IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 July 2024, accepted 25 July 2024, date of publication 30 July 2024, date of current version 16 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3435670

RESEARCH ARTICLE

Online Recruitment Fraud (ORF) Detection Using Deep Learning Approaches

NATASHA AKRAM[1], RABIA IRFAN[1], AHMAD SAMI AL-SHAMAYLEH[2], ADILA KOUSAR[1], ABDUL QADDOS[3], MUHAMMAD IMRAN[3], AND ADNAN AKHUNZADA[4], (Senior Member, IEEE)

[1]School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2]Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Al-Ahliyya Amman University, Amman 19328, Jordan
[3]Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad 45550, Pakistan
[4]Department of Data and Cybersecurity, College of Computing and Information Technology, University of Doha for Science and Technology, Doha 24449, Qatar

Corresponding authors: Adnan Akhunzada (adnan.adnan@udst.edu.qa) and Natasha Akram (nakram.mscs19seecs@seecs.edu.pk)

The open access funding is provided by Qatar National Library. Besides, we also extend appreciation for the necessary support of Al-Ahliyya University.

ABSTRACT Most companies nowadays are using digital platforms for the recruitment of new employees to make the hiring process easier. The rapid increase in the use of online platforms for job posting has resulted in fraudulent advertising. The scammers are making money through fraudulent job postings. Online recruitment fraud has emerged as an important issue in cybercrime. Therefore, it is necessary to detect fake job postings to get rid of online job scams. In recent studies, traditional machine learning and deep learning algorithms have been implemented to detect fake job postings; this research aims to use two transformer-based deep learning models, i.e., Bidirectional Encoder Representations from Transformers (BERT) and Robustly Optimized BERT-Pretraining Approach (RoBERTa) to detect fake job postings precisely. In this research, a novel dataset of fake job postings is proposed, formed by the combination of job postings from three different sources. Existing benchmark datasets are outdated and limited due to knowledge of specific job postings, which limits the existing models' capability in detecting fraudulent jobs. Hence, we extend it with the latest job postings. Exploratory Data Analysis (EDA) highlights the class imbalance problem in detecting fake jobs, which tends the model to act aggressively toward the minority class. Responding to overcome this problem, the work at hand implements ten top-performing Synthetic Minority Oversampling Technique (SMOTE) variants. The models' performances balanced by each SMOTE variant are analyzed and compared. All implemented approaches are performed competitively. However, BERT+SMOBD SMOTE achieved the highest balanced accuracy and recall of about 90%.

INDEX TERMS Class imbalance, data augmentation, deep learning, employment scam, fraud detection, machine learning, online recruitment, SMOTE, transformer-based models.

I. INTRODUCTION

In the age of advanced technology, the internet has drastically transformed our lives in different ways. The traditional way to do any activity has now been switched online. Therefore, seeking a job and hiring employees have also switched online. An online recruitment system (E-recruitment) is an internet application, the benefits of which encompass productivity, easiness, and efficacy [1]. Most organizations prefer online recruitment systems to provide job opportunities to potential candidates [2]. Organizations publish job ads for their vacant positions through job portals, in which they mention job descriptions, including requirements, salary packages, offers, and facilities to be provided. Job seekers visit different online job advertising websites, seek job ads related to their interests, and apply for suitable jobs. The company then screens the CVs of applicants matching their requirements. The position is closed after fulfilling other formalities like interviewing and selecting potential candidates. The trend of posting online job advertisements was inflated during the global pandemic of COVID 2019. According to the World Economic Outlook Report, the International Monetary Fund (IMF) estimated that

The associate editor coordinating the review of this manuscript and approving it for publication was Yiqi Liu.

the unemployment rate increased to 13% at the peak time of the COVID-19 pandemic in 2020. These statistics were only 7.3% in 2019 and 3.9% in 2018. During the outbreak, many companies decided to post job openings online to provide facilities to job seekers [3]. But, where a facility is provided to the public, it also allows online fraudsters to take advantage of their pessimism.

An employment scam is one of the considerable problems in the realm of online recruitment fraud (ORF). Although an online recruitment system benefits job seekers and recruiters, it can also be deleterious for them if it is not administered carefully. It is inauspicious for job seekers in terms of losing their privacy, money, or even their current job sometimes. Moreover, fraudsters also breach the credibility of well-reputed companies by defacing their reputation in the job market [4]. The fraudsters are using sophisticated methods to involve people in the scam, and making it very difficult for them to distinguish between real/fake job advertisements. According to the survey conducted by Flex Jobs [5], about 52% of the aspirants did not know ORFs, whereas the rest had only preliminary knowledge about them. Another survey recently accompanied by Action Fraud [6], it is investigated that more than 67% of people are now interested in looking for a job online. Still, they need to be aware of the increased number of job scams.

Multiple studies were conducted to detect ORF. Authors in [7] and [8] applied traditional machine learning algorithms to classify job postings as fraudulent/non-fraudulent. The work [9] and [10] used ensemble-based machine learning techniques to improve classification accuracy. The authors in [11] first performed downsampling to handle the imbalance problem and then used an Artificial Neural Network (ANN) based model for classification. Authors in [12] extracted features by using TF-IDF, and after oversampling data, applied Random Forest (RF) to improve accuracy. Researchers in [13] created their own dataset and proposed context-based behavioral features to test them on conventional machine learning algorithms to get predictions. Upon reviewing the underlying study, it is noticed that many machine learning approaches have been used for ORF detection. Nowadays, the trend is moving towards implementing transformer-based deep learning techniques to get promising results compared to traditional machine learning algorithms; however, for ORF detection advanced deep-learning approaches have yet to be explored in their full capacity to solve this problem. Therefore, this research aims to analyze and alarm people about rapidly growing employment scams and to detect ORF by implementing transformer-based deep learning models. Consequently, people would not fall into the trap of job scams anymore. So by detecting ORF, the people wasting their time and money on those fraudulent activities can be more careful.

In this research, we presented a novel dataset of fake job postings labeled as ''fraudulent'' for fake job postings and ''non-fraudulent'' for legitimate job postings. The proposed data is a combination of job postings from three different sources. We use ''Fake Job Postings[1] as a primary dataset and add publicly available job postings of Pakistan[2] and the US[3] to extend the dataset with the latest job postings. We have done this because the existing benchmark datasets are outdated and limited due to knowledge of specific job postings, which limits the capability of existing models in detecting fraudulent jobs. After preparing the dataset, Exploratory Data Analysis (EDA) was performed on this data. Through EDA, it was identified that the dataset has an imbalanced class distribution. Imbalance class distribution can be defined as the ratio of the number of samples in the minority class to the number in the majority class [14]. It may cause high predictive accuracy for frequent classes and low predictive accuracy for infrequent classes. Class imbalance problem occurs in various real-world domains, including anomaly detection [15], face recognition [16], medical diagnosis [17], text classification [18], and many others. SMOTE [19] gained extensive popularity as an oversampling technique. Almost 85 different SMOTE variants have been introduced in the literature and are recently used by various researchers to handle class imbalance problems in multiple domains.

The objective of this research is to investigate Online Recruitment Fraud (ORF) and to overcome the possible issues in implementing the system. The significant contributions of this study are mentioned as follows:

- Job postings from three different sources are collected and combined to present a novel dataset.
- It is observed from Exploratory Data Analysis (EDA) that the class distribution from the collected dataset is highly imbalanced. Ten top-performing SMOTE variants are implemented to balance the class distribution ratio.
- Transformer-based deep learning models are implemented on the dataset to detect whether a job posting is fraudulent or non-fraudulent.
- Comparative analysis of implemented models is conducted on both imbalanced and balanced datasets.

