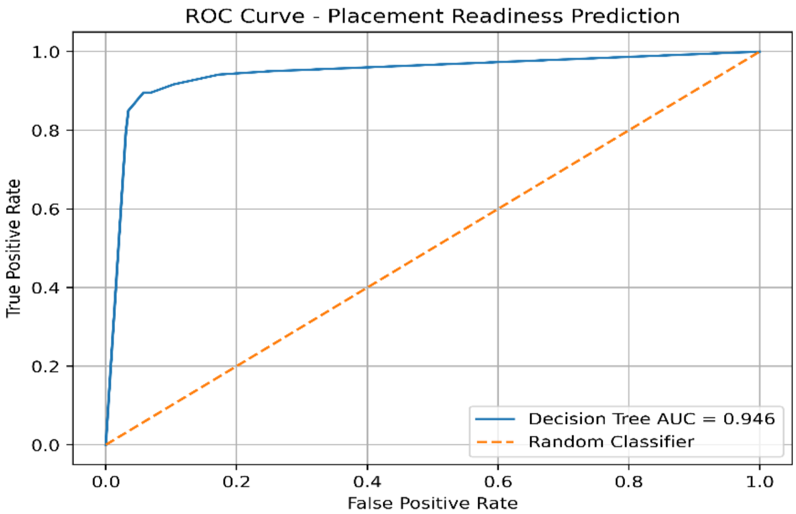

Figure 4: ROC Curve of the Decision Tree Model

4.5 Placement Prediction Results

Once trained, the Decision Tree went straight into the web app to handle live placement calls. A student finishes the aptitude, technical, and HR rounds, the scores get crunched, and out comes a Ready or Needs Improvement verdict. From there the system flags weak spots and builds out recommendations through the Skill Gap Analysis and Personalized Learning Roadmap pieces — giving students a straight read on where they stand and what to work on before recruitment season. Figure 5 shows what that output looks like.

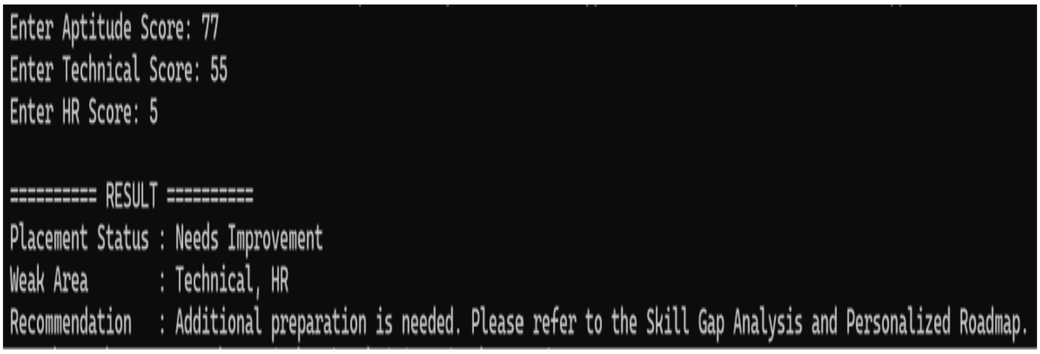

Figure 5: Placement Prediction Output

V. CONCLUSION

The goal behind this framework was a single platform that walks students through the whole of campus placement prep aptitude checks, technical evaluation, HR interview practice, readiness prediction, and guidance built around the individual. A Decision Tree handles the readiness call from assessment scores, while rule-based NLP turns technical and HR answers into usable feedback, and the system rounds it out by flagging skill gaps and mapping a roadmap to close them. Built on Python, Django, SQLite, HTML, CSS, and JavaScript, it's a practical, easy-to-run tool, and the results back up its reliability useful both for students chasing readiness and institutions trying to lift employability numbers.

VI. FUTURE ENHANCEMENTS

Looking ahead, the framework can be extended further through the following additions:

- **Human-like AI Interviewer:** Build an AI interviewer that can ask dynamic follow-up questions during technical and HR interviews, making the overall experience feel closer to a real interview.
- **AI-Based Resume Analysis:** Add a resume evaluation module that checks ATS compatibility along with skills, projects, internships, and certifications, and suggests relevant improvements.
- **Emotion and Facial Expression Analysis:** Apply computer vision to analyse facial expressions, eye contact, and confidence levels during mock interviews for a more well-rounded evaluation.
- **Explainable AI (XAI):** Bring in XAI techniques to clearly explain the factors behind each placement prediction, improving transparency and helping users understand the results.

