Neural Processing Letters (2022) 54:2219–2247
https://doi.org/10.1007/s11063-021-10727-z

Detection of Fake Job Postings by Utilizing Machine Learning and Natural Language Processing Approaches

Aashir Amaar[1] · Wajdi Aljedaani[2] · Furqan Rustam[1] · Saleem Ullah[1] · Vaibhav Rupapara[3] · Stephanie Ludi[2]

Accepted: 18 December 2021 / Published online: 4 January 2022

Abstract
The modern era is about everything that can be handled virtually in human life, such as online banking, education, security, job, etc. This increase in technology use also makes it easy for a scammer to loot people and make money quickly. A popular scam nowadays is fake job advertisements. People apply for these fake job vacancies, pay application fees to scammers, send their data to the scammers, and end up with a scam and waste their money. For this purpose, we proposed a methodology that uses natural language processing and supervised machine learning techniques to detect fraudulent job ads from online recruitment portals. We used two feature extraction techniques to extract the features from data: Term Frequency-Inverse Document Frequency (TF-IDF) and Bag-of-Words (BoW). In the study, we used six machine learning models to analyze whether these job ads are fraudulent or legitimate. Then, we compared all models with both BoW and TF-IDF features to analyze the classifier's overall performance. One of the challenges in this study is our used dataset. The ratio of real and fake job posts samples is unequal, which caused the model over-fitting on majority class data. To overcome this limitation, we used the adaptive synthetic sampling approach (ADASYN), which help to balance the ratio between target classes by generating the number of sample for minority class artificially. We performed two experiments, one with the balanced dataset and the other with the imbalanced data. Through experimental analysis, ETC achieved 99.9% accuracy by using ADASYN as over-sampling ad TF-IDF as feature

✉ Furqan Rustam
furqan.rustam1@gmail.com

Aashir Amaar
aashiramaar97770@gmail.com

Wajdi Aljedaani
wajdialjedaani@my.unt.edu

Vaibhav Rupapara
vaibhav.rupapara.sept@gmail.com

Stephanie Ludi
stephanie.ludi@unt.edu

1 Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan, Pakistan

2 University of North Texas, Denton, USA

3 Florida International University, University Park, USA

extraction. Further, this study also performs an in-depth comparative analysis of our proposed approach with state-of-the-art deep learning models and other re-sampling techniques.

Keywords Fake ads detection · Feature extraction · Machine learning · NLP · Online fake job posts

1 Introduction

Fake job ads can harm the reputation of the platform through misleading information for the user. Primarily the misleading information is spread across the internet, for instance, on Google Plus, Twitter, Facebook, and other web apps [1]. On the one hand, social media websites are an advancement in the IT field, while on the other hand, one can easily manipulate these platforms to forge information such as posting false jobs.

The fake job posting is currently a painful issue on social media platforms. Once fake job posts have initially been sown on the internet by their producers, the individuals can access them like wildfire. Typically, users might share the information on their personal media channels by making posts using Twitter or Facebook. Further liaison, for instance, 'likes' on a post, also triggers the social media algorithm; consequently, the information becomes visible to several users. This phenomenon is called 'organic reach' [2]. There is confirmation that fraudulent data is spread more rapidly and widely through social media networks as users can independently post it without any validation of information [3].

When humans interact with fraudulent information online, they get influenced by it as they believe it would be correct. Effective methods of social data processing recommend that there are various ways of persuasion [4] When data is shared on the internet, it is acceptable to be rapid and casual. Most of the time, people do not intend to verify the information. If there is a suggestion of humans using the liaison features of the social media network in a nearly careless and automated manner [5]. In such a condition, a peripheral way to attract is most important [6]. Humans should generally consider heuristics cues to like or share any information [7].

This study focuses on solving the issue of fake jobs posted online by utilizing Machine Learning (ML), and Natural Language Processing (NLP) approaches. To conduct this study, we utilized a dataset contains more than 17,000 fake and real job descriptions. The dataset was available online on the official Kaggle website. Then, we used two different feature extraction approaches Bag-of-words (BoW) and Frequency–Inverse Document Frequency (TF-IDF). BoW gives simple features based on term frequency while TF-IDF gives weighted features also based on term frequency. After that, we used adaptive synthetic (ADASYN) in our proposed approach to resolving the imbalanced dataset problem. Finally, we apply six supervised machine learning classifiers, namely MP (Multilayer Perceptron), RF (Random Forest), KNN (K-Nearest Neighbor), SVM (Support Vector Machine), ET (Extra Tree), and LR (Logistic Regression), for observing fake job posts. These classifiers help detect fake job posts from a significant number of job ads. The performance evaluation matrices, for instance, Recall, Accuracy, F measure, and Precision, were used to assess all classifiers' predictions. The main points of our work include:

– Data performed in this study is highly imbalanced with the ratio 1:17 between target classes. We used the ADASYN technique to solve the imbalance dataset problem that was existed in the dataset and not covered in prior studies.

- Applying pre-processing technique and supervised ML classification algorithms to solve the problem in the two categories balance dataset and imbalanced dataset.
- Feature extraction approaches, i.e., TF-IDF and BoW, are used to analyze and compare the performance of classifiers.
- An in-depth comparison of our proposed approach with state-of-the-art deep learning models such as long short term memory (LSTM), Convolutional neural networks (CNN), and Gated Recurrent Unit (GRU).

The rest of this paper is organized as follows: Sect. 2 discusses the related works. The study methodology describes in Sect. 3, such as experiment overall, data collection, data preprocessing, feature engineering, machine learning, oversampling, and evaluation parameters. Following by Sect. 4, where we present and discuss the study results. Finally, Sect. 5 concludes the paper and highlights the direction of future work.

2 Related Work

In this section, we present an overview of prior studies that are related to our work. Table 1 shows a summary of the systematic analysis studies in related work. According to the various studies, fake reviews detection, fake content and misleading information detection, fake reviews detection, email spam detection, review spam detection, and fake news detection have drawn critical attention in the field of online fraud detection. In-text classification field, several researchers provide diverse types of approaches and techniques.

Zhang et al. [8] described the text classification process by using supervised machine learning classifiers. This study illustrated the main theoretical problems that occur in a text classification problem. The authors emphasized that it is vital to perform a clean data process by using different techniques such as stop word removing, stemming, and eliminating extra spaces before applying data on specific models. Feature selection approaches, for instance, TF-IDF, are more critical, and the aim is to reduce the dimensionality by eliminating features that are treated as irrelevant for the classification process.

In this study, [9] illustrated a primary pre-processing technique in the process of text classification where selecting features can make a text model more accurate, scalable, and efficient. Usually, the best technique for selecting features considers both algorithm and field properties. The Naïve Bayes model is simple and more sensitive to the approach of feature selection. Trail and analysis of text classification with the Naïve Bayes model provide the best performance using feature selection techniques. Their results showed two types of features evaluation metric CDM and MOR for the Naïve Bayes (NB) classifier, which applied on multi-class text collections and compared MOR and CDM with EOR, MC-OR, WOR, and three variations for multi-class datasets.

In [10], the authors described the bag-of-words technique, which is the most famous representation approach of the classification process. The central concept is to quantize the obtained certain point into optical words and show each image by the histogram of optical words. The author highlighted that problem which is demonstrated how to use BoW for image classification. They used five different popular datasets in their study. Three of the datasets are small-scale datasets, including 15 Scenes, while the two reaming datasets are large. In this way, a clustering model, namely K-means, is mainly used for producing optical words. Even if several classes have displayed the best results using a bag-of-words depiction for object arrangement, the general studies on equity of the bag-of-words approach are spotless due to the adversity offered by using heuristic clustering. They showed a statistical structure

that concluded the bag of words depiction. In this method, the creation of optical words was made by a statistical act that used a clustering model, while the experimental achievement is aggressive to the clustering-based approach. Based on the structure, we build two models that do not depend on the clustering base method while performing aggressive achievement in the object arrangement when related to a cluster-based bag of words approach representations.

