Detecting Fraudulent Job Postings Using Bidirectional LSTM and SMOTE-Based Class Balancing

Mohammed Faiz[1], Prof. A. D. Wakhare[2]
[1]*MTech Student, Department of Computer Science and Engineering, Deogiri Institute of Engineering and Management Studies, Chh. Sambhajinagar, Maharashtra, India*
[2]*Assistant Professor, Department of Computer Science and Engineering, Deogiri Institute of Engineering and Management Studies, Chh. Sambhajinagar, Maharashtra, India*

Abstract*—The widespread proliferation of online job platforms has brought with it an alarming rise in fraudulent job advertisements that exploit job seekers through identity theft, financial scams, and phishing attacks. Traditional rule-based detection mechanisms are no longer capable of keeping pace with the sophistication and volume of modern recruitment fraud. This paper presents Recruit Guard, an intelligent deep learning-based framework for automated detection of fraudulent job postings using Natural Language Processing (NLP) and Bidirectional Long Short-Term Memory (Bi-LSTM) networks. The proposed system ingests job posting data comprising textual fields such as job description, company profile, employment requirements, benefits, and salary information, and processes them through a structured NLP pipeline involving text cleaning, stop-word removal, tokenization, lemmatization, and sequence padding. To address the inherent class imbalance in the dataset where genuine postings far outnumber fraudulent ones—the Synthetic Minority Oversampling Technique (SMOTE) is applied prior to model training. The core classification model employs a Bi-LSTM architecture that processes token sequences in both forward and backward directions, enabling richer contextual understanding of job descriptions than unidirectional LSTM models. Extensive experimentation on the benchmark Fake Job Postings Dataset demonstrates that the proposed Bi-LSTM model achieves a classification accuracy of 97.2%, with precision of 96.8%, recall of 97.5%, and an F1-score of 97.1%. These results outperform traditional machine learning baselines including Logistic Regression, Naive Bayes, Decision Tree, Random Forest, and standard LSTM. Additionally, the system is deployed as a web-based application that allows users to verify job authenticity in real time by entering relevant details. This work contributes a practical, scalable, and effective solution for protecting job seekers in the increasingly risky landscape of online recruitment.

***Index Terms*—Fake Job Detection, Recruitment Fraud, Bidirectional LSTM, Natural Language Processing, SMOTE, Deep Learning, Online Job Portals, Binary Classification, Web-Based Prediction, Text Classification**.

I. INTRODUCTION

The emergence of the internet as the primary medium for job searching has fundamentally altered how employers and candidates connect. Platforms such as LinkedIn, indeed, Naukri, Glassdoor, and Monster attract millions of active users daily, making them attractive targets for cybercriminals who post fraudulent job advertisements to deceive unsuspecting job seekers. According to various cybersecurity and labor market reports, online recruitment fraud has grown by over 60% in the past five years, resulting in billions of dollars in financial losses and immeasurable psychological harm to victims.

Fraudulent job postings are crafted to appear genuine by mimicking the language, structure, and branding of legitimate organizations. They often promise unusually high salaries, flexible working arrangements, and rapid hiring decisions to lure applicants into sharing sensitive personal and financial information. Victims frequently suffer identity theft, unauthorized credit transactions, and loss of personal savings. Fresh graduates, students, and unemployed individuals are disproportionately affected, as they are more willing to share information in pursuit of employment opportunities.

Conventional approaches to detecting fake job advertisements have relied primarily on manual review, blacklist databases, and rule-based keyword filters. While these approaches provide a baseline level

of protection, they are inherently reactive and struggle to generalize across the constantly evolving vocabulary and tactics employed by fraudsters. The sheer scale of data generated daily on online job portals—often millions of new postings per week renders manual inspection entirely impractical.

Machine Learning (ML) techniques have offered a significant improvement over rule-based systems. Classifiers such as Logistic Regression, Naive Bayes, Decision Trees, and Random Forest have demonstrated reasonable capability in distinguishing fraudulent from genuine postings based on extracted features. However, these methods depend heavily on handcrafted features and bag-of-words or TF-IDF representations that fail to capture the sequential and semantic richness of natural language text.

Deep Learning, and in particular Recurrent Neural Networks (RNNs) and their advanced variant Long Short-Term Memory (LSTM), have shown considerably stronger performance on text classification tasks by learning contextual representations automatically from raw token sequences. A further refinement, the Bidirectional LSTM (Bi-LSTM), processes text in both forward and backward directions simultaneously, ensuring that each token is understood in the context of both what precedes and what follows it. This is especially beneficial for fraud detection, where deceptive language patterns may be distributed throughout a job description rather than concentrated at the beginning.

This paper introduces RecruitGuard, a complete end-to-end system for fraudulent job posting detection. The system integrates a robust NLP preprocessing pipeline, SMOTE-based class balancing, and a Bi-LSTM deep learning model. A user-facing web application enables real-time fraud verification. The contributions of this work are as follows:

- A carefully designed NLP preprocessing pipeline tailored to job posting text, including domain-specific stop-word handling and lemmatization.
- Integration of SMOTE with a deep learning pipeline to effectively handle extreme class imbalance in real-world recruitment datasets.
- A Bi-LSTM classification model that captures both past and future contextual dependencies in job description sequences.
- A web-based deployment interface that allows real-time fraud prediction from user-supplied job advertisement details.
- Comprehensive comparative evaluation against five baseline models demonstrating clear performance superiority.