The rest of the paper is comprised of the following sections: Section II mentions a detailed recap of work that has already been done related to the underlying study. Section III exhibits characteristics of the dataset, proposed methodology, and framework of implemented models. Section IV illustrates the experimental results, and discusses the critical findings of the study. In the end, Section V concludes the presented study with limitations and some future recommendations.

II. RELATED WORK

This section reviews multiple studies related to Online Recruitment Fraud (ORF) detection. Moreover, as it is mentioned earlier that the collected dataset for this research

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

[2] https://www.kaggle.com/datasets/zusmani/pakistans-job-market

[3] https://www.kaggle.com/datasets/promptcloud/indeed-job-posting-dataset

has a class imbalance problem associated with it; hence, the literature related to handling class imbalance problem is also reviewed in this section.

A. ORF DETECTION TECHNIQUES

To detect fake job postings, Vidros et al. [7] officially released the first dataset, ''Employment Scam Aegean Dataset'' (EMSCAD), and applied traditional machine learning classifiers on it to detect ORF. They performed two types of experiments and compared their results. The first experiment consists of six different classifiers, Naive Bayes (NB), Zero Rule (ZeroR), One Rule (OneR), Logistic Regression (LR), J48, and Random Forest (RF). The best classifier of this experiment is RF, with the highest precision of 91.4%. For the second experiment, the empirical ruleset model is used. LR, J48, and RF classifiers gave a precision of 90.6% for the empirical ruleset modeling. Dutta and Bandyopadhyay [8] also applied machine learning algorithms to the ''fake job postings'' dataset. NB, Multi-Layer Perceptron (MLP), K-Nearest Neighbor (KNN), and Decision Tree (DT) are used as single classifier-based predictions. RF, Adaptive Boosting (AdaBoost), and Gradient Boosting (GB) classifiers are used as ensemble classifier-based predictions. DT achieved the highest accuracy of 97.2% among single classifier-based predictions, whereas, the RF classifier outperforms with an accuracy of 98.27% among ensemble classifier-based predictions. Another work to detect ORFs was published by Alghamdi and Alharby [9]. They applied Support Vector Machine (SVM) for the determination of relevant features present in the dataset. For the classification task, they used an ensemble-based RF classifier. The precision accomplished by this research is 97.2%, considered high and adequate. Lal et al. [10] used three ensemble techniques, Maximum Vote, Majority Vote, and Average Vote, and applied them to three baseline classifiers, RF, LR, and J48, to build ORF Detector. The extracted features are categorized into three basic categories: contextual, linguistic, and metadata. The accuracy achieved by the proposed ORFDetector is 95.5%. Nasser et al. [11] used an imbalanced dataset and tried to tackle this problem by downsampling majority class records. Artificial Neural Network (ANN) is used in this paper to detect Online Recruitment Frauds. The accuracy achieved by the proposed model is 93.64%. A study by Habiba et al. [20] applied different data mining techniques to the EMSCAD dataset. They have evaluated both traditional Machine Learning (ML) and Deep Learning (DL) classifiers. RF was the outperformer, with the highest accuracy of 96.5% among ML classifiers, and Deep Neural Network (DNN) has the highest accuracy of 99% among DL models. Another study by Lokku et al. [12] used the ''EMSCAD'' dataset. After data cleaning and preprocessing steps, features were extracted by using the TF-IDF. As the dataset is imbalanced, they did some work related to balancing the data by increasing data points of the minority class and then used the RF classifier on the balanced dataset. They secured an accuracy of 99% by using this approach. Nindyati and Nugraha [13] also researched to eliminate the problem of employment scams. They created a dataset named as Indonesian Employment Scam Detection Dataset (IESD). They proposed context-based behavioral features to predict whether there is a scam in online job vacancy descriptions. They tested their proposed features on six machine learning algorithms, NN, SVM, LR, DT, NB, and KNN. An accuracy of 90% is attained by using behavioral features. Alandjani et al. [21] used two features set on machine learning models, DT, NB, RF, and KNN, to classify job advertisements and compare them. It is noticed that KNN gave promising results in this research work.

	job_content	fraudulent
0	marketing intern marketing we're food, and we'...	0
1	customer service cloud video production succes...	0
2	commissioning machinery assistant cma valor se...	0
3	account executive washington dc sales our pass...	0
4	bill review manager spotsource solutions llc i...	0

FIGURE 1. Data samples.

B. DATA AUGMENTATION TECHNIQUES

To balance class distribution in data, Gosain and Sardana [22] proposed four oversampling techniques; Synthetic Minority Oversampling Technique (SMOTE), Borderline-SMOTE, ADASYN, and Safe Level SMOTE with various classification models, NB, KNN, and SVM. These oversampling techniques and models were implemented on six different datasets. The performance of different oversampling techniques on various datasets has been evaluated. SLSMOTE is considered to be the outperformer in this study. Akhbardeh et al. in [23] experimented with seven logbook datasets from the domain of facility, aviation, and automotive. They used four methods to handle the class imbalance problem: undersampling, oversampling, feedback loop, and random downsampling loop. The models used for classification are Bidirectional Encoder Representations from Transformers (BERT), Long Short-Term Memory (LSTM) networks, Convolutional Neural Network (CNN), and DNN. The feedback loop achieved better results on all models and datasets mentioned above. Ah-Pine and Soriano-Morales [24] worked on three publicly available imbalanced datasets of Twitter: Health Care Reform (HCR), Obama-McCain Debate, and ImagiWeb (IW). They used three oversampling techniques to overcome the class imbalance problem: ADASYN, SMOTE, and Borderline-SMOTE. Supervised learning models like l1 penalized LR and DT are used for classification. ADASYN exhibited peculiar behavior and gave more stable results. David et al. [25] worked on the ''SocIal Media And Harassment (SIMAH)'' dataset to handle class imbalance issues. Three different experimental sets have been built for comparison: BERT+LSTM, BERT+FNN, and BERT+SMOTE+LSTM. Results show that the model BERT+SMOTE+LSTM gave better results than the other

