Research

Fraud-BERT: transformer based context aware online recruitment fraud detection

Khushboo Taneja[1] · Jyoti Vashishtha[1] · Saroj Ratnoo[1]

Received: 2 June 2024 / Accepted: 29 January 2025
Published online: 06 February 2025

Abstract
Online recruitment facilitates the automatic hiring process for recruiters and provides convenience to job seekers via online job platforms. Parallelly, it has given rise to malicious use of such platforms by fraudsters who post fake jobs and steal money and personal information from innocent job seekers. It is difficult to detect fake jobs manually, as these are meticulously crafted to mimic legitimate ones. Previously, various machine learning approaches have employed Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction methods for this task. However, these methods are non-contextual and show skewness in results due to imbalances in data distribution. This paper presents Fraud-BERT, a transformer-based contextual framework leveraging Bidirectional Encoder Representations from Transformers (BERT) via transfer learning approach and evaluates it on a highly imbalanced fake job dataset, which is popularly named as Employment Scam Aegean Dataset (EMSCAD). The dataset is available on Kaggle website. The superiority of the proposed method is demonstrated by a comparative analysis with conventional methods. The results of the study conclude that the proposed method is more robust in tackling imbalanced data and, it has significantly out-performed existing state-of-the-art studies with F1 score of 0.93 and 99% accuracy.

Keywords Fake job detection · Text classification · Transformer · Large language model · Transfer learning

1 Introduction

Online recruitment is the process of recruiting employees through various job seeking platforms such as Naukri.com, LinkedIn, etc. via internet. It has gained popularity among HR practitioners due to its cost effectiveness. It facilitates the automation of employment procedures for businesses, including arranging interviews, screening applications, and other communication services. Similarly, it provides convenience to job seekers who can search and apply for different kind of jobs that matches their profiles through these platforms. However, the emergence of job seeking platforms has established yet another media for internet fraud i.e., Online Recruitment Fraud (ORF). Fraudsters skilfully craft fake job advertisements and post these to various job portals. Allured by the lucrative financial offers, flexible work hours, etc., innocent job seekers reveal their private information such as contact addresses, bank account numbers, social security numbers, etc. to fraudsters. Fraudsters can even sell their data to third parties and grab a lot of money from innocent job seekers.

The COVID pandemic has left many people unemployed. Many young adults began searching various websites and platforms for work as the economy began to open up following two years of setbacks caused by the pandemic.

✉ Khushboo Taneja, khushbootaneja@gmail.com; Jyoti Vashishtha, jyoti.vst@gmail.com; Saroj Ratnoo, ratnoo.saroj@gmail.com | [1]Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar 125001, India.

Discover Computing (2025) 28:9 | https://doi.org/10.1007/s10791-025-09502-8

Fraudsters took advantage of this opportunity to trap innocent people through fake job advertisements. More recently, they have also found a new way to scam people via sending a text SMS or a WhatsApp message, promising lucrative job opportunities [1]. According to a recent report of May 2023, "a chat-based direct hiring platform stated that 56% of young job seekers aged between 20 and 29 years in country are impacted by job scams during their job hunt" [2]. ORF can be considered as a cyber-crime that adversely impacts the security and social well-being of people. It may also harm the reputation of a legitimate organization leading to smaller applicant pool. There is an obvious need for solutions to safeguard the security of those involved in the recruitment process given these costs to society.

Text classification is a challenging task. Unlike ordinary text classification, distinguishing between genuine and fake job adverts can be challenging even for human professionals because these are intentionally crafted to imitate legitimate recruitment information. Additionally, these are also consisting of a lot of missing information and exaggerations such as higher rate of salary, lower education requirements, etc. Several machine learning (ML) techniques have been proposed in the literature to solve this problem [3–14]. Most of them are based on Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) text representation methods which are derived from one-hot representation technique in which the size of vocabulary depends upon total number of words present in the sentence. These methods represent text with high dimensional vectors which are computationally expensive to process. Moreover, these techniques are context-free and cannot preserve the order of words. A few studies have also focussed on using deep learning (DL) models such as RNN, GRU, etc. in combination with Word2Vec [10, 12]. Word2Vec is a pre-trained model developed by Google. It is trained on very large corpora of text and represents text with low dimensional dense vectors. However, Word2Vec can only represent static context of a word and DL models need a lot of data for training to give accurate results. In this domain, we generally have to deal with imbalanced data. The disadvantage of traditional ML and DL algorithms is that they cannot handle data imbalance problem and show biasness towards majority class. The previous results obtained using these methods indicate very low F1 and recall scores due to large number of misclassification errors. A few research studies have also studied the role of data resampling in this domain and applied oversampling techniques such as synthetic minority over-sampling (SMOTE) and adaptive synthetic sampling (ADASYN) to balance the dataset but could not achieve the optimal results [12, 14].

In a nutshell, we can say that to address the aforementioned limitations, we need a solution whose foundation builds upon the ability to understand the dynamic context and pragmatics of a language. Recently, transformer [15] has become a state-of-the-art (SOTA) architecture and it has been widely adopted by researchers to develop neural language models. It has changed the complete Natural Language Processing (NLP) landscape. The present research in NLP has spawned several transformer language models that are extensively pre-trained on massive amount of data [16, 17]. These models are known as Large Language Models (LLMs). Some of the most popular LLMs nowadays are BERT [18], T5 [19], ChatGPT [20], etc. where the T in each model stands for transformer. These models can be reused to solve various tasks and can also be applied in different domains via a process known as transfer learning.

In this research paper, we have proposed Fraud-BERT, a BERT based transfer learning approach for identification of fake jobs. Unlike traditional approaches like BoW, TF-IDF and Word2Vec, BERT employs WordPiece tokenizer which works on the principle of subword-based tokenization algorithm. It splits the rare words into smaller meaningful subwords that empowers BERT to capture the nuances of the domain or language. BERT is based on original transformer architecture [15] that has in-built "self-attention" mechanism which helps BERT to establish textual relationship and understand the dynamic context of words. It can handle different types of text data including structured data such as sentences, paragraphs and documents as well as unstructured text data which consist of informal text from platforms like Twitter, Instagram, etc. It can also resolve the issue of data imbalance without application of any data resampling technique [21]. Identification of fake job posts is challenging because these are very skilfully crafted by fraudsters. By exposing a LLM like BERT, the unique language patterns in this domain can be learned efficiently by the model.

To fulfil our objective, we have adopted BERT via fine-tuning it on fake job posts dataset [22]. Fine-tuning is a process in which weights of the original BERT model can be modified according to the specific task or domain. The results obtained in the study has shown remarkable performance in terms of precision (P), recall (R), F1, accuracy and area under ROC curve (AUC) scores as compared to traditional approaches and SOTA studies on this dataset. The significant contributions of this research work can be listed as follows.

- We have proposed a novel transformer-based BERT language model and modified its parameters according to domain of study by fine-tuning it on a highly imbalanced fake job posts dataset.
- A comparative analysis of the proposed model is made with conventional ML and DL models with results indicating superiority of proposed method over traditional approaches.

