FAKE JOB SCAM DETECTION USING AI AND ML

IntelliHireX Employment Intelligence Hub: A Hybrid Framework for Detecting Fake Job and Internship Offers

Sushama Kolhe, Kesar Thakur, Suraj Khillare, Vaishnavi Shete, Yogeshchandra Puranik

Department of Master of Computer Applications, Progressive Education Society's Modern College of Engineering, Pune, Maharashtra, India

Published November 2026

Abstract

Online job portals have made hiring easier, but they have also made it easier for scammers to post fake jobs and internships that collect personal details, bank information or "training fees" from job seekers. Many fake adverts now read like professional ones, so keyword filters and text-only classifiers often miss them. A scammer can copy a genuine company's advert word for word and change only the contact email, which a text model cannot detect. This paper presents IntelliHireX Employment Intelligence Hub, a hybrid framework that checks a job offer from several sides instead of trusting the text alone. A TF-IDF and Linear Support Vector Machine (SVM) model reads the posting. Separate modules check the company against Indian Ministry of Corporate Affairs (MCA) records, the recruiter's email and website domain, the company's official career page, suspect identifiers that can be reported to the National Cyber Crime Reporting Portal (NCRP), and fee-related clauses in offer letters. A transparent 100-point score combines these signals into a Low, Medium or High risk level and lists the reasons. On the public EMSCAD dataset of 17,880 postings, the text model reached 99.11% accuracy, 94.34% precision, 86.71% recall, an F1-score of 90.36% and a ROC-AUC of 0.9908 on a held-out test set of 3,576 postings, with all preprocessing fitted on training data only. The registry, domain and document modules have not yet been evaluated on labelled data. The system is deployed as a FastAPI service with a web interface.

Keywords: fake job detection; employment scams; natural language processing; Linear SVM; TF-IDF; MCA registry; NCRP; explainable risk scoring

1 Introduction

Job portals such as LinkedIn, Indeed and Glassdoor, together with many remote-internship sites, now sit at the centre of hiring. Students and fresh graduates use them heavily, which makes them an easy target. Scammers publish adverts that appear to come from real companies and then ask for money (a "security deposit" or "training kit" fee), collect identity documents, or use the applicant's bank account to move money. For an applicant, the loss can be financial, but it can also be the exposure of identity documents that may be misused long after the advert has disappeared.

Most platforms still depend on manual review or fixed keyword lists. Manual review is slow, so a fake advert can reach hundreds of applicants before it is removed. Keyword lists fail because modern scam adverts avoid obvious warning words and copy the polished language of genuine companies. Machine learning on the advert text has improved detection, but it has a clear blind spot. If a scammer copies a genuine advert from a large company and replaces only the contact email with a free webmail address or a look-alike domain, the text is still genuine and a text-only model passes it. The deception lies in *who* is offering the job, not in *what* the advert says.

India also has an official channel for reporting cyber fraud, the National Cyber Crime Reporting Portal (NCRP), and a public source of company registration data, the master data of the Ministry of Corporate Affairs (MCA). These sources are rarely connected to the tools that job seekers use to judge an offer. This gap motivated the present work. The central idea is simple: a trustworthy offer should survive checks on the text, on the company, on the recruiter's contact channel and on the terms of the offer, and the job seeker should be told in plain language which checks failed and why.

The contributions of this paper are as follows:

1. **Leakage-safe text classifier.** A TF-IDF and Linear SVM model for recruitment fraud, evaluated on highly imbalanced data with all preprocessing fitted on the training partition only and no oversampling.
2. **Text-independent verification.** Five verification modules that do not depend on the advert wording: an MCA company lookup over 2,215,712 records, recruiter email and domain analysis, a live career-page check, NCRP identifier support, and an offer-letter clause scan.
3. **Explainable risk score.** A 100-point score that shows which signals fired, in place of an opaque probability, together with an analysis of what the score can and cannot conclude.
4. **Working system.** A REST API and web application that deliver the result and a downloadable report to non-technical users.

The remainder of the paper is organised as follows. Section 2 reviews related work and Section 3 identifies the research gap. Section 4 formulates the problem and the threat model. Section 5 summarises the theoretical background. Section 6 presents the methodology, Section 7 the verification modules and Section 8 the risk engine. Section 9 describes the implementation. Section 10 reports the results, Section 11 the testing, and Sections 12 and 13 discuss limitations and responsible use. Section 14 concludes the paper and Section 15 outlines the future scope.

2 Literature survey

2.1 Online recruitment fraud

Online recruitment fraud (ORF) refers to fake job offers published to deceive applicants. Reported motives include advance-fee collection, harvesting of personal and identity data, and recruiting unwitting people to move funds through their own accounts. ORF is closely related to phishing and spam because it relies on social engineering, but it differs in one important way: the message is a public advert that a victim actively seeks out and trusts because of the platform on which it appears. Early work therefore borrowed techniques from email spam filtering and treated the problem as supervised text classification.

2.2 Review of related studies

2.2.1 Joju (2026)

"Fake Job Detection System Using Machine Learning and NLP" was published in IJIRT [16]. The author uses the Kaggle fake job posting data of 17,880 records, of which only 4.84% are fake. The title, company profile, description and requirements are joined into one text, converted with TF-IDF (unigrams and bigrams) and classified with Logistic Regression, Random Forest and Random Forest with SMOTE. Lowering the decision threshold from 0.5 to 0.32 raises recall for the fake class from 68% to 84%, while precision falls from 99% to 86%. The final model reaches 99% accuracy, an F1-

score of 0.85 and a ROC-AUC of 0.984 on a test set of 3,576 postings. The system runs as a Streamlit web application that labels a posting as Fake, Suspicious or Genuine, highlights suspicious keywords and produces a PDF report that can be sent by email. This paper is close to ours in its use of TF-IDF, a test set of the same size, threshold tuning, a three-level output and a downloadable report. The difference is that it still judges the posting text alone and does not check the company or the recruiter.

2.2.2 Radhika et al. (2026)

"Detecting Recruitment Scams Using NLP and Machine Learning Techniques" was published in IJESAT [17]. The authors use the Kaggle fake job posting data with the title, company profile, description, requirements, benefits and other fields of each posting. They balance the classes with SMOTE-type methods and compare Naive Bayes, Logistic Regression, Random Forest, SVM and LSTM. Their results table shows about 85%, 88%, 92%, 90% and 94% accuracy for these models, with LSTM the highest. The system also gives a confidence score and a dashboard where reported fraudulent postings can be managed. The authors point out that widely used datasets such as EMSCAD date from 2012 onwards, which is also a limit of our work.

2.2.3 Pitumpe and Rahmati (2026)

"Anansi: Scalable Characterization of Message-Based Job Scams" is an arXiv paper [18]. It does not classify postings. It studies job scams that reach people by SMS, WhatsApp and Telegram, where the victim is asked to do simple online "tasks", such as rating products, and is later asked to deposit cryptocurrency. The authors built a pipeline that uses large language models to chat with scammers, and ran it for ten months. It collected 29,209 messages from 7,028 scammer phone numbers, got replies from 1,901 of them, and traced more than $12 million in cryptocurrency sent to the scammers' wallet addresses. The study also found heavy reuse of message templates, website designs and hosting servers, frequent use of low-cost domain endings such as .vip and .club, and copying of well-known brand names. Only 29.6% of the 135 scam websites it found were listed by VirusTotal, which shows that existing blocklists miss many job scams. The work was done in the United States and still needs human help at some steps. For our project it shows that sender, domain and payment details can expose a scam even when the message looks professional, which supports our domain, email and NCRP identifier checks.