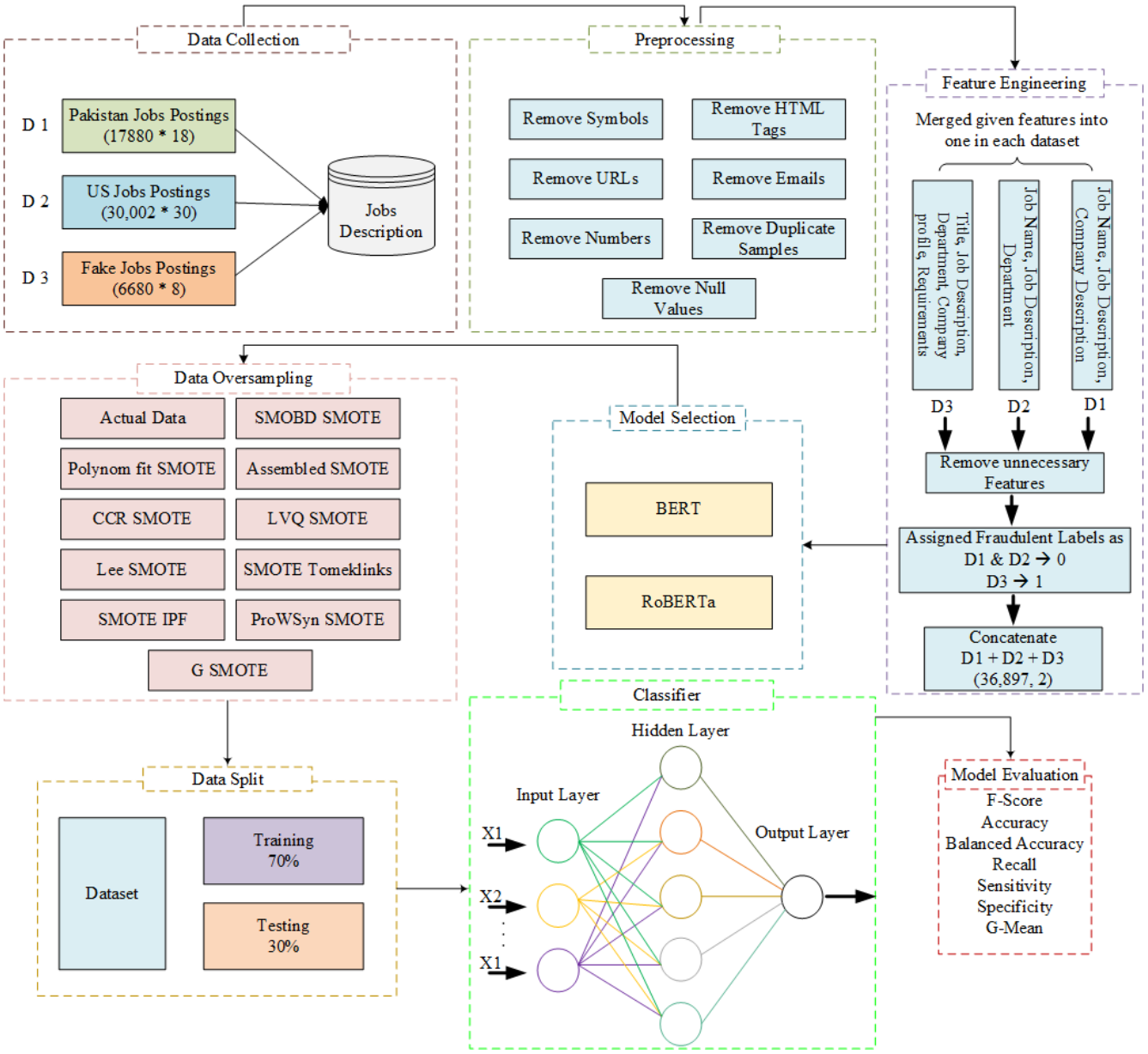


FIGURE 2. The proposed methodology.

models. Singla et al. [26] handled the class imbalance problem in Online Transaction Fraud (OTF) detection. Three datasets having transactional data, namely Credit Card, Banksim, and IEE CIS, are used. DNN architecture containing two hidden layers has been set up for all datasets with a different set of hyperparameters, and the performance of DNN significantly improved compared to other baseline methods.

C. CRITICAL ANALYSIS

Many machine learning approaches have been used for Online Recruitment Fraud (ORF) detection; however, advanced deep-learning approaches have yet to be explored in their full capacity to solve this problem. Employment scam is one of the significant issues drastically increasing day by day, as thousands of job advertisements are posted daily by scammers on various job portals or social media platforms. Scammers not only harm the privacy of the candidates, but the candidates also suffer in terms of loss of money and even their current job sometimes. Hence, there is a need to detect illegitimate job postings to restrain people from being scammed. For better detection of job advertisements, advanced deep learning approaches must be applied, so that job seekers seek only legitimate job offers of their interest posted by authentic companies. It is also observed from the above-mentioned literature that most of the work done related to fraud detection problems is intended to improve classification accuracy. Very high accuracies have indeed

been achieved but with poor recall. However, due to the class imbalance problem, accuracy does not represent the accurate picture of the story. It can be misleading that we get high predictive accuracy for the majority class and fail to seize the minority class, so we cannot rely only upon it as an evaluation metric. There is a need to improve balanced accuracy and recall to capture the situation truly. Furthermore, it is identified from Exploratory Data Analysis (EDA) that our collected dataset has a class imbalance problem, so the literature review helped us identify some top-performing SMOTE variants to be selected for experimenting in this regard.

The methodology we will follow to handle the issues mentioned above is presented in the next section in detail.

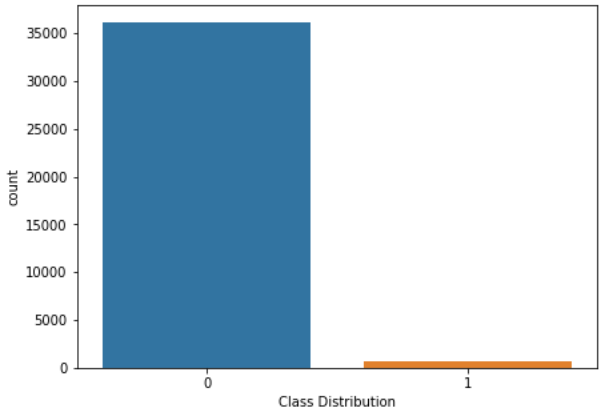


FIGURE 3. Amount of real vs. fake job posts.

III. PROPOSED METHODOLOGY

This section discusses the different phases involved in the underlying research. Firstly, datasets from three different sources are integrated to propose a final version of the dataset. An Exploratory Data Analysis (EDA) is performed to identify that the dataset has an imbalanced class distribution. A detailed discussion is given in the section III-C to show the importance of different features. Second, necessary steps in the preprocessing phase are performed on the proposed data. The special symbols, URLs, emails, numbers, HTML, tags, duplicate records and samples that contain null values are removed in the preprocessing phase to clean the dataset.

Thirdly at the feature engineering phase, only required and relevant features are selected and merged as a single feature named ''Job_Content''. This process is repeated for each dataset D1, D2, and D3 as shown in Fig. 2. Then, fraudulent and non-fraudulent labels are assigned as D1, D2 to '0' for non-fraudulent jobs and D3 to '1' for fraudulent job posting. Later in the next step of the feature engineering phase, all three datasets D1, D2, and D3 are concatenated to generate a finalised dataset as shown in Fig. 2.

The dataset is encoded in phase four through BERT/RoBERTA to generate the contextual vectors. Then data is augmented using different SMOTE variants to get a balanced class distribution in the fifth phase. We chose to use only the encoder part of the BERT/RoBERTa model because contextual information across entire sequences is essential for ORF detection. The ability of the BERT/RoBERTa model to grasp long-range dependencies is particularly relevant for identifying subtle patterns indicative of fraudulent activities. Moreover, it has been found in the literature review that the tasks requiring contextual understanding, such as natural language processing, demonstrate the superior performance of transformer-based models.

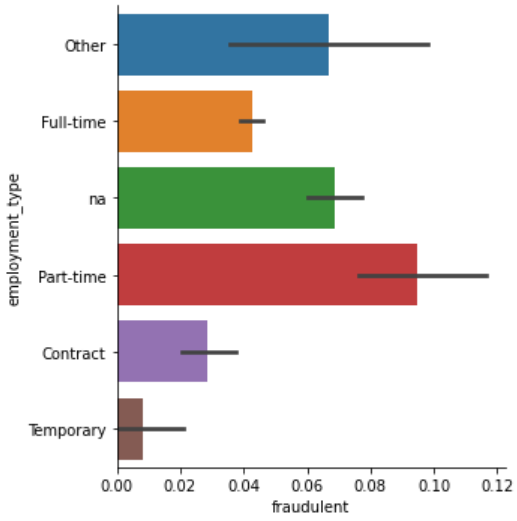
Lastly, classification is performed to detect fraudulent job postings. Fig. 2 shows the flow diagram of the proposed methodology. The details about each of the phases are discussed below:

A. DATA ACQUISITION

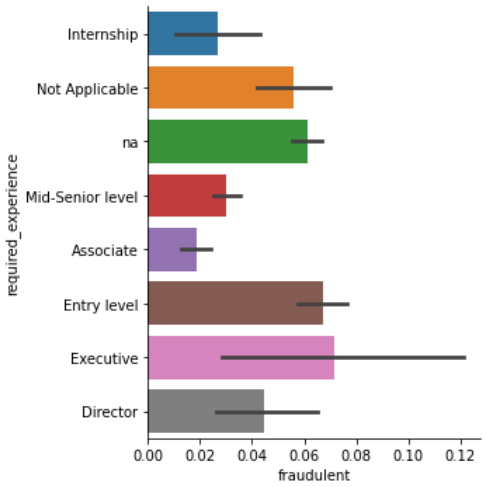
To address the underlying problem, we present a novel dataset of fake job postings labeled as ''fraudulent'' for fake and ''non-fraudulent'' for legitimate job postings. The proposed data is a combination of job postings from three different sources mentioned as follows:

- ''Fake Job Postings'' dataset [27] containing almost 17,880 real-life job postings advertised between 2012 and 2014 in different countries was collected. Eighteen features represented a particular job posting in this data.
- ''US Job Postings'' dataset [28] containing almost 30,000 job advertisements published from July 2019 to August 2019 and belonging to different cities in the United States was collected. Thirty features represented a particular job posting in this data.
- ''Pakistan Job Postings'' dataset [29] containing about 7000 job advertisements published during COVID-19 from December 2019 to March 2021 and belonging to different cities in Pakistan was collected. Nine features represented a particular job posting in this data.

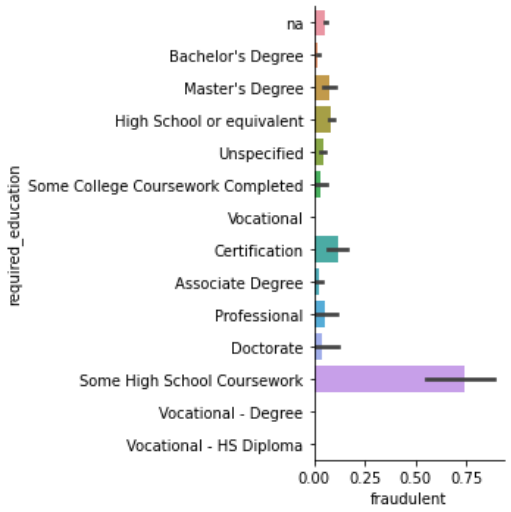
We add publicly available job postings of Pakistan and the US in the ''Fake Job Postings'' dataset. The reason to extend the ''Fake Job Postings'' dataset is that its job postings are pretty outdated, and limited due to knowledge of specific job postings, which limits the capability of existing models in detecting fraudulent jobs. Therefore, we enhance this dataset with the latest job postings of Pakistan and the US to get a better realization of this problem. All textual columns of the aforementioned datasets are combined into a single column to get a prediction. The shape of the final data is now changed, shown in Fig. 1. It has only two columns. The first is ''job-content,'' representing the job description, whereas the second column, ''fraudulent,'' represents the class label. It can either be ''0'' for non-fraudulent or ''1'' for fraudulent. The rest of the columns do not take part in making predictions. They have been kept for analysis purposes only. In the next section, the preprocessing steps performed to clean our data are discussed in detail.



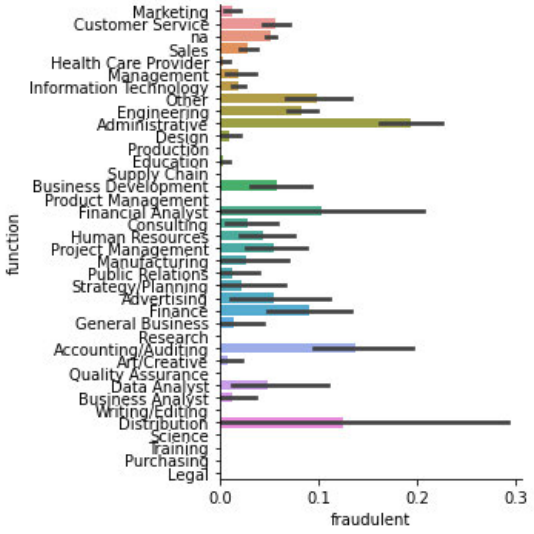
(a) w.r.t Employment Type.



(b) w.r.t Required Experience.



(c) w.r.t Required Education.



(d) Fraudulency of Job Postings.

FIGURE 4. **Fraudulency of job postings.**

B. DATA PREPROCESSING

Data preprocessing is a crucial step to transform raw data in a way suitable for any machine learning and deep learning task. In this phase, we only keep a useful portion of data and remove unnecessary data. We used *neattext*[4] python library for preprocessing task. Various preprocessing steps are performed, which include the extraction of hashtags, HTML tags, URLs, email addresses, special characters, and duplicate and null values from the data because such words do not affect the orientation of the text. Lowercasing all available text is also necessary to preserve the consistent flow of the text. After getting cleaned data, it is split into training and testing sets with a ratio of 80:20. Exploratory Data Analysis (EDA) is performed in Section III-C to examine the patterns present in the data.

[4]https://pypi.org/project/neattext/

C. EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) is an analytical process of scrutinizing the data for better insight. It is important to understand the data and its main characteristics, visualize it to deal with outliers present in the data, examine the data distribution to discover patterns and trends and find correlations between attributes. Therefore, it has always been a good practice to analyze data before giving it to any model for prediction. We have analyzed over 30 features from the given datasets to identify the most impactful elements for improving model performance in ORF detection. The key features identified are Job Title,

Job Description/Function, Company Profile, Job Requirement, Department, and Employment Type. These features are crucial as they provide comprehensive insights into job postings, which are essential for accurately detecting fraudulent listings. Job requirements are further categorized into required education and required experience, providing granular details that enhance model precision. We have created comparison graphs as shown in Fig. 4 for the four major features that significantly impact the detection of fraudulent job postings, illustrating their importance in our analysis.

To analyze illegitimate job postings, we explored our proposed data to understand it well and extract different patterns. From Fig. 4(a), it has been analyzed that the job advertisements specified as ''part-time'' w.r.t employment type are more likely to be fraudulent, with a fraudulency rate of more than 90%. The job postings in which no employment type is mentioned have a fraudulency rate of about 70%. In contrast, those in which employment type is specified as ''temporary,'' ''contract,'' and ''full-time,'' are less likely to be fraudulent, having fraudulency rates of about 10%, 30%, and 50%, respectively. From Fig. 4(b), it is observed that the job postings requiring experience as ''executive'' are more likely to be fraudulent, with a fraudulency rate of about 70%. The job postings that required ''no experience'' and ''entry-level experience'' have a fraudulency rate of about 60%. In contrast, those that require experience of ''mid-senior,'' ''internship,'' and ''associative'' level are less likely to be fraudulent, having a fraudulency rate between 20-30%.

From Fig. 4(c), it is observed that the job postings specified as ''some high school coursework'' are more likely to be fraudulent, having a fraudulency rate of about 75%. In contrast, the other categories have more or less the same fraudulent rates of not more than 25% in terms of required education. From the list of job functions shown in Fig. 4(d), ''administrative'' has the highest fraudulent rate of about 20% as compared to other job functions. It has been observed through EDA that our data is highly imbalanced. This problem occurs when we have a large number of instances of one class but very few instances of another class; this situation is termed a class imbalance problem, and our dataset suffers from a class imbalance problem as presented in Fig. 3. There are two classes of data, i.e., fraudulent and non-fraudulent. ''Fraudulent'' is a minority class represented by 1, whereas ''non-fraudulent'' is a majority class represented by 0. The imbalance ratio of the majority to minority class is 36162:735. Only 2% of the instances belong to the minority class, and the rest belong to the majority class. It shows that the class distribution is extremely skewed, which can cause severe issues. It may cause high predictive accuracy for frequent classes and low predictive accuracy for infrequent classes. Accuracies achieved from imbalanced data might be very high with poor recall value. This situation represents biases towards the majority class.