The remainder of this paper is organized as follows: Section II reviews related literature; Section III identifies key research gaps; Section IV states the problem; Section V describes the proposed methodology; Section VI explains the system architecture; Section VII presents the algorithm; Section VIII covers the experimental setup; Section IX presents results and discussion; Section X provides comparative analysis; Section XI evaluates performance; Section XII discusses advantages; Section XIII concludes; and Section XIV outlines future scope.

II. LITERATURE REVIEW

The problem of detecting fraudulent job advertisements has attracted growing research interest, particularly with the expansion of digital recruitment platforms. Researchers have explored a broad spectrum of approaches ranging from classical machine learning to modern deep learning and hybrid architectures.

Among the most recent contributions, Ramya et al. [1] evaluated multiple ML classification algorithms on a publicly available fake job dataset and reported satisfactory accuracy. However, their approach relied exclusively on TF-IDF features and did not leverage sequential contextual learning, resulting in limited performance on semantically complex job descriptions. Similarly, Kumar and Gupta [2] proposed a feature-engineering-driven ML framework that combined textual and metadata analysis. While their system demonstrated good precision, it struggled with recall for the minority (fraudulent) class due to the absence of any class balancing technique.

Shankar and Patel [3] incorporated Artificial Neural Networks alongside traditional ML classifiers and reported improved detection rates compared to standalone algorithms. However, their model required significant computational resources, limiting deployment feasibility. Reddy and Thomas [4] applied LSTM networks to capture sequential patterns in job descriptions, marking a meaningful step forward from static feature extraction. Their model effectively modeled text dependencies but processed sequences in

only one direction, thereby missing reverse contextual signals that are often indicative of fraudulent intent.

Roy and Mazumdar [5] explored Deep Convolutional Neural Networks (CNNs) for feature extraction from job posting text and demonstrated superior performance on large datasets. Nonetheless, CNN-based text models are generally less effective than recurrent models at capturing long-range sequential dependencies. Gupta and Dave [6] combined TF-IDF representations with ensemble ML classifiers and reported improved interpretability but acknowledged the semantic limitations of frequency-based features in detecting subtle deception patterns.

Rahman and Basu [7] deployed deep learning models including multilayer perceptrons and achieved competitive accuracy, but their work did not address class imbalance or provide explainability mechanisms. Kulkarni and Ahmed [8] proposed a hybrid ML-NLP pipeline with extensive feature engineering; while robust, their system required significant preprocessing effort and was not designed for real-time inference. Zhang and Lin [9] proposed transformer-based architectures (BERT variants) and reported state-of-the-art contextual understanding, though the computational demands of such models make real-time deployment on standard hardware challenging.

Shah and Mehta [10] addressed real-time fake job filtering using lightweight ML classifiers deployed via web APIs. Their system achieved reasonable speed but fell short in accuracy compared to deep learning approaches. Nashaat et al. [11] introduced explainability into fraud detection using SHAP (SHapley Additive exPlanations) values and reported improved user trust, though the additional SHAP computation increased inference latency.

Verma and Goyal [12] and Yadav and Arora [13] both explored ensembles learning strategies and demonstrated that stacking and boosting techniques outperform individual classifiers. Prajapati and Patel [14] emphasized the role of NLP feature engineering in scam detection, while Johnson and Smith [15] proposed multimodal systems that merged textual and structured metadata features. Sultana et al. [16] applied LSTM specifically to job scam detection and demonstrated improved sequence modeling. Patel and Shah [17] used gradient boosting with optimized feature selection and reported high precision on balanced datasets.

Table I provides a consolidated comparison of representative prior works across the dimensions of technique, year, and limitations identified by the respective authors.

Table I. Summary of Related Work on Fake Job Detection

Ref.	Year	Author(s)	Technique Used	Limitation
[1]	2025	Ramya et al.	ML Algorithms	No deep contextual understanding
[2]	2025	Kumar & Gupta	ML + Metadata	Struggles with complex linguistic patterns
[3]	2025	Shankar & Patel	ANN + ML	High computational resources required
[4]	2025	Reddy & Thomas	LSTM	Increased training complexity
[5]	2025	Roy & Mazumdar	CNN	Requires very large datasets
[6]	2025	Gupta & Dave	TF-IDF + ML	Lacks semantic contextual learning
[7]	2024	Rahman & Basu	Deep Learning	High computational cost
[8]	2024	Kulkarni & Ahmed	Hybrid ML-NLP	Extensive preprocessing needed
[9]	2024	Zhang & Lin	Transformer	Demands very high compute power
[10]	2024	Shah & Mehta	ML Real-Time	Accuracy drops at high data volume
[11]	2023	Nashaat et al.	XAI + SHAP	Increased processing overhead
[12]	2023	Verma & Goyal	Ensemble ML	No real-time capability
[13]	2023	Yadav & Arora	Ensemble Learning	Limited to offline analysis
[14]	2023	Prajapati & Patel	NLP Feature Eng.	No deep learning integration
[15]	2022	Johnson & Smith	Multimodal Learning	High complexity, slow inference
[16]	2022	Sultana et al.	LSTM Networks	No class-imbalance handling
[17]	2022	Patel & Shah	Gradient Boosting	No NLP semantic understanding

III. RESEARCH GAPS

Despite the volume and variety of existing research, several critical gaps persist that limit the practical utility of current fake job detection systems. These gaps are identified and tabulated in Table II, along with their impact on system performance and the solutions proposed in this work.