- Unlike traditional approaches, the proposed approach mitigates the issue of data imbalance without any need of extra data pre-processing steps.
- Our model has shown significant improvement over SOTA existing studies on the chosen dataset.

The remainder of the paper is organized as follows. Section 2 describes the related work. Section 3 illustrates the materials and methods utilized in this study. Section 4 presents the experimental settings for developing various models in this work. Section 5 indicates the results obtained. Section 6 and 7 presents the integration challenges and limitations of the proposed method respectively. Section 8 concludes the study with its future scope.

2 Related works

Considering the adverse impact of fake job posts to the lives of common people, several research studies have been conducted in the past in this domain. Vidros et al. (2017) was the first group of researchers that published a real-life dataset of 17,880 job ads which consists of 17,014 instances of legitimate jobs and 866 instances of fraudulent jobs [3]. This dataset is known as Employment Scam Aegean Dataset (EMSCAD) and became popular for subsequent research studies in this field. It is manually annotated and contains some duplicate and blank entries. The authors also declared that some entries in the dataset are misclassified, but their number is insignificant. One of the major limitation or challenge is to address class imbalance in this dataset. The authors used traditional BoW modelling for text representation and proposed various machine learning-based classification techniques such as Logistic Regression (LR), Random Forest (RF), Decision Trees (DT), J48, Naïve Bayes (NB), ZeroR and OneR for the automatic detection of fraudulent jobs. They also performed the empirical analysis and highlighted the different characteristics of fake job posts like missing information, short and sketchy textual content. Mahbub and Pardede [4] have manually designed a set of contextual feature space and utilized various ML algorithms including J48, JRip and NB to classify fake and legitimate posts. Their results indicated JRip as the highest performing model with an accuracy of 96.19% and 0.915 recall score. Alghamdi and Alharby [5] have proposed an ensemble method in which Support Vector Machine (SVM) is used for feature selection and then the ensemble Random Forest classifier is trained with the extracted features with an accuracy of 97.41%. In the same line of work, Lal et al. [6] developed an ensemble of J48, LR and RF. Their work showed the superiority of ensemble classifiers over base classifiers with an accuracy of 95.4% and 94.4% F1 score.

Dutta and Bandyopadhyay (2020) performed this task by converting text into encoding categories. Their work compared various single ML based classifiers and ensemble classifiers. The results of the experiments showed that best performance is delivered by RF classifier with an accuracy of 98.27% [7]. Nasser and Alzaanin [8] extracted the textual features using TF-IDF method and developed various ML models including NB, DT, SVM, RF, and k-Nearest Neighbour (KNN). Shibly et al. [9] designed models with "two-class boosted decision trees" and "two-class decision forest algorithms" using Microsoft azure ML studio. Their results indicated that former is the highest performing algorithm with an accuracy of 93.8% and a recall of 0.73. Anita et al. [10] have also proposed ML algorithms such as KNN, RF, LR and DL algorithm such as Bi-LSTM for automated fake job detection. Their results show the superiority of Bi-LSTM over ML algorithms.

Chiraratanasopa and Chay-intr (2022) designed a set of features for this task based on the criteria of exaggeration, credibility and missing information. The transformed set of features is then used to train ML algorithms including SVM, KNN and DT. The proposed method achieved the highest accuracy of 97.64%, precision of 0.97 and recall of 0.99 [11]. Amaar et al. (2022) studied the role of data resampling in this domain. The authors proposed ADASYN in combination with TF-IDF as the best technique to train ML classifiers and obtained the accuracy of 99.9% with extra tree classifier algorithm. However, they have measured the performance of their classifiers on resampled test set and it is considered as a major limitation of their work. Hence their results don't reflect the true performance of the models developed during the study [12].

Naudé et al. (2023) designed and validated a ML system for categorizing identity theft, corporate identity theft as well as multi-level marketing amongst fraudulent job advertisements. The authors have used four categories of features which are empirical rule set-based features, BoW models, word embeddings and transformer-based models for different ML classifiers. Their results showed that a Gradient Boosting (GB) model integrated with empirical rule-set based features, pos tags and BoW vectors achieved the superior results [13]. Afzal et al. (2024) further explored the impact of feature selection along with data resampling method such as SMOTE. The authors have used the traditional TF-IDF method for feature extraction. They also applied the Chi2 and PCA methods for feature selection and used SMOTE oversampling technique to train various ML algorithms including ETC, LR, RF, NB, KNN and ensemble model. The results indicate that the proposed method has achieved an accuracy of 98.4% with a F1 score of 0.91 [14]. Table 1 summarizes the existing studies for fake job dataset in the literature.

From the study of existing literature, we found that most of the previous studies are based on rule-based approach and ML techniques using BoW and TF-IDF methods for data representation. However, rule-based approach needs domain expert and also time consuming. Manually designing a set of features is a complex task. It is not a reliable method and cannot be generalized. On the other hand, BoW and TF-IDF methods are non-contextual and high dimensional representation of text. Many studies have shown skewed performance on imbalanced data due to low recall and F1 scores. Some researchers have worked on balanced subset of data and even obtained good results but in real world we have to deal with imbalanced data. Researchers have also employed data resampling for balancing the dataset but could not achieve the optimal results.

To overcome the above-mentioned limitations, we have proposed a context-enabled general-purpose transformer-based framework using BERT named as Fraud-BERT for fake job identification on imbalanced dataset. BERT is a pre-trained LLM that is based on transformer architecture. It is considered as a major milestone in NLP research due to its ability to understand the dynamic textual content. It can process text bidirectionally. BERT and its variants like RoBERTa [23] and DistilBERT [24] are widely adopted by NLP research community for various domains and applications such as text classification [25], sentiment analysis [21], semantic search [26]. Researchers have proposed BERT in several related domains such as fake news detection [27] and cyber bullying analysis [28]. However, it has not been applied in the area of online recruitment fraud detection. Hence the goal of this study is to explore the potential of a language model for fake job dataset. We have also performed a comparative study between the proposed model and the traditional models. For impartial comparison, we have also developed several classification models based on traditional machine learning algorithms such as RF, NB, SVM, LR and Bidirectional Long Short-Term Memory (Bi-LSTM). Among these, Bi-LSTM is a popular DL model employing word embedding for sequence categorization due to its ability to process textual data in forward and backward direction. To the best of our knowledge, only one study has utilized it in the existing literature. To fill this gap also, we have also built Bi-LSTM model and compared it with the proposed model.

3 Materials and methods

Subsection 3.1 of this section presents description of dataset, subsections 3.2 and 3.3 provides a brief introduction to pre-trained language models and transfer learning approach respectively. Subsection 3.4 discusses the proposed model architecture and methodology adopted in this work.