Therefore, accuracies attained from imbalanced data might not be true because these are often misleading accuracies. Ten top-performing SMOTE variants are implemented on the embedding of minority class to get a balanced ratio. The details of which are given in the next section.

D. DATA AUGMENTATION TECHNIQUES - SMOTE VARIANTS

Data augmentation refers to the process of generating new training examples from existing data. Generally, there are two approaches to overcoming the class imbalance problem; the algorithm-level approach [30] and the data-level approach [31]. The algorithm-level approach aims to enhance learning tasks with respect to the minority class by fine-tuning the traditional classification algorithm. To balance class distribution through the data-level approach, undersampling, oversampling, and hybrid techniques are usually used. Undersampling eliminates some instances from the majority class while oversampling adds some instances to the minority class to overcome class disparity. As the process of undersampling involves the removal of examples from the majority class, it can cause a loss of useful information that might be significant for creating rule classifiers. The remaining examples can be biased and might not represent a true population. Thereby cause to give inaccurate results on test data. We need excessive data for machine and deep learning models for better training, but removing instances will reduce the data size in undersampling. Based on these grounds, undersampling could not be more favorable for the underlying study. Therefore, we prefer to use the oversampling technique.

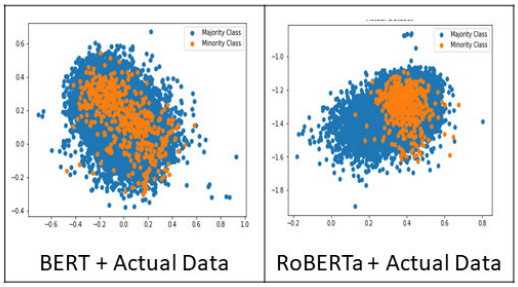


FIGURE 5. Data distribution of actual data.

SMOTE is used to generate synthetic samples for the minority class of job postings dataset. Initially, $X_{\min}$ the total number of samples from minority class, $y_{\min}$ is a corresponding label, are identified from the input dataset X and their corresponding labels y as shown in Eq. 1.

$$X_{\min}, y_{\min} = \text{minority class samples from } X \text{ and } y \quad (1)$$

$$x_{\text{synth}} = x_i + \lambda \times (x_{nn} - x_i) \quad (2)$$

The Eq. 2 is the interpolation method for SMOTE algorithm where $0 < \lambda < 1$ is a random number, x_i is random sample from dataset and x_{nn} nearest random sample of x_i.

The dataset X, labels y along with minority class $X_{\min}$ and $y_{\min}$ are given as input to the SMOTE algorithm 1 in step 1 and the output is the number of generated synthetic samples X_{synth} and y_{synth} at step 2. The SMOTE

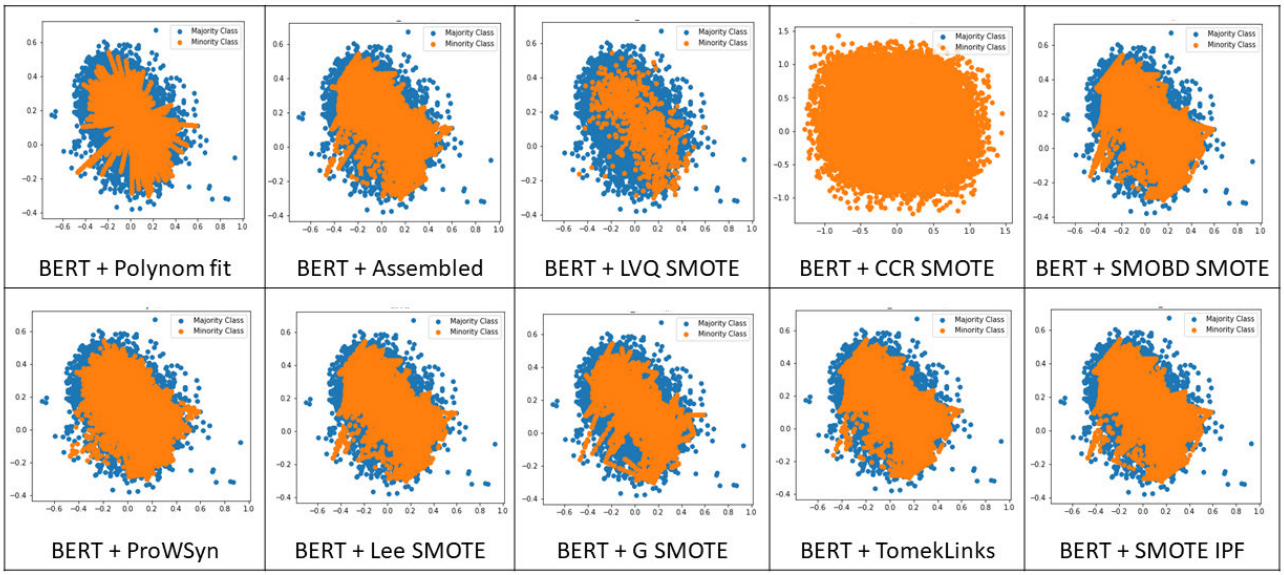


FIGURE 6. Data distribution of BERT+SMOTE variants.

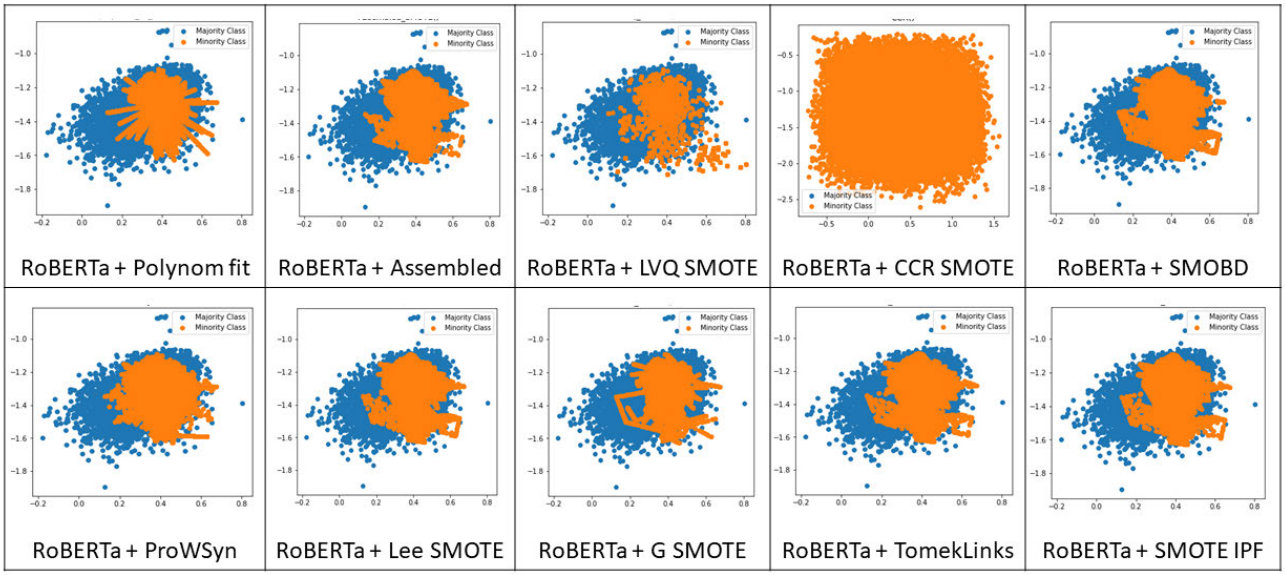


FIGURE 7. Data distribution analysis of RoBERTa+SMOTE variants.

algorithm 1 takes three additional parameters (k, strategy, interpolation) to specify the neighborhood selection strategy and the interpolation method, allowing for a more generic and flexible implementation of different SMOTE variants.

