IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Authenticity Verification Of Job And Internship Postings Using Ml And Nlp

[1]Dr.G.Aparna, [2]Atthota Jyothika, [3]Kommineni Phani Sai, [4]Muddeti Gayatri Abhilasha, [5]Pottolla Pranusha

[1]Associate Professor, Hyderabad Institute of Technology and Management, Medchal Telangana

[2]UG Student, Hyderabad Institute of Technology and Management, Medchal, Telangana

[3]UG Student, Hyderabad Institute of Technology and Management, Medchal, Telangana

[4]UG Student, Hyderabad Institute of Technology and Management, Medchal, Telangana

[5]UG Student, Hyderabad Institute of Technology and Management, Medchal, Telangana

Abstract: Job and internship fraud is a growing concern on digital recruitment platforms, often leading to financial loss and exploitation of applicants. This paper presents an AI-based authenticity verification system designed to detect and flag suspicious job and internship postings. The proposed model leverages Natural Language Processing (NLP) techniques to analyze posting content, recruiter details, and linguistic cues, while machine learning classifiers are trained on datasets of genuine and fraudulent postings to identify anomalies. The system generates a reliability score and alerts users to potential risks, thereby ensuring a safer application process. Experimental results demonstrate that this approach significantly improves trust and security in online career ecosystems, providing a practical solution to the increasing problem of fraudulent employment opportunities.

Keywords: Job Authenticity Verification, Internship Fraud Detection, Machine Learning, Natural Language Processing, Online Career Platforms.

1.INTRODUCTION

The increasing reliance on digital recruitment platforms has transformed how individuals seek employment and internship opportunities. While these platforms provide accessibility and convenience, they have also become a medium for fraudulent postings that exploit unsuspecting applicants. Such scams often result in financial losses, identity theft, and a decline in trust toward online career services. The challenge is particularly severe for students and early-career professionals who may struggle to distinguish between authentic and deceptive postings.

Conventional verification mechanisms, including manual moderation and user reporting, are limited in scalability and effectiveness given the high volume of postings generated daily. Therefore, an intelligent automated system is essential to ensure the authenticity of employment opportunities on online platforms. This paper presents an AI-driven authenticity verification system that employs Machine Learning (ML) and Natural Language Processing (NLP) techniques to analyze job and internship postings. By extracting linguistic features, recruiter metadata, and anomaly patterns, the proposed system assigns a reliability score and flags suspicious entries. The approach enhances applicant safety, reduces fraud-related risks, and improves trust in digital recruitment ecosystems.

2. LITERATURE SURVEY

The detection of fraudulent job and internship postings has been widely studied using both classical machine learning and modern deep learning techniques. Early approaches primarily relied on traditional text-processing methods such as TF–IDF and n-gram features combined with classifiers like Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines (SVM). These methods demonstrated baseline effectiveness in distinguishing authentic postings from fraudulent ones [1], [2].

Recent studies have explored deep learning models that capture contextual and sequential patterns in job descriptions. Bidirectional Long Short-Term Memory (Bi-LSTM) networks have shown improved performance by leveraging sequential dependencies within text data, leading to higher accuracy and AUC scores compared to classical models [3].

Furthermore, transformer-based architectures such as BERT have advanced the field by providing richer semantic understanding and detecting subtle linguistic cues often present in deceptive postings [4], [5].

Researchers have also emphasized the importance of metadata features such as recruiter information, salary ranges, and contact patterns. Handling class imbalance using techniques like SMOTE has been identified as essential for robust performance. Despite promising progress, challenges remain in cross-platform generalization and real-world scalability, motivating the hybrid approach proposed in this work.

3. METHODOLOGY

The proposed authenticity verification system follows a structured hybrid pipeline that integrates Natural Language Processing (NLP), machine learning (ML), and deep learning (DL) techniques to accurately detect fraudulent job and internship postings. The overall workflow consists of data acquisition, preprocessing, feature engineering, model development, and evaluation. The design focuses on maximizing detection accuracy while maintaining scalability for real-world recruitment platforms.