3.1 Dataset description

Nowadays, EMSCAD dataset is also named as fake job dataset. We have used the fake job dataset in this study and downloaded it from Kaggle [22]. It consists of 17,880 job ads. The dataset is highly imbalanced with 17,014 instances of legitimate jobs and 866 instances of fake jobs. It consists of 18 attributes which are described in Table 2. Our dataset consists of both textual as well as meta information about the jobs. In this study we have used fraudulent as target variable. It is a binary variable with two values: 0 stands for legitimate job and 1 stands for fake job. Figure 1 shows the imbalanced distribution of dataset.

3.1.1 Data pre-processing

We have analysed and pre-processed our data in the beginning before performing the experiments. After analysing the dataset, we found that the binary variables such as telecommuting, has_company_logo and has_questions have minimal or no correlation with fraudulent attribute. So, initially we dropped these columns from the dataset and later we also dropped job_id and salary_range attributes which can be considered irrelevant for a text classification problem. We have selected all the remaining 12 independent attributes and concatenated them into single text column as shown in Table 3. The text is also comprised of null values and other noisy data such as special characters, numbers, stop words, punctuations etc. We performed the removal of such content and obtained a cleaned dataset.

Table 1 Summary of existing studies on fake job dataset

Reference	Year	Models	Method	Results (Highest Performing Model)	Merit	Demerit
[3]	2017	ZeroR, OneR, NB, J48, DT, RF, LR	BoW	RF (F1: 0.912, P: 0.914, R: 0.912, AUC: 0.97)	Low computational cost due to simple models	Used a small balanced subset of data
[4]	2018	J48, JRip, NB	Rule-based	JRip (Accuracy: 96.19, P: 0.935, R:0.915)	Less computational cost	Rule-based approach, Need human expertise
[5]	2019	RF	Ensemble	Accuracy: 97.41%	Less computational cost	Model over-fitting
[6]	2019	LR, J48, RF, Ensemble	TF-IDF	Ensemble (Accuracy: 95.4%, F1: 94.4%)	Over-fitting is reduced with the use of ensemble learning	Low accuracy
[8]	2020	NB, KNN, RF, DT, SVM	Ensemble	RF (Accuracy: 98.2%)	Simple techniques are used	Model over-fitting
[9]	2021	Two class decision forest, Two class boosted decision trees	–	Accuracy: 93.8%, R: 0.75, F1: 0.73)	Less computational cost	Low F1 and recall score
[10]	2021	KNN, LR, RF, Bi-LSTM	TF-IDF, Word2Vec	Bi-LSTM (Accuracy: 0.98, P: 0.90, R: 0.81, F1: 0.85)	Simple techniques are used	Low F1 and recall score
[11]	2022	KNN, SVM, DT	Rule-based	SVM (Accuracy: 97.64%, R: 0.97, F1: 0.99)	High performance	Time consuming, not reliable, need domain expert
[12]	2022	ETC, LR, SVM, GRU, LSTM, CNN	TF-IDF, Word2Vec	ETC (F1: 0.99, Accuracy: 99.9%)	Use of data resampling technique	Model overfitting, model is evaluated on resampled test set
[13]	2023	SGD, SVM, RF, GB	TF-IDF	GB (F1: 0.88)	Good use of SOTA techniques	Low F1 score
[14]	2024	ETC, RF, LR, NB, KNN, Ensemble	TF-IDF	ETC (Accuracy: 98.4%, F1: 0.91, P: 0.97, R: 0.86, AUC: 0.85)	Use of feature selection and SOTA resampling method	Low recall and AUC scores

Table 2 Dataset description

Attribute	Description
job_id	Indicates the unique job id
title	Describes the title of job
location	Shows the geographical location of job posts
salary_range	Gives details of salary packages
department	Provides information about working department such as sales
company_profile	Presents information about the company
description	A detailed description about job
requirements	Lists the requirements to be fulfilled for the post
benefits	States the proposed benefits of applying for job
telecommuting	Indicates whether the job requires telecommuting works
has_company_logo	Whether the company has its own company logo or not
has_questions	Whether the company has asked questions or not
employment_type	Categorizes the type of employment provided such as full-time, part-time, contractual, etc
required_experience	Information about type of previous experience required for the job
required_education	States the educational qualification required for the job
industry	Domain of the company like healthcare, IT, real estate, etc
function	Type of profession the company follows like engineering, research, etc
fraudulent	Whether the job is legitimate or fraudulent

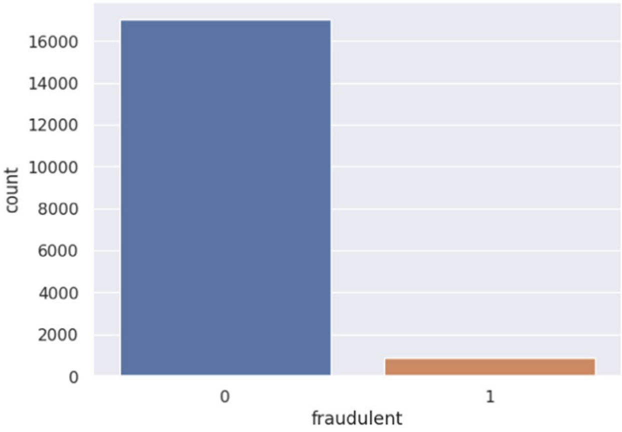


Fig.1 Imbalanced distribution of dataset

Table 3 Pre-processed dataset (sample)

Fraudulent	Text
0	account executive washington dc dc washington…
0	customer service cloud video production nz…
0	commission machinery assistant cma ia wever…
0	market intern ny new york marketing …
0	bill review manager fl fort worth spotsource…

3.2 Pre-trained language model

This section introduces the background of proposed method. A pre-trained language model can be defined as a model which is trained in two phases: pre-training and fine-tuning. During the pre-training phase, the model is trained on one or more NLP tasks with huge amount of unsupervised data. The process of pre-training allows the model to learn complex language representations. In the fine-tuning step, weights of the general purpose pre-trained model obtained during pre-training phase can be modified according to the downstream NLP task. The underlying principle behind these language models is to predict the next word given a set of previous words. In unsupervised learning, probabilistic language modelling is used to estimate probability density. The main objective of a language model is to calculate joint probability $P(s)$ as given in Eq. (1). For a given sentence $s = (s_1, s_2, \ldots, s_n)$, the joint probability can be calculated as follows: [29]

$$P(s) = \prod_{k=1}^{N} P(s_k|s_{<k}) \tag{1}$$

After this, the conditional probability distribution is evaluated by neural network. The autoregressive forward factorization function given in Eq. (2) is utilized to improve the probability of pre-training.

$$\begin{aligned} max_\theta \log p_\theta(s) &= \sum_{k=1}^{N} \log p_\theta(s_k|s_{<k}) \\ &= \sum_{k=1}^{K} \frac{\exp\left(h_\theta\left(s_{1:k-1}\right)^T e\left(s_k\right)\right)}{\sum_{s'} \exp\left(h_\theta\left(s_{1:k-1}\right)^T e(s')\right)} \end{aligned} \tag{2}$$

where $h_\theta(s_{1:k-1})$ is the hidden state vector representation according to parameter θ and $e(s_i)$ represents the embedding vector of word s_i..