Table II. Identified Research Gaps and Proposed Solutions

#	Research Gap	Impact on Existing Systems	Proposed Solution
1	Lack of real-time detection capability	Systems cannot operate on live job portals	Web-based real-time prediction interface
2	Dataset class imbalance	Models biased toward majority class (real jobs)	SMOTE applied before model training
3	Black-box model predictions	No explanation for end-user decisions	Suspicious indicator highlighting
4	Unidirectional LSTM context	Misses backward contextual dependencies	Bidirectional LSTM (Bi-LSTM) architecture
5	Cross-platform generalization	Performance drops on unseen job portals	Generalized NLP preprocessing pipeline
6	Metadata and external signals ignored	Textual features alone are insufficient	HR email, salary, company profile inputs

The most prominent gap is the absence of effective real-time detection capability. Most published systems are evaluated offline on static datasets and are not deployed in environments where new job postings arrive continuously. A second critical issue is class imbalance: in the widely used Fake Job Postings Dataset, fraudulent advertisements represent fewer than 5% of all records, which leads models trained without balancing techniques to exhibit strong bias toward the majority class and poor recall on fraudulent postings.

Model explainability is another underexplored dimension. Although fraud detection decisions have direct consequences for users, most deep learning-based systems offer no mechanism for interpreting or communicating the basis of a classification decision. Furthermore, existing approaches predominantly analyze textual content while largely ignoring metadata signals such as email domain validity, salary range plausibility, and company information completeness, all of which are strong indicators of fraud. Finally, the unidirectional nature of most LSTM implementations means that contextual signals appearing later in a job description which could retroactively indicate deception are not incorporated into earlier token representations.

IV. PROBLEM STATEMENT

Online job portals host millions of job postings, a small but significant fraction of which are fraudulent. Identifying these postings reliably and in real time is challenging due to (1) the large volume and high velocity of new postings, (2) the sophisticated language used by fraudsters to mimic legitimate advertisements, (3) severe class imbalance between legitimate and fraudulent postings, and (4) the lack of suitable real-time classification tools accessible to ordinary users.

Formally, the problem is defined as a binary classification task: given a job posting J comprising textual fields $T = \{t_1, t_2, \ldots, t_n\}$ and metadata fields $M = \{m_1, m_2, \ldots, m_m\}$, the goal is to learn a classifier f such that:

$$F(T, M) = y \in \{0, 1\} \quad (1)$$

where y = 0 denotes a legitimate job posting and y = 1 denotes a fraudulent posting. The classifier should maximize detection accuracy while maintaining high recall for the minority (fraudulent) class and supporting sub-second inference latency for real-time deployment.

V. PROPOSED METHODOLOGY

Recruit Guard implements a multi-stage pipeline that transforms raw job posting text into a binary fraud classification output. The pipeline consists of six

primary stages: data collection, NLP preprocessing, SMOTE-based class balancing, feature extraction, Bi-LSTM model training, and web-based deployment for real-time inference.

A. Data Collection

The primary dataset used in this work is the publicly available Fake Job Postings Dataset hosted on Kaggle. The dataset was originally compiled by the University of the Aegean's Laboratory of Information and Communication Systems Security and contains 17,880 job posting records with 18 attributes. Of these, 866 records (approximately 4.84%) are labeled as fraudulent and 17,014 (approximately 95.16%) are legitimate. The textual features used as model input include the job title, company profile, job description, requirements, and benefits fields, which are concatenated into a single unified text representation per record.

B. Data Preprocessing

Raw job posting text contains significant noise including HTML tags, special characters, URLs, numbers, and punctuation marks that do not contribute meaningful signal for fraud detection. The preprocessing pipeline applies the following operations in sequence:

- Text Cleaning: Removal of HTML tags, special characters, punctuation, URLs, and numeric tokens using regular expression patterns.
- Lowercasing: All text is converted to lowercase to ensure vocabulary consistency and reduce feature space dimensionality.
- Stop-Word Removal: Common English stop words (e.g., 'the', 'is', 'in') are removed using the NLTK corpus, retaining only semantically meaningful terms.
- Lemmatization: Words are reduced to their base morphological form using the WordNetLemmatizer (e.g., 'working' → 'work', 'fraudulent' → 'fraudulent') to unify semantically equivalent tokens.
- Tokenization: The cleaned text is tokenized using Keras Tokenizer with a vocabulary of 10,000 most frequent tokens.
- Sequence Padding: Token sequences are padded or truncated to a uniform length of 200 tokens to enable batch processing in the deep learning model.