REFERENCES

[1] D. K. Dake, “Information technology graduates employability prediction model in a low-income country using tree-based machine learning classifiers,” Discover Global Society, vol. 3, art. 158, 2025, doi: 10.1007/s44282-025-00304-3.

[2] O. Saidani, L. J. Menzli, A. Ksibi, N. Alturki, and A. S. Alluhaidan, “Predicting student employability through the internship context using gradient boosting models,” IEEE Access, vol. 10, pp. 47513–47527, 2022, doi: 10.1109/ACCESS.2022.3170421.

[3] P. Aruna, V. S. Abinayashri, K. S. Abinayaa Sree, and A. Meghashri, “Student Placement Prediction Using Machine Learning Algorithms,” in Proc. Innovations in Power and Advanced Computing Technologies (i-PACT), 2023.

[4] S. Wu, Y. Cao, J. Cui, R. Li, H. Qian, B. Jiang, and W. Zhang, “A Comprehensive Exploration of Personalized Learning in Smart Education: From Student Modeling to Personalized Recommendations,” arXiv:2402.01666, 2024.

[5] R. Kadu, P. J. Assudani, T. Mukewar, J. Kapgate, and R. Bijekar, "Student Placement Prediction and Skill Recommendation System using Machine Learning Algorithms," in Proc. International Conference on Inventive Computation Technologies (ICICT), 2024.

[6] P. J. Uppalapati, M. Dabbiru, and V. R. Kasukurthi, "AI-driven mock interview assessment: leveraging generative language models for automated evaluation," International Journal of Machine Learning and Cybernetics, vol. 16, pp. 10057–10079, 2025, doi: 10.1007/s13042-025-02529-9.

[7] J. M. Bershika and G. Nancy, "AI-Based Virtual Interviewer System Using NLP and Emotion Detection," in Sustainable Computing and Intelligent Systems, Lecture Notes in Networks and Systems, vol. 1929, Springer, 2026, pp. 69–85, doi: 10.1007/978-3-032-22911-3_6.

[8] V. Matzavela and E. Alepis, "Decision tree learning through a predictive model for student academic performance in intelligent m-learning environments," Computers and Education: Artificial Intelligence, vol. 2, art. 100035, 2021, doi: 10.1016/j.caeai.2021.100035.

[9] G. ElSharkawy, Y. Helmy, and E. Yehia, "Employability Prediction of Information Technology Graduates using Machine Learning Algorithms," International Journal of Advanced Computer Science and Applications, vol. 13, no. 10, pp. 359–367, 2022, doi: 10.14569/IJACSA.2022.0131043.

[10] N. Mezhoudi, R. Alghamdi, R. Alharbi, F. Alsubait, and M. W. Almutairi, "Employability prediction: a survey of current approaches, research challenges and applications," Journal of Ambient Intelligence and Humanized Computing, 2021, doi: 10.1007/s12652-021-03276-9.

[11] K. Rai, P. Kumar, and S. Sood, "Anticipating Placement Status of Engineering Students Using Machine Learning Based Prediction Models – A Case Study of Computer Science Stream," Proc. ICIDCA, 2023, pp. 181–185, doi: 10.1109/ICIDCA56705.2023.10100042.

[12] "Student Placement Prediction Using Supervised Machine Learning," in Proceedings of the International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN), IEEE, 2022.

[13] I. O. Muraina, M. A. Adebayo, and A. O. Adeyemi, "Application of Decision Tree Algorithm for Predicting Students' Academic Performance in an Advanced Programming Course," International Journal of Human and Education Practice, vol. 5, no. 8, pp. 35–45, 2022.

[14] A. M. Hajjem and T. El-Diraby, "Comparative Study of Regressor and Classifier with Decision Tree Using Modern Tools," Procedia Computer Science, vol. 192, pp. 3658–3667, 2022.

[15] C. Kim, J. Choi, J. Yoon, D. Yoo, and W. Lee, "Fairness-Aware Multimodal Learning for Automatic Video Interview Assessment," IEEE Access, vol. 11, pp. 118364–118377, 2023, doi: 10.1109/ACCESS.2023.3325891.