Unlike conventional static pre-trained word embeddings such as GloVe, Word2Vec and FastText, transformer-based pre-trained language models preserve the dynamic context of word by establishing relationships between different words in the sentence using multi-headed self-attention mechanism. Nowadays, these are popularly known as large language models as they are pre-trained on sufficiently large and diverse corpora of text. In this study, we have used BERT which is pre-trained on English Wikipedia (2500 million words) and BooksCorpus (800 million words). Several existing transformer-based models process text in unidirectional manner but the bidirectional pre-training in BERT allows it to capture both the left and right context of words for complex language understanding. It can be fine-tuned for a particular task or domain with the addition of task specific layer on top of it. BERT model is available in two sizes: $BERT_{base}$ trained with 110 M parameters and $BERT_{large}$ trained with 340 M parameters. In this paper, we have trained BERT with *Masked Language Modeling* objective that masks 15% text in the input sequence and predicts masked tokens from the contextual representation of given sequence. Given a text sequence s, masked tokens are predicted from real tokens according to Eq. (3). Here, S_{masked} indicates the masked tokens and $S_{non-masked}$ indicates the real tokens.

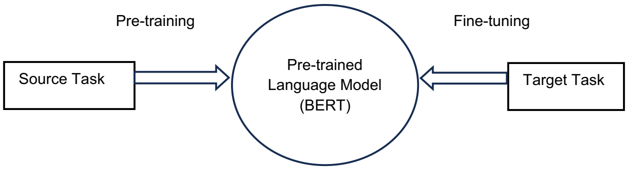
$$\begin{aligned} max_\theta \log p_\theta\left(S_{masked}|S_{non-masked}\right) &\approx \sum_{k=1}^{N} m_k \log p_\theta(S_k|S_{non-masked}) \\ &= \sum_{k=1}^{N} m_k log \frac{\exp\left(H_\theta\left(S_{non-masked}\right)_k^T e\left(s_k\right)\right)}{\sum_{s'} \exp(H_\theta\left(S_{non-masked}\right)_k^T e(s'))} \end{aligned} \tag{3}$$

Here, H_θ represents the hidden vector of sentence s of length k calculated by transformer with parameter θ. The value of m_k as 1 indicates that s_k is masked token.

3.3 Transfer learning

Transfer learning is a process in which the knowledge obtained by a pre-trained language model during training on source task can be transferred or utilized to solve target task. Figure 2 shows the transfer learning setup in which BERT is first pre-trained on source task such as *Masked Language Modeling* using a very large corpora of text, then the knowledge acquired by BERT during pre-training phase is utilized to solve target task or downstream task such as text classification by fine-tuning. During fine-tuning, the weights of the original BERT are modified according to our dataset, eliminating the need to build task specific NLP model from scratch.

Fig. 2 Transfer learning setup



3.4 The proposed model architecture

In this research work, we have adapted $BERT_{base}$ via fine-tuning and proposed a novel transformer-based framework for fake job identification. BERT utilizes the encoder part of original transformer architecture to produce contextualized embeddings. First, $BERT_{base}$ is loaded with pre-trained parameters and then all of the parameters are updated using our labelled dataset to begin the fine-tuning process. Figure 3 shows the proposed model architecture. Before passing any input data to the model, we prepared it according to the format understandable by BERT. Hence, the input sentence is first converted into tokens with the help of BERT WordPiece tokenizer having a vocabulary of size 30 k. It splits a word into several sub-words (tokens). A piece of word is prefixed with ## (as shown in Fig. 3). Use of sub-words in place of words substantially reduces the total vocabulary size of BERT (only 30 k) and there are fewer possible out-of-vocabulary (OOV) tokens.

