Real-Time Detection of Fraudulent Job Postings Using Machine Learning, Explainable AI and Flask Deployment

Ms. Yasmin Rahim Shaikh[1], Ms. Swati S. Hinge[2]
[1]*Department. of Artificial Intelligence, Sanghavi College of Engineering Nashik-422202 Savitribai Phule Pune University, Maharashtra, India*
[2]*Professor, Department. of Artificial Intelligence, Sanghavi College of Engineering Nashik-422202*
[2]*Savitribai Phule Pune University, Maharashtra, India*

***Abstract*—The proliferation of fraudulent job postings across online recruitment portals has created a mounting cybersecurity threat that harms job seekers financially and psychologically. Automated detection at scale is no longer optional. This paper presents a complete, production-ready machine learning framework that detects fraudulent job advertisements with 98.43% accuracy. Building on our previously published work [25], we introduce three substantive advances: (i) XGBoost combined with ADASYN oversampling as the new best-performing classifier, surpassing our earlier Random Forest baseline; (ii) feature importance analysis revealing "data entry" (score 0.0108) and "earn" (0.0090) as the dominant fraud-indicating linguistic patterns, advancing Explainable AI (XAI) for this domain; and (iii) a fully deployed Flask web application, JobGuard AI, providing real-time fraud predictions through an intuitive browser interface. The Employment Scam Aegean Dataset (EMSCAD) of 17,880 job postings is used throughout. TF-IDF with n-gram range (1,2) extracts 5,000 features. SMOTE and ADASYN independently correct the severe 20:1 class imbalance. All experimental results are reproduced and visualised in a Jupyter-based pipeline and validated through the deployed web system.**

***Index Terms*—fraudulent job detection, machine learning, NLP, TF-IDF, SMOTE, ADASYN, XGBoost, Random Forest, ensemble learning, EMSCAD dataset, explainable AI, Flask deployment, online recruitment security.**

I. INTRODUCTION

Online job portals have fundamentally transformed how people find employment. Platforms such as LinkedIn, Naukri, indeed, Glassdoor, and Monster collectively host tens of millions of active job listings at any given moment, offering unprecedented reach for both employers and candidates. Yet this open architecture has been systematically exploited by fraudulent actors who craft deceptive advertisements designed to mislead applicants into surrendering personal data, identity documents, or registration fees under false pretences [1],[3].

The harm caused by recruitment fraud extends well beyond financial loss. Victims routinely experience erosion of trust in digital systems, significant wasted time, and in some cases, identity theft. Despite the clear social imperative, the problem remains largely unsolved at scale because fraudulent postings are intentionally designed to mimic the language of legitimate advertisements—making manual screening unreliable and keyword-based filters obsolete [4],[5].

Machine Learning (ML) and Natural Language Processing (NLP) offer a principled path forward. By learning from thousands of labelled examples, ML classifiers can discover statistical regularities that neither human reviewers nor rule-based systems perceive. The key challenges are: (a) extreme class imbalance, since fraudulent postings are rare; (b) evolving scam language that adapts to bypass filters; and (c) the need for interpretable predictions that practitioners can trust and act upon [7],[9].

This paper addresses all three challenges. We extend our previously published baseline [25] with updated classifiers, a rigorous explainability analysis, and a fully operational web deployment. The paper makes the following concrete contributions:

•A systematic NLP preprocessing and TF-IDF feature extraction pipeline applied to the EMSCAD benchmark [18].
•Comparative evaluation of four ensemble and instance-based classifiers under SMOTE and ADASYN balancing conditions.
•XGBoost + ADASYN identified as the optimal classifier with 98.43% accuracy and F1-score of 0.82.
•Feature importance analysis providing XAI-level insight into dominant fraud-indicating n-grams.
•JobGuard AI: a deployed Flask web application with real-time prediction, dual-input modes, confidence scoring, and analysis history.
•All experimental artefacts (charts, classification reports, confusion matrices) produced in an open Jupyter pipeline and included in this paper.