2.2.4 Saranya et al. (2026)

"A Systematic and Comprehensive Survey on Deep Learning-Based Fake Recruitment Email Detection" was published in IJCRT [19]. The paper reviews ten studies that move from classical machine learning and NLP methods to deep learning (LSTM) and transformer models such as BERT and DistilBERT. It also covers duplicate-job detection and explainable AI. The authors conclude that models which understand context detect scams better than traditional ones, but they need more computing power, and open problems remain in imbalanced data and changing scam tactics. Their own evaluation table lists 89.5% accuracy and an F1-score of 86.2%. The paper deals with recruitment emails, and its interest in explainable results supports the way we show reasons with every score.

2.2.5 Ashok Kumar et al. (2024)

"Detection of Fake Online Recruitment Using Machine Learning" was published in IJSDR [20]. The paper describes a system that picks features from job descriptions, qualifications and company details and uses TF-IDF with models such as support vector machines, random forests and deep neural networks. The idea is to place the system on a job platform so that suspicious postings are filtered

out. The training table in the paper lists an accuracy of 96.45% for Random Forest, and a web page lets the user enter job details and get a prediction. Like most of the studies above, it looks at the posting only and does not check the company or the recruiter.

2.2.6 Pillai (2023)

"Detecting Fake Job Postings Using Bidirectional LSTM" was published in IRJMETS and is also available on arXiv [3]. The author trained a bidirectional LSTM, which reads the text of a posting in both directions, on the Kaggle job posting data of 17,880 postings and reported 98.71% accuracy with a ROC-AUC of 0.91. For the fake class the paper reports a precision of 0.89, a recall of 0.83 and an F1-score of 0.86, so about one in six fake postings is still missed. Neural models can capture more context than word counts, but they need more computing power and training time. They also share the weakness of simpler models: if the advert text is genuine and only the contact details are fake, the text gives no warning.

2.2.7 Ullah and Jamjoom (2023)

"A Smart Secured Framework for Detecting and Averting Online Recruitment Fraud Using Ensemble Machine Learning Techniques" appeared in PeerJ Computer Science [21]. The authors use four ensemble methods: AdaBoost, XGBoost, Voting and Random Forest. They worked on the public Kaggle copy of the EMSCAD data with 17,880 job records and prepared it by handling missing values, class imbalance and feature scaling. AdaBoost gave the best accuracy at 98.374%, followed by XGBoost and Voting at 97.835% and Random Forest at 97.669%. We use the same dataset, but we also report precision, recall and F1-score for the fake class, because accuracy alone hides how many scams are missed.

2.2.8 Goyal et al. (2021)

"Spy The Lie: Fraudulent Jobs Detection in Recruitment Domain using Knowledge Graphs" is a paper from the PreCog group at IIIT Delhi [22]. The authors argue that fake postings often contain facts that do not fit together, such as a very high salary for a job with no experience needed, and that text-only methods ignore the relations between such details. Their framework, FRJD, has three parts: a fact-checking module that builds two knowledge graphs from facts taken from genuine and fraudulent postings and scores new facts (the TransH method worked best); a contextual module that uses DistilBERT; and meta-data features such as the number of skills and the length of the posting. They used 70,000 genuine and 70,000 fraudulent postings from a large Indian recruitment platform, which gave about 4 million facts, and also tested on EMSCAD. They report an F1-score of 0.96 on 157,880 postings. The paper notes that some facts, such as demands for visa fees, appear in both genuine and fraudulent postings and are not separated well. The main data is proprietary, so we could not repeat the result, and the facts are checked only against other postings, not against an outside record such as a company registry.

2.2.9 Anand and Sharma (2020)

"Job Portal End Detection of Fake Job Posting Using Machine Learning" was published in the Compliance Engineering Journal [23]. The authors took a Kaggle job-posting dataset of 17,881 rows and built a decision tree in RapidMiner with a 70/30 split of training and test data. The model works on a few signs, mainly the company's CIN (registration) number, an official email id and a security-deposit demand, and it reaches 97.50% accuracy. Their argument is that most fake posters have no CIN, no official email domain and ask for a large deposit. This is the closest paper to our design,

because our company check, email check and offer-letter scan look for the same signs. The difference is that we check them as separate steps and add a text model and a report.

2.2.10 Lal et al. (2019)

"ORFDetector: Ensemble Learning Based Online Recruitment Fraud Detection" was presented at the 12th International Conference on Contemporary Computing (IC3) [2]. The authors proposed ORFDetector, an ensemble learning model, and reported about 95.4% accuracy and a 94% F1-score on annotated postings.

2.2.11 Vidros et al. (2017)

"Automatic Detection of Online Recruitment Frauds: Characteristics, Methods, and a Public Dataset" appeared in Future Internet [1]. The authors built and released the Employment Scam Aegean Dataset (EMSCAD), which holds 17,014 genuine and 866 fraudulent job adverts published between 2012 and 2014; the labels were put in by the staff of the platform that supplied the adverts. It remains the most widely used public dataset for this problem and is the one used here. The paper also describes how online recruitment fraud works and tries ready-made classifiers on the data. For our project it gives us the data and shows how small the fraudulent part is.

2.3 Summary of the surveyed studies

Table 1 summarises the surveyed studies.

Table 1. Summary of the surveyed studies (as stated by authors).

Study	Year	Method	Reported result
Joju [16]	2026	TF-IDF with Logistic Regression, Random Forest and Random Forest + SMOTE, with threshold tuning	99% accuracy, 86% precision, 84% recall, F1-score 0.85
Radhika et al. [17]	2026	Naive Bayes, Logistic Regression, Random Forest, SVM, LSTM with class balancing	LSTM best, about 94% accuracy
Pitumpe and Rahmati [18]	2026	LLM-driven chats with scammers; analysis of messages, domains, hosting and wallets	29,209 messages; more than $12 million traced to wallets
Saranya et al. [19]	2026	Survey of deep learning and transformer methods for recruitment emails	89.5% accuracy, 86.2% F1-score in its table
Ashok Kumar et al. [20]	2024	TF-IDF with SVM, Random Forest and neural networks	96.45% accuracy (Random Forest)
Pillai [3]	2023	Bidirectional LSTM	98.71% accuracy, ROC-AUC 0.91
Ullah and Jamjoom [21]	2023	AdaBoost, XGBoost, Voting, Random Forest	98.374% accuracy (AdaBoost)
Goyal et al. [22]	2021	Knowledge-graph fact checking (TransH) with DistilBERT and meta-data features	F1-score 0.96 on 157,880 postings
Anand and Sharma [23]	2020	Decision tree on CIN number, official email id and security deposit	97.50% accuracy
Lal et al. [2] (ORFDetector)	2019	Ensemble learning	About 95.4% accuracy, 94% F1-score
Vidros et al. [1]	2017	EMSCAD dataset and standard classifiers	17,014 genuine and 866 fraudulent adverts released

The surveyed papers show a clear movement. Early and student-level work uses classical classifiers such as decision trees, Naive Bayes, SVM and Random Forest [1, 16, 20, 23]. Later work uses ensembles [2, 21] and then deep learning and transformer models [3, 17, 19, 22]. Reported accuracy ranges from about 85% to 99%. The recent studies [16, 18] also show where the field is heading: scams now reach people through messaging apps and fake websites, and a tool used by job seekers needs to be explained and delivered as a working application, not only as a model.