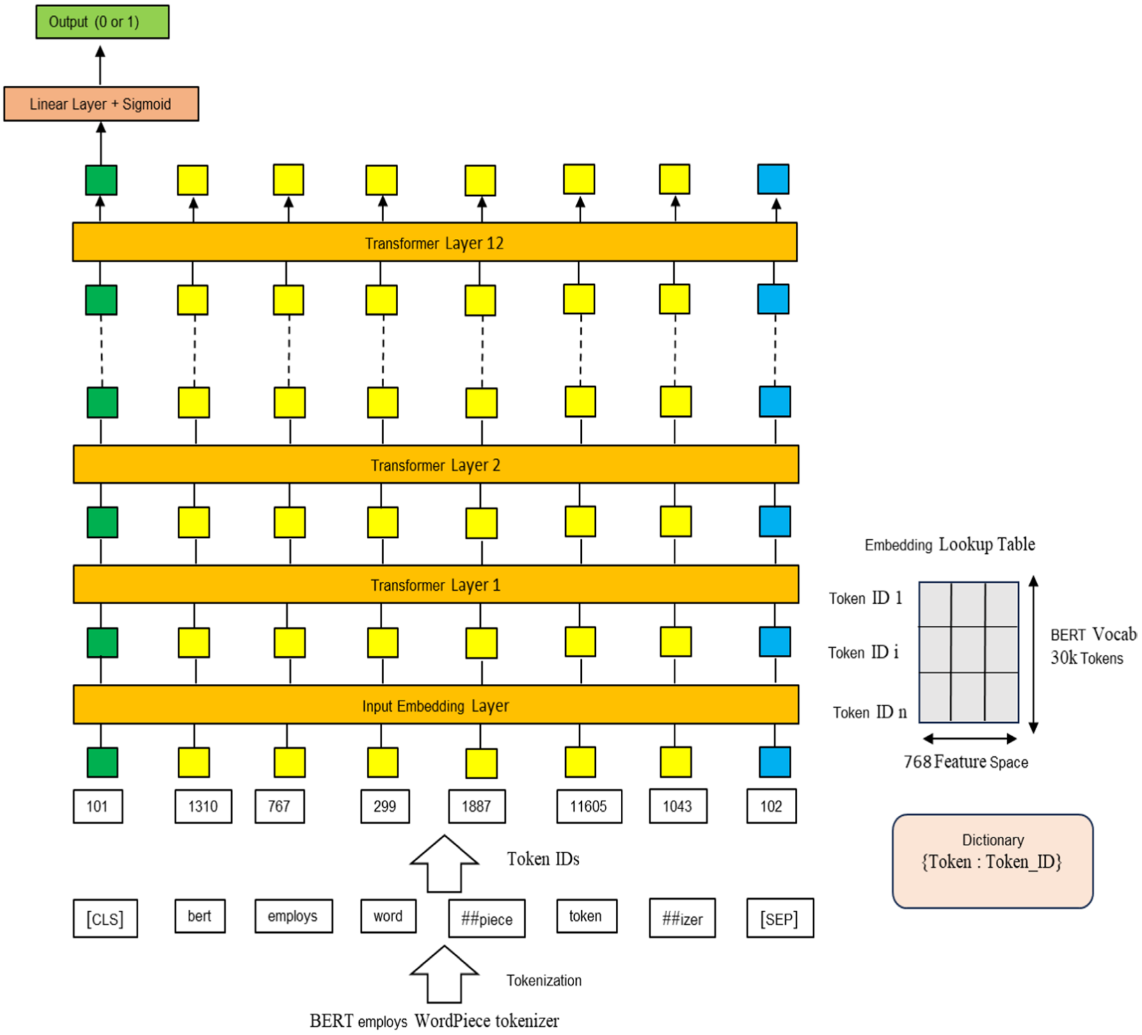


Fig. 3 Proposed model architecture

After this, we add some special tokens such as [CLS], [SEP] and [PAD] to the sequence. [CLS] token is added in the beginning and it presents the summary of the whole sentence. [SEP] token is used to separate two sentences or as the end token. A BERT model can process a maximum sequence length of 512. The sequence length can also be pre-defined according to the dataset. [PAD] token is used to pad the sentences shorter than the chosen sequence length and the longer sentences will get truncate. An attention mask is used to differentiate between [PAD] tokens and the real tokens. Each token is then encoded to its ID that corresponds to its index in an embedding lookup table (as shown in Fig. 3).

In the input embedding layer, the input embedding for each token is calculated as the sum of token embedding, segmentation embedding and positional embedding. Token embeddings are obtained from the embedding lookup table where rows represent all possible token IDs in the vocabulary and columns represent the token embedding size. Segmentation embedding indicates whether a given token belongs to the first or second sentence. Positional embedding indicates the position of tokens in a sentence. Unlike the original transformer, BERT does not use trigonometric functions; instead, it learns positional embeddings from absolute ordinal position.

These input embeddings are then passed to transformer layers. Each layer employs the multi-headed self-attention mechanism to understand the dynamic contextual relationships between words and passes its output to the upper layer. The last transformer layer will output the embedding vector having dimension 768 for each token. The output embedding corresponding to [CLS] token represents the pooled sequence embedding and passed to linear layer with sigmoid activation function to predict the correct label. Figure 4 illustrates the layout of methodology adopted for proposed work.

4 Experiment

In this study, the experiments are carried out in *Google Colab* using python programming language and single *A100 GPU*. Along with the proposed model, we have also applied standard conventional ML models such as LR, SVM, NB, RF and DL algorithm Bi-LSTM. These models are considered as baseline models in this research work. In each case we divided the dataset in a ratio of 80:20 for training and testing. The rest of the implementation details are given in the following subsections.

4.1 Baseline models

The baseline models are implemented using *Scikit-learn*, *Keras* and *Tensorflow* python libraries. ML models are developed using *Scikit-learn* and text data is represented using the traditional TF-IDF feature extraction method. We have used the multinomialNB and SVC (with linear kernel) as variants of NB and SVM respectively with default hyper-parameter settings in each case. *Keras* and *Tensorflow* libraries are used to import deep learning layers for building Bi-LSTM model. For Bi-LSTM, we have used an embedding layer with vec_size 300, two Bi-LSTM layers (each with 60 units), two dropout layers (with dropout rate 0.2) and one dense layer for classification. We have used Adam optimizer with binary cross-entropy

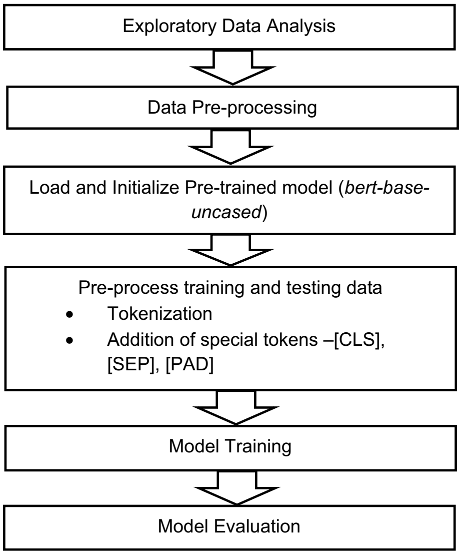
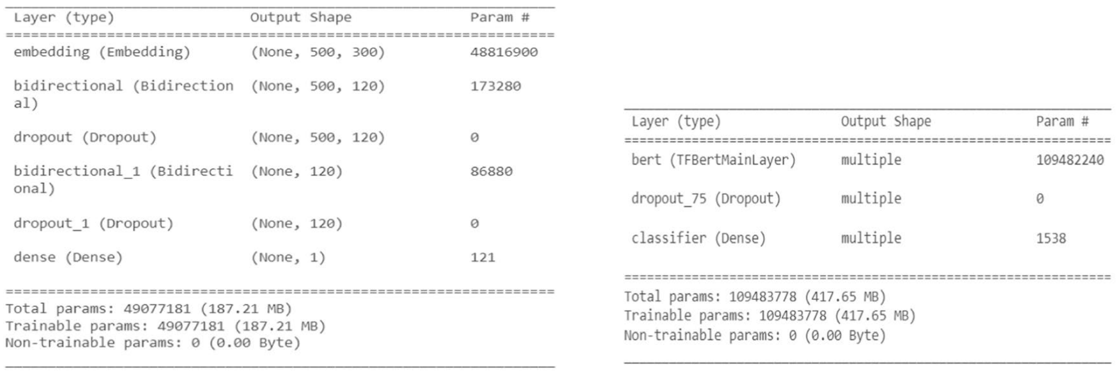

Exploratory Data Analysis
Data Pre-processing
Load and Initialize Pre-trained model (*bert-base-uncased*)
Pre-process training and testing data
• Tokenization
• Addition of special tokens –[CLS], [SEP], [PAD]
Model Training
Model Evaluation

Fig. 4 The proposed methodology



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 500, 300)          48816900

 bidirectional (Bidirection  (None, 500, 120)          173280
 al)

 dropout (Dropout)           (None, 500, 120)          0

 bidirectional_1 (Bidirecti  (None, 120)               86880
 onal)

 dropout_1 (Dropout)         (None, 120)               0

 dense (Dense)               (None, 1)                 121

=================================================================
Total params: 49077181 (187.21 MB)
Trainable params: 49077181 (187.21 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

(a)

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bert (TFBertMainLayer)      multiple                  109482240

 dropout_75 (Dropout)        multiple                  0

 classifier (Dense)          multiple                  1538

=================================================================
Total params: 109483778 (417.65 MB)
Trainable params: 109483778 (417.65 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

(b)

Fig. 5 **a** Bi-LSTM; **b** BERT

Table 4 Hyper-parameter values

Hyper-parameter	Value
Learning rate	1e–5
Maxlen	510
Batch size	6
Epochs	2

loss function. These parameters have been decided based upon dataset size and standard DL practioner's guidelines. The model is trained for 2 epochs with a batch size of 128. We have also experimented with greater number of epochs but it led model to overfit with greater difference between training and validation accuracy. Figure 5a shows the summary of Bi-LSTM model.