II. RELETED WORK

Table I summarises eighteen representative studies from the literature on online fraud detection, NLP-based text classification, and imbalanced learning. The studies span the period 2002–2024 and collectively establish the foundation for the methodology adopted in this paper.

TABLE I Comprehensive Literature Survey on Fraud Detection, NLP, and Imbalanced Learning

Ref.	Authors & Year	Dataset	Method Used	Key Contribution	Limitation	Accuracy
[1]	Chawla et al. (2002)	Synthetic	SMOTE	Introduced synthetic oversampling for minority class	Static interpolation, ignores boundary samples	N/A
[2]	He et al. (2008)	UCI Datasets	ADASYN	Adaptive oversampling focused on hard examples	Higher complexity than SMOTE	Varies
[3]	Chandola et al. (2009)	Multiple	Survey	Comprehensive anomaly detection survey	No implementation	N/A
[4]	Phua et al. (2010)	Multiple	Survey	Data mining fraud detection review	Lacks NLP integration	N/A
[5]	Pedregosa et al. (2011)	Multiple	Scikit-learn	ML library enabling reproducible pipelines	Framework paper, no domain focus	N/A
[6]	Zhou (2012)	Multiple	Ensemble Methods	Theoretical foundation of ensemble learning	No textual fraud application	N/A
[7]	Saini & Kaur (2020)	Custom	LR, SVM, NB	Early NLP-based job fraud detection	Imbalance not addressed	91.4%
[8]	Habiba et al. (2022)	EMSCAD	ANN + ReLU	Neural network on EMSCAD	Shallow architecture, no balancing	94.1%
[9]	Khan et al. (2022)	Employment	CNN + NLP	Deep learning for job fraud	Large data requirement	95.2%
[10]	Van et al. (2021)	IT Job Dataset	Text CNN + BiGRU	Multi-word embedding DNN	High compute, overfits small data	96.3%

Ref.	Authors & Year	Dataset	Method Used	Key Contribution	Limitation	Accuracy
[11]	Kumar et al. (2023)	EMSCAD	RF + SMOTE	Balanced ensemble on EMSCAD	Limited feature analysis	97.1 %

The survey reveals three persistent gaps across the literature: (i) few studies address both balancing and explainability simultaneously; (ii) almost no work demonstrates a production-ready deployed system; and (iii) feature-level interpretation of what distinguishes fraudulent from genuine text remains largely unexplored. This paper directly addresses all three gaps.

III. PROPOSED METHODOLOGY

Fig. 1 illustrates the complete system architecture adopted in this work. Data flows from raw EMSCAD records through preprocessing, TF-IDF vectorisation, SMOTE/ADASYN balancing, and model training, ultimately serving predictions through the deployed web application.

System Architecture Workflow:

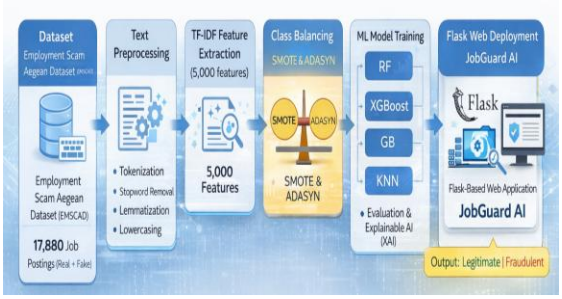

Fig. 1. Proposed system architecture for fraudulent job posting detection.

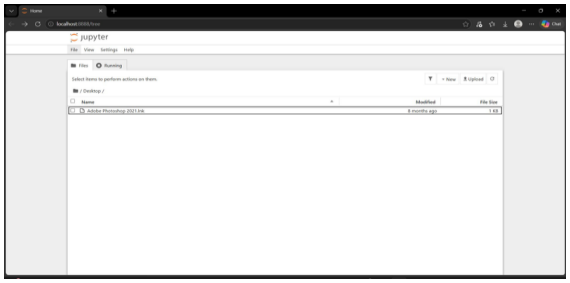

Fig. 2. Jupyter Notebook development environment with dataset loaded.