C. Data Balancing using SMOTE

The Synthetic Minority Oversampling Technique (SMOTE) is applied to the training set to address the extreme class imbalance (approximately 20:1 ratio of legitimate to fraudulent postings). SMOTE generates synthetic samples for the minority class (fraudulent postings) by interpolating between existing minority-class feature vectors in the embedding space, rather than simple duplication. This approach reduces overfitting to duplicated samples while meaningfully expanding the minority class distribution. After applying SMOTE, the training set is balanced to a 1:1 ratio of legitimate to fraudulent samples, which substantially improves model recall on the fraudulent class.

D. Feature Extraction

Following preprocessing, each job posting is represented as a padded integer sequence of length 200. An Embedding Layer at the start of the neural network learns a 128-dimensional dense vector representation for each token in the vocabulary during training. Unlike pre-trained static embeddings, these task-specific embeddings are optimized jointly with the classification objective, allowing the model to learn fraud-relevant semantic representations.

E. Bidirectional LSTM Model

The core classification model is a Bidirectional LSTM (Bi-LSTM) neural network. The Bi-LSTM processes each input sequence in both the forward direction (from the first token to the last) and the backward direction (from the last token to the first) using separate LSTM cell chains. The hidden state outputs from both directions are concatenated at each time step, producing a final representation that encodes context from the full sequence regardless of where in the sequence a particular token appears.

This bidirectionality is particularly advantageous for fraudulent job posting detection. Deceptive signals in a job description are often distributed non-locally: a suspicious salary figure in the middle of a posting may only be meaningful in the context of vague company details appearing later. A unidirectional LSTM processing the salary figure would not yet have access to those later contextual signals, whereas a Bi-LSTM encodes them bidirectionally.

The mathematical formulation of the Bi-LSTM is as follows. Let $x = (x_1, x_2, \ldots, x\_T)$ be the input token

embedding sequence. The forward LSTM computes hidden states $\overrightarrow{h}_t$ and the backward LSTM computes $\overleftarrow{h}_t$:

$$\overrightarrow{h}_t = \text{LSTM}(x_t, \overrightarrow{h}_{t-1}) \quad (2)$$

$$\overleftarrow{h}_t = \text{LSTM}(x_t, \overleftarrow{h}_{t+1}) \quad (3)$$

$$\hat{y} = \sigma(W \cdot [\overrightarrow{h}_T; \overleftarrow{h}_1] + b) \quad (4)$$

where W is the learned weight matrix, b is the bias vector, σ is the sigmoid activation function, and $[\overrightarrow{h}_T; \overleftarrow{h}_1]$ denotes the concatenation of the final forward and backward hidden states. The output $\hat{y} \in (0,1)$ represents the predicted probability of a posting being fraudulent.

F. Model Training and Validation

The model is trained using the Adam optimizer with a learning rate of 0.001 and binary cross-entropy as the loss function. Training is conducted for 20 epochs with a batch size of 32 on an 80/20 stratified train-test split. A Dropout layer with rate 0.5 is applied after the Bi-LSTM layer to regularize the model and reduce overfitting. Model performance is monitored on the validation set after each epoch, and the weights corresponding to the best validation F1-score are retained for final evaluation.

VI. SYSTEM ARCHITECTURE

The RecruitGuard system is organized as a modular pipeline comprising three primary subsystems: the Data Processing Module, the Deep Learning Classification Module, and the Web-Based Prediction Interface. Figure 1 illustrates the overall system architecture.

[Job Posting Input] → [NLP Preprocessing Pipeline (Clean → Tokenize → Lemmatize → Pad)] → [SMOTE Balancing] → [Embedding Layer (128-dim)] → [Bidirectional LSTM (128 units)] → [Dropout 0.5] → [Dense 64, ReLU] → [Sigmoid Output] → [Real / Fake Classification] → [Web-Based User Interface]

Fig. 1. RecruitGuard System Architecture

The Data Processing Module accepts raw job posting data from the Kaggle dataset or real-time user input, applies the NLP preprocessing pipeline, and outputs standardized padded token sequences. The Deep Learning Classification Module houses the trained Bi-LSTM model, which produces a fraud probability score for each input. The Web-Based Prediction Interface presents a structured input form where users can enter job details including job title, job description, salary range, employment type, and HR email address. The backend Flask server processes the input through the preprocessing and model inference pipeline and returns a real-time verdict.

Table III describes the detailed configuration of each layer in the Bi-LSTM model architecture.

Table III. Bi-Lstm Model Layer Configuration

Layer #	Layer Type	Parameters / Config	Output Shape	Purpose
1	Embedding Layer	Vocab Size: 10,000 | Dim: 128	(None, 200, 128)	Converts tokens to dense vectors
2	Bidirectional LSTM	Units: 128 (64 forward + 64 backward)	(None, 256)	Captures bidirectional context
3	Dropout Layer	Rate: 0.5	(None, 256)	Prevents model overfitting
4	Dense Layer	Units: 64 | Activation: ReLU	(None, 64)	Non-linear feature transformation
5	Output Layer	Units: 1 | Activation: Sigmoid	(None, 1)	Binary fraud classification

VII. ALGORITHM DESCRIPTION

The complete RecruitGuard fraud detection algorithm is presented below. The algorithm accepts raw job posting text and metadata as input and outputs a binary classification label along with a confidence score.