4.2 Proposed model

We have implemented the proposed model using python's *ktrain* [30] library and downloaded *bert-base-uncased* with pre-trained parameters from Hugging Face.[1] According to our pre-processed dataset we need to process long sequences, so we set the maximum sequence length as 510. The sentences shorter than this length are padded with [PAD] tokens and the longer sentences get truncated. Hence there is only a little loss of information. We trained the model for 2 epochs with a learning rate of 1e-5. The process of fine-tuning was finished in just 20 min. Figure 5b represents the summary of the proposed model. The details of hyperparameters for training our model are mentioned in Table 4. Here maxlen indicates the maximum sequence length.

4.3 Model evaluation metrics

In this research study, we are working on an imbalanced dataset for which accuracy cannot be considered as a sufficient performance metric. Besides accuracy, we have also chosen precision (P), recall (R), F1-score (F1), area under ROC curve (AUC) and used their macro averaged value to measure the effectiveness of the proposed approach. We have also performed the comparison of the proposed model with baseline models using these metrics.

[1] https://huggingface.co/models

Table 5 Comparison of proposed model with baseline models

Model	P	R	F1	Accuracy	AUC
Proposed Model	**0.94**	**0.92**	**0.93**	**0.99**	**0.99**
Bi-LSTM	0.91	0.86	0.89	0.98	0.88
SVM	0.98	0.84	0.89	0.98	0.98
NB	0.48	0.50	0.49	0.95	0.85
LR	0.98	0.70	0.78	0.97	0.98
RF	0.99	0.82	0.88	0.98	0.98

The bold values highlight the results of proposed method

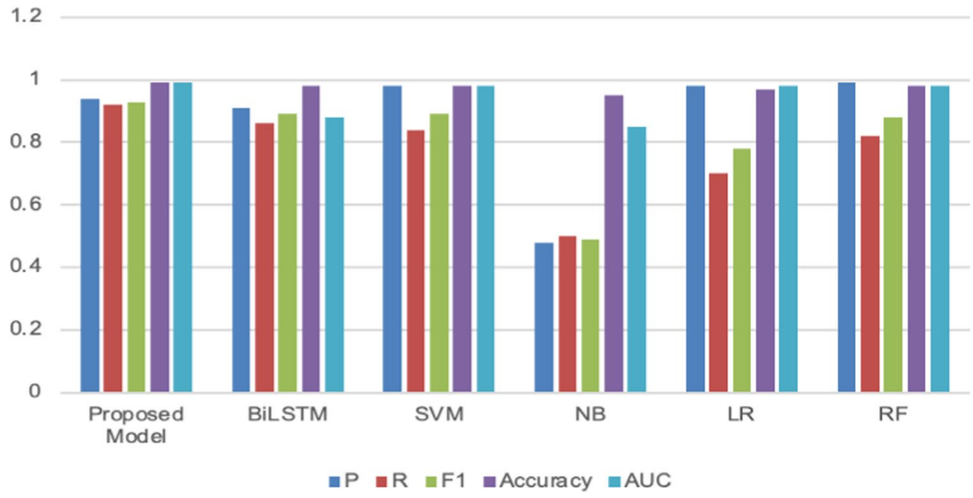


Fig. 6 Performance comparison of proposed model over baseline models

5 Result and discussions

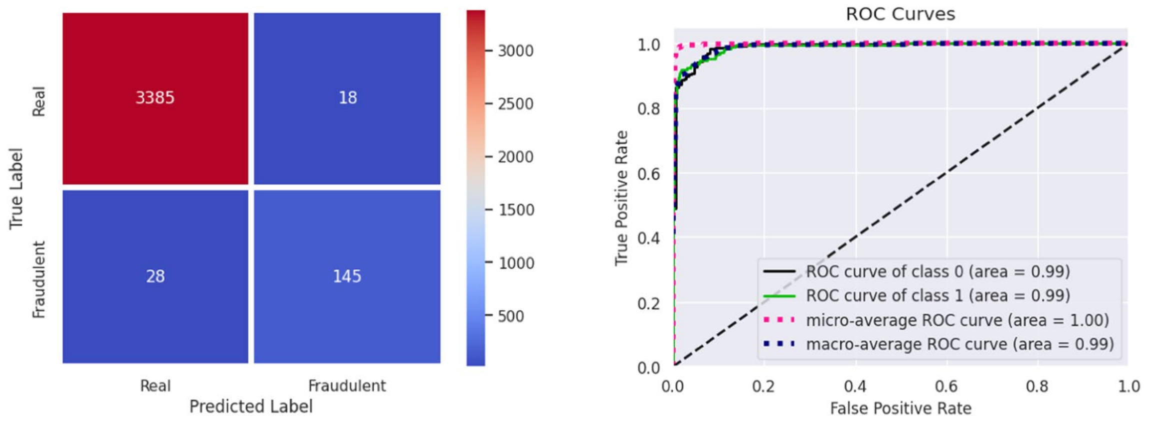
5.1 Performance evaluation of proposed model against baseline models

The performance of the proposed approach in comparison to baseline models is shown in Table 5. It can be clearly observed from the table that the proposed model has achieved the highest recall (0.92) and F1 score (0.93) as compared to traditional models. It indicates that our model is least affected by data imbalance. We have also obtained superior performance in terms of precision (0.94), accuracy (0.99) and AUC (0.99) scores. SVM is the second-best performing model with F1 score (0.89), recall (0.84) and accuracy (0.98) but it took more than one hour during training to obtain these results. Bi-LSTM has obtained good accuracy but low recall and AUC scores which shows the data hungry nature of deep learning model and inability to deal with data imbalance problem. On the other hand, our model took approximately 20 min training time and gave remarkable results. Figure 6 graphically illustrates the results obtained in the study.

The high recall and F1 scores obtained in the proposed model indicate a smaller number of misclassification errors (false positives and false negatives). This can also be verified with the help of confusion matrix as shown in Fig. 7a. We have also plotted the ROC curve that shows the AUC value for the proposed model as shown in Fig. 7b.