At step 4, a random sample x_i is selected from minority class X_{min} and then k nearest neighbors are calculated using a selection strategy depending on the type of SMOTE variant. At step 6, a random neighbor x_{nn} sample is selected from the neighborhood. A synthetic sample is generated using an interpolation method at step 7. The interpolation method varies for different variants of SMOTE. At step 8, generated synthetic sample x_{synth} is added to the X_{synth} and its label is assigned as minority class to y_{synth}.

From step 3 to step 9, each step is repeated until the required number of synthetic samples are generated. At the end of step 10, synthetic samples X_{synth} and y_{synth} are combined with X_{min} and y_{min} to produce the final balanced dataset.

SMOTE gained extensive popularity as an oversampling technique. Authors in [33] carried out an extensive comparison and evaluation of 85 different SMOTE variants on 104 imbalanced datasets. After a comprehensive analysis, they ranked ten variants as top performers by the average

Algorithm 1 SMOTE for Oversampling of Dataset [32]

Input ← Input dataset X and y, $X_{\min}$, $y_{\min}$, Number of synthetic samples N_{synth}, k (neighborhood size), *strategy* (neighborhood selection strategy), *interpolation* (interpolation method)
Output ← Dataset with synthetic samples X_{synth}, Labels y_{synth}; $X_{\text{synth}} \leftarrow \{\}$, $y_{\text{synth}} \leftarrow \{\}$
for i **do** from 1 to N_{synth}
    Select a random sample x_i from $X_{\min}$
    Find its k nearest neighbors $NN(x_i)$ using *strategy*
    Select a random neighbor x_{nn} from $NN(x_i)$
    Generate a synthetic sample using (eq.)
    Add x_{synth} to X_{synth} and assign its label as the minority class to y_{synth}
end for
Combine $X_{\min}$ with X_{synth} and $y_{\min}$ with y_{synth} to get the final dataset and labels.

score of different metrics achieved on all datasets. Hence, we used these ten top-performing SMOTE variants, including Polynom fit SMOTE [34], ProWSyn SMOTE [35], SMOTE IPF [36], Lee SMOTE [37], SMOBD SMOTE [38], G SMOTE [39], CCR SMOTE [40], LVQ SMOTE [41], Assembled SMOTE [42], and SMOTE Tomeklinks [43] to balance the class distribution of the underlying data. Data distribution analysis is performed on the actual data and on the data balanced by these SMOTE variants to know the behavior of instances in a sample space. Fig. 5 shows highly imbalanced data distribution of actual data for BERT and RoBERTa. After implementing various SMOTE variants, it is noticed from Fig. 6 and Fig. 7 that both models among each SMOTE variant have a different data distribution. Some variants fully balanced the class distribution, and some of them partially balanced it, because each SMOTE variant uses a different approach to oversample the minority class. CCR SMOTE didn't perform well on our data, and it can be observed that minority data points dominate the majority class for both models leading to imprecise results. However, BERT+SMOBD SMOTE and RoBERTa+G SMOTE balanced the class distribution very well. Its impact on results will be shown in Section IV-B.

E. TECHNIQUES USED FOR ORF DETECTION

As discussed earlier in Section III-C, the dataset exhibits a significant imbalance, resulting in high but misleading accuracy scores with low recall values. Presently, there is a growing inclination towards employing transformer-based deep learning methodologies, which offer more promising outcomes compared to traditional machine learning algorithms. Hence, the focus of our research centres around the implementation of transformer-based deep learning models for both imbalanced and balanced datasets. A transformer-based model refers to a neural network architecture that incorporates an encoder-decoder structure featuring a multi-headed self-attention mechanism. This type of model possesses the ability to capture extensive contextual information pertaining to individual words, thus enhancing its performance efficiency when compared to conventional neural networks. Within the scope of this study, we employ two well-known transformer-based models, namely BERT and RoBERTa, as the underlying frameworks.

1) BERT

BERT [44] is a transformer-based deep learning approach to pre-train bidirectional models from unlabeled contents. A pre-trained BERT model can be acclimated by varying output layers to design state-of-the-art models for a vast range of problems. BERT has been used to perform different tasks, including text classification [45], [46], [47], [48] sentiment analysis [49], [50], text summarization [51], question answering [52], text generation [53], [54], text clustering [55], document classification [32], text similarity [56], neural machine translation [57], sequence labeling [58] and many others.

We used the BERT model to learn contextual representations of words in a sentence with the objective of masked language modeling (MLM). Two different variants of BERT, $BERT_{base}$ and $BERT_{large}$ have been proposed and trained for various applications. In our problem scenario, we used the $BERT_{base}$ model to encode the text sentences for the classification task. The number of hidden units in our setting for $BERT_{base}$ are specified as 768, with 12 transformer layers, 12 self-attention heads in each layer and 110M total number of parameters.

The BERT model architecture has several layers, including an input embedding layer, multiple transformer encoder layers, and a final output layer. The input layer converts each token of the input sequence into a vector representation. These vectors are then fed into the transformer encoder layers, which use self-attention mechanisms to capture the relationships between different words in a sentence. The output of the transformer encoder layers is passed through a classification layer, which produces the final output of the model. One of the key features of using the BERT model is its ability to handle variable-length input by using positional embeddings to encode the position of each token in a sequence. Additionally, BERT uses a special classification symbol ([CLS]) token to represent the entire input sequence, which is used for tasks such as text classification. Similarly, a separator ([SEP]) token indicates a clause symbol which is used for separating two sentences. It is used at sentence-level embedding.

$$X = \{x_1, x_2, \ldots, x_n\} \tag{3}$$

Let X be the input sequence of tokens, where n is the sequence length.

$$E = \text{Embedding}(X) = \text{Embedding}(\{x_1, x_2, \ldots, x_n\}) \tag{4}$$

where $E \in \mathbb{R}^{V \times d_{\text{model}}}$ is the matrix of embedded vectors, V is the vocabulary size and d_{model} is the dimensionality of the