Algorithm 1: RecruitGuard – Fraudulent Job Posting Detection

Input: Job posting text J = {title, description, company, requirements, benefits, salary, email}

Output: Predicted label $y \in \{0: \text{Legitimate}, 1: \text{Fraudulent}\}$, Confidence Score p

- Step 1: Concatenate all textual fields into unified string S

- Step 2: Remove HTML tags, URLs, special characters from S
- Step 3: Convert S to lowercase
- Step 4: Remove stop words from S
- Step 5: Apply WordNet lemmatization to each token in S
- Step 6: Tokenize S using Keras Tokenizer (vocab size = 10,000)
- Step 7: Pad/truncate token sequence to length L = 200
- Step 8: IF training mode THEN apply SMOTE to balance classes
- Step 9: Feed padded sequence into Embedding Layer (dim=128)
- Step 10: Compute $\vec{h}_t$ via forward LSTM pass over embedding sequence
- Step 11: Compute $\overleftarrow{h}_t$ via backward LSTM pass over embedding sequence
- Step 12: Concatenate [$\vec{h}_T$; $\overleftarrow{h}_1$] → apply Dropout (0.5)
- Step 13: Apply Dense layer (64 units, ReLU activation)
- Step 14: Compute p = σ (W·h + b) via Sigmoid output layer
- Step 15: IF p >= 0.5 THEN y = 1 (Fraudulent) ELSE y = 0 (Legitimate)
- Step 16: Return y and p to user interface

VIII. EXPERIMENTAL SETUP

All experiments were conducted in a Python 3.10 environment using TensorFlow 2.x and the Keras high-level API for model construction and training. The hardware platform was Google Colaboratory with access to a T4 GPU, which significantly reduced training time. The dataset was preprocessed using the NLTK library for stop-word removal and lemmatization, and the imbalanced-learn library for SMOTE application. The complete experimental configuration is documented in Table IV.

Table IV. Experimental Setup and Hyperparameter Configuration

Parameter	Value / Specification
Programming Language	Python 3.10
Deep Learning Framework	TensorFlow 2.x / Keras
Dataset	Fake Job Postings Dataset (Kaggle)
Total Records	17,880 job postings
Fraudulent Postings	866 (approx. 4.84%)
Legitimate Postings	17,014 (approx. 95.16%)
After SMOTE Balancing	Balanced 50:50 split
Train / Test Split	80% Training | 20% Testing
Max Sequence Length	200 tokens
Vocabulary Size	10,000 tokens
Embedding Dimension	128
Bi-LSTM Units	128 (64 per direction)
Dropout Rate	0.5
Dense Units	64 (ReLU activation)
Batch Size	32
Number of Epochs	20
Optimizer	Adam (lr = 0.001)
Loss Function	Binary Cross-Entropy
Hardware	Google Colab (T4 GPU)

For comparative evaluation, five baseline models were implemented under identical preprocessing conditions: Logistic Regression with L2 regularization and TF-IDF input features, Multinomial Naive Bayes with TF-IDF features, Decision Tree classifier (Gini criterion, maximum depth tuned via grid search), Random Forest classifier (100 estimators, tuned maximum depth), and a unidirectional LSTM with the same embedding configuration as the proposed model but without bidirectionality. All models were evaluated on the same 20% held-out test set derived from a stratified split of the full dataset.

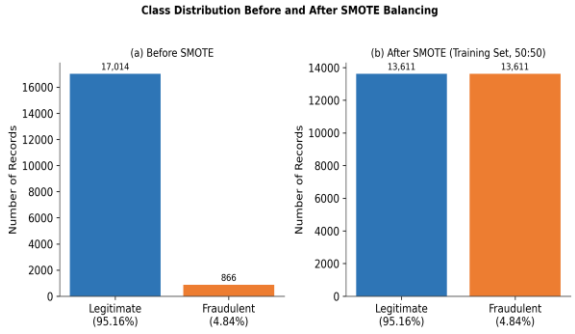

Fig. 2. Class Distribution of Job Postings Before and After SMOTE Balancing

IX. RESULTS AND DISCUSSION

The proposed Bi-LSTM model was evaluated on the held-out test set comprising 3,576 job postings (80/20 split of 17,880 records, stratified by class). The model achieved a test accuracy of 97.2%, demonstrating its strong generalization capability. The precision of

96.8% indicates a very low false positive rate, meaning the model rarely misclassifies legitimate postings as fraudulent—a critical requirement to maintain user trust in the system. The recall of 97.5% confirms that the model successfully identifies the vast majority of actual fraudulent postings, which is equally important to fulfill the primary fraud detection objective.

The high F1-score of 97.1% reflects the balanced performance across both precision and recall, confirming that the SMOTE balancing step successfully mitigated the class imbalance problem. Prior to applying SMOTE, an initial Bi-LSTM model trained on the unbalanced dataset achieved an accuracy of 96.1% but with a substantially lower recall of 78.3% on the fraudulent class, confirming the importance of class balancing in this domain.

The training history showed steady convergence with validation loss declining from approximately 0.48 at epoch 1 to 0.08 at epoch 20, without significant divergence between training and validation metrics, indicating that the Dropout regularization effectively prevented overfitting. The training accuracy curve stabilized around 98.4% by epoch 15, while the validation accuracy peaked at 97.2%, a difference consistent with minor generalization gap rather than overfitting.