Accuracy figures from different papers cannot be compared directly. Studies use different splits, different handling of class imbalance and, in some cases, different copies of the data. Because only about 5% of postings are fake, accuracy is also a weak measure on its own: a classifier that labels every posting genuine scores 95.16% accuracy on EMSCAD while catching no scams. This paper therefore concentrates on precision, recall and F1-score for the fake class.

2.4 Data leakage in text classification

Data leakage occurs when information from outside the training data influences the model, which inflates the measured performance [4]. In text classification a common source is fitting the vocabulary or TF-IDF statistics on the full corpus before splitting, or applying an oversampling technique such as SMOTE [5] before the split so that synthetic points derived from test examples enter training. Both problems are easy to introduce and hard to notice because the reported scores look excellent. The present work avoids them by construction, as described in Section 6.2.

3 Research gap

The surveyed studies concentrate on classifying the posting. Real scams, however, often depend on things outside the advert text: a fake domain, a free email address, a WhatsApp or Telegram contact, or a company that does not legally exist. As far as we could tell from the scope of these studies, they do not combine text classification with company-registry, domain, career-page or cybercrime-identifier checks. Only Anand and Sharma [23] use signs from outside the advert text, namely the company registration number, an official email id and a security deposit, and they use them as inputs to a small decision tree. Goyal et al. [22] go beyond plain text by checking facts and meta-data inside the posting, but only against other postings. Pitumpe and Rahmati [18] study scam websites, domains and wallets in depth, but as a measurement study of scammers and not as a tool a job seeker can use before applying. None of the surveyed papers checks the company in a registry, looks for the vacancy on the official career page, or scans an offer letter. The studies also usually return a class label or a probability, which a job seeker cannot easily interpret; a few [16, 17] add a confidence score and keywords, but these come from the text model alone. Table 2 summarises how IntelliHireX Employment Intelligence Hub differs in scope.

Table 2. Scope of the surveyed studies [1–3, 16–23] compared with IntelliHireX Employment Intelligence Hub.

Aspect	Surveyed studies [1–3, 16–23]	IntelliHireX Employment Intelligence Hub
Main input	Job posting text and posting fields (scammer messages and websites in [18])	Posting text, recruiter details, career-page URL, offer letter
Company registration check	Not covered (only [23] uses the CIN number as a model input)	MCA records (2.21 million companies, six states)

Aspect	Surveyed studies [1–3, 16–23]	IntelliHireX Employment Intelligence Hub
Recruiter email and domain check	Not covered (only [23] uses an official email id as a model input)	Free-email and domain-mismatch checks
Official career page check	Not covered	Live page fetch, job-title search
Cybercrime identifier support	Not covered	NCRP / I4C identifier preparation and links
Offer-letter screening	Not covered (only [23] uses a security-deposit sign)	Rule-based fee and bond clause scan
Output	Class label, probability or confidence score with keywords	Low / Medium / High with reasons

4 Problem formulation and threat model

4.1 Problem statement

Job seekers, especially students and fresh graduates, cannot easily tell whether an online job or internship offer is genuine. Existing filters and text-only classifiers examine the advert text and can miss scams where the text is genuine but the recruiter's identity, company registration, contact domain or offer terms are fake. Given a job posting and the related recruiter details, the system must: (i) estimate from the text whether the posting looks fraudulent; (ii) check outside signs of legitimacy, namely company registration, recruiter email and domain, the official career page and the terms of an offer letter; (iii) combine all evidence into an explainable Low, Medium or High risk level; and (iv) deliver the result through a usable web application and a downloadable report.

4.2 Scam patterns considered

The design targets four recurring patterns, each of which leaves a different trace:

- **Advance-fee scams.** The applicant is asked to pay a deposit, kit or training fee before joining. The trace appears in the offer-letter terms and in payment instructions that point to unregistered bank or UPI accounts.
- **Identity harvesting.** The applicant is asked for identity or banking documents early. The trace appears in free-email contact channels and in companies with no registration record.
- **Impersonation of real companies.** A genuine advert is copied and the contact details are replaced. The text is clean; the trace appears in the email domain, which does not match the company's website, and in the absence of the vacancy on the official career page.
- **Repeat offenders.** The same phone number, email address or messaging handle is reused across scams. The trace is a prior cybercrime report, which is why NCRP identifier support is included.

4.3 Design requirements

From the problem statement we derived five requirements. **Explainability:** every result must show which signals added points and why. **Graceful degradation:** a check that cannot be completed, such as an unreachable website, must be reported as "unable to verify" and must not stop the other checks. **Responsiveness:** web requests to company sites use a 5-second timeout. **Safety of language:** results

are advisory and never accuse a person or company of a crime. **Maintainability:** each check is a separate module with its own test script, so that data sources can be replaced independently.

5 Theoretical background

5.1 TF-IDF weighting

TF-IDF represents a document as a vector in which each dimension corresponds to a term and the value reflects how characteristic that term is of the document [6, 7]. With sublinear term frequency, the weight of term w in document d is

$$\text{tfidf}(w,\ d) = (1 + \ln tf) \times \left(\ln \left(\frac{1 + N}{1 + df} \right) + 1 \right) \tag{1}$$

where tf is the count of w in d, df is the number of documents containing w, and N is the number of documents. Sublinear scaling dampens the effect of terms repeated many times, which is useful because scam adverts often repeat phrases such as “urgent” or “immediate joining”. Vectors are length-normalised. Using word unigrams and bigrams lets the model capture short phrases (for example, “registration fee”) that a single-word model would split apart.

5.2 Linear support vector machine

A linear SVM [8] learns a weight vector $\mathbf{w}$ and bias b so that the hyperplane $\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$ separates the two classes with the largest possible margin. With an L2 penalty and squared hinge loss, the training objective is

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_n c_{y_n} \left(\max\left(0,\ 1 - y_n \left(w^{\mathrm{T}} x_n + b \right) \right) \right)^2 \tag{2}$$

where $y_{\mathrm{n}} \in \{-1, +1\}$, C controls the trade-off between margin width and training errors, and c_{y} is a per-class weight. Text vectors are high-dimensional and sparse, a setting in which linear models are known to work well because the classes are often close to linearly separable in the full term space. Training is a convex problem with a single global optimum, and efficient solvers exist for sparse data [9]. The decision function $\mathbf{w}^{\mathrm{T}}\mathbf{x} + b$ gives a signed distance from the boundary, not a probability. This is why IntelliHireX Employment Intelligence Hub reports a label and reasons rather than a percentage confidence.

5.3 Naive Bayes and Random Forest baselines

Multinomial Naive Bayes models the probability of each term given the class and assumes terms are independent given the class. It is fast and a standard baseline for text, but its independence assumption ignores phrase structure. Random Forest [10] averages many decision trees trained on random subsets of data and features. It handles nonlinear interactions well but does not scale gracefully to tens of thousands of sparse features, so a dimensionality-reduction step is normally applied first. Truncated SVD projects the sparse TF-IDF matrix onto a small number of dense components, at the price of discarding information that does not lie along the leading directions.

5.4 Handling class imbalance by weighting

Instead of creating synthetic minority examples, the loss of each training example is multiplied by a class weight. For class j the weight is

$$w_j = \frac{N}{k n_j} \tag{3}$$

where N is the number of training postings, $k = 2$ is the number of classes and n_j is the number of training postings in class j. With 693 fake and 13,611 genuine training postings, each fake posting receives a weight of about 10.3 and each genuine posting a weight of about 0.53, so errors on the rare class count roughly twenty times more. Weighting leaves the data unchanged, so no synthetic point derived from a test example can enter training.