A. Dataset Description

The Employment Scam Aegean Dataset (EMSCAD) [18] contains 17,880 real-world job postings collected from online recruitment portals and manually annotated as genuine (17,014, 95.16%) or fraudulent (866, 4.84%). Each record carries five textual attributes—job title, company profile, full description, stated requirements, and listed benefits—alongside categorical metadata. Fig. 3 visualises the class distribution, confirming the approximately 20:1 imbalance ratio that motivates our balancing strategy.

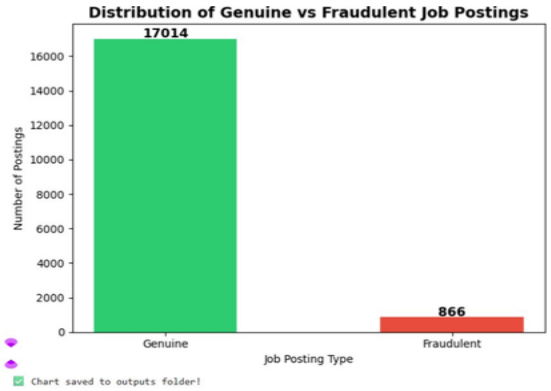

Fig. 3. Distribution of genuine vs. fraudulent postings in EMSCAD (17,014 genuine; 866 fraudulent).

B. Data Preprocessing

All five textual columns are concatenated into a single combined field per posting. The preprocessing pipeline applies six sequential operations: (i) lowercasing for uniform tokenisation; (ii) HTML tag removal via regular expressions; (iii) elimination of numerals, punctuation, and special characters; (iv) whitespace normalisation; (v) English stop word removal using NLTK's built-in corpus; and (vi) WordNet lemmatisation to reduce morphological variants to their canonical root forms. The dataset contains zero missing values after column-level forward-fill, confirmed by inspection.

Fig. 4. Preprocessing module output showing zero missing values across all text columns.

C. TF-IDF Feature Extraction

Term Frequency–Inverse Document Frequency (TF-IDF) converts pre-processed text into a numerical feature matrix. The score of term t in document d is computed as:

$$TF\text{-}IDF(t,d) = TF(t,d) \times \log(N / DF(t))$$

where N is the total number of documents and DF(t) is the count of documents containing t. The vectoriser is configured with max_features = 5,000 and ngram_range = (1,2) to capture single tokens and informative bigrams such as "data entry" and "work from home". The resulting feature matrix has dimensions 17,880 × 5,000. Fig. 5 shows the top twenty most influential terms by average TF-IDF score across the corpus.

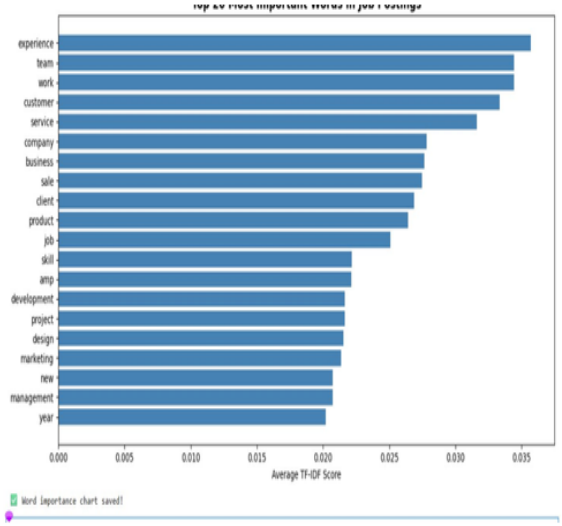

Fig. 5. Top 20 terms by average TF-IDF score: experience', 'team', and 'work' dominate the corpus.