3.1 Data Collection

A labeled dataset containing both genuine and fraudulent job and internship postings was utilized. The data was compiled from publicly available recruitment datasets and supplemented with synthetically generated fraudulent samples to improve class balance and diversity. Each record in the dataset typically contains the job title, company profile, job description, requirements, salary information, recruiter contact details, and posting metadata. To ensure model robustness, duplicate entries and incomplete records were removed. The dataset was then shuffled and split into training, validation, and testing subsets using an 80:10:10 ratio. Stratified sampling was applied to preserve the proportion of fraudulent and genuine postings across all splits.

3.2 Data Processing

Raw job descriptions often contain noise, inconsistent formatting, and irrelevant symbols. Therefore, a comprehensive preprocessing pipeline was implemented to standardize the textual data. First, text normalization was performed by converting all characters to lowercase and removing HTML tags, URLs, special characters, and numeric noise that do not contribute to semantic meaning. Stop-word removal was applied using the NLTK stop-word corpus to eliminate frequently occurring but semantically weak words.

Next, tokenization was performed to split the text into individual tokens. Lemmatization using spaCy was applied to reduce words to their base form while preserving contextual meaning. Compared to stemming, lemmatization provides better linguistic consistency for downstream models. Finally, the cleaned text was transformed into numerical representations using two complementary techniques: TF–IDF vectorization to capture term importance Word embeddings (Word2Vec/BERT embeddings) to capture semantic relationships This dual representation helps the system detect both surfacelevel and contextual fraud patterns.

3.3 Feature Engineering

To improve discriminative power, additional derived features were computed from the cleaned dataset.

Linguistic features include: Lexical diversity (type–token ratio), Average sentence length, Frequency of promotional or urgency keywords, Part-of-speech (POS) distribution patterns,Readability scores, Fraudulent postings often exhibit exaggerated language, urgency cues (e.g., "apply immediately"), and abnormal writing patterns. Capturing these signals improves classification performance.

Behavioral metadata features include: Recruiter posting repetition score, Domain trust score, Salary deviation index, All features were concatenated into a unified feature vector before model training.

3.4 Model Development

The system evaluates both classical machine learning models and deep learning architectures to identify the most effective approach.

Classical Machine Learning Models
The following baseline models were implemented using scikitlearn:
Logistic Regression, Random Forest, Support Vector Machine (SVM). These models were trained on TF–IDF and engineered metadata features. Hyperparameters such as regularization strength, tree depth, and kernel type were optimized using grid search with cross-validation.

Deep Learning Models
To capture contextual semantics, two deep learning approaches were implemented.

Bi-LSTM:
A Bidirectional Long Short-Term Memory network was trained on Word2Vec embeddings. The bidirectional architecture enables the model to capture both past and future context within job descriptions. Dropout regularization was applied to prevent overfitting.

Transformer-Based BERT Model:
A pretrained BERT base model was fine-tuned on the job posting dataset. The final classification head was replaced with a dense layer followed by softmax activation. Fine-tuning allows the model to adapt pretrained linguistic knowledge to the fraud detection task. Training was performed using the Adam optimizer with an empirically selected learning rate. Early stopping based on validation loss was used to prevent overfitting.

3.5 Evaluation and Verification

Models were evaluated using Accuracy, Precision, Recall, F1- score, and ROC-AUC. A reliability score was generated for each posting, and threshold-based flagging was applied. Crossvalidation ensured robustness and generalization.

IV. IMPLEMENTATION

The proposed authenticity verification system was implemented as a modular machine learning pipeline designed for scalability, reproducibility, and real-time deployment. The implementation phase converts the conceptual methodology into a working prototype capable of detecting fraudulent job and internship postings with high accuracy.

4.1 Development Environment

The system was developed using Python 3.10, selected for its extensive ecosystem in natural language processing and machine learning. The implementation leveraged multiple specialized libraries to handle different stages of the pipeline. Natural language preprocessing tasks such as tokenization, stop-word removal, and lemmatization were performed using NLTK and spaCy. Classical machine learning models were implemented using scikit-learn, while deep learning architectures including Bi-LSTM and BERT were developed using TensorFlow/Keras and PyTorch respectively.

