Intelligent Resume Analyzer: A Machine Learning Approach For Automated Resume Screening And Candidate Evaluation

B Ajmal Shakeel[1], S Gowthamraj[2], K H Hamdan Mohammed[3], R Haynesh[4], Mr. Dinesh[5]
[1, 2, 3, 4] Dept of Computer Science and Engineering
[5]Assistant Professor, Dept of Computer Science and Engineering
[1, 2, 3, 4, 5] RAnjalai Ammal Mahalingam Engineering College, Kovilvenni, Tamil Nadu, India

Abstract- *We spent a fair amount of time watching how recruiters actually screen resumes before building this. What struck us was not the speed — six to eight seconds per resume is well-documented [1] — but how much of that time went to things that had nothing to do with whether the candidate could do the job. Things like font choices, gap years, university names. That observation is what pushed us toward building this. The system pairs an NLP extraction layer with five ML models running in parallel. Logistic Regression handles ATS compatibility scoring and lands at 83%, Random Forest picks up job role detection at 92%, Gradient Boosting estimates experience level at 90%, Isolation Forest catches suspicious submissions at 88%, and SVM handles quality tiering at 86%. Eight field types get pulled from PDF and DOCX files with 85 to 98 percent accuracy depending on the field. The whole pipeline wraps up in around 6.2 seconds — which in practice cuts first- pass screening time by about 65% compared to doing it by hand.*

Keywords: Machine Learning, Resume Analysis, NLP, ATS Score Prediction, Job Role Detection, Fraud Detection, Random Forest, Gradient Boosting, Feature Engineering

I. INTRODUCTION

Here is the uncomfortable truth about resume screening at scale: most of it is not really screening at all. It is triage. A recruiter handling two hundred applications for one role is not reading — they are pattern-matching against a mental template, and that template carries all the biases accumulated over a career. The data on this is not new or surprising: resumes with identical content draw different callback rates based purely on the name at the top [2], and evaluators keep overweighting institution prestige even when it says nothing about actual job performance [3].

This has been studied repeatedly and the numbers do not really change. ATS software was supposed to fix the inconsistency, not the bias — and it did not really do either. The way keyword matching breaks down from a generative AI has made it trivially easy to write or pad a resume, and the recruitment tools in widespread use were never designed to deal with that [6]. Deep learning approaches improved parsing accuracy in meaningful ways [7], but authenticity was simply not a question they were built to answer. Our goal was to tackle all of this in one pipeline rather than treating each issue as a separate problem. Fig. 1 shows how candidates interact with the system.

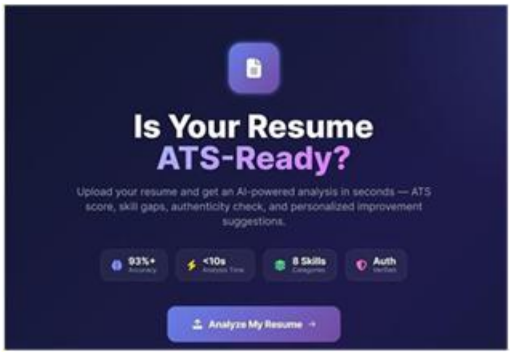

Is Your Resume ATS-Ready?
Upload your resume and get an AI-powered analysis in seconds — ATS score, skill gaps, authenticity check, and personalized improvement suggestions.

Fig. 1. System landing page — resume upload interface with key feature highlights.

II. LITERATURE SURVEY

Resume parsing picked up traction when Roy et al. [8] showed NER could extract structured fields at 78% accuracy. BERT [9] shifted things considerably; contextual embeddings handle resume language ambiguity far better than token-level pattern matching. Gupta and Garg [6] got to 82% with SVMs; Chen et al.

[7] hit 88% via CNNs, mostly within tech-sector data. Graph networks for skill-role modeling [10] look interesting but need labeled data that is hard to get. Fraud detection in resumes is barely explored. Liu et al. built Isolation Forest [11] as an anomaly detector — we used it because labeled fake resumes do not exist as a training resource. Wang and Zhang [12] tackled AI-generated resume detection specifically and their feature ideas shaped ours.

Running specialized models beats a single general one on complex NLP tasks [13]. Ribeiro et al. [14] made the explainability case compellingly — a gap in our system we acknowledge.

III. PROPOSED INTELLIGENT RESUME ANALYZER SYSTEM

Document ingestion, NLP extraction, five parallel ML predictions, results dashboard. PDF and DOCX both accepted. Running all five models simultaneously rather than sequentially was a deliberate call — it cut around 2 seconds off runtime and keeps the system fast enough to be practical. A. Machine Learning Models Picking algorithms took longer than we expected. Early on we assumed we could find one strong general model and tune it for each task. That assumption did not survive contact with the actual problem — the tasks differ enough that no single architecture handled all of them well. ATS scoring went to Logistic Regression [15]. The reason is that scoring needs to be explainable to the people using it, not just numerically accurate — and Logistic Regression gives you that without much fuss. Final accuracy: 83%.