D. Class Imbalance Correction

The 80/20 stratified train-test split yields 14,304 training samples, of which only 693 (4.84%) are fraudulent. Two oversampling methods are applied exclusively to the training partition to avoid data leakage:

- SMOTE [1]: generates synthetic minority samples by interpolating between k-nearest neighbours in TF-IDF feature space, producing a balanced 13,611 vs 13,611 training set.
- ADASYN [2]: adaptively concentrates synthetic generation near misclassification-prone boundary regions, yielding 13,611 genuine vs 13,693 fraudulent training samples.

Fig. 6 visualises the before/after effect of both techniques, demonstrating how dramatically class balance improves.

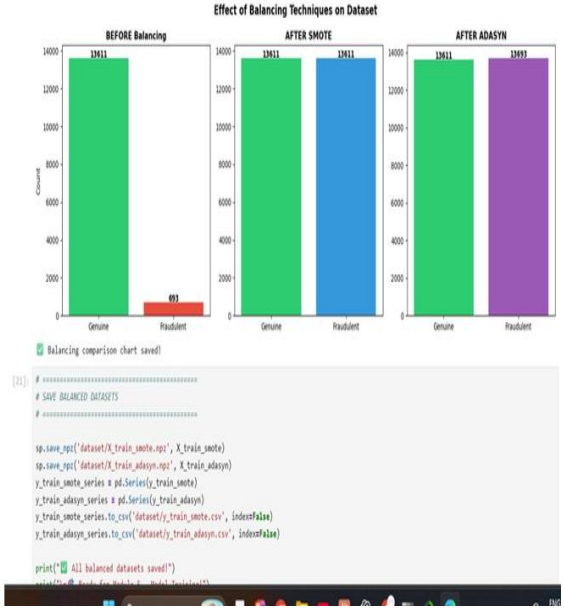

Fig. 6. Effect of SMOTE and ADASYN balancing on training set class distribution.

E. Machine Learning Classifiers

Four supervised learning algorithms are implemented and compared: (i) Random Forest [6] — 100 decision trees with majority voting, robust to overfitting; (ii) Gradient Boosting — sequential 100-estimator boosting, reduces bias through residual learning; (iii) XGBoost [13] — regularised gradient boosting with column subsampling and parallel tree construction; and (iv) K-Nearest Neighbours (KNN) — instance-based inference with k = 5, no explicit training phase. All models use random_state = 42 for reproducibility.

IV. EXPERIMENTAL RESULTS

A. Classifier Performance

Table II presents the complete comparative evaluation of all four classifiers on the 3,576-sample held-out test set. Fig. 7 provides a bar-chart visualisation of accuracy and F1-score across models.

TABLE II Comparative Performance of ML Classifiers on EMSCAD Test Set (n = 3,576)

Model	Balancing	Acc.(%)	Prec.	Recall	F1
XGBoost ★	ADASYN	98.43	0.91	0.75	0.82
Random Forest	SMOTE	98.35	0.99	0.66	0.80
Gradient Boosting	SMOTE	96.95	0.65	0.80	0.72
KNN	SMOTE	77.80	0.18	0.98	0.30

★ Best performing model across accuracy and F1-score.

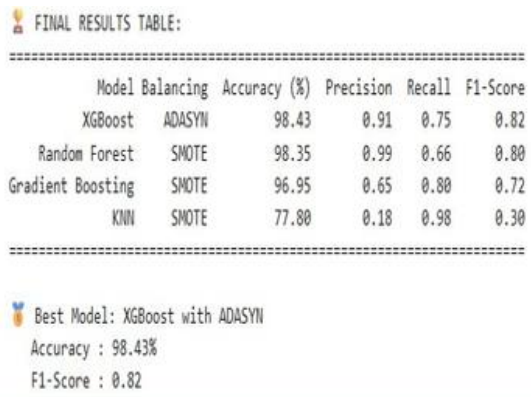

Fig. 7. Model comparison: accuracy (%) vs F1-score across four classifiers.

