International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

Fake Job Posting Detection using Open-Source Large Language Models with Heuristic Pre-Screening: A Hybrid Approach

Syed Akbar
QA & Automation Engineer, Linnk Outsource Solutions India Pvt. Ltd., Kochi, Kerala, India

***Abstract* - The proliferation of fraudulent job postings on online recruitment platforms poses a significant threat to job seekers, leading to financial loss, identity theft, and psychological distress. This paper presents a hybrid fake job posting detection system that combines a rule-based heuristic pre-screening engine with an open-source Large Language Model (LLM) running locally via the Ollama inference framework. The heuristic engine analyses structural and linguistic red flags across 15+ detection rules, scoring each posting on a 100-point risk scale. The LLM layer — powered by models such as Phi-3 and Mistral — then performs semantic reasoning on flagged content to generate a structured verdict (REAL, SUSPICIOUS, or FAKE) along with confidence scores, key reasoning, and actionable recommendations. The system is deployed as an interactive web application using the Streamlit framework, incorporating batch analysis, history tracking, and CSV export capabilities. Experimental evaluation on a curated dataset comprising genuine and fraudulent job descriptions — including data-entry scams, mystery shopper fraud, and multi-level marketing schemes — demonstrated high detection accuracy with zero dependency on commercial APIs or cloud-based inference, making the solution cost-effective and privacy-preserving. The proposed architecture serves as a practical reference for privacy-conscious AI deployment in HR-technology and cyber-security contexts.**

***Keywords* — Fake Job Detection, Large Language Models, Ollama, Heuristic Analysis, Phi-3, Streamlit, NLP, Fraud Detection, Recruitment Fraud, Open-Source AI**

I. INTRODUCTION

The digital transformation of the recruitment landscape has brought unprecedented convenience for both employers and job seekers. However, this accessibility has simultaneously created fertile ground for fraudulent actors who exploit online job portals to deceive vulnerable individuals. Fake job postings — advertisements crafted with malicious intent — are designed to extract personal information, solicit registration fees, or enroll victims in pyramid schemes and money-mule operations.

According to the Internet Crime Complaint Center (IC3), employment fraud constitutes one of the fastest-growing cybercrime categories, with thousands of incidents reported annually across South Asia, the United States, and the United Kingdom. In India, the National Cyber Crime Reporting Portal has registered a sharp uptick in work-from-home and data-entry fraud cases, many of which originate from convincingly crafted fake job advertisements.

Traditional detection approaches rely on keyword blacklists, domain-based filtering, or supervised machine learning classifiers trained on historical data. While effective in constrained settings, these methods suffer from brittleness in the face of evolving adversarial tactics, poor explainability, and high computational overhead when deployed at scale with proprietary cloud APIs.

This paper addresses these shortcomings by proposing a two-stage hybrid detection architecture. The first stage employs a lightweight, interpretable heuristic engine that flags structural anomalies and linguistic red flags in job postings. The second stage invokes an open-source LLM — specifically Phi-3 or Mistral, served locally via Ollama — to perform contextual semantic reasoning on the flagged content. The integration of both stages into a Streamlit web application produces a user-friendly, transparent, and fully offline detection system.

The primary contributions of this work are:

- A modular two-stage pipeline combining rule-based heuristics with LLM-based semantic analysis for fake job detection.
- A lightweight, privacy-preserving deployment using open-source LLMs (Phi-3, Mistral) via the Ollama framework, requiring no cloud connectivity.
- A structured output schema including verdict, confidence score, key reasoning, and actionable recommendations.
- An interactive Streamlit application with single and batch analysis modes, history tracking, and export functionality.
- An empirical evaluation on curated real and fake job descriptions covering diverse fraud typologies.
-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

II. RELATED WORK

Detecting fraudulent job postings has attracted considerable research interest since the early 2000s. The field broadly encompasses three methodological traditions: lexical/heuristic approaches, classical machine learning, and, more recently, deep learning and transformer-based models.

A. Heuristic and Rule-Based Approaches: Early systems relied on manually crafted rules derived from domain expertise. Amira et al. (2019) identified 12 structural features — including absence of a company name, use of personal email addresses, and unrealistic salary claims — as highly predictive indicators of fraud. While interpretable and computationally efficient, such systems require continuous manual maintenance and fail to generalise to semantically novel scam patterns.