5.5 Evaluation metrics

Let TP, FP, FN and TN denote true positives, false positives, false negatives and true negatives, with the fake class as positive. The metrics used are

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

Precision answers “of the postings flagged as fake, how many truly were?”; recall answers “of the fake postings, how many were caught?”. F1 is their harmonic mean. ROC-AUC measures how well the decision scores rank fake postings above genuine ones, independent of any threshold. We additionally report the Matthews correlation coefficient (MCC) and balanced accuracy, which remain informative under heavy imbalance.

6 Proposed methodology

6.1 Overview

IntelliHireX Employment Intelligence Hub works in two parts. The first is a text classifier trained once on a public dataset. The second is a set of checks that run for every new job offer. A risk engine then combines both parts. Figure 1 shows the overall flow: the posting, recruiter details and an optional offer letter enter the system, the text model and five verification checks run independently, and all signals feed the risk engine, which produces the final level and a report.

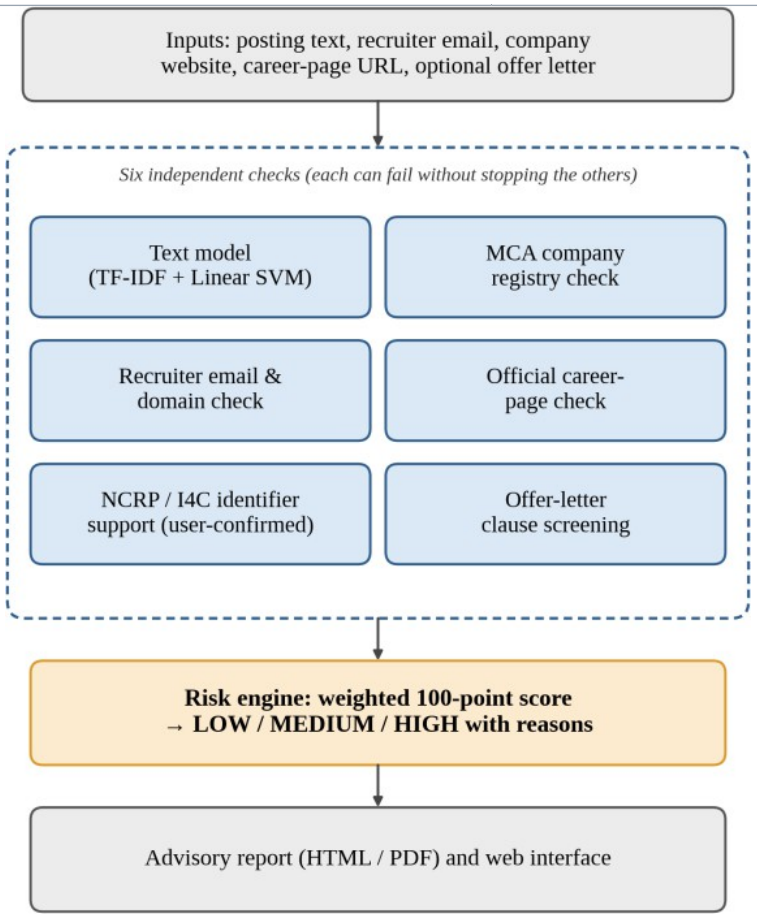


Figure 1. Overall flow of IntelliHireX Employment Intelligence Hub. Each check runs independently and sends its signal to the risk engine.

6.2 Splitting the data without leakage

The split precedes every fitted operation, as summarised in Figure 2. Specifically:

1. The data was split first: 80% for training and 20% for testing, stratified on the label with a fixed random seed of 42.
2. The job_id column was dropped because it carries no real information.
3. Missing-value handling, the TF-IDF vocabulary and the TF-IDF weights were learned from the training part only and then applied unchanged to the test part.

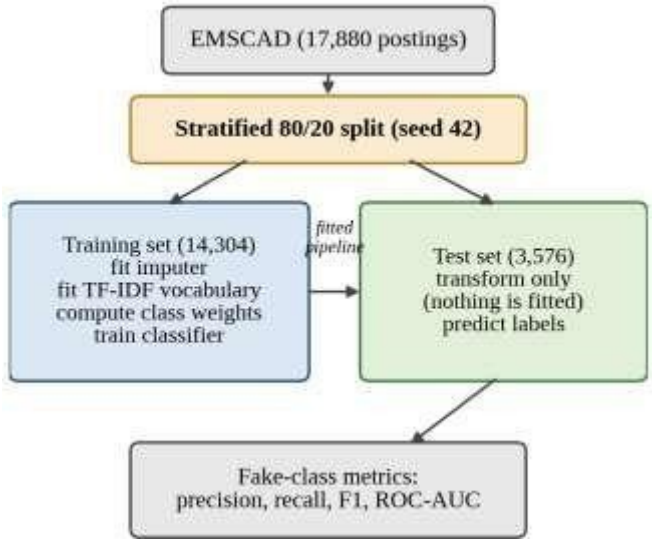


Figure 2. Leakage-safe evaluation pipeline. Every fitted component sees the training partition only.

1. SMOTE was not used. Imbalance is handled inside the classifiers by class weighting (Equation 3).

6.3 Dataset

The EMSCAD data [1] contains 17,880 job postings: 17,014 legitimate (95.16%) and 866 fraudulent (4.84%). This imbalance matches real life, where most adverts are genuine. Each posting has free-text fields (title, company profile, description, requirements, benefits) and simple fields (telecommuting, has_company_logo, has_questions, employment_type, required_experience, required_education, industry, function). The text fields were combined into a single cleaned text column in the processed dataset file used for training.

The fake share is 4.84% in the full set, 4.85% in training and 4.84% in testing, confirming that stratification preserved the class ratio (Table 3; the training counts follow from the totals and the test counts).

Table 3. Stratified train–test partition.

Partition	Genuine	Fake	Total
Training (80%)	13,611	693	14,304
Test (20%)	3,403	173	3,576
Full dataset	**17,014**	**866**	**17,880**

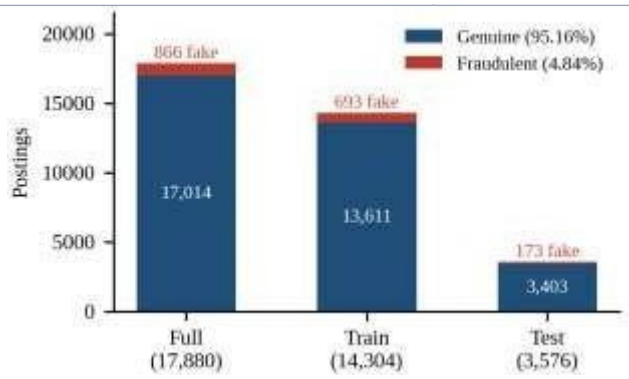


Figure 3. Class distribution of the full dataset and of the stratified training and test partitions.

The model files are built as scikit-learn pipelines [11], so the vocabulary and imputation values are fitted inside the training step and stored with the model. This makes accidental refitting on test data unlikely.

6.4 Feature preparation

Text. The title, company profile, description, requirements and benefits of each posting are joined into one document and cleaned. The TF-IDF step lowercases the text and strips accents, uses word unigrams and bigrams with sublinear weighting, keeps terms that appear in at least two postings, and limits the vocabulary to 20,000 features. **Categorical fields.** Missing values are filled with "Not Specified" and one-hot encoded; categories seen fewer than twice are grouped together. **Numeric fields.** Missing yes/no fields are filled with the most common training value.

6.5 Models compared

Three classifier families were compared on the same split. Their main settings are listed in Table 4.