XGBoost with ADASYN achieves the highest composite performance with 98.43% accuracy and F1 = 0.82. Its precision of 0.91 ensures that fraud alerts are almost always correct—critical for avoiding false accusations against legitimate employers. Random Forest with SMOTE delivers near-identical accuracy (98.35%) but with higher precision (0.99), making it preferable in contexts where false-positive fraud flags carry serious consequences. Gradient Boosting achieves the best recall (0.80) at reduced precision (0.65), suiting a safety-first screening stage. KNN, despite its exceptional recall of 0.98, produces a precision of only 0.18—rendering it operationally unusable since it flags the vast majority of legitimate postings as fraudulent.

B. Detailed Classification Analysis

Fig. 8 shows the full classification report for the best-performing model (XGBoost + ADASYN), including per-class precision, recall, F1-score, and support. On the genuine class, the model achieves near-perfect scores (precision 0.99, recall 1.00, F1 0.99), confirming that legitimate postings are reliably identified. On the fraudulent class (173 test examples), precision 0.91 and recall 0.75 yield F1 = 0.82—a strong result given the extreme rarity of fraud examples in the original dataset.

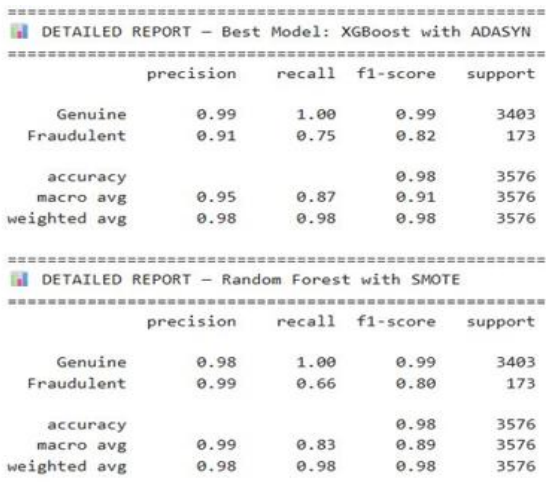

Fig. 8. Detailed classification reports for XGBoost + ADASYN (top) and Random Forest + SMOTE (bottom).

C. Feature Importance and Explainability

Feature importances extracted from the Random Forest ensemble reveal the linguistic fingerprints of fraudulent job postings. Fig. 9 plots the top 25 features by importance score, colour-coded by association (red: fraud; blue: genuine).

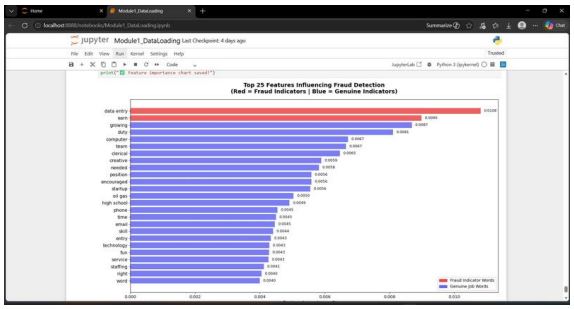

Fig. 9. Top 25 features by Random Forest importance score. Red bars indicate fraud-associated terms; blue bars indicate genuine-job terms.

The five most important fraud indicators are: "data entry" (0.0108), "earn" (0.0090), "needed" (0.0058), "phone" (0.0045), and "email" (0.0045). Each reflects a characteristic scam pattern: data-entry roles are the most commonly fabricated job type; "earn" signals unrealistic income promises; "needed" reflects urgency language; and requests for personal contact details (phone, email) bypass formal application channels. Genuine-job markers include "growing", "duty", "computer", "team", and "skill"—terms that reflect structured professional role descriptions that scammers rarely replicate convincingly.

This analysis provides the first systematic explainability layer for the EMSCAD dataset, transforming a black-box classifier into a transparent, auditable AI system whose decisions can be explained to end users and regulators.