Vidros et al. [11] conducted research in the area of online recruitment frauds. Employment scan is the most typical case in the field of online recruitment frauds (ORF). It is challenging to differentiate between legitimate and fraudulent job ads. They used a dataset, "Employment Scam Aegean Dataset" (EMSCAD), containing real-life job posts. To overcome the problem of ORF, the authors used bag-of-words modeling for features and machine learning classifiers such as random forest (RF), naïve base (NB), logistic regression (LR), and decision tree (DT) classifiers. Their results show that using an RF classifier can achieve 90% accuracy.

Ahmed et al. [12] proposed a model for detecting fraudulent information. Primarily authors utilized datasets collected from news articles. They used n-gram analysis techniques (uni-gram, bi-gram, tri-gram, and four-gram) with different machine learning classifiers, namely LR, SGD, DT, SVM, LSVM, KNN. They investigated different types of feature extraction approaches. TF-DIF approach gives better performance with the linear support vector machine (LSVM) classifier with an accuracy of 92%. In addition, Ahmed et al. [13] focused on opinion spam. The authors highlighted that opinions spam and fake news are a phenomenon that both contain misleading information. Writing and spreading fake news or beliefs on social media channels is easy, but it has affected both things and stores alike. Opinion spam and fake news have become a challenging problems nowadays. In the experiment, they used a public dataset that contains 800 numbers of truthful reviews and 800 are fake reviews written in the English language. They propose a methodology that uses the term frequency-inverted frequency (TF-IDF) to detect features and apply different machine learning models such as SVM, KNN, SGD, LSVM, and DT. The authors used TF-IDF as uni-gram, bi-gram, tri-gram, and four-gram LSVM with bi-gram features, where their result showed 90% accuracy.

Another study by Shawni Dutta and Samir Bandyopadhyay [14] discussed fake job advertisements. A fake job advertisement is typical to appear on web pages when the users surf the internet. They used the dataset "fake job postings" that contain the variables namely title, location, department, company_profile, description, requirement, benefits, employment_type, required_education, industry, and function. To avoid fake advertisements, machine learning models are used to detect fraudulent job posts, and the results of those models are comparable to check which models provided better performance. The authors used various types of classifiers, namely naïve base classifier (NB), multilayer perceptron (MLP), k-nearest neighbor (KNN), and random forest classifier (RF). Their results showed that the Random forest classifier provided the optimal accuracy from the other classifiers, achieving 98% accuracy.

A recent study by Shibly et al. [15] Boosted decision tree and random frost was utilized in Microsoft Azure Machine Learning Studio to propose a model. Their findings revealed that the boosted decision tree outperformed random frost, attaining (0.954) accuracy. In another study, Anita et al. [16] identified fake job postings using machine learning (KNN, RF, LR) and deep learning (Bi-LSTM) algorithmic approaches. The authors compared the classification algorithms; their findings showed that Bi-LSTM delivered the highest accurate result in detecting bogus jobs.

The summary of related work is present in Table 1. The primary motivation for us through this literature is the model good-fit on the balanced dataset. Many researchers have done work on fake jobs positive and did not consider the data balancing, causing models over-fitting on majority class data. We have done work on data balancing and deployment of state-of-the-art machine learning models to achieve high accuracy and a good F1 score.

Table 1 Summary of the systematic analysis studies in related work

Study	Year	Purpose	Approach	Source	Dataset	Techniques
Zhang et al. [8]	2008	Recognize the multi-words' efficacy	ML	Daviddlewis	19,403 Valid texts	Information Gain, SVC
Chen et al. [9]	2009	Detect online recruitment forgery	ML	Reuters 21578, Chinese text classification	2816 Job ads	CDM, MOR, NB
Wang and Huang [10]	2015	BoW used to heighten the efficiency of the image classifier	ML	Caltech-256 object category Calif. Inst. Technol.54	15 Scenes Caltech 256	SVM
Vidros et al. [11]	2017	Detect online recruitment forgery	ML	Employment Scam Aegean	17,880 Job ads	BoW, ZeorR, LR, DT, RFC, J48, OneR single rule, NB
Ahmed et al. [12]	2017	Detect online fake news	ML	Kaggle	12,600 Truthful news	TF-IDF, SVM, LSVM, KNN, DT, SGD LR
Vidros et al. [13]	2018	Detect opinion spam and fake news	ML	Kaggle Truthful reviews	1600	TF-IDF
Dutta and Bandyopadhyay [14]	2020	Detect online fake job advertisements	ML	Kaggle	17,880 job posts	RFC, Adaboost, GBC, KNN, MLP, DT
Shibly et al. [15]	2021	Detect online fake job advertisements	ML	Kaggle	17,880 job posts	boosted decision tree, RF
Anita et al. [16]	2021	Detect online fake job advertisements	ML	Kaggle	17,880 job posts	KNN, RF, Bi-LSTM
This work	2021	Evaluating the performance of ML on fake job ads data	ML	Kaggle	14,640 Records	LR, RF, SVC, DT, ETC, GBC, BoW, TF-ID, ADASYN, SMOTE, LSTM, CNN, GRU

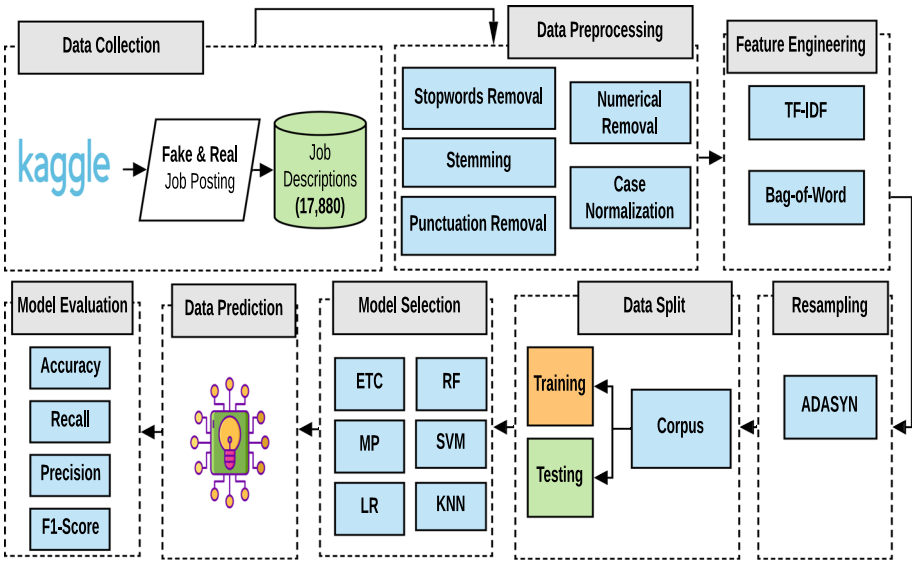

Fig. 1 Overview approach of data preparation

3 Study Methodology

In this section, we describe the study methodology that we used to perform the experiment. We discuss the data collection process, the selection of feature engineering, the six supervised machine learning classifiers, the oversampling techniques, and the evaluation parameters.

3.1 Study Overview

Figure 1 presents an overall approach of our study as a staged approach for solving the classification problem. At the first stage, the data were collected from the Kaggle website. Then, all the collected data were preprocessing. In the prepossessing stage, text data are prepared to be cleaned for further analysis, where we performed tokenization to job ads, and then digital values are detached from ads. This is because numerical values are not valuable for the study and do not affect our prediction result. Also, the preprocessing stage handled the case normalization, punctuation removal in the job ads. Afterward, we used the PorterStemmer [17,18] to perform the stemming where we bring the root of the word, for instance, the word *"doing"*, *"did"*, and *"done"*, because to be *"do"*. We explain in detail the processing of the preprocessing in Sect. 3.3.

After the data because ready for the experiment, feature engineering approaches were implemented on both the training and the testing process to choose important features from the text. We used Bag of word and TF-IDF approaches for classifying textual data. The word frequency serves as training data for the model [19]. After we used the feature engineering technique, we split the corpus into two subgroups for testing and training. We distribute the corpus into the ratio of 80% and 20%. The 80 percent of data is utilized for training, while the rest of the data is used for testing purposes.