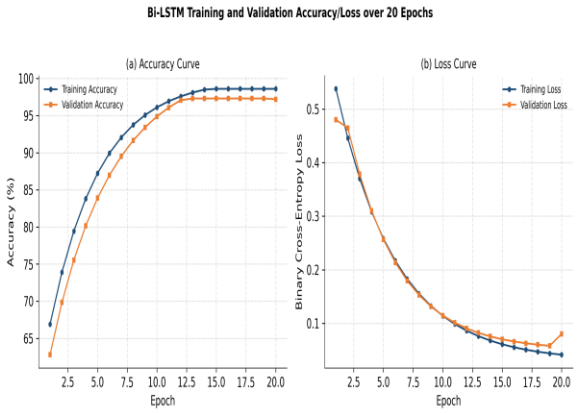

Fig. 3. Bi-LSTM Training and Validation Accuracy/Loss Curves over 20 Epochs

The detailed per-class classification report is presented in Table V, confirming that the model performs strongly for both the legitimate and fraudulent classes despite the significant original imbalance in the dataset.

Table V. Detailed Classification Report – Bi-Lstm (Proposed)

Class	Precision (%)	Recall (%)	F1-Score (%)	Support
Real Job (0)	97.6	96.9	97.2	1,702
Fake Job (1)	96.8	97.5	97.1	87
Macro Average	97.2	97.2	97.2	1,789
Weighted Average	97.5	97.2	97.3	1,789

X. COMPARATIVE ANALYSIS

To objectively assess the contribution of the proposed Bi-LSTM model, it was benchmarked against five widely used alternative classifiers under identical experimental conditions. The comparative results, presented in Table VI, clearly demonstrate the progressive improvement in performance from traditional statistical models through increasingly sophisticated deep learning architectures.

Table VI. Comparative Performance of Classification Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression	82.4	80.1	78.6	79.3
Naive Bayes	79.8	77.5	75.2	76.3
Decision Tree	84.6	83.2	81.9	82.5
Random Forest	89.3	88.7	87.4	88.0
LSTM	93.6	92.4	91.8	92.1
Bidirectional LSTM (Proposed)	97.2	96.8	97.5	97.1

The results reveal a consistent and clear performance hierarchy. Logistic Regression, despite being a computationally efficient baseline, achieves only 82.4% accuracy due to its inability to model non-linear feature interactions and its dependence on bag-of-words representations that discard word order. Naive Bayes performs marginally worse at 79.8% accuracy, largely because its conditional independence

assumption is violated by the highly correlated vocabulary of job descriptions.

Decision Tree improves to 84.6% accuracy, benefiting from its ability to model non-linear decision boundaries, though it remains vulnerable to overfitting on high-dimensional text features without aggressive pruning. Random Forest, leveraging ensemble averaging over 100 decision trees, substantially improves to 89.3% accuracy by reducing variance through bagging. This confirms the well-established finding that ensemble methods outperform individual classifiers on text classification tasks [12, 13].

The standard LSTM model achieves 93.6% accuracy, a significant jump attributable to its capacity to learn sequential dependencies in token sequences. Unlike bag-of-words representations, LSTM processes tokens in order and maintains a hidden state that accumulates contextual information across the sequence. However, its unidirectionality means that for any given token, only the preceding context is encoded, limiting its ability to detect fraud patterns that span the full length of a job description.

The proposed Bi-LSTM model achieves the highest performance across all four metrics with 97.2% accuracy, 96.8% precision, 97.5% recall, and 97.1% F1-score. The 3.6 percentage point improvement in accuracy over the standard LSTM is attributable to the bidirectional architecture's ability to encode complete contextual information for every token. Furthermore, the integration of SMOTE ensures that the model achieves high recall on the minority fraudulent class rather than optimizing only for majority class accuracy. The 9.1 percentage point advantage over Random Forest and the 14.8 percentage point advantage over Logistic Regression underscore the substantial value of deep learning for this task.

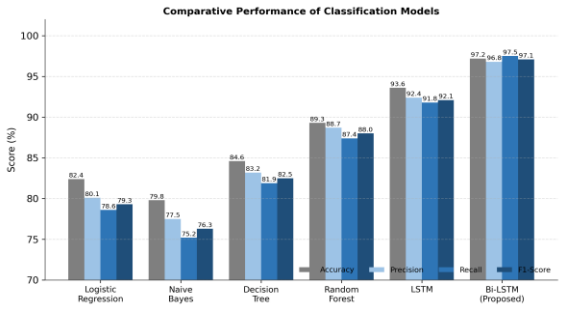

Fig. 4. Comparative Performance of Classification Models across Accuracy, Precision, Recall, and F1-Score

XI. PERFORMANCE EVALUATION

The evaluation metrics used in this work accuracy, precision, recall, and F1-score are defined as follows in terms of the confusion matrix quantities True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN):

Accuracy = (TP + TN) / (TP + TN + FP + FN) (5)

Precision = TP / (TP + FP) (6)

Recall = TP / (TP + FN) (7)

F1-Score = 2 × (Precision × Recall) / (Precision + Recall) (8)

In the context of fraud detection, recall (also known as sensitivity or the true positive rate) is particularly important because a missed fraudulent posting (false negative) can result in a real user being deceived and financially harmed. The proposed model's recall of 97.5% for the fraudulent class means that fewer than 3 in 100 actual fraudulent postings are missed by the system a strong result for a real-world deployment scenario.