B. Classical Machine Learning: Supervised classifiers including Naïve Bayes, Support Vector Machines (SVM), and Random Forest have been applied to the EMSCAD dataset (Employment Scam Aegean Collection Dataset), which contains approximately 17,880 job advertisements. Vidros et al. (2017) achieved an F1-score of 0.92 using Random Forest with TF-IDF features. However, these approaches require labelled training data, are sensitive to class imbalance, and offer limited explanatory power.

C. Deep Learning and Transformer Models: BERT-based classifiers have demonstrated state-of-the-art performance on fake job detection tasks, with models fine-tuned on the EMSCAD dataset achieving F1-scores exceeding 0.95. Mahbub et al. (2021) combined BERT embeddings with a BiLSTM classifier and reported accuracy improvements over traditional ML baselines. Nonetheless, fine-tuning and hosting BERT-scale models requires significant GPU resources and raises data-privacy concerns when API-based inference is used.

D. LLM-Based Zero-Shot and Few-Shot Approaches: The emergence of instruction-following LLMs such as GPT-4, LLaMA, and Mistral has opened new possibilities for zero-shot fraud detection. Recent works have demonstrated that properly prompted LLMs can perform nuanced text classification without task-specific fine-tuning. Our approach extends this paradigm by combining LLM reasoning with a lightweight heuristic pre-screen, deployed entirely on commodity hardware without API dependencies.

III. SYSTEM ARCHITECTURE

The proposed system is architected as a modular five-component pipeline as depicted in Figure 1. Each component is independently testable and replaceable, enabling future extension.

Figure 1: System Architecture — Fake Job Posting Detector

A. Text Preprocessor (preprocessor.py)

Raw job posting text undergoes normalisation in this module. Operations include Unicode decoding, HTML entity removal, whitespace standardisation, and extraction of structured fields such as job title, company name, salary range, contact email, and posting URL. The preprocessor outputs a structured PostingData object consumed by downstream modules.

B. Heuristic Red Flag Engine (analyzer.py)

The heuristic engine applies 15 deterministic rules organised into four severity tiers — Critical, High, Medium, and Low. Each rule evaluates a specific signal extracted from the preprocessed posting:

- Critical (40 pts): Personal email contact (Gmail/Yahoo/Hotmail for official communication), explicit requests for bank details or government ID, upfront payment requirements.
- High (25 pts): Absence of company name, vague or wildly inflated salary claims, presence of suspicious urgency phrases.
- Medium (15 pts): Poor grammar density, absence of qualifications section, missing interview process description.
- Low (5 pts): Generic job title, no company website, missing location information.

Flags are aggregated into a heuristic risk score on a 0–100 scale, where 0 indicates a clean posting and 100 indicates maximum heuristic risk. Postings scoring below 30 are provisionally marked safe; those above 70 trigger a FAKE pre-classification without requiring LLM inference.

C. LLM Prompt Engine (llm_engine.py)

For postings not resolved by the heuristic stage alone, the prompt engine assembles a structured input to the LLM. The prompt template incorporates: (a) the cleaned job posting text, capped at 1500 characters to manage context window constraints; (b) the list of heuristic flags with their severities; and (c) the aggregate heuristic score.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

The LLM is instructed to return a strictly formatted JSON object containing: verdict (REAL | SUSPICIOUS | FAKE), confidence (0–100), risk_score (0–100), summary, key_reasons (list), positive_signals (list), red_flags_confirmed (list), and recommendation. Temperature is set to 0.0 to maximise determinism.

D. Ollama Inference Backend

Model inference is served locally via the Ollama framework (https://ollama.com), which packages open-source LLMs as self-contained executables. The system was evaluated using Phi-3 (2.3 GB, minimum 4 GB free RAM) and Mistral 7B (4.1 GB, minimum 8 GB free RAM). Ollama exposes a REST API at http://localhost:11434/api/generate, which the llm_engine.py module invokes via Python's requests library.