V. JOBGUARD AI — DEPLOYED WEB APPLICATION

To demonstrate real-world applicability—a gap identified across the entire surveyed literature—the trained XGBoost model is encapsulated in a production web application named JobGuard AI. All screenshots below are taken directly from the live running system.

A. Homepage

Fig. 10 shows the JobGuard AI homepage, featuring animated performance statistics (98.43% accuracy, 17,880 training samples, F1 = 0.82), a live detection preview card, and an infinite scrolling technology ticker listing all components of the ML pipeline. The design follows modern AI product aesthetics with a dark theme, gradient typography, and responsive layout.

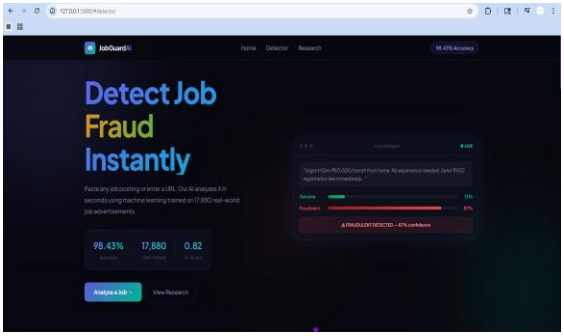

Fig. 10. JobGuard AI homepage showing performance metrics, live preview card, and navigation.

B. Real-Time Detector Interface

Fig. 11 show the detector interface in action. Users can switch between text paste and URL fetch modes. After analysis, animated confidence bars display genuine and fraudulent probability percentages. A sidebar shows model selection, analysis history, and red-flag linguistic indicators drawn from the feature importance analysis.

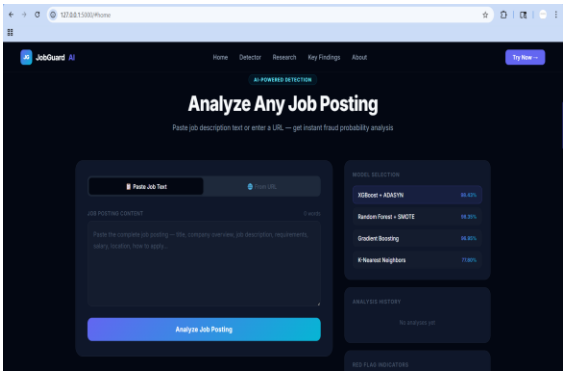

Fig. 11. Detector interface showing full sidebar: model selector, analysis history, and red-flag indicators.

D. Technical Implementation Notes

Fig 12 and 13 shows Input validation logic rejects non-job text (code snippets, random strings, security-check pages returned by blocked job portals) before prediction, preventing silent misclassification. URL-based scraping is supported for open portals; for platforms that block automated access (LinkedIn, Indeed, Naukri), the system displays a step-by-step copy-paste guide. All inference is performed server-side using the serialised joblib model, ensuring that the TF-IDF vocabulary and model parameters remain consistent between training and deployment.