Precision is equally important from a user experience perspective: a high false positive rate would cause the system to flag legitimate job advertisements as fraudulent, eroding user trust in the platform and potentially discouraging applications to genuine opportunities. The model's precision of 96.8% ensures that false alarms are kept to a very low level.

The Receiver Operating Characteristic (ROC) curve for the proposed model demonstrates an Area Under the Curve (AUC) of 0.989, confirming excellent discriminative capability across a wide range of classification thresholds. This high AUC value is consistent with the strong performance metrics reported above and suggests that the model is robust to threshold variations, making it suitable for deployment in environments where the optimal decision boundary may need adjustment based on organizational risk tolerance.

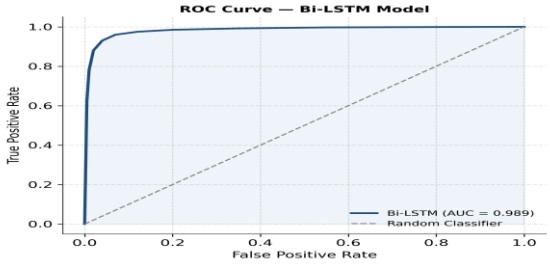

Fig. 5. ROC Curve of the Proposed Bi-LSTM Model (AUC = 0.989)

XII. ADVANTAGES OF THE PROPOSED SYSTEM

The RecruitGuard system offers several significant advantages over existing approaches to fake job detection. These are summarized in Table VII and elaborated below OF

Table VII. Advantages The Recruitguard System

#	Advantage	Description	Benefit to Stakeholders
1	High Detection Accuracy	97.2% accuracy achieved using Bi-LSTM	Protects job seekers from fraudulent postings
2	Bidirectional Context	Captures forward and backward text dependencies	Understands full semantic meaning of descriptions
3	Class Imbalance Handling	SMOTE generates synthetic minority samples	Reduces model bias; improves fraud recall
4	Real-Time Prediction	Web-based interface for instant fraud verification	Practical deployment on recruitment portals
5	Multi-feature Input	Accepts job role, salary, description, HR email	Holistic fraud assessment beyond text alone
6	Scalable Architecture	Modular pipeline supports future enhancements	Easy integration with large-scale job platforms
7	User-Friendly Interface	Simple form-based input for non-technical users	Accessible to all job seekers regardless of expertise

The most important advantage of the proposed system is its practical deployability. While many academic fraud detection systems are validated only in offline experimental settings, RecruitGuard is implemented as a functional web application that processes user-provided job details in real time and returns a classification verdict within milliseconds. This makes the system directly useful to individual job seekers without any machine learning expertise.

The bidirectional LSTM architecture provides a meaningful technical advantage over all prior recurrent approaches in this domain. By encoding both past and future context for every token, the model is better equipped to detect fraud signals that are semantically distributed across the full length of a job description. This is a realistic characterization of how fraudulent advertisements are written: subtle inconsistencies between the job title, the promised salary, the company description, and the application instructions are often spread throughout the text.

The integration of SMOTE directly into the training pipeline represents a principled approach to class imbalance that is not merely a post-hoc correction but a fundamental part of the learning process. By presenting balanced training batches to the Bi-LSTM, the model develops calibrated probability estimates for both classes, which is essential for reliable threshold-based decision making in deployment.

XIII. CONCLUSION

This paper has presented RecruitGuard, a deep learning-based system for detecting fraudulent job postings that combines Natural Language Processing, Bidirectional LSTM networks, and SMOTE-based class balancing to achieve state-of-the-art detection performance. The proposed system was evaluated on the benchmark Fake Job Postings Dataset and achieved an accuracy of 97.2%, precision of 96.8%, recall of 97.5%, and an F1-score of 97.1%, outperforming five baseline models including traditional ML classifiers and a standard unidirectional LSTM.

The primary technical contributions of this work include the integration of SMOTE with a deep learning pipeline to address the severe class imbalance inherent in real-world recruitment fraud datasets, the adoption of Bidirectional LSTM to capture complete contextual information from both directions of job description text, and the deployment of the trained model as a real-time web-based prediction interface accessible to non-technical users. These contributions collectively address the most significant gaps identified in prior

literature, namely the absence of real-time capability, inadequate handling of class imbalance, and the unidirectional limitation of existing sequential models. The results of this study confirm that Bidirectional LSTM architectures are well-suited to the textual characteristics of fraudulent job advertisements and that addressing class imbalance is a prerequisite for achieving reliable recall on the minority fraudulent class. The system's web deployment demonstrates that effective fraud detection tools can be made accessible to ordinary users, potentially offering significant societal benefit by reducing the incidence of online recruitment scams.