For deployment, the Flask framework was used to create a lightweight web interface that enables real-time prediction. The experiments were conducted on a workstation equipped with an Intel i7 processor, 16 GB RAM, and optional NVIDIA GTX GPU acceleration to speed up deep learning training.

4.2 Dataset Organization

The collected dataset of job and internship postings was organized into structured tabular format. Each record contains textual fields (job title, description, requirements) and metadata fields (company details, recruiter email, salary information). Prior to training, the dataset underwent a cleaning phase in which Duplicate postings were removed, Null or incomplete entries were filtered, Label consistency was verified, Class distribution was analyzed .The cleaned dataset was divided into training, validation, and testing sets using stratified splitting to maintain class balance. Data loaders were configured to support batch-wise processing for deep learning models.

4.3 Text Preprocessing Pipeline

A robust preprocessing pipeline was implemented to standardize the raw job posting text before feature extraction. First, text normalization was applied by converting all characters to lowercase and removing HTML tags, URLs, punctuation, and special symbols. Stop-word removal was performed using the NLTK stop-word corpus to eliminate semantically weak words. Tokenization was then applied to split sentences into individual tokens. Lemmatization using spaCy reduced words to their base form while preserving contextual meaning. This step improves vocabulary consistency and reduces feature sparsity. After cleaning, the processed text was transformed into numerical vectors using two parallel approaches: TF–IDF vectorization for classical models, Word2Vec/BERT embeddings for deep learning models. Maintaining consistent preprocessing between training and inference was critical to ensure stable real-world performance.

4.4 Metadata Feature Extraction

In addition to textual content, structured metadata features were extracted because fraudulent postings often exhibit suspicious recruiter behavior. The following features were engineered: Recruiter email domain validation, Company website presence check, Salary anomaly detection, Posting repetition frequency, Missing company profile indicators. Categorical metadata attributes were encoded using one-hot encoding, while numerical features were normalized using Min–Max scaling. These structured features were concatenated with textual vectors to form a hybrid feature representation.

4.5 Model Training Pipeline

To perform comparative analysis, both classical machine learning models and deep learning architectures were trained. For classical models, Logistic Regression, Random Forest, and Support Vector Machine were implemented using scikit-learn. Hyperparameters were optimized using grid search with crossvalidation. For sequential modeling, a Bidirectional Long Short-Term Memory (Bi-LSTM) network was trained using Word2Vec embeddings. The architecture includes embedding, bidirectional LSTM, dropout, and dense output layers.

The transformer-based model was built by fine-tuning a pretrained BERT base model. The original classification head was replaced with a task-specific dense layer followed by softmax activation. Fine-tuning was performed using the Adam optimizer with a low learning rate to preserve pretrained linguistic

knowledge. To address class imbalance, SMOTE oversampling was applied to classical models, while class-weighted loss functions were used for deep learning models.

4.6 Model Evaluation and Validation

To ensure strong generalization performance, multiple overfitting control techniques were incorporated. Data augmentation served as the primary regularization mechanism by increasing effective dataset diversity. Validation monitoring helped detect early signs of overfitting by comparing training and validation loss trends. In addition, dropout regularization was optionally used within the classification head. Early stopping criteria could also be applied to halt training when validation performance stopped improving. These combined strategies improved the robustness of the final model.

4.7 Real-Time Prediction Engine

A dedicated inference pipeline was developed to support realtime fraud detection. The trained model weights were serialized and loaded into the prediction module. When a user submits a job posting, the system performs the following steps:

1. Text preprocessing using the same pipeline as training.
2. Metadata extraction and encoding.
3. Feature vector construction.
4. Model inference.
5. Reliability score generation.
6. The output includes the predicted class (genuine or fraudulent) along with a probability-based reliability score between 0 and 1.