Table 4. Classifier configurations.

Model	Configuration
Naive Bayes	Multinomial; equal prior probability for both classes; TF-IDF input
SVD + Random Forest	TF-IDF and category features reduced to 75 dense components by Truncated SVD; 150 trees; balanced class weights
Linear SVM	L2 penalty; squared hinge loss; balanced class weights; fixed random seed 42; TF-IDF and category features

Models were chosen by the F1-score of the fake class, with recall, ROC-AUC and precision used to break ties. No hyperparameters were tuned. Because the choice among the three models used the test-set scores, the reported results are slightly optimistic; a separate validation set would remove this effect. We state this limitation again in Section 12.

7 Verification modules beyond text

Each module below works independently of the text model, because the text model alone missed 23 of the 173 fake postings in the test set (Section 10). A module that cannot reach its data returns an explicit "unable to verify" status instead of a guess, and the remaining modules continue to run.

7.1 Company registration check (MCA)

Genuine recruiters work for registered companies. The company name is searched in a table of 2,215,712 records derived from MCA master data [12] for six states. Table 5 gives the size of each state's contribution.

Table 5. MCA records by state.

State	Records	Share
Maharashtra	755,653	34.1%
Delhi	507,637	22.9%
Karnataka	258,323	11.7%
Tamil Nadu	251,291	11.3%
Gujarat	222,915	10.1%
Telangana	219,893	9.9%
Total	**2,215,712**	**100%**

Matching is deliberately conservative, because a false match would give a fake company a clean registry result. Algorithm 1 shows the steps. Names are first cleaned (extra spaces and punctuation removed, case ignored) and legal endings such as PRIVATE LIMITED, PVT LTD, LIMITED and LLP are replaced by a standard form. An exact match is tried first, then a match with the legal ending removed from both names. Only if both fail is a fuzzy similarity ratio computed over close candidates, based on edit distance [13]. A fuzzy match is accepted only if the best ratio is at least 0.90 *and* it beats the second-best candidate by at least 0.05. The margin rule prevents acceptance when two registered companies have very similar names, a common situation in company registries. When a match is found the record's status (ACTIVE or INACTIVE) is read, since a struck-off company is also a warning sign.

A company that is not found is not declared fraudulent. The result is "not found in the six-state data" with a warning that a genuine company registered elsewhere, or abroad, would also be absent.

7.2 Recruiter email and domain check

The module separates the registrable domain from the recruiter's email and from the company website using the tldextract library, which uses the public suffix list so that domains such as co.in are handled correctly. The email is placed in one of four groups (Table 6) and the company website is checked with a light DNS and HTTP test.

Table 6. Recruiter email categories.

Category	Meaning
COMPANY_DOMAIN	Email domain matches the company's website domain
FREE_EMAIL	Public webmail provider such as Gmail, Yahoo, Hotmail or ProtonMail
OTHER_DOMAIN	A corporate-looking domain that differs from the company's website
INVALID	The email address is not syntactically valid

A free webmail address is not proof of fraud, since small employers sometimes use one, but it removes the recruiter's link to any organisation that could be verified. A domain that differs from the company website is a stronger sign of possible impersonation, for example a look-alike spelling of a well-known company.

7.3 Official career-page check

The user supplies the company's careers URL. The module downloads the page with an HTTP GET request, a 5-second timeout and a custom User-Agent, removes scripts, styles and navigation menus, cleans the visible text and searches it for the job title with case-insensitive matching. The result is one of FOUND, NOT_FOUND, UNABLE_TO_VERIFY or INVALID_URL. A vacancy that the company does not list on its own career page is a warning sign, but a page that cannot be reached is only "unable to verify" and earns fewer points. Pages built mainly with client-side JavaScript may not expose the job title in a simple fetch; this is acknowledged as a limitation.

7.4 NCRP / I4C identifier support

The National Cyber Crime Reporting Portal [14] is the official channel for reporting cyber fraud in India.

IntelliHireX Employment Intelligence Hub does not read any government database. The function prepare_ncrp_check() accepts an email, phone number, website link, social-media link and messaging handle, cleans each one (for example, "+91 90000 00000" becomes "+919000000000"), labels its type (EMAIL, PHONE, URL and so on), sets its status to NOT_CHECKED and returns the official portal link. The user opens the portal, searches the identifiers there and reports the answer. The function record_ncrp_result() then stores it: MATCH becomes USER_CONFIRMED_MATCH and NOT_FOUND becomes USER_CONFIRMED_NOT_FOUND, while any other value raises an error. Only a user-confirmed match adds risk points. This design keeps the tool honest about what it has actually verified.

7.5 Offer-letter screening

An uploaded PDF or DOCX file is read with pypdf or python-docx and the text is scanned with rules and keyword patterns for three groups of clauses: payment demands before joining (deposits, training kits, registration fees); payments to unregistered bank or UPI accounts for buying equipment; and heavy penalty or bond clauses in unpaid internship agreements.

Each clause found is listed with its text so that the applicant can see exactly what triggered the warning. The scan is rule-based and has not been evaluated on a labelled collection of offer letters.

8 Explainable risk engine

8.1 Scoring

Instead of showing a single probability that users cannot interpret, IntelliHireX Employment Intelligence Hub adds points for each warning sign and shows which signs fired. The total is capped at 100:

$$\text{Score} = \min\left(100, \sum_i p_i s_i\right)$$

(6)

The points for each signal are listed in Table 7 and shown graphically in Figure 4.

Table 7. Points assigned to each signal.

Check	Condition	Pts	Why it matters
Text model	Model predicts FAKE	+40	Wording looks like known scams
MCA	Company not found in six-state data	+15	No registration in major hubs
MCA	Found, but status INACTIVE	+10	Company closed or struck off
Email	Free email provider	+10	Recruiter not tied to an organisation
Email	Domain differs from company website	+15	Possible impersonation
Email	Email is not valid	+15	Contact details unusable
Career page	Job title missing on official page	+10	Vacancy may not exist
Career page	Page unreachable or timed out	+5	Cannot be verified
NCRP	User confirms an NCRP match	+30	Identifier already reported

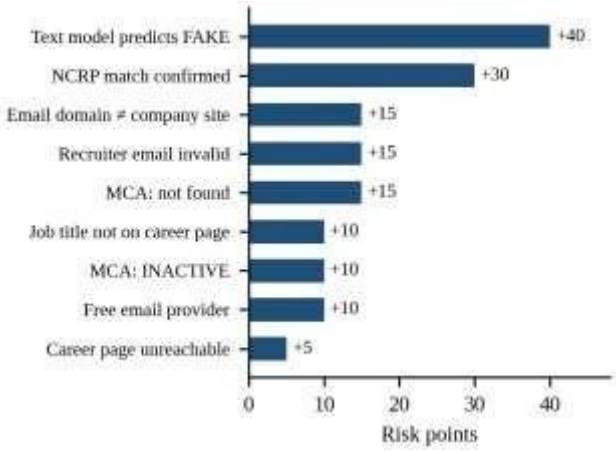


Figure 4. Risk points assigned to each signal. The weights were set by judgment and not learned from data.

8.2 Risk levels

Scores map to three levels. **Low (0–24):** the company, domain and text all look normal, or only a single minor sign appears. **Medium (25–49):** several warning signs appear together or one serious sign appears alone, and the user is asked to take care. **High (50 or more):** several signs combine, and the user is advised not to share personal details or pay any money. To avoid counting the same problem twice, a free email address is scored once as a free-email signal and is not also counted as a domain mismatch.