5.2 Comparison with SOTA studies

A comparative analysis of proposed model is also performed with existing studies on the same dataset as shown in Table 6. It can be observed from the table that previous studies have used the traditional ML algorithms. Vidros et al. [3] have obtained good performance on a smaller balanced corpus of dataset but it does not promise its validity for the original dataset. Chiraratanasopa and Chay-intr (2022) has used rule-based approach which is very time consuming and need domain expert [11]. Amar et al. (2022) have studied the role of data resampling and also



(a)

(b)

Fig. 7 **a** Confusion Matrix; **b** ROC Curve

Table 6 Comparison with state-of-the-art studies

Reference	Year	Highest performing model	P	R	F1	Accuracy (%)	AUC
[3]	2017	RF	0.914	0.912	0.912	91.22	0.97
[4]	2018	JRip	0.935	0.915	–	96.19	–
[5]	2019	RF	–	–	–	97.41	0.95
[8]	2020	RF	–	–	–	98.2	–
[9]	2021	Two class decision forest	0.72	0.75	0.73	93.8	–
[10]	2021	Bi-LSTM	0.90	0.81	0.85	98	–
[11]	2022	SVM	–	0.97	0.99	97.64	–
[12]	2022	ETC	–	–	0.99	99.9	–
[13]	2023	GB	–	–	0.88	–	–
[14]	2024	ETC	0.97	0.86	0.91	98.4	0.85
Ours	**2024**	**Fraud-BERT**	**0.94**	**0.92**	**0.93**	**99**	**0.99**

The bold values highlight the results of proposed method

reported the results of their study measured on a resampled test set which does not reflect the true performance and considered as a major drawback of their study [12]. Afzal et al. [14] have applied feature selection technique along with data resampling but could not achieve optimal results. Here, we can state that our approach advances the SOTA results on the original imbalanced dataset without the application of any resampling technique for data balancing.

6 Integration challenges and strategies

BERT is a large size model. It consists of millions of parameters which leads to high memory and computational requirements, making it difficult to deploy in resource-constrained environments like mobile devices or low-latency applications. Some other challenges related to integration of BERT-based models into real-world applications include inference latency, lack of interpretability and requirement of high computational resources including such as powerful GPUs or TPUs. However, by employing strategies like model compression, efficient fine-tuning, domain-specific adaptation, and leveraging scalable cloud services such as Google Colab, Azure, etc., these challenges can be mitigated, enabling the effective use of BERT.

6.1 Compatibility with existing systems and data interoperability

Upgrades to the BERT architecture or tokenization techniques may cause version incompatibilities when utilizing third-party libraries or deploying in production. Following strategies can be employed to mitigate this risk.

- *Version Control:* Carefully track and document the versions of BERT models and tokenizers used in training and production to avoid compatibility issues.
- *API Compatibility:* Use stable, well-maintained libraries like Hugging Face's Transformers for model deployment, as they ensure backward compatibility and provide updates for different versions of pre-trained models.

7 Limitations and potential risks

7.1 Computational overhead

BERT-based model consists of millions of parameters, this requires high computational resources for training and inference. High resource consumption can limit deployment in real-time applications or on devices with limited computational capabilities (e.g., mobile phones, edge devices). Also, the self-attention mechanism in BERT operates quadratically with respect to the input sequence length, making inference slow, especially for long texts, this can lead to poor user experience. However, use of light weight BERT variants such as DistilBERT [24] can speed up the inference with a little less accuracy.

7.2 Possible failure points

BERT can process a sequence with maximum length of 512 tokens. Truncating long texts to fit within the token limit can cause important contextual information to be lost, leading to inaccurate predictions.

7.3 Trade-offs between accuracy and efficiency

We have used $BERT_{base}$ in this study. More complex versions of BERT such as $BERT_{large}$ may yield superior performance but are computationally expensive to train and deploy. Using larger models in production may not be feasible in terms of time and resources, especially for startups or small organizations. They may also become too slow for real-time applications.

7.4 Ethical implications

BERT models inherit biases from the data, they are trained on. This can lead to biased or unfair predictions, particularly for tasks involving sensitive attributes like gender, race, or socioeconomic status.

7.5 Data privacy and security

Fine-tuning BERT model on sensitive data raises concerns about privacy and data security. Techniques such as federated learning and differential privacy [31, 32] may mitigate this risk at the cost of reduced accuracy. The balance between model security and predictive power needs to be carefully managed.

7.6 Scalability for large datasets

BERT is a powerful model. It can scale well for large datasets but also requires high computational resources such as powerful GPUs/TPUs which are available on cloud platforms. In this study, we have also worked on a large dataset and used *A100* GPU available on *Google Colab*. However, various techniques such as distributed training, data and model parallelism, knowledge distillation, etc. can be utilized to address scalability issues. We can also use more efficient

BERT variants such as DistilBERT [24] (which has been kept under consideration for future research) to reduce the computational burden. But there is often a trade-off between accuracy and efficiency for large datasets.

8 Conclusion and future scope

Online recruitment fraud is a serious cyber-crime that is affecting the lives of many innocent people at an alarming rate. It has not only incurred the loss of privacy and security of an individual but also the financial loss. In the literature, many researchers have proposed different ML approaches for automatic detection of fake job posts using traditional BoW and TF-IDF methods for feature extraction. However, these methods are not reliable as they are not able to understand the context of text. In the domain of fraud detection, it is very important to understand the textual context as fraudsters take immense care while crafting the fake text messages to imitate the pattern of legitimate ones and it is difficult even for a human expert to differentiate between real and fake post. Understanding this need, this paper presents transformer-based contextual framework using BERT, a large language model via transfer learning approach. We have modified the parameters of original BERT by fine-tuning it according to the domain of study. Since, a small amount of data is sufficient to fine-tune BERT model, our approach is least affected by imbalanced nature of data. The results also indicate that we have achieved the highest performance as compared to baseline models as well as existing SOTA studies.

In this study, we have used BERT model. It has high computational requirements and cannot process long sequences. In the future, we would like to use other variants of BERT such as DistilBERT [24], XLNET [33], DeBERTa [34], etc. DistilBERT is a compressed variant of BERT that retains 97% of the language understanding capabilities while being 60% faster and smaller. XLNet is based on Transformer-XL architecture, which has the ability to handle long dependencies. DeBERTa is an improved variant of BERT with disentangled attention mechanism and enhanced mask decoder. Furthermore, integrating Internet-of-Things (IoT) and Artificial Intelligence (AI) for ORF detection and prevention can be more promising and proactive approach. IoT devices can provide real-time, continuous data streams that capture behavioural, geolocation, and device usage patterns, while AI models can analyse this data for suspicious activity and anomalies. By combining these technologies, recruitment platforms can detect fake job postings, prevent identity fraud during candidate verification, and identify fraudulent recruiters.

Author contributions Khushboo Taneja conceptualised the study, performed the experiments and wrote the original manuscript. Saroj Ratnoo and Jyoti Vashishtha gave the expert advice, supervised the work and also reviewed the manuscript. All authors read and approved the final manuscript.

Funding Not applicable.

Data availability The dataset used in this work is publicly available and downloaded from Kaggle website (https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction).

Code availability The code for the proposed work is publicly available at https://github.com/GJU-CSE/Online-Recruitment-Fraud-Detection.

Declarations

Informed consent Not applicable.

Competing interests All authors declare that they have no conflict of interest.

Research involving human and/or animals Not applicable.