Table 2 Attributes details of used dataset

Features	Description
Jod_id	The unique job id
Job_title	Describes the job advertisements
Job_location	The geographical location of job posts
Company_department	The corporate department like sales
Salary_package	The salary package starts from 50,000 dollars
Company profile	Short representation of the company
Job_description	A brief explanation of the post is applying
Job_requirement(s)	List of requirements for the post
Job_benefits	Describes the benefits proposed
Work from Home	True for working-from-home posts
Company logo	Emblem of the company
FAQs	The queries about employees
Employment form	Permanent or contract-based
Experience requirement	The Company required experience
Required education	The employment education
Industry	Company name or department type
Function	Employ job states

Table 3 Sample of data from the dataset

Title	Location	Department	Fraudulent
Marketing intern	US, NY New York	Marketing	0
Customer service—cloud video production	NZ, Auckland	Success	0
Commissioning machinery assistant (CMA)	US, IA Wever	Sales	0
IC&E technician	US, Stockton CA	Oil and energy	1

3.2 Data Collection

In this study, we obtained a benchmark dataset[1] "fake job postings" from a well-known website kaggle.[2] Our used dataset is highly imbalanced as the dataset contains 17,880 advertisement records where 17,014 are legitimate, and 866 are fraudulent job posts.

In this study, we used eleven of the seventeen independent variables for experiments: the company's profile, location, job description, title, department, benefits, requirements, type of employment, qualification, and function. The content columns of the dataset are shown in Table 2. Fraudulent is our dependent variable. For learning features, we select all other text variables such as job_title, job_location, etc., present in Table 3. We merge them into one column known as "text", which is shown in the following Table 4. After merging the selected independent variables in one column, "text", we perform some preprocessing techniques on the text data and apply the feature engineering methods.

[1] https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction.

[2] https://www.kaggle.com/.

Table 4 Sample of merged data

Text	Fraudulent
Marketing intern US, NY New York marketing	0
Customer service—cloud video production	0
NZ, Auckland success	
Commissioning Machinery Assistant (CMA)	0
The US, IA Wever sales	
IC&E Technician US, Stockton CA oil and energy	1

3.3 Data Preprocessing

This study used text data for the training of machine learning models. This text dataset contains a large amount of ambiguous content, which is not helpful for machine learning models. Sometimes, this ambiguity can create complexity in the training of the model, which can reduce the accuracy. Preprocessing of text consistent on following steps.

3.3.1 Stop-Words Removal

Stopwords are the part of the text which are helpful for a human to understand the sentence in terms of proper grammar, but they didn't add much information in the sentence such as he, it, do, us, the, etc. So removal of these stopwords can reduce the complexity in the learning feature set. We used these stopwords using the NLTK library.

3.3.2 Punctuation Removal

Punctuation is a part of the sentence and helps the reader to understand the message that is being conveyed clearly, but in a machine learning task, it has no meaning as it creates complexity in the learning procedure of machine learning models. We remove the punctuation marks such as !.,¿¡/([-=+&%$#)] using regular expressions.

3.3.3 Numerical Removal

Numbers in the text are also not valuable for text classification because the number has no specific meaning and not consider as strong features in text classification [20]. We remove these numbers, which help to reduce the size of the features set and help in a good fit of models.

3.3.4 Stemming

Stemming is an important part of preprocessing because sometimes words in the text are in different forms but have the same meaning. For example, the words 'go', 'going', 'goes', and 'gone' have a similar meaning, but in machine learning, they are considered as different words, resulting in complexity. To solve this problem, we used the stemming technique to convert each word into its root form. For example, stemming will convert 'go', 'going', 'goes', and 'gone' into their root for 'go' by removing the 'es', 'ing', and 'ne' from the end. We used the porter stemmer library for stemming of text [21].

Table 5 Sample of data after preprocessing

Before preprocessing	After preprocessing
Marketing intern, New York marketing	Market intern new york market
Customer service—cloud video	Customer service cloud video
Production, Auckland success	Production Auckland success
Commissioning machinery	Commissioning machinery
Assistant (CMA), wever sales	Assistant wever sale
Commissioning machinery	Commissioning machinery
Assistant (CMA), wever sales	Assistant wever sale
Technician, Stockton oil and energy	Technician Stockton oil energy

Table 6 Results of BoW approach on pre-processed sample dataset

Doc.	Market	Intern	New	York	Technician	Stockton	Oil	Energy
1	2	1	1	1	0	0	0	0
2	0	0	0	0	1	1	1	1

3.3.5 Case Normalization

We convert each word into the lower case because the different words have the same meaning,. However, machine learning models are different from learning, such as 'Go,' and ''go' are the same in meaning but are different for machine learning models. Therefore, we convert each word into the lower case so that this problem can be solved. We used tolower() python function to perform this task. Table 5 showing the sample data after preprocessing.

3.4 Feature Extraction

The feature engineering technique focuses on achieving purposeful features from the dataset to be utilized by classification algorithms of ML. In other words, features are formed and extracted from standard ones [22]. Previous study [23] concluded that extraction of features could enhance the achievement of ML models. Feature engineering is a way for feature extraction from unprocessed data; the ML models become more consistent and accurate. We used two feature engineering techniques: Bag-of-Word (BoW) and Frequency–Inverse Document Frequency (TF-IDF).

3.4.1 Bag-of-Words

The Bag-of-word technique focuses on the feature extraction from data that the text contains. We selected this approach because it is more suitable for study, for instance, text classification and modeling of language. We have utilized Count-Vectorizer for feature extraction that operates on the term frequency that calculates the frequency of token instances and generates the rare token matrix [24]. A bag of words consists of word sets and features along with a value that shows the instance of an individual feature [25]. The results of BoW technique on sample data shown in Table 6.

Table 7 Results of TF-IDF approach on pre-processed sample dataset

Doc.	Market	Intern	New	York	Technician	Stockton	Oil	Energy
1	0.75	0.37	0.37	0.37	0	0	0	0
2	0	0	0	0	0.5	0.5	0.5	0.5

3.4.2 TF-IDF

Term frequency-inverted document frequency (TF-IDF) is another approach for extracting features. This technique is commonly utilized in the evaluation of feature selection methods for text classification, and text analysis [26]. It allocates weight to the terms existing in a document following the inverse frequency of document and frequency of terms [20,27]. Greater weighted score terms are treated as more important [22]. TF-IDF can be described in the form of an equation:

For information retrieval, one of the approaches used is Term Frequency-Inverse Document Frequency [23]. Through this technique, relevant words are extracted to summarize a document. As it is evident from its name, it checks how frequently a word is encountered in a document [20]. This approach utilizes a frequency of term, and inverse frequency of document to extract features [22]. The words that are not frequent are represented by IDF. If a word is uncommon in one document, however, it also appears in other documents, then it has significance for both documents. Set of documents represented here by U. Inverse document frequency is calculated as follows:

$$tfidf_{t,d,D} = tf_{t,d} \cdot idf_{t,D} \tag{1}$$

where the $tf_(t, d)$ is frequency of term t in document d, N is number of documents, and D_i, t is the number of document contain the term t.

Here $\boldsymbol{t}$ represents all words (terms); each document is represented by $\boldsymbol{d}$; while set of all documents is represented by $\boldsymbol{D}$. A sequence of words (n-words) is represented by n-gram. The weighted word matrix represents the output of TF-IDF. The words frequency of corpus balances out the meaning of terms, which increases with the count of words, with TF-IDF vectorizer.

TF-IDF approach is also employed for classifying text by extracting features. The word frequency is utilized for training the classification model. The term frequency (TF) is not considerate about whether a word is frequent or not compared to TF-IDF. As TF-IDF adds less weight to frequent terms. The results of TF-IDF technique on sample data shown in Table 7.

3.5 Supervised Machine Learning Models

This section discusses the supervised machine learning models utilized in this study, and how these models are implemented. In this study, we used NLTK [28] and Scikit-learn library[3] to implement supervised ML models. Supervised ML models are generally utilized for solving problems of regression and classification [29]. In this study, six supervised machine learning models have been used for the future forecasting process of fraudulent job ads. The algorithms utilized are Random Forest (RF), Linear Regression (LR), Support Vector Machine (SVM),

[3] https://sci-kit-learn.org/stable/.

