Spot the Scam: AI Job Fraud Detection System

David Nguyen
University of North Carolina at Chapel Hill
snghoang@unc.edu

David Pu
University of North Carolina at Chapel Hill
davidpu@unc.edu

Alice Nicole Karakachian
University of North Carolina at Chapel Hill
akaraka@unc.edu

Sai Ananya Channamsetti
University of North Carolina at Chapel Hill
csaiana@unc.edu

Lakshmi Gayathri Divakaruni
University of North Carolina at Chapel Hill
dlakshmi@unc.edu

Abstract

In our interconnected world, job seekers can find more opportunities online. While this increases the efficiency with which workers and employers can find each other, online platforms also increase the risk of fraud and monetary exploitation. Numerous algorithms have been designed to mitigate the impact of fraudulent job postings on vulnerable job seekers. Our approach combines classical machine learning pipeline with modern transformer-deep learning pipeline. Our strategy allows us to leverage the strengths of both paradigms and select the best-performing model for product deployment.

1 Introduction

In the digital age, employers and job seekers have increased opportunities to interface on various popular online platforms. While technology has increased the ease with which information about employment opportunities can be circulated, there has also been an increased risk of fraud. Fraudulent online job postings are often circulated with ill-intent: to exploit the vulnerabilities of desperate job-seekers and steal personal information or money through illegitimate up-front charges [1]. Neural networks, NLP, SVM, random forest, decision trees, N-Gram and TF-IDF feature extraction methods have previously been employed to analyze job descriptions and recruiting company information to accurately flag fake job ads [4, 5, 3]. These approaches have shown promising results, achieving high accuracy, but certain ambiguous or poorly formatted job descriptions still present a challenge [4].

To improve the accuracy of the identification of fraudulent job postings, and to enhance the user experience, we developed ***Spot The Scam***, a full-stack application that applies both classical and transformer models to this modern problem.

2 Methods

The Spot the Scam system uses a dual-track architecture that trains a classical machine learning pipeline alongside a transformer-based deep learning pipeline. Specifically, logistic regression, linear SVM, LightGBM with config-driven grids, and DistilBERT fine-tuned with HF Trainer (AMP and early stopping) are used.

Logistic regression is a common supervised machine learning algorithm used in various classification tasks. This approach is particularly preferred for binary classification tasks, as a fraudulent job detector requires. Previous research has shown the promise of using logistic regression for this specific task, yielding high accuracy results of more than 98% [2]. Our project builds on this previous work, and logistic regression is used as the baseline model to compare with the performance of other classical and transformer models.

We used the two Kaggle CSV datasets "Real/Fake Job Posting Prediction" and "Fraudulent Job Posting". Links to the datasets are provided in the Appendix A. First, the datasets were ingested, de-duplicated with text hashing that removes roughly 32% of unique samples, cleaned by stripping HTML and URL artifacts, normalized for whitespace, and imputed with a "<missing>" token for absent fields. All textual attributes are concatenated into a unified "text_all" field. The corpus was then partitioned with a stratified split into training (70%, ~12,516 samples), validation (15%, ~2,682 samples), and testing (15%, ~2,682 samples). The test partition remains untouched until final model selection to preserve an unbiased performance estimate.

In the classical modeling track, feature vectors combine TF-IDF text representations with normalized tabular signals, including message length, uppercase ratio, currency and URL counts, scam-term indicators, and metadata such as company-logo presence and telecommuting flags. The LightGBM variants would consume only the tabular block, while other classical models use TF-IDF plus tabular. Training follows an exhaustive grid, with L2-regularized logistic regression with class-balanced weights over $C \in \{0.1,\ 1.0,\ 10.0\}$, class-balanced linear SVM over the same C grid with decision scores mapped to probabilities via sigmoid calibration when needed, and a 32-combination LightGBM sweep over num_leaves, learning_rate, n_estimators, subsample, and colsample_bytree. An additional XGBoost sweep runs a capped set of calibrated variants over depth, learning rate, estimators, regularization, and scale_pos_weight to capture the nonlinear structures. Altogether, 38 classical configurations would complete in roughly two to three hours. Each candidate is evaluated on the validation split, paired with both Platt (sigmoid) and isotonic calibration, and retains the calibration variant with the lowest validation Brier or expected calibration error. Decision thresholds are then tuned on the validation set to maximize the F1, which helps address class imbalance without touching the test set.