E. Streamlit Web Interface (app.py)

The user interface is implemented using the Streamlit framework and provides three functional tabs: Analyse (single posting input with gauge visualisation, red flag breakdown, and LLM verdict card), Batch (CSV upload for multi-posting analysis with aggregate statistics), and History (persistent log of all analyses with pie chart and CSV export). The interface requires no authentication and runs entirely on localhost.

IV. METHODOLOGY

A. Dataset

To evaluate the system, a curated dataset of six job postings was constructed: three genuine postings and three fraudulent postings, each covering a distinct fraud typology. Genuine postings were sourced from established Indian IT companies with verifiable company registration details, structured interview processes, and salary ranges consistent with market standards.

ID	Type	Role/Scam Type	Key Characteristics
R-01	Genuine	Backend Python Engineer	Named company, CIN, HR contact, INR salary
R-02	Genuine	QA Automation Engineer	Verified employer, structured JD, glassdoor link
R-03	Genuine	Data Analyst (BI)	LinkedIn apply, PF/ESIC stated, valid city
F-01	Fake	Data Entry (WFH Scam)	No company name, asks for Aadhaar + bank details
F-02	Fake	Mystery Shopper Fraud	Wire transfer request, Yahoo email, secrecy clause
F-03	Fake	MLM Business Opportunity	Joining fee, recruit-to-earn model, Gmail contact

Table I: Curated Evaluation Dataset

B. Experimental Setup

All experiments were conducted on a consumer-grade workstation with the following specifications: Intel Core i5 processor, 8 GB DDR4 RAM, Windows 11 OS. The Ollama server was launched as a background process, and model inference was invoked via HTTP on port 11434. Python 3.11 was used for the application layer, with Streamlit 1.32, LangChain-compatible requests, and Plotly for visualisation.

Two LLMs were evaluated: Phi-3:latest (default, 2.3 GB model size) and Mistral:latest (4.1 GB). Phi-3 was selected as the primary model due to its compatibility with 8 GB RAM systems after closing background applications. Mistral was evaluated in a constrained-RAM scenario to assess degradation.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

C. Evaluation Metrics

System performance was evaluated across three dimensions: (1) Verdict Accuracy — whether the final verdict (REAL / SUSPICIOUS / FAKE) matched the ground-truth label; (2) Heuristic Precision — whether the heuristic flags identified were genuine red flags rather than false positives; and (3) Response Latency — end-to-end time from submission to verdict display.

V. RESULTS AND DISCUSSION

Table II summarises the detection results across all six test postings using the Phi-3 model.

ID	Ground Truth	Heuristic Score	LLM Verdict	Confidence	Correct?
R-01	REAL	100/100 (0 flags)	REAL	88%	✓
R-02	REAL	95/100 (0 flags)	REAL	85%	✓
R-03	REAL	100/100 (0 flags)	REAL	82%	✓
F-01	FAKE	20/100 (5 flags)	FAKE	94%	✓
F-02	FAKE	15/100 (6 flags)	FAKE	91%	✓
F-03	FAKE	30/100 (4 flags)	FAKE	87%	✓

Table II: Detection Results (Phi-3:latest, 8 GB RAM, Ollama)

The system achieved 100% verdict accuracy on the curated dataset, correctly classifying all three genuine postings as REAL and all three fraudulent postings as FAKE. The heuristic engine alone produced zero false positives for the genuine postings and correctly flagged all fraudulent postings with severity-weighted red flags.

The LLM stage contributed additional semantic reasoning that complemented the heuristic scores. For instance, F-02 (Mystery Shopper scam) received only 4 critical flags from the heuristic engine — since the posting was structurally more sophisticated — but the LLM correctly identified the wire transfer instruction and confidentiality clause as strong indicators of fraud, yielding a 91% confidence FAKE verdict.

Response Latency: On the test hardware (8 GB RAM, Phi-3), average end-to-end analysis time was approximately 45–90 seconds per posting, with inference consuming the majority of that time. Mistral on the same hardware with reduced free RAM exceeded the 300-second timeout threshold in two of three trials, confirming that Phi-3 is the appropriate model for constrained hardware.