8.3 Properties of the scoring scheme

Three properties follow directly from the weights and thresholds. First, a text-model prediction of FAKE alone contributes 40 points and therefore yields Medium, never High. High requires at least 10 further points from the other checks, so the system never rests a High rating on the text model alone; this is intentional, given that the text model produced 9 false alarms among 3,403 genuine test postings. Second, a user-confirmed NCRP match alone contributes 30 points, which is Medium; it

needs at least 20 further points to reach High. Third, no single registry, email or career-page signal can reach 25 points by itself, so isolated minor irregularities such as a free email address or a missing registry entry stay Low. These properties make the output conservative: it escalates when evidence from independent sources agrees.

The points were set by hand according to the seriousness of each sign. They were not learned from data, so they are a reasonable starting point rather than an optimum. Replacing them with weights learned from labelled cases is part of the future work.

8.4 Illustrative scenarios

Table 8 applies Algorithm 2 to six constructed scenarios, and Figure 5 plots the resulting scores. The scenarios illustrate how the scheme behaves; they are not measured results on real postings. Scenario S6 represents the impersonation case that motivated the work: the copied text is genuine and scores nothing, yet the free email address, the absent vacancy and a reported identifier together reach High.

Table 8. Constructed scenarios scored by Algorithm 2.

ID	Signals present	Score	Level
S1	Genuine text; company active; company-domain email; title found	0	Low
S2	Genuine text; company not found (15); free email (10)	25	Medium
S3	Text FAKE (40); all other checks clean	40	Medium
S4	Genuine text; not found (15); free email (10); page unreachable (5)	30	Medium
S5	Text FAKE (40); domain mismatch (15); company INACTIVE (10)	65	High
S6	Genuine text; free email (10); title missing (10); NCRP match (30)	50	High

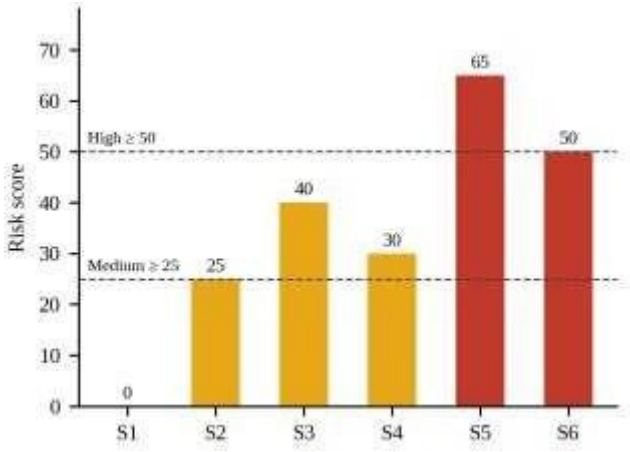


Figure 5. Scores of the constructed scenarios in Table 8 against the Medium and High thresholds.

9 System implementation

9.1 Technology stack

The backend is written in Python 3.13 with FastAPI, which handles asynchronous requests well, and Pydantic v2 for validating request and response data. The machine-learning part uses scikit-learn

[11]. The web interface uses HTML5, CSS and JavaScript. Table 9 lists the main libraries with the minimum versions given in the project's requirements file.

Table 9. Software and libraries.

Library	Version	Used for
Python	3.13	Programming language
FastAPI / Uvicorn	0.110 / 0.29	REST API and web server
scikit-learn	1.3	TF-IDF, models, metrics
pandas / NumPy	2.0 / 1.24	Data handling
joblib	1.3	Saving and loading the model
matplotlib	3.8	Confusion-matrix images
pypdf / python-docx	5.0 / 1.1	Reading offer letters
python-multipart	0.0.9	File upload handling
ReportLab	4.0	PDF report generation
requests	2.31	Fetching web pages
pytest	8.0	Running tests
SQLite	built in	User and session storage

9.2 Architecture

Figure 6 shows the layers of the system. The web application talks to the FastAPI service through REST calls. The service calls the individual modules, which use the trained model file, the MCA company data, company websites and uploaded documents. User accounts are kept in a SQLite database. The cybercrime portal is opened by the user and is not called by the system. Communication uses JSON over HTTP, and offer letters are uploaded as multipart/form-data.

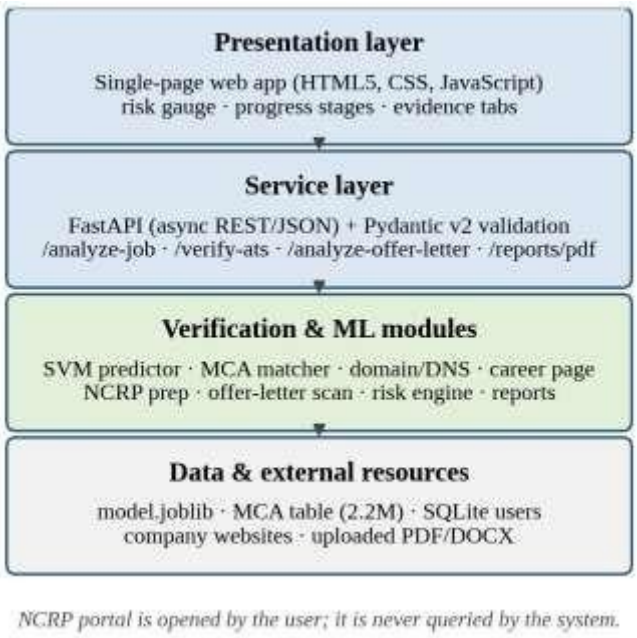


Figure 6. Layered architecture of IntelliHireX Employment Intelligence Hub. The NCRP portal is opened by the user; it is never queried by the system.

9.3 Model training module

The training script loads the processed dataset, builds one scikit-learn pipeline per model so that the TF-IDF vocabulary and the imputation values are learned from the training part only, evaluates each model on the test part, saves a classification report and a confusion-matrix image for each model, and stores the selected model together with its metadata. Algorithm 3 summarises the steps. The Linear SVM does not produce probabilities, so its ROC-AUC is computed from the decision scores.

9.4 API endpoints

Table 10. Main API endpoints.

Endpoint	Function
POST /api/analyze-job	Runs the text model, MCA and domain checks and builds the risk report
POST /api/verify-ats	Checks public Applicant Tracking System pages (for example Lever)
POST /api/analyze-offer-letter	Scans an uploaded PDF or DOCX for risky clauses
POST /api/reports/pdf	Creates a downloadable PDF assessment report

9.5 Web interface and reports

The single-page interface shows live progress while the checks run, a speedometer-style gauge for the 100-point score, and separate tabs that explain each warning in plain language. Two report functions, render_report() and render_pdf(), produce an HTML and a PDF report with the same content in the same order: the overall assessment with the score out of 100, the job and employer details, the explanation of why the result was given, the positive indicators found, the warning signs (each with what was detected, why it matters and the recommended action), the checks that could not be verified, the recommended next actions and a disclaimer. The disclaimer states that the report is an automated screening aid and not a legal finding, and points to the official complaint portal.

9.6 Security and resource requirements

Passwords are stored as PBKDF2 hashes, and user accounts and sessions are kept in SQLite. The system trains and runs on an ordinary CPU-only computer because a Linear SVM needs no GPU. As recommended, not tested, settings, a dual-core 2 GHz processor, 8 GB of RAM and about 1 GB of free storage are sufficient for the datasets, model file and packages, and an internet connection is needed for the domain and career-page checks.