The transformer track fine-tunes a DistilBERT sequence classifier on the concatenated text_all field. Specifically, sequences are limited to 128 tokens, with batch size being 16, learning rate being 3×10^{-5}, weight decay being 0.01, warmup ratio being 0.1, and gradient accumulation of 1. Training runs for up to three epochs with early stopping on validation loss, using mixed precision when the platform supports it and falling back to full precision otherwise. This configuration adapts the pre-trained language model to fraud detection while controlling overfitting through short runs and validation-based stopping.

After both tracks are complete, all calibrated candidates are ranked by their validation F1. The top model, whether classical, XGBoost-based, or transformer, is evaluated once on the held-out test set to obtain an unbiased performance estimate. The selected model's artifacts, including weights, calibration method, and the validation-optimized threshold, are exported for production deployment so the live system reflects the best-performing configuration observed during validation and confirmed on the test partition.

The system provides a chatbot experience via an external large language model rather than training an LLM in-house. It invokes the Gemini API with a structured prompt that includes a fixed system role (explaining how it should act as a fraud-detection assistant), the running conversation history, the user's submitted job posting fields, and the trained model's outputs (probability, decision label, threshold and gray-zone policy, and the top explanatory features). This context-engineering and prompt-engineering strategy keeps the fraud-detection models purpose-built and lightweight while offloading open-ended, conversational explanation and summarization to the managed LLM service, which ensures that the chatbot produces high-quality and accurate responses.

3 Results

The final model is a classical ensemble (`ensemble_top3`) using TF–IDF and tabular features, selected over transformer-based baselines by its superior validation F1. Overall, the model exhibits strong discrimination and ranking on both validation and test splits, with only modest degradation on the held-out test set consistent with expected generalization.

Table 1: Performance of `ensemble_top3` on validation and test splits.

Split	F1	Precision	Recall	ROC AUC	PR AUC	Brier
Validation	0.8561	0.9297	0.7933	0.9890	0.9053	0.0103
Test	0.7721	0.8537	0.7047	0.9863	0.8659	0.0143

As summarized in Table 1, the ensemble delivers strong performance on the validation split and maintains robust metrics on the unseen test split, with Brier scores indicating probabilistic predictions that are suitable for downstream decision-making.

Validation-time threshold optimization selected an operating point at a decision threshold of approximately 0.5802 with a gray-zone width of 0.10, improving F1 relative to the default threshold of 0.5 while preserving a balanced precision–recall trade-off. Decision quality on the test set is illustrated by the confusion matrix and precision–recall curve in Figures 1 and 2, which together show effective fraud recall at tolerable false-positive rates for the intended application.

Calibration diagnostics in Figure 3 show that predicted probabilities closely track observed frequencies, with a low expected calibration error on the test set (ECE ≈ 0.0066). The probability-versus-text-length plot in Figure 4 shows no systematic dependence of fraud scores on document length, indicating that the TF–IDF plus tabular stack remains stable across typical posting sizes. The latency and throughput benchmark in Figure 5 demonstrates end-to-end latency in the tens of milliseconds at small batch sizes, while throughput scales to the hundreds of requests per second at larger batches before tail latency becomes the primary constraint, satisfying interactive inference requirements and supporting higher-throughput batch scoring when needed.

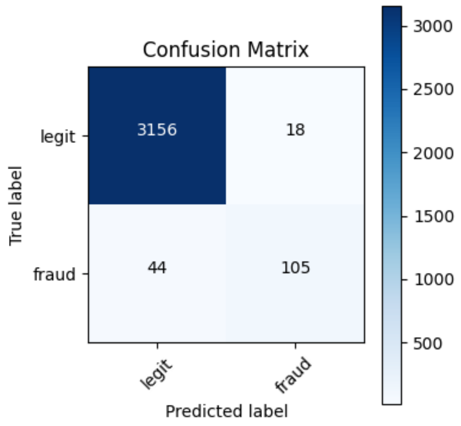

Figure 1: Confusion matrix on the test split at the selected decision threshold.

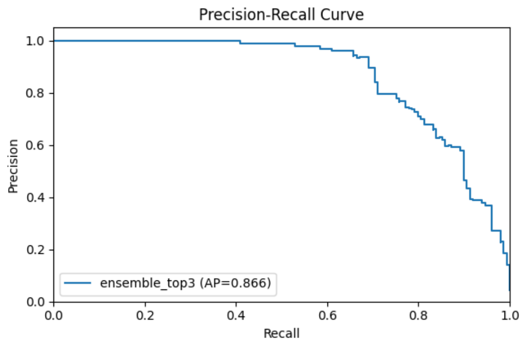

Figure 2: Precision–recall curve for `ensemble_top3` on the test split.