Table 8 Hyper-parameters of machine learning algorithms used in our study

ML algorithms	Hyperparameters
Random forest (RF)	n_estimators = 300
	Random_state = 5
	Max_depth = 300
	Random_state = 52
Logistic regression (LR)	Multi_class = 'multinomial'
	C = 3.0
	Solver = 'liblinear'
Support vector machine (SVM)	C = 2.0
	Kernel ='linear'
Extra trees classifier (ETC)	No. of estimators = 200
	Max tree_depth = 8
	Random states = 2
K-nearest neighbor (KNN)	n_neighbors = 6
	Weight = 'uniform'
Multilayer perceptron (MP)	Optimize adam
	Binary cross-entropy
	Loss dropout = 0.2

Extra Tree Classifier (ETC), K-nearest neighbor (KNN), and Multilayer perceptron (MLP). Table 8 present the hyper-parameters used in each algorithm.

3.5.1 Random Forest (RF)

Random forest technique utilizes an ensemble model that is based on trees. In this approach, several sub-trees (decision trees) help in providing thorough prediction [30]. While training, subtrees are generated that are independent. These trees are trained by using the "Bagging" approach. The training data set is provided by bootstrap samples [30]. Moreover, the combined results of these provide final predictions. This value is used with different outcomes. These outcomes are employed in ensemble classification. In the period of training, the dataset is subdivided into samples by substitution. This dataset is referred to as a bootstrap sample. Here the sizes of the training dataset and actual sample are equal [31]. During classifying, similar to other classification models, RF also employs the decision tree technique for choosing the root for the tree to construct the whole forest.

With the help of this approach, tree attributes are collected [32]. The results are merged through the process of voting. Previous research suggests a variety of assembly mechanisms [33,34]. A number of ensemble classifiers, for instance, bagging [35] and boosting [36], are generally used. The approach of bagging minimizes any variations in the classification model. Here Random Forest id represented as follows:

$$rf = mode\{T_1(z), T_2(z), T_3(z), \ldots, T_n(z)\} \tag{2}$$

$$rf = mode\left\{\sum_{i=1}^{n} T_i(z)\right\} \tag{3}$$

The number of trees is represented by n; prediction value of all trees, compiled through voting, is indicated by rf. $T_1(z)$, $T_2(z)$, $T_3(z)$, ..., $T_n(z)$ represent the trees in Random Forest.

3.5.2 Logistic Regression (LR)

Logistic Regression is a statistical approach, quite often employed by machine learning to perform regression analysis. With the help of this logical model, the relationship between covariates or predictors (independent variables) and outcome variables (dependent variables) can be estimated [37]. To deal with problems of differential classification, where two class values cause negative issues, this model is preferable [38]. Logistic Regression can utilize the data for predicting the category of input variables based on independent variables (i.e., predictors). This model is a form of supervised classification, where the probability of outcome variable either increases or decreases [39]. The response variable (outcome variable) is of dual nature. It implies that this model can result in more than two possible distinct classes or outcomes [38]. The following equation depicts the logistic function:

$$g(x) = \frac{L}{1 + e^{-m(z-z_o)}} \tag{4}$$

where

- Euler Number is represented by e.
- The x-value of the sigmoid midpoint is represented by z_o.
- Maximum value of curve is represented by L.
- The curve's steepness is represented by m.

The range of z_o is $-\infty$ to $+\infty$ of real numbers. When z approaches positive infinity, the g graph approaches L. And when x approaches negative infinity ($-\infty$), it approaches to 0. This is how the logistic function's S-curve is formed.

3.5.3 Support Vector Machine (SVM)

SVM is another ML model that is utilized in classifiers and regression algorithms [24,40]. Regression of SVM starting from the non-parametric method, which builds upon mathematical notation. This is where kernel transformation allows inputting desired data Support vector machine determines the problem of regression with the help of linear functions. On the other hand, for the non-linear regression problems, it inputs x, which is a vector to z (i.e., n-dimensional feature space). The non-linear approach completes the mapping procedure, and then the space tests the linear regression. To insert the theory in a machine learning context by a variety of training data (xn) with "N" number of considerations and "yn" as an observed response. The following equation denotes the linear function:

$$f(x) = x\beta^{'} + b \tag{5}$$

The goal is to create the possible function value with the β for f(x), and with 0β in terms of basic model values. Thus, the issue can be adjusted in the formula as follows:

$$f(x) = \frac{1}{2}\beta^{'}\beta \tag{6}$$

With all certain cases of values of the residuals are not greater than ϵ as shown in the equation:

$$\forall n : |y_n - (x_n^{'}\beta + b)| \leq \epsilon \tag{7}$$

3.5.4 Extra Trees Classifier(ETC)

ETC is named randomized trees, which uses many decision trees to fit them on the different sub-samples from a dataset and before applying a balanced method to increase the accuracy and minimize the over-fitting problem. The tree's size is composed of its delinquency parameters, but these can compose by arranging parameter values. The extra tree model is working in the same way as the random forest model [41], by randomizing the particular decisions and is reducing the learning process from the data to manage the over-fitting problem.

3.5.5 K-Nearest Neighbor (KNN)

KNN is quite a simple model utilized in ML for analysis of regression and classification. The technique uses the data and arranges the new data indicates based on a distance function, and the data is referred to as the class with the nearest neighbors. In this experiment, the k-nearest neighbor model bestows pledge results where the value of k is equal to five (k=5). It means that it checks the five nearest neighbors and selects the base of the majority or nearest distance.

3.5.6 Multilayer Perceptron (MLP)

MPL approach lies under the artificial neural networks (ANNs), and it uses neural network architecture to solve the classification problem. The MLP is a feed-forward class of ANNs [42]. MLP consists of a basic of three layers of nodes which are (1) input layer, (2) hidden layer, and (3) output layer. The flow of the data is one-way directed, either forward or backward. This occurs from the input layer to the output layer. The architecture of MLP is shown in Fig. 2.

Neurons use the non-linear activation function to lean from the data except the input layer nodes, multiple layers, and non-linear activation function make different MLP from linear perceptron It can distinguish data that is not linearly separable.

3.6 Oversampling

Imbalanced datasets have a weak influence on the performance of the classification process of conventional classification models [43]. Dataset is called imbalanced where the size of samples of one class is larger or less than the other classes [44]. To handle the problem of class imbalance, various techniques are used for under and oversampling. Oversampling is a technique in data analysis that is used to accommodate the dataset of class distribution. Oversampling is used in the methodology of survey design, statistical sampling, and the machine-learning process. SMOTE and ADASYN are two well-known techniques of oversampling that are used to adjust the data. In this study, we used the ADASYN technique to adjust the dataset. After oversampling, we obtain a balanced dataset which is shown in Table 9.

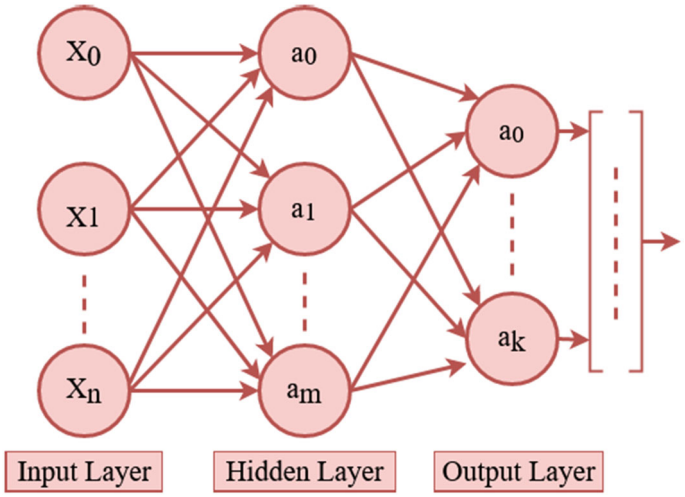

Fig. 2 MLP architecture

Table 9 Number of fraudulent, legitimate training and testing data

	Imbalanced	Balanced
Legitimate	17,014	17,014
Fraudulent	866	17,042
Training	14,304	27,244
Testing	3576	6812

After using both feature extraction techniques, machine learning classifiers are trained with meaningful features to obtain through feature-engineering approaches. The ML classification algorithms work along various hyper-parameters as shown in Table 8. When models are trained, the test data is utilized for checking the overall performance of classification or learning models.