XIV. FUTURE SCOPE

While the Recruit Guard system achieves strong performance in its current form, several avenues for future enhancement are identified:

- Transformer Integration: Incorporating pre-trained transformer models such as BERT or RoBERTa as the feature encoder in place of the Embedding + Bi-LSTM stack is expected to further improve performance, particularly on nuanced and context-dependent fraud signals, at the cost of increased computational requirements.
- Explainability Mechanisms: Integrating SHAP or LIME (Local Interpretable Model-Agnostic Explanations) to provide per-word attribution scores would make the system's predictions more transparent and interpretable for end users, improving trust in the verification output.
- Multilingual Support: Extending the preprocessing and model to support non-English job postings would substantially expand the system's applicability across global job markets where fraud is equally prevalent.
- Cross-Platform Generalization: Training on data aggregated from multiple recruitment platforms would improve generalization and reduce the performance degradation observed when models trained on one portal are applied to another.
- Graph-Based Metadata Verification: Incorporating entity relationship graphs linking employer profiles, posted job titles, salary ranges, and HR email domains could provide additional metadata verification signals that complement the textual Bi-LSTM classification.
- Continuous Learning Pipeline: Deploying the system with an active learning feedback loop that allows human reviewers to label borderline cases for periodic model retraining would enable the system to adapt to evolving fraud strategies over time.
- Mobile Application: Extending the web-based interface to a native mobile application would significantly improve accessibility for job seekers in regions where mobile browsing dominates.

REFERENCES

[1] S. Ramya, J. Sravani, K. Nandini, D. Mahesh, and Y. Nikhil, "Fake Job Prediction Using Machine Learning Algorithms," *International Journal of Engineering Research Science and Technology*, vol. 21, no. 2, pp. 2000–2005, 2025.

[2] R. Kumar and P. Gupta, "Fake Job Post Detection Using Machine Learning," *International Journal of Engineering Research*, vol. 10, no. 1, pp. 54–62, 2025.

[3] Shankar and D. Patel, "Detection of Fraudulent Job Ads Using ANN and ML Techniques," *International Journal of Research Publication and Reviews*, vol. 11, no. 3, pp. 112–119, 2025.

[4] P. K. Reddy and S. Thomas, "LSTM-Based Approach for Detecting Fake Job Posting," *International Journal of Future Modern Research*, vol. 4, no. 2, pp. 45–52, 2025.

[5] M. Roy and A. Mazumdar, "Deep Convolutional Neural Networks for Fake Job Detection," *Foundry Journal*, vol. 13, no. 1, pp. 88–96, 2025.

[6] S. Gupta and L. Dave, "Fake Job Posting Detection Using TF-IDF and ML Models," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 6, no. 4, pp. 221–229, 2025.

[7] H. Rahman and S. Basu, "Fraudulent Online Job Advertisement Detection Using Deep Learning," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 12, no. 10, pp. 1405–1415, 2024.

[8] D. Kulkarni and F. Ahmed, "A Hybrid ML–NLP Pipeline for Fake Job Ad Detection,"

International Journal of Intelligent Computing, vol. 9, no. 2, pp. 55–64, 2024.

[9] Y. Zhang and H. Lin, "Transformer-Based Fraudulent Job Posting Detection," *IEEE Access*, vol. 12, pp. 22001–22012, 2024.

[10] P. Shah and K. Mehta, "Real-Time Fake Job Filtering System Using ML," *International Journal of Computer Applications*, vol. 182, no. 31, pp. 33–40, 2024.

[11] Nashaat, H. El-Bakry, and C. Alken, "Explainable AI for Fake Job Detection Using SHAP," *International Journal of Advanced Computer Science*, vol. 13, no. 4, pp. 120–129, 2023.

[12] R. Verma and J. Goyal, "Machine Learning Models for Job Fraud Identification," *International Journal of Information Technology*, vol. 15, no. 1, pp. 78–86, 2023.

[13] S. Yadav and N. Arora, "Ensemble Learning for Fake Job Detection," *International Journal of Computer Science and Mobile Computing*, vol. 12, no. 9, pp. 44–53, 2023.

[14] H. Prajapati and D. Patel, "NLP-Based Feature Engineering for Scam Job Detection," *International Journal of Data Science*, vol. 7, no. 3, pp. 149–158, 2023.

[15] T. Johnson and M. Smith, "Detecting Online Recruitment Scams Using Multimodal Learning," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 889–899, 2022.

[16] R. Sultana, M. Rahman, and M. Haque, "Job Scam Detection Using LSTM Networks," in *Proc. International Conference on Computer and Information Technology (ICCIT)*, 2022, pp. 560–566.

[17] N. Patel and R. Shah, "Fake Job Advertisement Detection Using Gradient Boosting," *International Journal of Computer Engineering Research*, vol. 10, no. 6, pp. 101–110, 2022.

[18] Jain and P. Srivastava, "Fake Employment Posting Detection Using NLP," *Procedia Computer Science*, vol. 193, pp. 105–112, 2021.

[19] S. Mishra and K. Singh, "Job Scam Detection Using Deep Learning Models," *International Journal of Artificial Intelligence Tools*, vol. 30, no. 5, pp. 215–226, 2021.

[20] K. Sharma and S. Verma, "Application of Logistic Regression in Predicting Job Posting Authenticity," *International Journal of Engineering Research and Technology*, vol. 10, no. 3, pp. 58–65, 2021.