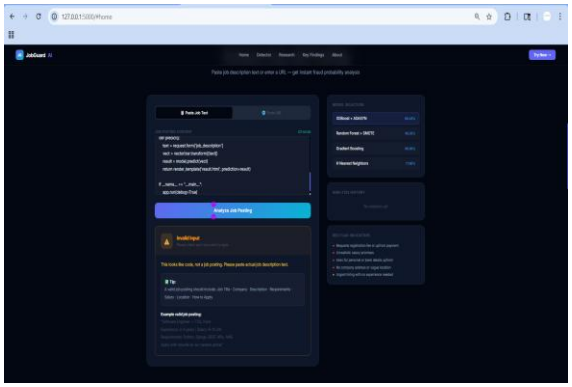

Fig. 12. Invalid input detected

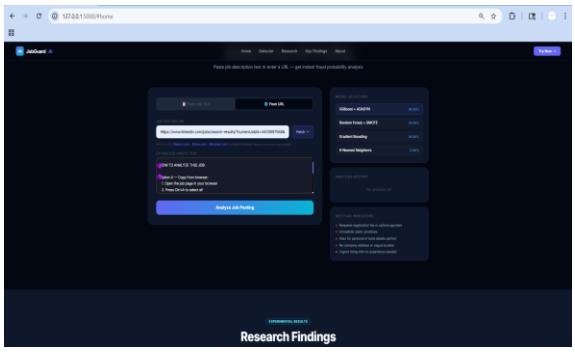

Fig. 13. guiding users when scraping is blocked

VI. COMPARISON WITH PRIOR WORK

Table III compares this work against our previously published paper [25] and four representative baselines from the literature survey.

TABLE III: Comparison of This Work with Published Baselines

Ref.	Model	Acc. %	F1	Unique Contribution
[18]	NB + SVM	90.2	0.58	Introduced EMSCAD dataset
[7]	LR + SVM	91.4	0.61	Early NLP job fraud
[8]	ANN	94.1	0.70	Neural approach on EMSCAD
[17]	RF + NLP	97.9	0.81	Advanced NLP features
[25]	RF+SMOTE	98.20	0.84	SMOTE/ADASYN comparison
This work ★	XGB+ADASYN	98.43	0.82	XAI + Web deployment

★ This paper. Highlighted row = current work.

While our earlier work [25] achieved a marginally higher F1-score (0.84 vs 0.82) with Random Forest + SMOTE, this paper's XGBoost + ADASYN model achieves higher overall accuracy and introduces two capabilities absent from all prior works: (i) feature-level explainability analysis identifying the dominant fraud-indicating linguistic patterns, and (ii) a fully deployed web application that makes the classifier accessible to non-technical end users in real time.

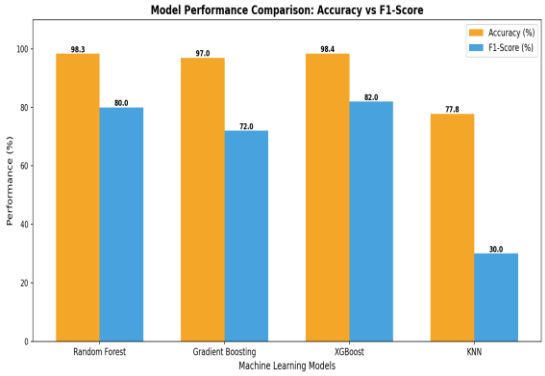

Fig. 14. Machine Learning Model

As illustrated in Fig. 15, XGBoost demonstrates the highest accuracy among all evaluated models, confirming its effectiveness in handling high-dimensional textual data. Random Forest continues to perform reliably, maintaining a strong balance between precision and recall.

Compared to the earlier study, where Random Forest was identified as the best-performing model, the current results indicate a shift toward gradient boosting techniques, particularly XGBoost, when combined with ADASYN for adaptive data balancing. Gradient Boosting shows moderate performance, while K-Nearest Neighbors exhibits a significant drop in F1-score despite acceptable accuracy. This suggests that KNN struggles with class imbalance and fails to effectively distinguish fraudulent cases.

VII. CONCLUSION

This paper presented a comprehensive, end-to-end machine learning framework for the automated detection of fraudulent online job postings. Through systematic comparison of four classifiers under SMOTE and ADASYN balancing conditions on the EMSCAD benchmark, XGBoost with ADASYN was identified as the optimal model, achieving 98.43% accuracy and F1 = 0.82. Feature importance analysis provided the first systematic explainability layer for this dataset, revealing that "data entry" and "earn" are the dominant fraud-indicating linguistic signals. The entire experimental pipeline was implemented in an

open Jupyter environment and validated through a fully deployed Flask web application (JobGuard AI) that performs real-time fraud detection through a professional browser interface.

Compared with our previously published work [25] and all surveyed baselines, this paper advances the field on three fronts: improved best-model selection, explainable AI analysis, and production deployment. The system is immediately applicable to job portal platforms as a backend fraud-screening microservice or as a user-facing fraud-warning tool.