3.6.1 ADASYN

Imbalanced learning occurs whenever some data distribution types significantly dominate the instance space compared to other data distributions. Some traditional methods to solve this problem are under-sampling and over-sampling. Under-sampling is where the majority class is downsampled to the same amount of data as the minority class. However, this is extensive data inefficient. The discarded data might have important information regarding the majority class. In over-sampling, the minority class is copied x times until its size is like the majority class. The most significant flaw here is the model will overfit the minority data because the same training examples appear many times. To avoid all the above problems, ADASYN can be used. ADASYN (Adaptive Synthetic) is an algorithm that generates synthetic data. The most significant advantages are not copying the same minority data and generating more data for "harder to learn" examples.

1. Calculate the ratio of minority to majority examples using: $d = \frac{m_s}{m_l}$. Where m_s and m_l are the number of minority and majority class examples, respectively. If d is lower than a certain threshold, initialize the algorithm.
2. Calculate the total number of synthetic minority data to generate using: $G = (m_l - m_s)\beta$. Here, G is the total number of minority data to generate. β is the ratio of minority to majority data desired after ADASYN. $\beta = 1$ means a perfectly balanced data set after ADASYN.
3. Find the k-Nearest Neighbors of each minority example and calculate the r_i value using $\frac{Numberofmajority}{k}$. After this step, each minority example should be associated with a different neighborhood. The r_i value indicates the dominance of the majority class in each specific neighborhood.
4. Normalize the r_i values so that the sum of all r_i values equals to 1.

$$\hat{r_i} = \frac{r_i}{\sum r_i}$$
$$\sum \hat{r_i} = 1$$

5. Calculate the number of synthetic examples to generate per neighborhood by using

$$G_i = G\hat{r_i}$$

6. Generate G_i data for each neighborhood using:

$$s_i = x_i + (x_{zi} - x_i)\lambda$$

3.7 Evaluation Parameters

To predict the performance of the models, we adopted different metrics for checking the performance, for instance, Accuracy, Precision, Recall, and F1 measure. These all parameters are used to classify the models of machine learning [45]. The confusion matrix is another metric for assessing the models' performance. Error matrix or confusion matrix is an approach, which sums up the performance of models [46]. A confusion matrix is mostly used to express the models' performance on several test datasets where the actual numbers are known.

TP or True Positives This term defines that the predicted or estimated values are correct; it implies that the prediction is correct as it is evident from both classes' values (i.e., yes). For instance, the actual class's value tells us that this job ad is real, and the same thing tells us predicted class.

TN or True Negatives This term defines that the predicted or estimated values are incorrect; it implies that the prediction is incorrect as it is evident from both classes' values (i.e., no). For instance, the actual class's value tells us that this job ad is fake, and the predicted class tells us the same thing. The false-negative and false posit values occur when our actual class belies with the predicted class in the process.

FP or False Positives Both classes' values differ from each other in this case, i.e., yes, in the actual class, while the predicted class holds a negative value. In a nutshell, the actual class's value displays that the job ad is fraudulent, whereas the predicted class tells otherwise.

FN or False Negatives Here is a difference between the values of the actual and predicted class, which contradicts the results. The actual class's value displays that the job ad is fraudulent, whereas the predicted class tells otherwise.

3.7.1 Accuracy

It is the better intuitive process to measure the models' performance by calculating the ratio of the correctly predicted number of observations to the overall number of observations. If we achieve high accuracy, so our classifier is excellent. Accuracy is a robust measuring parameter when we have balanced datasets and where the values of the dataset must be false negatives, and positive are approximately the same.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

3.7.2 Recall

It is measured by dividing the number of instances with the positive prediction by the overall number of instances having positive values in the actual class.

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

3.7.3 Precision

This metric involves the ratio of correctly predicted instances and has positive values and the total number of predicted positive instances of the dataset.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

3.7.4 F_1 Score

F1 score is also known as F measure. F1 score is an harmonic mean of both Recall and Precision metrics. F1 score is more convenient in comparison with accuracy metric on imbalanced dataset.

$$F_1 Score = 2 * \frac{Precsicion * Recall}{Precsicion + Recal} \tag{11}$$

In this study, we used all the above parameters for evaluating and analyzing the models' performance. We measured the models' accuracy, compared it with the other models, and introduced the best performer on fake job ads.

4 Results and Discussions

In this section, we describe our experimental results and discussion. As discussed in Sect. 3.1, our data is an imbalanced dataset. Therefore, we performed two experiments. The first experiment contains an imbalanced dataset, and the second experiment is performed on a balanced dataset. We used a bag-of-word and TF-IDF approaches for the feature extraction. Then, we applied the selected six machine learning classifiers, namely RF, ETC, SVM, LR, and KNN, to compare the performance of the classifiers. We analyze all the classifiers' results to understand which model is robust for providing better accuracy from others. If our models provide equal performance, we draw a confusion matrix and diagnose classification models' performance.

4.1 Experimental Results on Imbalanced Dataset

The used dataset is highly imbalanced which contain more record for the non-fraudulent class. This imbalanced dataset problem can cause model over-fitting. This section discusses the results of machine learning and deep learning models and shows the impact of imbalanced data on the model's performance.

4.1.1 Model Performance on the Imbalanced Dataset with BoW

This section contains the results of models with BoW and TF-IDF features on the imbalanced dataset. MLP outperforms all models in terms of correct prediction as MLP gives 3537 correct predictions out of 3576 predictions as shown in Table 11. LR gives just two more wrong predictions compared to the MLP model, which shows that MLP and LR perform well on the imbalanced dataset. There are many fluctuations in evaluation scores, such as MLP giving a 0.98 accuracy score and an 0.85 F1-score because the model gets over-fitted for majority class data and gives a more wrong prediction for minority class data. MLP gives 33 wrong predictions for minority class (fraudulent class) and provides only six wrong predictions for (non-fraudulent class) as shown in Table 11. These statistics show that the model is more promising on majority class data because it gets more train data for learning. The results of all models in terms of accuracy, precision, recall, and F1 score on the imbalanced dataset using BoW features are shown in Table 10.

4.1.2 Model Performance on the Imbalanced Dataset with TF-IDF

The performance of models with TF-IDF features on imbalanced dataset disuse in this section. Models with TF-IDF features perform better than BoW features because TF-IDF gives more weighted features, which can be suitable for learning models. MLP offers the highest correct predictions ratio on the imbalanced dataset with TF-IDF features 3540 out of 3576 predictions and gives only 36 wrong predictions as shown in Table 11. MLP gives three fewer wrong predictions with TF-IDF compared to with BoW features, which show the significance of TF-IDF compared with BoW. Overall, the model performance on the imbalanced dataset fluctuates in terms of all evaluation parameters scores. The accuracy score is high, and the F1-score is low, which shows the impact of model over-fitting on majority class data. The results of models on TF-IDF features without data balancing are shown in Table 10.

4.2 Models Performance with BoW and TF-IDF Features and After Balancing of the Dataset Using ADASYN Technique

This section presents results after balancing of data. The performance of machine learning models was very poor in terms of recall and F1-score on the imbalanced dataset. That is why we investigate the results of models after balancing the data. So we resolve the imbalanced dataset problem using the ADASYN technique and train machine learning models on the balanced dataset to show the significance of models for the fake job posting after balancing of data.