10 Experimental results and discussion

10.1 Experimental setup

All three models were trained on the same 14,304 postings and tested on the same 3,576 postings (3,403 genuine and 173 fake), using the pipeline in Figure 2. The fake class is treated as the positive class. All experiments ran on a standard CPU; no GPU was used. Model selection followed Section 6.5. Results come from a single stratified split with a fixed seed, so no confidence intervals are available.

10.2 Text model results

Table 11 gives the test-set scores, and Figure 7 compares precision, recall and F1-score.

Table 11. Results on the held-out test set (3,576 postings); precision, recall and F1 are for the fake class.

Model	Acc.	Prec.	Recall	F1	ROC-AUC
TF-IDF + Naive Bayes	0.9743	0.8140	0.6069	0.6954	0.9572
TF-IDF + SVD + RF	0.9771	0.9789	0.5376	0.6940	0.9678
TF-IDF + Linear SVM (selected)	**0.9911**	**0.9434**	**0.8671**	**0.9036**	**0.9908**

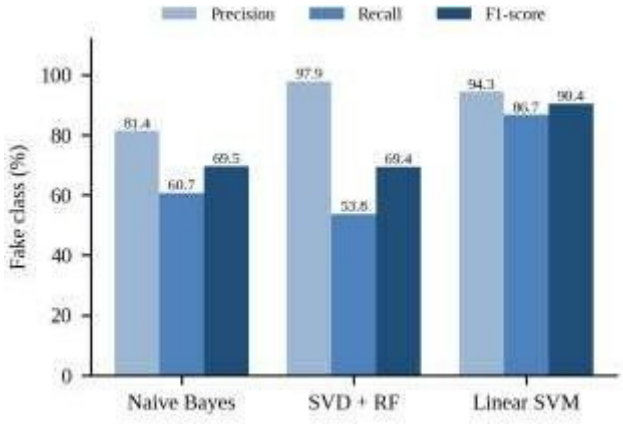


Figure 7. Fake-class precision, recall and F1-score of the three models on the test set.

The Linear SVM gave the best balance. Its F1-score of 90.36% was about 21 points above Naive Bayes (69.54%) and Random Forest (69.40%), and its recall of 86.71% was 26 and 33 points higher respectively. The Random Forest was the most precise model (97.89%) but caught only 53.76% of the scams, so it would let almost half of the fake postings through. Because recall matters most for a tool whose purpose is protecting applicants, and because the other modules add further evidence, the SVM is the better fit for the text-model role.

10.3 Confusion-matrix analysis

Figure 8 shows that the SVM correctly cleared 3,394 genuine postings and intercepted 150 fake ones, with 9 false positives and 23 false negatives. This gives a precision of 150/159 = 94.34% and a recall of 150/173 = 86.71%. Table 12 lists the confusion counts of all three models. For the SVM they are those reported by the training script; for Naive Bayes and Random Forest they were reconstructed from the reported precision, recall and test-set class sizes, and they reproduce the reported accuracy figures exactly.

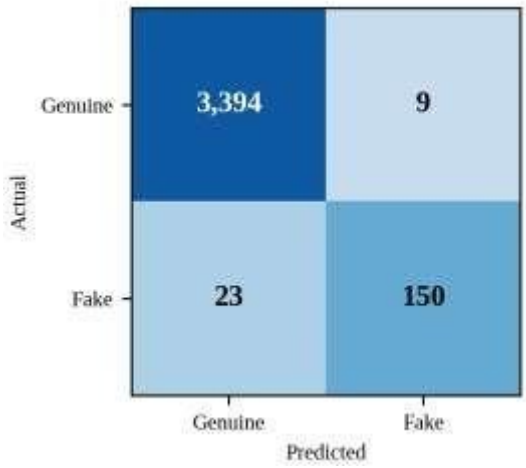


Figure 8. Confusion matrix of the Linear SVM on the test set. Colour is on a logarithmic scale.

Table 12. Confusion counts on the test set.

Model	TP	FP	FN	TN
Naive Bayes	105	24	68	3,379
SVD + Random Forest	93	2	80	3,401
Linear SVM	150	9	23	3,394

Table 13 gives additional measures for the SVM. Its false-alarm rate on genuine postings is 0.26%, so about 2.6 of every 1,000 genuine postings would be wrongly flagged by the text model. Its miss rate on fake postings is 13.29%. The Matthews correlation coefficient of 0.900 and the balanced accuracy of 93.22% confirm that the high accuracy of 99.11% is not an artefact of the majority class, a risk that the precision–recall view also highlights on imbalanced data [15].

Table 13. Additional measures for the Linear SVM.

Measure	Value
Specificity (genuine recall)	99.74%
Negative predictive value	99.33%
False-alarm rate (FP / genuine)	0.26%
Miss rate (FN / fake)	13.29%
Balanced accuracy	93.22%
Matthews correlation coefficient	0.900
F1-score of the genuine class	99.53%

10.4 Discussion

Why the linear SVM performed well. TF-IDF vectors of unigrams and bigrams are high-dimensional and sparse, and fraud postings may reuse recognisable phrases. In such a space a linear boundary with a wide margin is a natural fit, and class weighting directs the optimiser towards the rare class. We did not run ablation experiments, so these are plausible explanations rather than tested conclusions.

Why the Random Forest recalled less. The forest operated on 75 dense components rather than on the 20,000 TF-IDF features. Squeezing the features this far probably removed rare words that are useful for spotting fraud, although we did not test this directly. The very high precision and low recall suggest that the forest flags only postings that are far from the genuine majority in the compressed space.

The meaning of the 23 missed postings. The fake postings that the SVM missed are the reason for the rest of the framework. A text model alone will always let some scams through, particularly those that reuse genuine wording. The verification modules supply evidence that does not depend on the advert text. Whether they in fact recover these 23 cases cannot be said from the current data, because the modules need recruiter details that EMSCAD does not contain. This is the most important open evaluation question and is listed in Section 15.

Relation to earlier results. The scores in Table 1 cannot be compared with ours directly, because of the differences in splits, imbalance handling and data copies noted in Section 2. Our contribution is not a claim to beat those figures, but a leak-free reference result for a simple and fast model, plus the verification layer that text-only studies do not include.

11 System testing

11.1 Testing approach

The project has two types of testing. First, the text model was evaluated on a held-out test set (Section 10). Second, the verification modules have assertion-based test scripts, test_mca_verification.py and test_ncrp_verification.py. Each script runs the module on fixed inputs and uses assert statements to check the outputs; a failed assertion stops the script with an error, and the NCRP script prints "All NCRP verification tests passed." when every check succeeds. The scripts can be run directly with Python or with pytest.

11.2 Test cases

Table 14. Test cases of the MCA and NCRP modules.

ID	Module	Input	Expected result
T1	MCA	An existing company name, in full	found = true
T2	MCA	Same company with extra spaces, small letters and "pvt ltd"	found = true
T3	MCA	A company name that does not exist	found = false
T4	NCRP	Email only	Type EMAIL; lower-case value; NOT_CHECKED; portal link returned
T5	NCRP	Phone "+91 90000 00000"	Type PHONE; "+919000000000"
T6	NCRP	Website link only	Type URL
T7	NCRP	Email, phone, website, social link and handle together	Five identifiers, all NOT_CHECKED
T8	NCRP	No identifiers	Empty list; NOT_CHECKED
T9	NCRP	Result "MATCH" for an email	USER_CONFIRMED_MATCH
T10	NCRP	Result "NOT_FOUND" for a phone number	USER_CONFIRMED_NOT_FOUND

ID	Module	Input	Expected result
T11	NCRP	Invalid result “FAKE”	ValueError raised

Cases T1–T3 check that name normalisation tolerates formatting differences while not inventing matches. Cases T4–T8 check identifier typing and cleaning, and T9–T11 check that only the two permitted user answers are accepted. These are functional tests of module behaviour. They are not measurements of detection accuracy, and the text-model evaluation in Section 10 is the only accuracy measurement in this work. Tests for the domain, career-page and offer-letter modules, and end-to-end tests of the risk engine against Algorithm 2, are planned.