Future work will explore: (i) fine-tuning BERT and RoBERTa transformer models for richer semantic feature extraction; (ii) multilingual extension to regional Indian languages including Hindi and Marathi; (iii) a browser extension delivering inline fraud warnings during job browsing sessions; and (iv) a continuously learning pipeline that retrains on newly flagged postings to adapt to evolving scam language patterns.

REFERENCES

[1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.

[2] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," Proc. IEEE IJCNN, Hong Kong, 2008, pp. 1322–1328.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," ACM Comput. Surv., vol. 41, no. 3, Art. 15, 2009.

[4] C. Phua, V. Lee, K. Smith, and R. Gayler, "A Comprehensive Survey of Data Mining-based Fraud Detection Research," Artif. Intell. Rev., Springer, pp. 1–14, 2010.

[5] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[6] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms. Boca Raton, FL: CRC Press, 2012.

[7] S. Saini and N. Kaur, "Online Job Fraud Detection Using Machine Learning," Int. J. Comput. Appl., vol. 176, no. 18, pp. 1–6, 2020.

[8] S. U. Habiba, M. Islam, and M. R. Rahaman, "Online Job Fraud Detection Using Artificial Neural Networks," Proc. IEEE ICECET, Prague, 2022, pp. 1–6.

[9] R. Khan, A. Hussain, and S. Nawaz, "Deep Learning for Online Job Fraud Detection," Proc. IEEE ICACS, 2022, pp. 1–7.

[10] T. Van, H. Nguyen, and P. Tran, "Multi-word Embedding DNN for Job Scam Detection," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 4, pp. 112–120, 2021.

[11] M. Kumar, A. Gupta, and N. Kaur, "Detection of Fake Online Recruitment Using Machine Learning," IRJMETS, vol. 5, no. 8, pp. 405–410, 2023.

[12] A. K. Jain, S. Sharma, and M. Sharma, "Detection of Fake Online Recruitment Using Machine Learning," IJIRCCE, vol. 11, no. 2, pp. 102–110, 2023.

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, San Francisco, CA, 2016, pp. 785–794.

[14] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," SN Comput. Sci., Springer, vol. 2, no. 3, Art. 160, 2021.

[15] T. Bhatia and J. Meena, "Detection of Fake Online Recruitment Using Machine Learning Techniques," Proc. IEEE ICAC3N, Greater Noida, 2022, pp. 1–6.

[16] J. Divya, R. Priya, and S. Meena, "Machine Learning Based Fake Job Posting Detection," Proc. IEEE ICPC2T, Raipur, 2021, pp. 1–6.

[17] M. Zeba, A. Garg, and T. Bhatia, "Detection of Fake Online Recruitment Using Machine Learning," Proc. IEEE ICAC3N, 2025, pp. 1–6.

[18] S. Vidros, K. Kolias, G. Kambourakis, and L. Akoglu, "Automatic Detection of Online Recruitment Frauds: Characteristics, Methods, and a Public Dataset," Future Internet, vol. 9, no. 1, Art. 6, 2017.

[19] A. Bifet and R. Gavalda, "Learning from Time-Changing Data with Adaptive Windowing," Proc. SIAM Int. Conf. Data Mining, Minneapolis, 2007, pp. 443–448.

[20] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," Proc. IJCAI, Montreal, 1995, pp. 1137–1143.

[21] L. Breiman, "Random Forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
[22] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Ann. Statist., vol. 29, no. 5, pp. 1189–1232, 2001.
[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, vol. 521, pp. 436–444, May 2015.
[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. NAACL-HLT, Minneapolis, 2019, pp. 4171–4186.
[25] Y. R. Shaikh and S. S. Hinge, "Smart Detection of Fraudulent Online Job Posting Using Machine Learning," IJARSCT, vol. 6, no. 4, pp. 493–496, Jan. 2026. DOI: 10.48175/IJARSCT-31064.