Table 10 Machine learning models results using BoW and TF-IDF features on original dataset

BoW					TF-IDF				
Models	Accuracy	Precision	Recall	F1-score	Models	Accuracy	Precision	Recall	F1-score
RF	0.98	0.99	0.61	0.75	RF	0.98	0.99	0.62	0.76
ETC	0.98	0.99	0.68	0.81	ETC	0.98	0.99	0.65	0.78
LR	0.98	0.94	0.79	0.86	LR	0.97	1.00	0.39	0.56
SVM	0.98	0.88	0.80	0.84	SVM	0.98	0.97	0.73	0.83
MLP	0.98	0.94	0.78	0.85	MLP	0.98	0.97	0.79	0.85
KNN	0.90	0.31	0.84	0.45	KNN	0.90	0.86	0.78	0.82

Table 11 Confusion matrix values of machine learning models using BoW and TF-IDF features on original dataset

BoW							TF-IDF						
Models	TP	TN	FP	FN	CP	WP	Models	TP	TN	FP	FN	CP	WP
RF	3411	101	1	63	3512	64	RF	3411	102	1	62	3513	63
ETC	3411	113	2	51	3524	52	ETC	3411	107	1	57	3518	58
LR	3404	131	8	33	3535	41	LR	3412	64	0	100	3476	100
SVM	3395	132	17	32	3527	49	SVM	3409	102	3	44	3529	47
MLP	3406	131	6	33	3537	39	MLP	3409	131	3	33	3540	36
KNN	3113	138	299	26	3251	325	KNN	3392	128	20	36	3520	56

4.2.1 Models Performance After Balancing of the Dataset Using ADASYN Technique and BoW Features

The results using BoW features are discus in this section after balancing data using the ADASYN technique. According to results in Table 12, MLP outperforms all other models in terms of all evaluation parameters with 0.99, 0.99, 0.99, and 0.99 accuracy, precision, recall, and F1 score, respectively. RF also achieved 0.99 accuracy but low in terms of precision with a 0.98 score. ETC is also with 0.99 accuracy but low in terms of recall with 0.98 scores. MLP gives the highest correct prediction ratio compared to all other models with BoW features after balancing data. MLP gives 6751 correct predictions out of 6812 as shown in Table 13. ETC and RF are just behind in the race of accuracy with MLP and give 6744 and 6735 correct predictions, respectively, out of 6812 as shown in Table 13.

4.2.2 Models Performance After Balancing of Dataset Using ADASYN Technique and TF-IDF Features

The results with TF-IDF after balancing of data using ADASYN technique are shown in Table 12. The performance of models with TF-IDF features is more significant than BoW features after balancing data. ETC outperforms with TF-IDF features with a 0.999 accuracy score, and MLP also outperforms all other models in terms of sensitivity. All other models also improve their performance with TF-IDF features. The confusion matrix with TF-IDF feature on the balanced dataset with ADASYN technique is shown in Table 13. ETC gives the highest correct prediction ratio of the study, which is 6809 out of 6812, and MLP is just behind the ETC with 6799 correct predictions. Figure 3 showing the comparison between models performance on original data and after the ADASYN technique.

4.3 Performance Comparison with State-of-the-Art Deep Learning Models

The performance of deep learning models also evaluates on the fake job dataset. We deployed state-of-the-art deep learning models such as long-short terms memory (LSTM), convolutional neural networks (CNN) [47], gated recurrent unit (GRU) [48] on imbalanced and with the best ADASYN technique. The architectures of used models are shown in Fig. 4. Each model has an embedding layer as the input layer takes input data and passes them to the following layers. Each model end with an output layer consists of two neurons and a softmax

Table 12 Machine learning models results using BoW and TF-IDF features with ADASYN technique

BoW					TF-IDF				
Models	Accuracy	Precision	Recall	F1-score	Models	Accuracy	Precision	Recall	F1-score
RF	0.99	0.99	0.99	0.99	RF	0.99	0.99	0.99	0.99
ETC	0.99	0.99	0.99	0.99	ETC	0.999	0.999	0.999	0.999
LR	0.99	0.99	0.98	0.99	LR	0.99	0.99	0.98	0.99
SVM	0.99	0.99	0.98	0.99	SVM	0.99	0.99	0.98	0.99
MLP	0.99	0.99	1.00	0.99	MLP	0.99	0.99	1.00	0.99
KNN	0.85	0.77	1.00	0.87	KNN	0.85	0.77	1.00	0.87

Table 13 Confusion matrix values of machine learning models using BoW and TF-IDF features with ADASYN technique

BoW							TF-IDF						
Models	TP	TN	FP	FN	CP	WP	Models	TP	TN	FP	FN	CP	WP
RF	3382	3353	44	33	6735	77	RF	3403	3392	2	15	6795	17
ETC	3396	3348	30	38	6744	68	ETC	3404	3405	1	2	6809	3
LR	3365	3361	61	25	6726	86	LR	3346	3403	59	4	6749	63
SVM	3368	3361	58	25	6729	83	SVM	3346	3403	59	4	6749	63
MLP	3388	3363	38	23	6751	61	MLP	3392	3407	13	0	6799	13
KNN	2337	3383	1089	3	5720	1092	KNN	2432	3407	973	0	5839	973

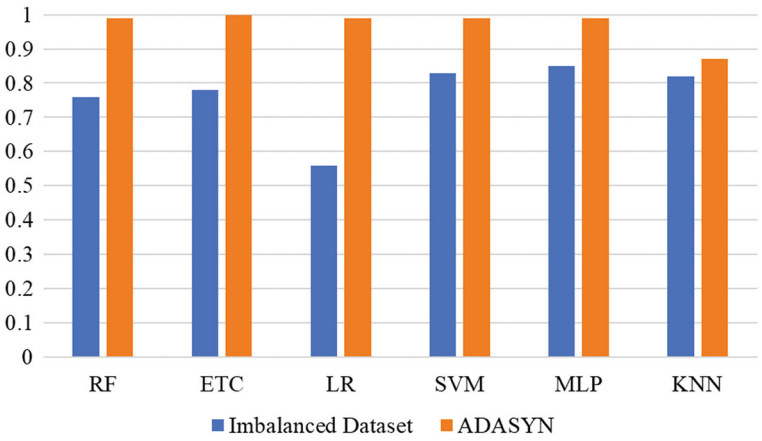

Fig. 3 Comparison of accuracy, precision, recall, and F1-score of all specified ML models with imbalanced dataset and ADASYN technique

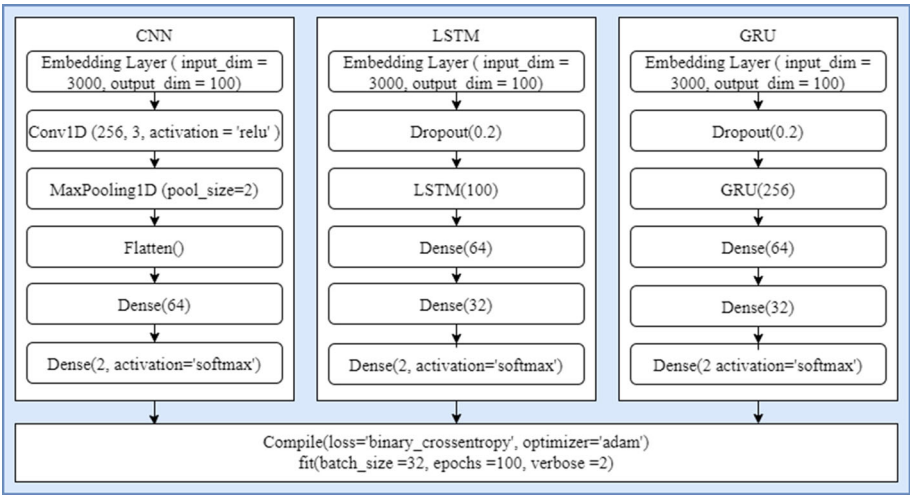

Fig. 4 The architectures of used deep learning models

activation function. We compile all models with binary cross-entropy loss function and adam optimizer [49]. Models fitted by 32 batch sizes and 100 epochs.

The performance of deep learning models is not significant compared to machine learning models on both imbalanced balanced data because of the small size of the dataset. Deep learning models required a large feature set for good-fit as compared to machine learning models. LSTM and CNN achieved the 0.86 F1-score on the imbalanced dataset, respectively, while with ADASYN, LSTM achieved a 0.98 F1-score as shown in Table 14. The confusion matrices value for all deep learning models is given in Table 15 showing the correct and wrong prediction ratio.