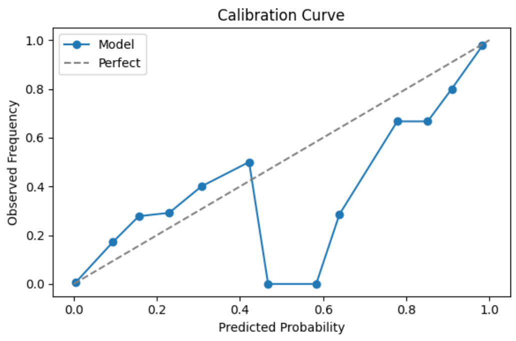

Figure 3: Calibration curve on the test split.

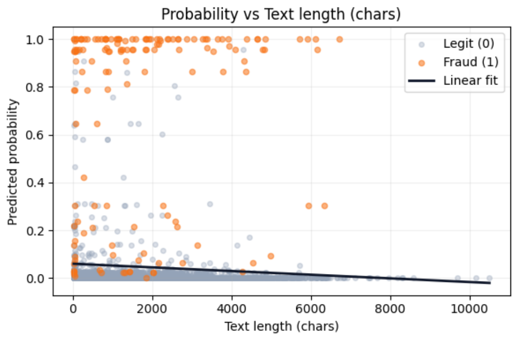

Figure 4: Predicted fraud probability as a function of text length on the test split.

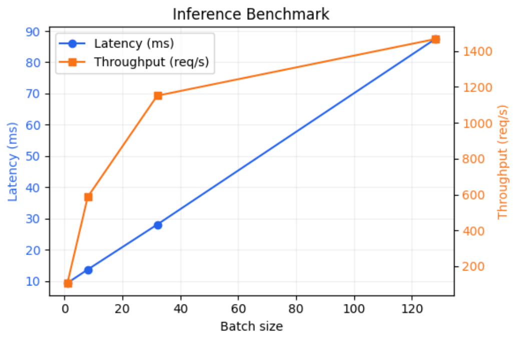

Figure 5: Inference latency and throughput as a function of batch size.

4 Conclusion

Our application, ***Spot The Scam*** aims to reduce the impact of fraudulent job postings on job seekers by implementing a traditional machine learning pipeline using transformer-based deep learning techniques. Our results showed an F1 score of 85.4% and precision of 91.6%. To further improve the model, it could be useful to add continuous data-quality and drift monitoring for text/tabular features and predicted probabilities. Additionally, one could add a "lightweight watchdog" that continuously measures input drift, score drift, and calibration error from recent traffic, raises alerts when thresholds are breached, and produces a small “what to retrain with” bundle. It is important to continue to improve applications to ensure robust fraud detection systems to protect job seekers online.

References

[1] Sultana Umme Habiba, Md Khairul Islam, and Farzana Tasnim. A comparative study on fake job post prediction using different data mining techniques. In *2021 2nd international conference on robotics, electrical and signal processing techniques (ICREST)*, pages 543–546. IEEE, 2021.

[2] Vijay Madaan, Neha Sharma, Raghubeer Singh Bangari, and Srinivas Aluvala. Fraudulent job posting detection using logistic regression. In *2024 International Conference on Information Science and Communications Technologies (ICISCT)*, pages 1–6. IEEE, 2024.

[3] Gayathri Malaichamy, Cristina Hava Muntean, and Anderson Augusto Simiscuka. Online job posting authenticity prediction with machine and deep learning: Performance comparison between n-gram and tf-idf. In *International Conference on Deep Learning Theory and Applications*, pages 143–162. Springer, 2024.

[4] K Ramakrishna Reddy, G Indrani, N Pavan Kumar, and K Vamshi Krishna. Fake job posting detection using machine learning algorithms. In *2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pages 822–827. IEEE, 2025.

[5] K. Suparna and R. Anil Kumar. A comparative study on fake job post prediction using different datamining techniques. *International Journal of Information Technology and Computer Engineering*, 12(3):317–323, 2024.

5 Appendix

Dataset 1: Real / Fake Job Posting Prediction `https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction`

Dataset 2: Fraudulent Job Posting `https://www.kaggle.com/datasets/subhajournal/fraudulent-job-posting`

Github Repository `https://github.com/hoangsonww/Spot-the-Scam-AI-Job-Fraud.git`