12 Limitations and threats to validity

- **Evaluation covers the text model only.** The MCA, domain, career-page and offer-letter checks have not been tested on labelled data, so their contribution to overall detection accuracy cannot yet be reported.
- **Dataset age and scope.** EMSCAD contains postings from 2012 to 2014 and is not specific to India. Scam wording changes over time, so the model may need retraining on newer data. This is the main threat to external validity.
- **Single split and model choice on the test set.** Results come from one stratified split, and the best of the three models was chosen using test-set scores. Cross-validation, repeated runs and a separate validation set would give a more reliable picture; the reported figures are slightly optimistic.
- **Registry coverage.** The MCA data covers six states. A genuine company registered elsewhere or abroad will show as not found. This is a limit of the data and not proof of fraud, and the system tells users so.
- **Dynamic career pages.** Pages built mainly with client-side JavaScript may not show the job title in a simple page fetch, which can produce a false NOT_FOUND.
- **Hand-set weights.** The points in Table 7 reflect judgment, not learned values, and the thresholds of 25 and 50 are likewise assumptions.
- **Score interpretation.** A Linear SVM gives a distance from the decision boundary and not a calibrated probability, so the system shows no percentage confidence. The risk score is a rule-based summary and not a probability of fraud.
- **Adaptive adversaries.** Scammers who learn what the checks look for may adapt, for example by registering a plausible domain. The layered design raises the cost of impersonation but does not make it impossible.

13 Responsible use and privacy

IntelliHireX Employment Intelligence Hub is an advisory tool. It reports facts such as a domain mismatch or a missing registry entry so that the job seeker can decide with care. It does not accuse any company or person of a crime, and every report carries a disclaimer to that effect and a pointer to the official complaint portal. This matters because a false alarm can damage a genuine employer’s reputation and because a missing registry entry may simply reflect the six-state coverage.

Users enter recruiter emails, phone numbers and offer-letter text, which can be sensitive. The system therefore uses these data only to produce the assessment, stores credentials as PBKDF2 hashes, and does not query any government system on the user's behalf. A deployment for wider use should add explicit data-retention rules, consent notices and encryption of any stored reports. The NCRP step is deliberately manual: the user, not the tool, decides what to search and what to report.

14 Conclusion

This paper presented IntelliHireX Employment Intelligence Hub, a system for detecting fake job and internship offers. Its Linear SVM text model, whose preprocessing was learned from the training data only, reached 99.11% accuracy and a 90.36% F1-score for the fake class on the EMSCAD test set. Since text alone missed 23 of 173 fake postings, the system adds checks on MCA registration, recruiter email and domain, the official career page, NCRP-related identifiers and offer-letter clauses, and combines everything in a 100-point score that explains itself. The work shows that a simple, fast text model becomes more useful to a job seeker when it is combined with checks on who is making the offer. The result is presented as advice, with reasons, and not as a final verdict.

15 Future scope

Future work includes:

- testing the non-text checks on labelled examples, including recovery of the text model's missed cases, and learning the point weights from data;
- reporting cross-validated results with a separate validation set, and evaluating on newer and India-specific datasets;
- extending the company database to all Indian states;
- using a headless browser for JavaScript-heavy career pages;
- allowing users to share scam reports so that new scam patterns can be flagged quickly.

Acknowledgment

The authors thank their project guide, Mr. Yogeshchandra Puranik, Assistant Professor, Department of MCA, for his guidance and support; Prof. Dr. Shivani A. Budhkar, Head of the MCA Department, and Prof. Dr. Mrs. K. R. Joshi, Principal, PES's Modern College of Engineering, Pune, for providing facilities; and the creators of the EMSCAD dataset and the open data sources used in this work.

References

[1] Vidros S, Kolias C, Kambourakis G, Akoglu L. Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset. *Future Internet*. 2017;9(1):6.

[2] Lal S, Jiaswal R, Sardana N, Verma A, Kaur A, Mourya R. ORFDetector: Ensemble learning based online recruitment fraud detection. In: *Proceedings of the 12th International Conference on Contemporary Computing (IC3)*; 2019. https://doi.org/10.1109/IC3.2019.8844879

[3] Pillai AS. Detecting fake job postings using bidirectional LSTM. International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS). 2023;5(3). *arXiv*:2304.02019

[4] Kaufman S, Rosset S, Perlich C, Stitelman O. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*. 2012;6(4):15.

[5] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*. 2002;16:321–357.

[6] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*. 1988;24(5):513–523.

[7] Manning CD, Raghavan P, Schütze H. *Introduction to information retrieval*. Cambridge (UK): Cambridge University Press; 2008.

[8] Cortes C, Vapnik V. Support-vector networks. *Machine Learning*. 1995;20(3):273–297.

[9] Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*. 2008;9:1871–1874.

[10] Breiman L. Random forests. *Machine Learning*. 2001;45(1):5–32.

[11] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*. 2011;12:2825–2830.

[12] Ministry of Corporate Affairs, Government of India. *Master data of registered companies* [Internet]. Open Government Data (OGD) Platform India. https://data.gov.in/

[13] Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*. 1966;10(8):707–710.

[14] Indian Cyber Crime Coordination Centre (I4C), Ministry of Home Affairs, Government of India. *National Cyber Crime Reporting Portal* [Internet]. https://cybercrime.gov.in/

[15] Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE*. 2015;10(3):e0118432.

[16] Joju D. *Fake job detection system using machine learning and NLP. International Journal of Innovative Research in Technology (IJIRT)*. 2026;13(2).

[17] Radhika E, Sri Manaswini P, Vishal Reddy K, Praveen G, Sai Siddu K. *Detecting recruitment scams using NLP and machine learning techniques*. International Journal of Engineering Science and Advanced Technology (IJESAT). 2026;26(4).

[18] Pitumpe A, Rahmati A. Anansi: Scalable characterization of message-based job scams [Preprint]. *arXiv*:2602.24223; 2026.

[19] Saranya A, Abinaya S, Kalanidhi A, Kannan M, Sharon Rufina B. *A systematic and comprehensive survey on deep learning-based fake recruitment email detection*. International Journal of Creative Research Thoughts (IJCRT). 2026;14(2).

[20] Ashok Kumar P, Likitha G, Nithin B, Akash A, Saikumar G. *Detection of fake online recruitment using machine learning*. International Journal of Scientific Development and Research (IJSDR). 2024;9(5).

[21] Ullah Z, Jamjoom M. A smart secured framework for detecting and averting online recruitment fraud using ensemble machine learning techniques. *PeerJ Computer Science*. 2023;9:e1234. https://doi.org/10.7717/peerj-cs.1234

[22] Goyal N, Sachdeva N, Kumaraguru P. *Spy the lie: Fraudulent jobs detection in recruitment domain using knowledge graphs* [Internet]. PreCog, IIIT Delhi; 2021. https://cdn.iiit.ac.in/cdn/precog.iiit.ac.in/pubs/2021_June_Spy_the_lie.pdf

[23] Anand P, Sharma V. *Job portal end detection of fake job posting using machine learning*. Compliance Engineering Journal. 2020;11(10).