References

1. Sengupta A. Got job offer on WhatsApp? Think hard before you reply or accept. 2022. https://www.indiatoday.in/technology/news/story/scammers-are-using-whatsapp-to-send-fake-job-offers-don-t-be-fooled-1988936-2022-08-17. (Accessed 15 Jan 2024)
2. Pathak P. WhatsApp private job scams on the rise in India: How to stay safe online, all your questions answered. 2023. https://www.financialexpress.com/life/technology-whatsapp-private-job-scams-on-the-rise-in-india-how-to-stay-safe-online-all-your-questions-answered-3097094/. (Accessed 18 Jan 2024)
3. 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):1–19. https://doi.org/10.3390/fi9010006.
4. Mahbub S, Pardede E. Using contextual features for online recruitment fraud detection. In: Proceedings of 27th International Conference on Information Systems Development. ISD 2018 Lund, Sweden 2018.
5. Alghamdi B, Alharby F. An intelligent model for online recruitment fraud detection. J Inf Secur. 2019;10(3):155–76. https://doi.org/10.4236/jis.2019.103009.
6. Lal S, Jiaswal R, Sardana N, Verma A, Kaur A, Mourya R. ORFDetector: Ensemble Learning Based Online Recruitment Fraud Detection. In: Proceedings of 12th international conference on contemporary computing, IC3 2019, pp. 1–5. https://doi.org/10.1109/IC3.2019.8844879.
7. Dutta S, Bandyopadhyay SK. Fake job recruitment detection using machine learning approach. Int J Eng Trends Technol. 2020;68(4):48–53. https://doi.org/10.14445/22315381/IJETT-V68I4P209S.
8. Nasser IM, Alzaanin AH. Machine learning and job posting classification: a comparative study. Int J Eng Informat Syst. 2020;4(9):6–14.
9. Shibly F, Sharma U, Naleer H. Performance comparison of two class boosted decision tree snd two class decision forest algorithms in predicting fake job postings. Annals of RSCB. 2021;25(4):2462–72.
10. Anita CS, Nagarajan P, Sairam GA, Ganesh P, Deepakkumar G. Fake job detection and analysis using machine learning and deep learning algorithms. Rev Gestão Inovação e Tecnologias. 2021;11(2):642–50. https://doi.org/10.47059/revistageintec.v11i2.1701.
11. Chiraratanasopha B, Chay-Intr T. Detecting fraud job recruitment using features reflecting from real-world knowledge of fraud. Curr Appl Sci Technol. 2022;22(6):1–12. https://doi.org/10.55003/cast.2022.06.22.008.
12. Amaar A, Aljedaani W, Rustam F, Rupapara V, Ludi S, Ullah S. Detection of fake job postings by utilizing machine learning and natural language processing approaches. Neural Process Lett. 2022;54(3):2219–47. https://doi.org/10.1007/s11063-021-10727-z.
13. Naudé M, Adebayo KJ, Nanda R. A machine learning approach to detecting fraudulent job types. AI & Soc. 2023;38(2):1013–24. https://doi.org/10.1007/s00146-022-01469-0.
14. Afzal H, Rustam F, Aljedaani W, Abubakar M, Ullah S, Ashraf I. Identifying fake job posting using selective features and resampling techniques. Multi Tools Applicat. 2023. https://doi.org/10.1007/s11042-023-15173-8.
15. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: Proceedings of the 31st International conference on advances in neural information processing systems, NeurIPS. NIP's17, pp. 6000–6010, Long Beach, California, USA 2017. https://doi.org/10.5555/3295222.3295349.
16. Taneja K, Vashishtha J. Recent advancements in natural language processing. J Criti Rev. 2020;7(19):6807–14.
17. Min B, Ross H, Sulem, Veyseh APB, Nguyen TH, Sainz O, Agirre E, Heintz I, Roth D. Recent advances in natural language processing via large pre-trained language models: a survey. ACM Comput Surv. 2023;56(2):1–40. https://doi.org/10.1145/3605943.
18. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota 2019. https://doi.org/10.18653/V1/N19-1423.
19. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ. Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. 2020;21:1–67. https://doi.org/10.48550/arxiv.1910.10683.
20. Ray PP. ChatGPT A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Int Things Cyber-Phys Syst. 2023;3:121–154. https://doi.org/10.1016/j.iotcps.2023.04.003.
21. Taneja K, Vashishtha J, Ratnoo S. Transformer based unsupervised learning approach for imbalanced text sentiment analysis of E-commerce reviews. Procedia Comput Sci. 2024;235:2318–31. https://doi.org/10.1016/j.procs.2024.04.220.
22. Real or fake job posting prediction–Kaggle. 2019. https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction.
23. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. RoBERTa: a robustly optimized bert pretraining approach. 2019. https://arxiv.org/abs/1907.11692v1. (Accessed 13 Jul 2021)
24. Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2019. http://arxiv.org/abs/1910.01108. (Accessed 22 May 2023).
25. Taneja K, Vashishtha J. Comparison of transfer learning and traditional machine learning approach for text classification. In: Proceedings of the 2022 9th International conference on computing for sustainable global development, INDIACom 2022, pp. 195–200, New Delhi, 2022. https://doi.org/10.23919/INDIACom54597.2022.9763279.
26. Taneja K, Vashishtha J, Ratnoo S. efficient deep pre-trained sentence embedding model for similarity search. Int J Comput Inf Syst Ind Manag Appl. 2023;15:605–15.
27. Kaliyar RK, Goswami A, Narang P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. Multimed Tools Appl. 2021;80(8):11765–88. https://doi.org/10.1007/s11042-020-10183-2.
28. Paul S, Saha S. CyberBERT: BERT for cyberbullying identification: BERT for cyberbullying identification. Multimed Syst. 2022;8(6):1897–904. https://doi.org/10.1007/s00530-020-00710-4.
29. Dai Z, Yang Z, Yang Y, Carbonell J, Le QV, Salakhutdinov R. Transformer-XL: Attentive language models beyond a fixed-length context. In: Proceedings of 57th Annu. Meet. Assoc. Comput. Linguist. Conf., pp. 2978–2988, 2020. https://doi.org/10.18653/v1/p19-1285.
30. Maiya AS. Ktrain: a low-code library for augmented machine learning. J Mach Learn Res. 2022;23(158):1–6.

31. Salam A, Abrar M, Ullah F, Khan IA, Amin F, Choi GS. Efficient data collaboration using multi-party privacy preserving machine learning framework. IEEE Access. 2023;11:138151–64. https://doi.org/10.1109/ACCESS.2023.3339750.
32. Salam A, Abrar M, Amin F, Ullah F, Khan IA, Alkhamees BF, Alsalman H. Securing smart manufacturing by integrating anomaly detection with zero-knowledge proofs. IEEE Access. 2024;12:36346–60. https://doi.org/10.1109/ACCESS.2024.3373697.
33. Yang Z, Dai Z,Yang Y, Carbonell J, Salakhutdinov R, Le QV. XLNet: Generalized autoregressive pretraining for language understanding. In: Proceedings of Adv. Neural Inf. Process. Syst., vol. 32, pp. 1–11, 2019.
34. He P, Liu X, Gao J, Chen W. DeBERTa: decoding-enhanced BERT with Disentangled Attention. 2020. https://arxiv.org/abs/2006.03654v4.