Table 14 Deep learning models results on balanced and imbalanced dataset

Imbalanced					ADASYN				
Models	Accuracy	Precision	Recall	F1-score	Models	Accuracy	Precision	Recall	F1-score
LSTM	0.98	0.91	0.82	0.86	LSTM	0.97	0.97	0.97	0.97
CNN	0.98	0.95	0.80	0.86	CNN	0.98	0.98	0.98	0.98
GRU	0.96	0.89	0.73	0.79	GRU	0.96	0.97	0.97	0.97

Table 15 Confusion matrix values of deep learning models on balanced and imbalanced dataset

Imbalanced							ADASYN						
Models	TP	TN	FP	FN	CP	WP	Models	TP	TN	FP	FN	CP	WP
LSTM	3381	111	21	63	3492	84	LSTM	3338	3274	117	77	6612	194
CNN	3393	106	9	68	3499	77	CNN	3352	3291	103	60	6643	163
GRU	3353	94	22	107	3447	129	GRU	3305	3229	150	122	6534	272

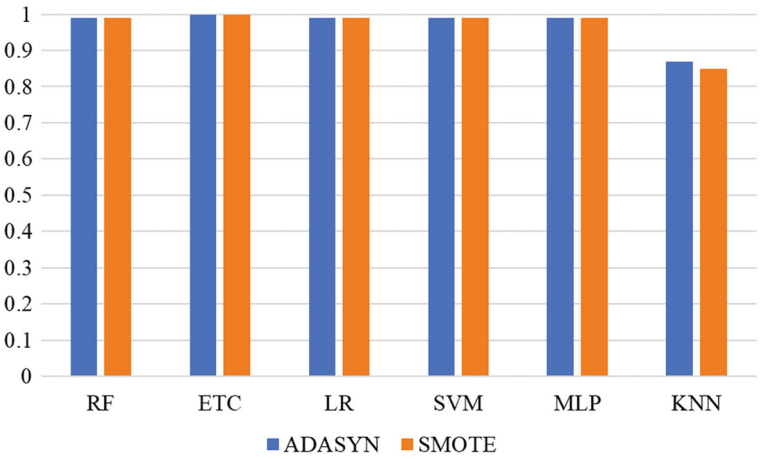

Fig. 5 Comparison of accuracy, precision, recall, and F1-score of all specified ML models with SMOTE and ADASYN technique

4.4 Models Performance with BoW and TF-IDF Features and After Balancing of the Dataset Using SMOTE Technique

The performance of machine learning models is also evaluated after balancing data using Synthetic Minority Oversampling TEchnique (SMOTE) in comparison with ADASYN [50]. SMOTE is also an oversampling technique used to generate the sample for minority class data. SMOTE used the k-nearest neighbor algorithm to create synthetic data. The results of models with BoW and TF-IDF feature using SMOTE technique shown in Table 16. The performance of all models is significant with as well as ADASYN. With BoW features, the accuracy of all models is 0.99, except the KNN with 0.84 in accuracy score. MLP gives the highest number of correct predictions, 6744 out of 6806 with BoW and SMOTE techniques as shown in Table 17. ETC is just behind the MLP with 6741 correct predictions.

The results with TF-IDF features are more significant as compared to BoW using the SMOTE technique. ETC with TF-IDF and SMOTE technique outperform all other models with the highest accuracy 0.998 accuracy score. ETC gives 6798 correct predictions out of 6806 predictions. MLP is just behind the ETC with 6796 correct predictions as shown in Table 17. Figure 5 showing the comparison between models performance after applying ADASYN and SMOTE techniques.

Table 16 Machine learning models results using BoW and TF-IDF features with SMOTE technique

BoW					TF-IDF				
Models	Accuracy	Precision	Recall	F1-score	Models	Accuracy	Precision	Recall	F1-score
RF	0.99	0.99	0.99	0.99	RF	0.99	1.00	0.99	0.99
ETC	0.99	0.99	0.99	0.99	ETC	0.998	0.998	0.999	0.998
LR	0.99	0.99	0.99	0.99	LR	0.99	1.00	0.99	0.99
SVM	0.99	0.99	0.99	0.99	SVM	0.99	0.99	0.99	0.99
MLP	0.99	0.99	0.99	0.99	MLP	0.99	0.99	0.99	0.99
KNN	0.84	0.84	0.84	0.84	KNN	0.86	0.89	0.86	0.85

Table 17 Confusion matrix values of machine learning models using BoW and TF-IDF features with SMOTE technique

BoW							TF-IDF						
Models	TP	TN	FP	FN	CP	WP	Models	TP	TN	FP	FN	CP	WP
RF	3397	3335	47	27	6732	74	RF	3442	3351	2	11	6793	73
ETC	3411	3330	33	32	6741	65	ETC	3439	3359	5	3	6798	8
LR	3387	3340	57	22	6727	79	LR	3333	3362	11	0	6795	11
SVM	3384	3342	60	20	6726	80	SVM	3416	3359	28	3	6775	31
MLP	3402	3342	42	20	6744	62	MLP	3434	3362	10	0	6796	10
KNN	2369	3362	1075	0	5731	1075	KNN	2473	3362	971	0	5835	971

Table 18 Comparison with other studies on the same dataset

Ref	Year	Model	Accuracy (%)
Dutta et al. [14]	2020	RF	98.27
Ranparia et al. [51]	2020	SNN	97
Keerthana et al. [52]	2021	MLP	71
Shibly et al. [15]	2021	RF	95.4
This work	2021	ETC	99.9

4.5 Proposed Approach Comparison with Previous Studies

In this study, we compared our results with the previous studies, such as study [14] used a machine learning approach on the same dataset and achieved 98.27% accuracy using the RF classifier. Another study [51] used the sequential neural network (SNN) for the fake job posting detection on the same dataset and achieved the highest 97% accuracy score. In comparison with this study, we get 99.9% accuracy using the TF-IDF features, ADASYN oversampling technique, ETC model. The comparison results are shown in Table 18.

5 Conclusion

This study has detected fraudulent job ads using different machine learning classifiers and two feature extraction approaches, namely BoW modeling and TF-IDF features. BoW and TF-IDF both were used with ADASYN oversampling technique to improve the accuracy and efficiency of our models. The existing dataset for experiments was highly imbalanced and caused for model over-fitting on majority class data. We used the oversampling technique with the features extraction technique to solve the model over-fitting problem and achieve a high accuracy score. Our proposed approach has achieved the highest accuracy score, 99.9%, using TF-IDF features and the ETC classifier with the ADASYN technique. We conclude that the imbalanced dataset caused for over-fitting of models. That's the reason the model shows poor performance on minority class data. We also concluded that TF-IDF features are more meaningful as compare to BoW simple features. We also used stat-of-the-arts, deep learning models to compare with machine learning models such as CNN, LSTM, and GRU. The performance of deep learning models is not significant as compared to machine learning models, and we concluded through the analysis and literature that deep learning models required a large dataset for good-fit. In comparison with ADASYN, we also used the SMOTE oversampling technique, which generated significant results for machine learning models. ETC with SMOTE and TF-IDF feature achieved 99.8% accuracy, which is very close to with ADASYN. This comparison concludes that oversampling generates more record in the dataset which help to generate a large feature set for the learning of the models which highly impact on improving the performance of learning models. In future work, we collect more datasets related to fake job postings and will apply more advanced deep learning models. We will also consider all attributes of the dataset for experiments.

References

1. Becker R (2017) Your short attention span could help fake news spread. https://www.theverge.com/2017/6/26/15875488/fake-news-viral-hoaxes-bots-information-overload-twitter-facebook-social-media