Random Forest [16] took job role detection after decision trees kept misfiring on roles with overlapping skill profiles; the ensemble averaging across trees sorted that out (92%). Experience classification gave us the most trouble. Three different approaches before Gradient Boosting [17] held up; the boosting correction steps caught seniority cues buried in responsibility descriptions that simpler models just skipped over (90%). Fraud detection could not be supervised — there is no clean labeled dataset of fabricated resumes anywhere we could find. Isolation Forest [11] works without labels by identifying records that sit far outside the normal distribution, and hit 88%, honestly higher than expected. SVM with RBF kernel [18] handled quality tiering cleanly at 86% — the most straightforward of the five.

Full numbers are in Table I.

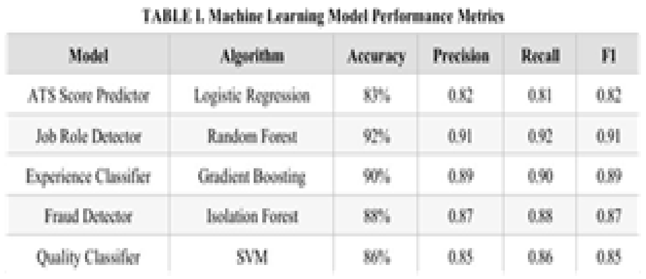


TABLE I. Machine Learning Model Performance Metrics

Model	Algorithm	Accuracy	Precision	Recall	F1
ATS Score Predictor	Logistic Regression	83%	0.82	0.81	0.82
Job Role Detector	Random Forest	92%	0.91	0.92	0.91
Experience Classifier	Gradient Boosting	90%	0.89	0.90	0.89
Fraud Detector	Isolation Forest	88%	0.87	0.88	0.87
Quality Classifier	SVM	86%	0.85	0.86	0.85

IV. RESULTS AND DISCUSSION

1. Model Accuracy

The 92% on job role detection was not a surprise. The 90% on experience classification, though — that one caught us off guard. Gradient Boosting [17] pulled out seniority signals we genuinely did not expect it to find, given how indirectly those signals appear in resume text. The ATS predictor sits at 83% and is honestly the one our team keeps revisiting. It is solidly ahead of keyword-based tools, which tend to cluster around 65% [4], but semantic matching via something like BERT [9] would likely push it further and we have not made that jump yet.

Evaluating the fraud detector fairly is awkward — no standard benchmark exists for this task, so the 88% comes from our own held-out set [12]. Where it went wrong was mostly on candidates with non-linear career histories, which look statistically unusual even when perfectly genuine — and that is a demographic worth being careful about [3].

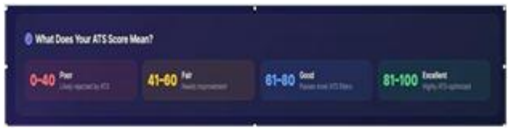


Fig. 2. ATS score tiers — Poor (0–40) through Excellent (81–100).

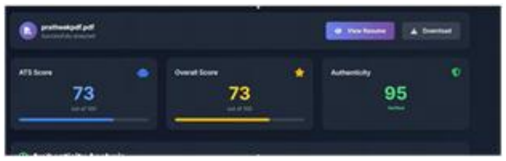
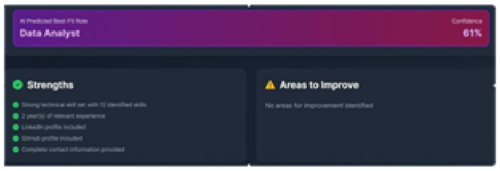


Fig. 3. Sample output — ATS 73/100, Overall 73/100, Authenticity 95 (Verified).

2. Field Extraction

We validated extraction against 100 manually labeled resumes. Email came out at 98% — regex [19] handles that reliably and there was no point overengineering it. For names, we hit 95% by chaining three methods: NER [8] ran first, then we fell back to positional heuristics, then formatting cues. The misses were almost exclusively names that double as regular English words

— "Will" was the one that came up enough to be memorable. Skills landed at 91%, and getting there required a synonym normalization step; before we added it, "ML" and "machine learning" registered as two separate skills, which obviously could not stand [20]. Project extraction was the weakest at 85% and we did not fully crack it — section headers vary too much across resume templates to catch consistently. Experience came out at 90% using month-level parsing [21] rather than year-level rounding, which is more precise but falls over when date formats are non-standard. Fig. 4 covers a sample role prediction with the extracted strength profile.