4.8 Flask Based Web Development

To demonstrate practical usability, the system was deployed using the Flask web framework. A simple user interface was designed where users can paste a job description or upload posting details. Upon submission, the backend automatically preprocesses the input, performs model inference, and displays the authenticity result in real time. The interface was tested for low latency and consistent predictions. The modular architecture allows easy future integration into job portals, recruitment platforms, and browser extensions.

4.9 System Integration

The final system integrates multiple components into a unified pipeline:
Data preprocessing module, Feature engineering module, Hybrid ML/DL prediction engine, Reliability scoring module, Flask-based user interface.

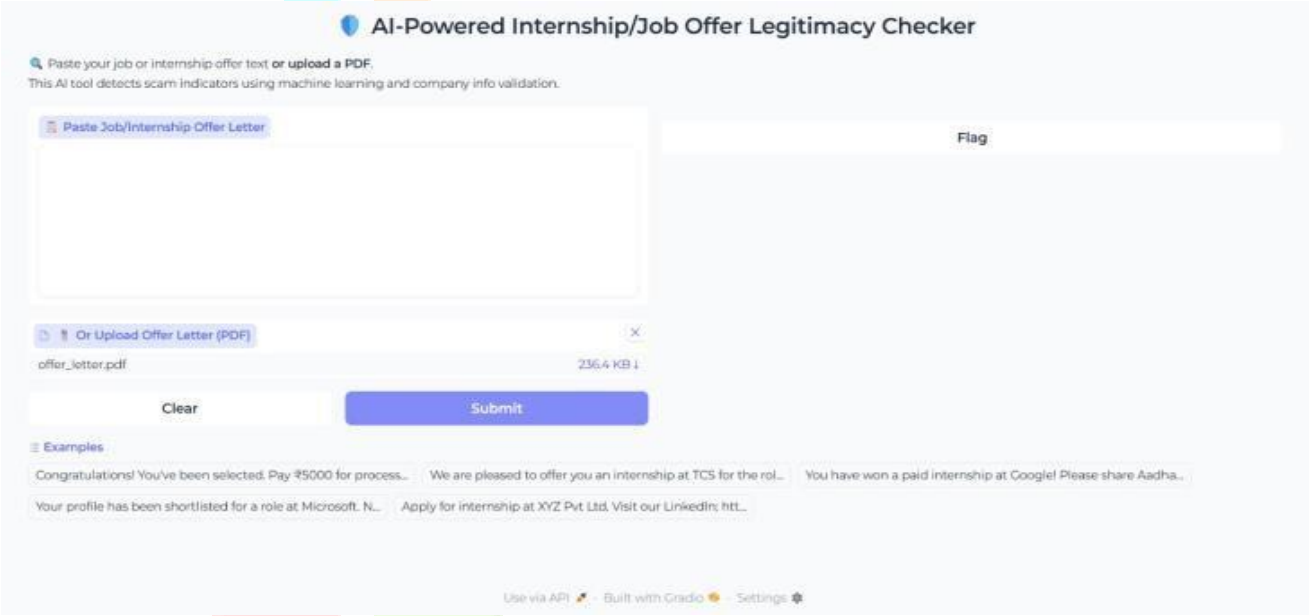
This end-to-end integration validates the feasibility of deploying AI-driven fraud detection in real-world recruitment ecosystems.