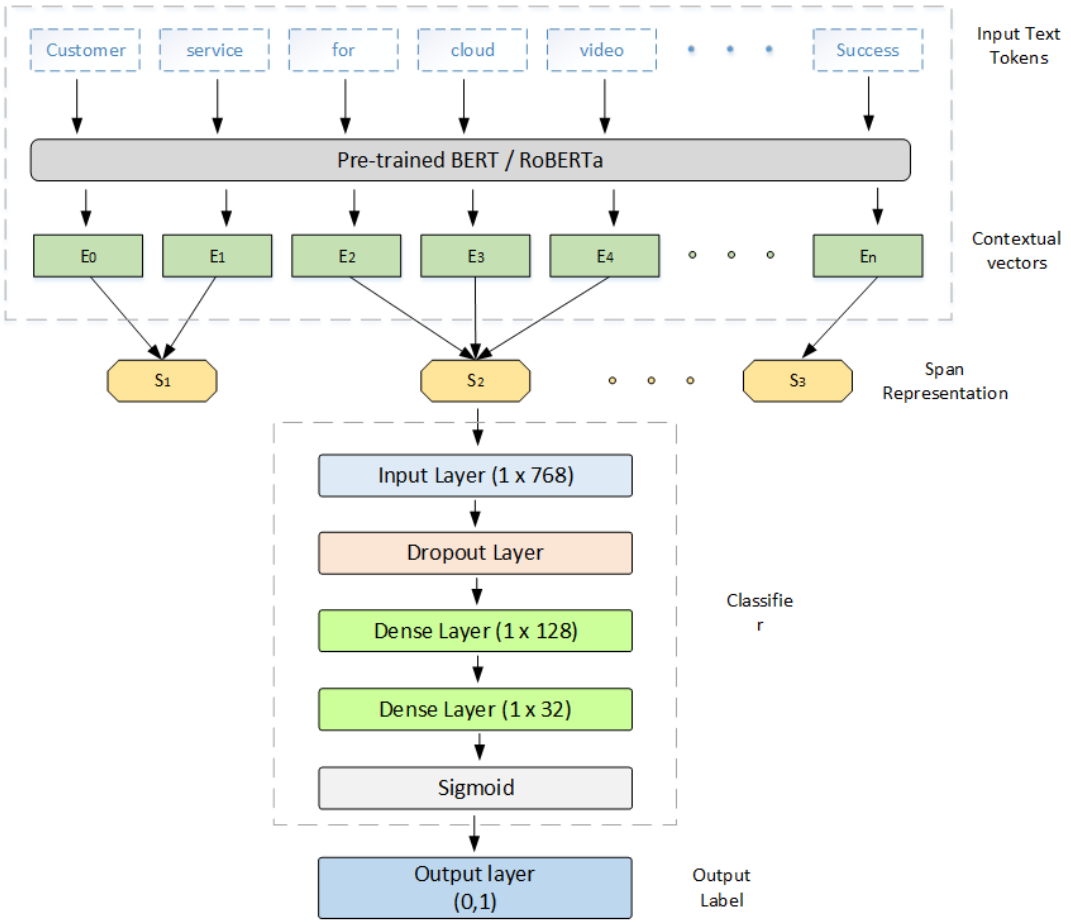


FIGURE 8. Model architecture.

embedding space.

$$Z = Fn_{\text{encoder}}(E) \tag{5}$$

where Z is the Transformer encoder consists of multiple layers.

$$Z_0 = E \quad \text{(initial embedding)} \tag{6}$$

$$Z_i = \text{LayerNorm}(Z_{i-1} + \text{MultiHeadAttention}(Z_{i-1})) \tag{7}$$

$$Z_i = \text{LayerNorm}(Z_i + \text{FeedForward}(Z_i)) \tag{8}$$

where $Z_i \in \mathbb{R}^{n \times d_{\text{model}}}$ is the output of the ith layer, L is the total number of layers, LayerNorm is layer normalization, Multi-Attention is the multi-head attention mechanism, and FeedForward is a feed-forward neural network with a non-linear activation function like ReLU. The Attention mechanism and position-wise feed-forward neural network can be defined as follows:

$$Q = Z_i \cdot W^Q$$
$$K = Z_i \cdot W^K$$
$$V = Z_i \cdot W^V$$

$$\text{AttentionScores} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{9}$$

$$\text{Multi-Attention}(Z_i) = \text{AttentionScores} \cdot V \tag{10}$$

$$\text{FeedForward}(Z_i') = \text{ReLU}(Z_i' \cdot W_1 + b_1) \cdot W_2 + b_2 \tag{11}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$ are learnable weight matrices, $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$, $b_1 \in \mathbb{R}^{d_{\text{ff}}}$, $W_2 \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$, $b_2 \in \mathbb{R}^{d_{\text{model}}}$ are learnable parameters, and d_k, d_{ff} are the dimensionality of the query, key, and value vectors and the hidden layer in the FeedForward, respectively.

The final output Z is a sequence of vectors, each corresponding to an input token, providing contextualized representations.

2) RoBERTa

A robust variant of BERT representation is proposed to improve end-task performance, and this modification of BERT is called RoBERTa [59]. We implemented RoBERTa architecture to fairly compare the results of the proposed methodology with BERT. During training, a random subset of tokens in each input sentence is masked, and the model is trained to predict the masked tokens based on the surrounding context. Unlike BERT, the dynamic masking technique is used in RoBERTa, where contiguous spans of tokens are masked together rather than individual tokens. This helps the model learn more effectively about context and relationships between words in longer sequences.

Algorithm 2 Proposed BERT/RoBERTa Model

Input ← Input data sentences X, labels Y
Output ← Accuracy, Specificity, Recall, F1-score, Balanced Accuracy, Sensitivity, G-Mean
Initialize Embedding Sequence
Initialize Pre-trained BERT/RoBERTa model parameters
function BPE(X)
    Initialize V with characters in X and ($< /CLS >$)
    Append text $X_{\text{tokenized}}$ with $< /CLS >$ to each word
    for size of V is less than desired **do**
        Compute frequency of all character pairs in V
        Merge most frequent pair into a single token
        Update vocabulary V
    end for
    Split text ($X_{\text{tokenized}}$) into characters
    Tokenize text using final vocabulary V
    return Tokenized text (X_{final})
end function
function Embedding(X_{final})
    Define vocabulary size V_s
    Define dimension space for embeddingd_{model}
    for $i = 0$ to X_{final} **do**
        Generate embedding vector for input token x_i
    end for
    return Embedded Matrix E
end function
function Fn$_{encoder}(E)$
    Define number of layers L for transformer encoder
    Initialize embedding operation (eq.6)
    for $i = 1$ to L **do**
        Calculate Q, K, V and Attention score (eq.9)
        Compute value of multi-headed attention (eq.10)
        Perform operation (eq.11)
        Perform multi-layer operation (eq.7) and (eq. 8)
    end for
    return Array of encoded vectors Z
end function
Split input data into training and testing sets
Training:
Initialize model parameters
while not converged **do**
    Sample a batch of training examples ($X_{\text{batch}}, Y_{\text{batch}}$)
    On training data: $features = \text{model}(X_{\text{batch}}, F,)$
    Compute predictions: $\hat{Y} = \text{Sigmoid}(W_{\text{out}} \cdot features + b_{\text{out}})$
    Compute: $loss = \text{binaryCrossEntropy}(Y_{\text{batch}}, \hat{Y})$
    Update model parameters using backpropagation
end while
Testing:
For test data: $features_{test} = \text{model}(X_{\text{test}}, F,)$
Compute predictions for test data: $\hat{Y}_{test} = \text{Sigmoid}(W_{\text{out}} \cdot features_{test} + b_{\text{out}})$
Compute performance metrics

We used 24 transformer layers in our RoBERTa model, with 768 hidden units and 16 self-attention heads in each layer. Each transformer layer has two sub-layers: a multi-head self-attention mechanism and a position-wise fully connected feed-forward dense network. The RoBERTa model uses the MLM, dynamically masks out some of the tokens in a sentence, and trains the model to predict the masked tokens. The input text is truncated or padded to a maximum sequence length of 512 tokens to ensure that all inputs have the same shape.

3) ALGORITHM DESCRIPTION

A detailed step-by-step operation performed for classifying fraudulent jobs is shown in an Algorithm 2. The algorithm begins with the input data sentences X and labels Y, aiming to output metrics such as Accuracy, Specificity, Recall, F1-score, Balanced Accuracy, Sensitivity, and G-Mean. Initially, the embedding sequence is set up and the parameters for a pre-trained BERT/RoBERTa model are initialized.

In the BPE (Byte Pair Encoding) function, the vocabulary V is initialized with characters in X and the end-of-word token ($< /CLS >$). Each word in the tokenized text $X_{\text{tokenized}}$ is appended with $< /CLS >$. The algorithm then repeatedly computes the frequency of all character pairs in V, merges the most frequent pair into a single token, and updates the vocabulary V until it reaches the desired size. The tokenized text is then split into characters and tokenized using the final vocabulary V, resulting in the final tokenized text X_{final}.