2. Simmons G (2017) Market incentives that drive fraud: the truth behind reach vs. frequency. https://medium.com/@gsimmons/incentives-for-fraud-reach-vs-frequency-52d62d49ccbf
3. Vosoughi S, Roy D, Aral S (2018) The spread of true and false news online. Science 359(6380):1146–1151
4. Cyr D, Head M, Lim E, Stibe A (2018) Using the elaboration likelihood model to examine online persuasion through website design. Inf Manag 55(7):807–821
5. Hayes RA, Carr CT, Wohn DY (2016) One click, many meanings: interpreting paralinguistic digital affordances in social media. J Broadcast Electron Media 60(1):171–187
6. Williams EJ, Beardmore A, Joinson AN (2017) Individual differences in susceptibility to online influence: a theoretical review. Comput Hum Behav 72:412–421
7. Cook J, Lewandowsky S, Ecker UK (2017) Neutralizing misinformation through inoculation: exposing misleading argumentation techniques reduces their influence. PLoS One 12(5):e0175799
8. Zhang W, Yoshida T, Tang X (2008) Text classification based on multi-word with support vector machine. Knowl Based Syst 21(8):879–886
9. Chen J, Huang H, Tian S, Qu Y (2009) Feature selection for text classification with Naïve Bayes. Expert Syst Appl 36(3):5432–5435
10. Wang C, Huang K (2015) How to use bag-of-words model better for image classification. Image Vis Comput 38:65–74
11. Vidros S, Kolias C, Kambourakis G, Akoglu L (2017) Automatic detection of online recruitment frauds: characteristics, methods, and a public dataset. Future Internet 9(1):6
12. Ahmed H, Traore I, Saad S (2017) Detection of online fake news using n-gram analysis and machine learning techniques. In: International conference on intelligent, secure, and dependable systems in distributed and cloud environments, pp 127–138. Springer
13. Ahmed H, Traore I, Saad S (2018) Detecting opinion spams and fake news using text classification. Secur Priv 1(1):e9
14. Dutta S, Bandyopadhyay SK (2020) Fake job recruitment detection using machine learning approach. Int J Eng Trends Technol 68.4(2020):48–53
15. Shibly F, Sharma U, Naleer H (2021) Performance comparison of two class boosted decision tree snd two class decision forest algorithms in predicting fake job postings. Ann Rom Soc Cell Biol 25(4):2462–2472
16. Anita C, Nagarajan P, Sairam GA, Ganesh P, Deepakkumar G (2021) Fake job detection and analysis using machine learning and deep learning algorithms. Revista Geintec-Gestao Inovacao e Tecnologias 11(2):642–650
17. Aljedaani W, Javed Y, Alenezi M (2020) LDA categorization of security bug reports in chromium projects. In: Proceedings of the 2020 European symposium on software engineering, pp 154–161
18. Aljedaani W, Nagappan M, Adams B, Godfrey M (2019) A comparison of bugs across the ios and android platforms of two open source cross platform browser apps. In: 2019 IEEE/ACM 6th international conference on mobile software engineering and systems (MOBILESoft), pp 76–86. IEEE
19. Joulin A, Grave E, Bojanowski P, Mikolov T (2016) Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759
20. Rustam F, Ashraf I, Mehmood A, Ullah S, Choi GS (2019) Tweets classification on the base of sentiments for us airline companies. Entropy 21(11):1078
21. Sugumar R (2018) Improved performance of stemming using efficient stemmer algorithm for information retrieval. J Glob Res Comput Sci 9(5):01–05
22. Bocca FF, Rodrigues LHA (2016) The effect of tuning, feature engineering, and feature selection in data mining applied to rainfed sugarcane yield modelling. Comput Electron Agric 128:67–76
23. Heaton J (2016) An empirical analysis of feature engineering for predictive modeling. In: SoutheastCon 2016, pp 1–6. IEEE
24. Eshan SC, Hasan MS (2017) An application of machine learning to detect abusive bengali text. In: 2017 20th International conference of computer and information technology (ICCIT), pp 1–6. IEEE
25. Ye X, Zheng Y, Aljedaani W, Mkaouer MW (2021) Recommending pull request reviewers based on code changes. Soft Comput 25(7):5619–5632
26. Hartmann J, Huppertz J, Schamp C, Heitmann M (2019) Comparing automated text classification methods. Int J Res Mark 36(1):20–38
27. Safdari N, Alrubaye H, Aljedaani W, Baez BB, DiStasi A, Mkaouer MW (2019) Learning to rank faulty source files for dependent bug reports. In: Big data: learning, analytics, and applications, vol 10989, p 109890B. International Society for Optics and Photonics
28. Alkhazi B, DiStasi A, Aljedaani W, Alrubaye H, Ye X, Mkaouer MW (2020) Learning to rank developers for bug report assignment. Appl Soft Comput 95:106667
29. Osisanwo F, Akinsola J, Awodele O, Hinmikaiye J, Olakanmi O, Akinjobi J (2017) Supervised machine learning algorithms: classification and comparison. Int J Comput Trends Technol (IJCTT) 48(3):128–138
30. Biau G, Scornet E (2016) A random forest guided tour. Test 25(2):197–227

31. AlOmar EA, Aljedaani W, Tamjeed M, Mkaouer MW, El-Glaly YN (2021) Finding the needle in a haystack: On the automatic identification of accessibility user reviews. In: Proceedings of the 2021 CHI conference on human factors in computing systems, pp 1–15
32. Xuan S, Liu G, Li Z, Zheng L, Wang S, Jiang C (2018) Random forest for credit card fraud detection. In: 2018 IEEE 15th international conference on networking, sensing and control (ICNSC), pp 1–6. IEEE
33. Alzubi OA, Alzubi JA, Alweshah M, Qiqieh I, Al-Shami S, Ramachandran M (2020) An optimal pruning algorithm of classifier ensembles: dynamic programming approach. Neural Comput Appl 32(20):91–107
34. Sun T, Zhou Z-H (2018) Structural diversity for decision tree ensemble learning. Front Comput Sci 12(3):560–570
35. Altman N, Krzywinski M (2017) Ensemble methods: bagging and random forests. Nat Methods 14(10):933–935
36. Kukkar A, Mohana R, Nayyar A, Kim J, Kang B-G, Chilamkurti N (2019) A novel deep-learning-based bug severity classification technique using convolutional neural networks and random forest with boosting. Sensors 19(13):2964
37. Curran PJ, Cole VT, Bauer DJ, Rothenberg WA, Hussong AM (2018) Recovering predictor-criterion relations using covariate-informed factor score estimates. Struct Equ Model Multidiscip J 25(6):860–875
38. Ruehle F (2020) Data science applications to string theory. Phys Rep 839:1–117
39. Alzubi J, Nayyar A, Kumar A (2018) Machine learning from theory to algorithms: an overview. J Phys Conf Ser 1142(1):012012
40. Hu X, Choi K, Downie JS (2017) A framework for evaluating multimodal music mood classification. J Assoc Inf Sci Technol 68(2):273–285
41. Ray S (2019) A quick review of machine learning algorithms. In: 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon), pp 35–39. IEEE
42. Rustam F, Reshi AA, Ashraf I, Mehmood A, Ullah S, Khan DM, Choi GS (2020) Sensor-based human activity recognition using deep stacked multilayered perceptron model, vol 8, pp 898–910. IEEE Access
43. Gosain A, Sardana S (2017) Handling class imbalance problem using oversampling techniques: a review. In: 2017 international conference on advances in computing, communications and informatics (ICACCI), pp 79–85. IEEE
44. Amin A, Anwar S, Adnan A, Nawaz M, Howard N, Qadir J, Hawalah A, Hussain A (2016) Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study. IEEE Access 4:7940–7957
45. Fang F, Wu J, Li Y, Ye X, Aljedaani W, Mkaouer MW (2021) On the classification of bug reports to improve bug localization. Soft Comput 25(11):7307–7323
46. Xu J, Zhang Y, Miao D (2020) Three-way confusion matrix for classification: a measure driven view. Inf Sci 507:772–794
47. Jamil R, Ashraf I, Rustam F, Saad E, Mehmood A, Choi GS (2021) Detecting sarcasm in multi-domain datasets using convolutional neural networks and long short term memory network model. PeerJ Comput Sci 7:e645
48. Dey R, Salem FM (2017) Gate-variants of gated recurrent unit (GRU) neural networks. In: (2017) IEEE 60th international midwest symposium on circuits and systems (MWSCAS), pp 1597–1600. IEEE
49. Zhang Z (2018) Improved adam optimizer for deep neural networks. In: 2018 IEEE/ACM 26th international symposium on quality of service (IWQoS), pp 1–2. IEEE
50. Rupapara V, Rustam F, Shahzad HF, Mehmood A, Ashraf I, Choi GS (2021) Impact of smote on imbalanced text features for toxic comments classification using RVVC model. IEEE Access
51. Ranparia D, Kumari S, Sahani A (2020) Fake job prediction using sequential network. In: 2020 IEEE 15th international conference on industrial and information systems (ICIIS), pp 339–343
52. Keerthana B, Reddy AR, Tiwari A (2021) Accurate prediction of fake job offers using machine learning. In: Bhattacharyya D, Thirupathi Rao N (eds) Machine intelligence and soft computing, pp 101–112. Springer