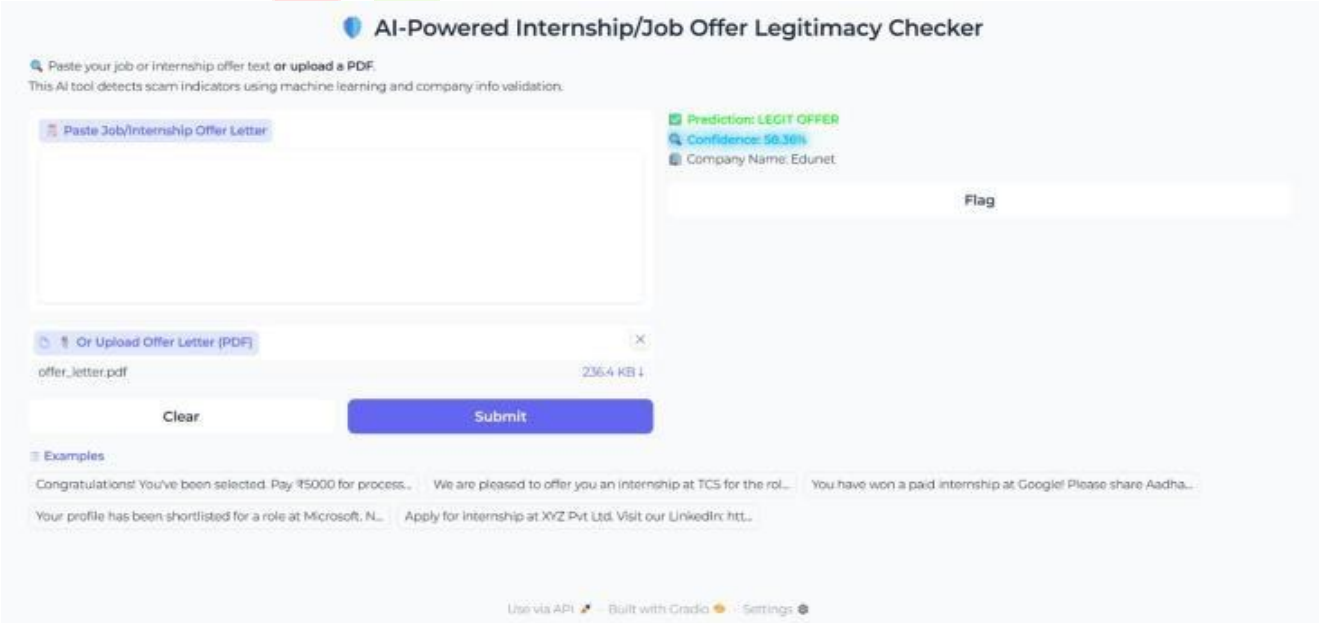
V. RESULT AND DISCUSSION

The system was evaluated on a benchmark dataset with an 80:20 train-test split and 10-fold cross-validation.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	ROC-AUC (%)
Logistic Regression	86.2	84.5	82.8	83.6	87.1
Random Forest	89.4	87.9	86.5	87.2	90.3
SVM	90.1	88.6	87.4	88.0	91.0
Bi-LSTM	92.7	91.5	90.2	90.8	93.5
BERT (Fine-tuned)	**95.6**	**94.8**	**93.7**	**94.2**	**96.8**

Deep learning models outperformed classical approaches. The fine-tuned BERT model achieved the highest performance, demonstrating superior capability in capturing complex linguistic and semantic patterns. The inclusion of metadata features further improved detection accuracy.





VI. CONCLUSION

This paper presented an AI-driven authenticity verification system for detecting fraudulent job and internship postings. By combining Machine Learning and Natural Language Processing techniques, the proposed framework effectively analyzes linguistic patterns and recruiter metadata to assign reliability scores. Experimental results demonstrate that the transformer-based model significantly improves fraud detection performance.

The system enhances applicant safety, reduces fraud risks, and strengthens trust in digital recruitment ecosystems. Future work includes multilingual support, cross-platform integration, real-time recruiter behavior analysis, and incorporation of explainable AI to improve transparency and user confidence.

VII. REFERENCES

[1] A. S. Pillai, "Detecting Fake Job Postings Using Bidirectional LSTM," arXiv preprint arXiv:2304.02019, 2023.
[2] S. Dutta and S. K. Bandyopadhyay, "Fake Job Recruitment Detection Using Machine Learning Approach," International Journal of Engineering Trends & Technology, vol. 68, no. 4, pp. 48–53, 2020.
[3] M. R. M., Mohan J., and Navin Jagadish P., "Fake Job Posting Detection," TIJER, vol. 10, no. 9, 2023.
[4] "Improving Fake Job Description Detection Using Deep Learning-Based NLP Techniques," Journal of Information and Telecommunication, 2024.
[5] "Detection of Fake Online Recruitment Using Machine Learning Techniques," IJSDR, 2024.