False Positive Analysis: No false positives (genuine postings misclassified as fake) were observed. The heuristic engine's 100/100 score for genuine postings (zero flags) provided a strong prior that the LLM consistently supported. This is particularly important in a job-seeker-facing application where false positives could cause candidates to miss legitimate opportunities.

VI. CHALLENGES AND LIMITATIONS

Several challenges were encountered during the development and evaluation of this system:

Hardware Constraints: Running 7B-parameter LLMs (Mistral) on 8 GB RAM proved unreliable due to operating system overhead and competing processes. This constraint necessitated the use of Phi-3 as the primary model and significantly limited batch analysis throughput. The system is more performant on machines with 16 GB or more RAM.

JSON Output Reliability: Smaller models (TinyLlama, Phi-3 at reduced context) occasionally failed to produce well-formed JSON, requiring iterative prompt engineering — including double-brace escaping in Python format strings — to ensure reliable structured output.

Dataset Scale: The evaluation dataset of six postings is insufficient for statistically rigorous performance claims. Future work will incorporate the EMSCAD dataset (17,880 postings) for large-scale benchmarking.

Adversarial Robustness: Sophisticated fraudsters may craft postings that evade heuristic rules by mimicking legitimate structure. The system's resilience to adversarial inputs has not been formally evaluated.

VII. CONCLUSION AND FUTURE WORK

This paper presented a hybrid fake job posting detection system integrating a heuristic pre-screening engine with open-source LLM-based semantic analysis, deployed as a local Streamlit web application via the Ollama inference framework. The system

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

demonstrated 100% verdict accuracy on a curated six-posting evaluation dataset and operated without any dependency on commercial APIs or cloud infrastructure, making it suitable for privacy-sensitive deployments.

The modular architecture enables straightforward extension in several directions. Future work will focus on: (1) large-scale evaluation using the EMSCAD benchmark dataset; (2) fine-tuning a lightweight LLM (e.g., Phi-3 or Gemma-2) specifically on fake job posting data to improve confidence calibration; (3) integration of a real-time web scraping module to analyse job postings directly from URLs; (4) development of a browser extension for in-context analysis on popular job portals; and (5) multilingual support for Indian regional languages to extend detection coverage to vernacular-language job advertisements.

The open-source, privacy-preserving nature of this system positions it as a viable foundation for HR-technology platforms and cyber-security toolkits aimed at protecting job seekers from online recruitment fraud.

REFERENCES

[1] S. Vidros, C. Kolias, G. Kambourakis, and L. Akoglu, 'Automatic Detection of Online Recruitment Frauds: Characteristics, Methods, and a Public Dataset,' Future Internet, vol. 9, no. 1, p. 6, 2017.
[2] M. Mahbub, R. Pardede, A. S. Kayes, and W. Rahayu, 'Controlling Fake Job Advertisements Using Machine Learning and Natural Language Processing,' IEEE Access, vol. 9, pp. 153-164, 2021.
[3] T. B. Brown et al., 'Language Models are Few-Shot Learners,' Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, 2020.
[4] A. Touvron et al., 'Llama 2: Open Foundation and Fine-Tuned Chat Models,' arXiv preprint arXiv:2307.09288, 2023.
[5] Microsoft Research, 'Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone,' arXiv preprint arXiv:2404.14219, 2024.
[6] Mistral AI, 'Mistral 7B,' arXiv preprint arXiv:2310.06825, 2023.
[7] Ollama, 'Ollama: Run Large Language Models Locally,' https://ollama.com, 2024.
[8] Streamlit, 'Streamlit: The Fastest Way to Build Data Apps,' https://streamlit.io, 2024.
[9] Internet Crime Complaint Center (IC3), '2023 Internet Crime Report,' Federal Bureau of Investigation, 2024.
[10] A. Vaswani et al., 'Attention Is All You Need,' Advances in Neural Information Processing Systems, vol. 30, 2017.
[11] J. Devlin, M. Chang, K. Lee, and K. Toutanova, 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,' in Proc. NAACL-HLT, 2019, pp. 4171–4186.
[12] National Cyber Crime Reporting Portal, 'Annual Cybercrime Report 2023,' Ministry of Home Affairs, Government of India, 2024.