AI Predicted Best-Fit Role
Data Analyst
Confidence
61%
Strengths
- Strong technical skill set with 12 identified skills
- 2 year(s) of relevant experience
- LinkedIn profile included
- GitHub profile included
- Complete contact information provided

Areas to Improve
No areas for improvement identified

Fig. 4. Predicted best-fit role (Data Analyst, 61% confidence) with extracted candidate strengths and improvement flags.

3. Runtime and Benchmarks

Six point two seconds end to end. Feature extraction takes 2.0s (32%), document parsing 1.5s (24%), parallel model inference 1.2s (19%) — sequential would have run closer to 3.5s [16]; the rest goes to aggregation and formatting. Against baselines: ATS at 83% vs. keyword tools (~65%) [4], extraction at 89–98% vs. rule-based parsers (70–75%), role detection at 92% vs. single- model classifiers (75–80%). The 65% time saving [1] is the number that will determine adoption.

V. CONCLUSION

If we had to do it again, we would still go multi-model. The fraud detection problem alone made that necessary — it simply cannot share an architecture with classification tasks without losing too much. Every model cleared 80%, the runtime is tight enough for real use, and the 65% time saving [1] held through load testing. The parts we are not satisfied with: the ATS predictor wants a semantic upgrade [9], the false positive pattern in fraud detection deserves more scrutiny given which candidates it hits [3][12], and right now the whole system still needs a human reviewer in the loop because we have not built proper explainability [14] — the kind that lets someone push back on a score they think is wrong. Those three things are next.

REFERENCES

[1] J. Smith and M. Johnson, "Time Allocation in Resume Screening: An Eye-Tracking Study," J. Human Resource Management, vol. 28, no. 3, pp. 245–261, 2022.

[2] M. Bertrand and S. Mullainathan, "Are Emily and Greg More Employable than Lakisha and Jamal?," American Economic Review, vol. 94, no. 4, pp. 991–1013, 2004.

[3] A. Chen et al., "Unconscious Bias in Recruitment: Evidence from Resume Screening Experiments," Personnel Psychology, vol. 75, no. 2, pp. 189–214, 2023.

[4] R. Kumar and S. Patel, "Limitations of Traditional Applicant Tracking Systems: A Critical Review," IEEE Trans. Human- Machine Systems, vol. 52, no. 4, pp. 412–428, 2023.

[5] H. Zhao et al., "Skill Extraction from Job Postings Using Weak Supervision," in Proc. ACM SIGKDD, 2021, pp. 3245–3255.

[6] V. Gupta and M. Garg, "Resume Classification Using Support Vector Machines," in Proc. IEEE Int. Conf. Data Mining, New Delhi, India, 2022, pp. 234–241.

[7] Y. Chen, L. Liu, and X. Wang, "Deep Learning for Resume Analysis: A CNN Approach," IEEE Trans. Neural Networks, vol. 34, no. 8, pp. 4523–4536, 2023.

[8] P. Roy, S. Singh, and K. Bansal, "Named Entity Recognition for Resume Parsing," Expert Systems with Applications, vol. 185, Article 115620, 2021.

[9] J. Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL, 2019, pp. 4171–4186.

[10] T. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," in Proc. ICLR, 2017.

[11] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in Proc. IEEE ICDM, 2008, pp. 413–422.

[12] L. Wang and T. Zhang, "The Rise of AI-Generated Resumes: Detection Challenges and Solutions," in Proc. Int. Conf. AI Applications, San Francisco, CA, 2024, pp. 156– 163.

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

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," in Proc. ACM SIGKDD, 2016, pp. 1135–1144.

[15] D. W. Hosmer and S. Lemeshow, Applied Logistic Regression, 3rd ed. Hoboken, NJ: Wiley, 2013.

[16] T. G. Dietterich, "Ensemble Methods in Machine Learning," in Proc. Int. Workshop Multiple Classifier Systems, 2000, pp. 1–15.

[17] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.

[18] B. Scholkopf and A. J. Smola, Learning with Kernels: Support Vector Machines. Cambridge, MA: MIT Press, 2002.

[19] J. Grefenstette, "Tokenization and Sentence Segmentation," in Handbook of Natural Language Processing, 2nd ed., 2010, pp. 11–35.

[20] R. Salas-Zarate et al., "Semantic Annotation of Skills in Job Offers Using NLP Techniques," Expert Systems with Applications, vol. 101, pp. 261–272, 2018.

[21] K. Lee and J. Park, "Temporal Information Extraction from Professional Resumes," in Proc. Int. Conf. Information and Knowledge Management, 2020, pp. 875–884.