The Embedding function defines the vocabulary size V_s and the dimensional space for embedding d_{model}. For each token in X_{final}, an embedding vector is generated, resulting in the embedded matrix E.

In the Fn$_{encoder}$ function, the number of layers L for the transformer encoder is defined. An embedding operation is initialized, and for each layer, the queries (Q), keys (K), values (V), and attention scores are calculated. The multi-headed attention values are then computed, followed by specific operations within the layer and additional multi-layer operations. This results in an array of encoded vectors Z.

The input data is split into training and testing sets. During training, model parameters are initialized, and the algorithm iterates until convergence. In each iteration, a batch of training examples ($X_{\text{batch}}, Y_{\text{batch}}$) is sampled, and features are extracted using the model. Predictions are computed using a sigmoid function, and the binary cross-entropy loss is calculated. Model parameters are then updated using backpropagation.

For testing, features are extracted from the test data using the model, and predictions are computed for the test data using a sigmoid function. Finally, performance metrics are calculated to evaluate the model's performance.

4) IMPLEMENTATION DETAILS

The implementation detail of BERT/RoBERTa models, as done in our case, is given below:

First, the input jobs description is preprocessed to match the required input format expected by the BERT/RoBERTa model. For this purpose, the Byte Pair Encoding (BPE) algorithm is used for tokenizing the text into subwords, and then the subwords are converted to their corresponding numerical representations. Next, a fully connected dense neural network is added on top of the pre-trained BERT/RoBERTa model for the classification task. The flow diagram of the dense network is shown in Fig. 8. A linear layer on top of the final hidden state of the BERT/RoBERTa

model passed the feature vector to the following dense layers for training the classifier to predict whether a job is fraudulent or not.

TABLE 1. Parameter description of the BERT and RoBERTa.

Model	Parameters	Value
BERT/ RoBERTa	No. of transformer layers	12/24
	No. of attention heads	12/16
	Hidden units	768
	Total parameters	110M
	Max sequence length	512
	Vocabulary size	30,000 (default)
	Dropout rate	0.1
Dense Network	No. of Neurons	128
	Hidden layers	02
	Batch size	32
	No. of Epochs	10
	Learning rate	0.001
	Activation function	Sigmoid
	Loss function	Binary cross entropy
	Optimizer	Adam

For fine-tuning the hyperparameters, a default learning rate of 0.001, a batch size of 32, and an early stopping mechanism are used during fine-tuning to avoid overfitting or underfitting. The selected parameter values for the BERT/RoBERTa are adopted from default configurations that are empirically validated for broad applicability in the natural language processing domain. The choice of transformer layers, attention heads, and dropout settings optimizes both convergence and generalization, making these parameters ideal for diverse applications without extensive customization. We fine-tuned the Dense Network model on different hyper-parameters including learning rate, and batch size to analyze the generated results. After running a series of experiments, it has been found that the provided parameters give better performance. So, we have chosen the parameters for BERT/RoBERTA models and Dense network as specified in Table 1.

The pre-trained BERT/RoBERTa model is then fine-tuned on our dataset. During fine-tuning, the weights of the pre-trained model are updated based on the labeled data, while the pre-trained weights are used as a starting point to minimize the loss function on the task-specific labeled dataset. After BERT/RoBERTa model has been fine-tuned on the classification task, it is tested for inference on test data and outputs a predicted label based on the learned representation of the input text.

F. EVALUATION PARAMETERS

Most of the literature we reviewed mainly focused on improving accuracy; however, this may not be enough metric for measuring the model's performance because it does not take incorrectly identified predictions into account. If a fraudulent post is predicted real, it causes a consequential problem. Therefore, it is important to consider false negative and false positive observations into account as well for the compensation of misclassification. We picked out the following evaluation metrics to measure the performance of the proposed models.

1) Accuracy is the ratio of correctly identified predictions by the total number of input instances calculated by using Eq. 12.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{12}$$

where TP represents the number of True Positive samples predicted by the model. Similarly, TN, FP and FN represent the True Negative, False Positive and False Negative samples respectively.

2) Balanced accuracy is a performance measure especially used when the classes involved in a data are imbalanced. For our case, where the data is extremely imbalanced, we considered balanced accuracy as a significant evaluation metric and computed from Eq. 13. We tried to improve balanced accuracy for true evaluation.

$$BalancedAcc. = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{13}$$

3) Sensitivity determines the ability of a model to predict true positive observations of all the involved classes.

4) Specificity determines the ability of a model to predict true negative observations of all the involved classes. Specificity is calculated from the Eq. 14.

$$Specificity = \frac{TN}{TN + FP} \tag{14}$$

5) A recall is a measure of determining all positive samples which is calculated by using Eq. 15. Having high accuracy with low recall indicates the biases of a model. To mitigate biases, we improved recall for all of the experiments performed in this research. In this way, we got a true set of results.

$$Recall = \frac{TP}{(TP + FN)} \tag{15}$$

6) F1-score is the harmonic mean of precision and recall as given in Eq. 16.

$$F1 - score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}, \tag{16}$$

7) The Geometric mean (G-mean) determines the balancing factor between the majority and minority classes. It is also another important metric in a case when given classes are highly imbalanced. So, in our case, the value of the G-mean is very considerable; therefore, we tried to improve G-mean for all of the experiments we performed as computed using Eq. 17.

$$G - mean = \sqrt{Sensitivity \times Specificity} \tag{17}$$

IV. EXPERIMENTAL RESULTS

This section is divided into two parts. The first part contains type error analysis performed to study the impact of using SMOTE variants on the predictive models. Transformer-based classification models, i.e., BERT and RoBERTa, were

TABLE 2. Confusion metrics of BERT+SMOTE variants.

Methodology	Actual Real vs. Predicted Real	Actual Real vs. Predicted Fake	Actual Fake vs. Predicted Real	Actual Fake vs. Predicted Fake
BERT + Actual Data	10827 99.79%	22 0.21%	**152** **68.77%**	69 31.23%
BERT + Polynom fit SMOTE	10586 97.57%	263 2.43%	57 25.79%	164 74.21%
BERT + Assembled SMOTE	10643 98.10%	206 1.90%	46 20.81%	175 79.19%
BERT + LVQ SMOTE	10802 99.56%	47 0.44%	114 51.58%	107 48.42%
BERT + CCR SMOTE	10808 99.62%	41 0.38%	117 52.94%	104 47.06%
BERT + SMOBD SMOTE	10560 97.33%	289 2.67%	**39** **17.64%**	182 82.36%
BERT + ProWSyn SMOTE	10515 96.92%	334 3.08%	54 24.43%	167 75.57%
BERT + Lee SMOTE	10820 99.73%	29 0.27%	108 48.86%	113 51.14%
BERT + G SMOTE	10637 98.04%	212 1.96%	47 21.26%	174 78.74%
BERT + SMOTE Tomeklinks	10590 97.61%	259 2.39%	46 20.81%	175 79.19%
BERT + SMOTE IPF	10742 99.01%	107 0.99%	79 35.74%	142 64.26%

implemented on imbalanced and balanced data, and the achieved results are compared in the second part. Different evaluation metrics, as discussed in the previous section, have been used to measure the performance of implemented frameworks. All implemented approaches showed up to mark performances.

A. TYPE ERROR ANALYSIS

We have conducted an analysis of type errors to investigate the impact of employing different SMOTE oversampling techniques on predictive models. Our analysis involved a comparison with the performance of models that do not utilize any SMOTE variant. In this context, two distinct types of errors may arise: Type I and Type II. Type I errors occur when a genuine job posting (representing the majority class) is mistakenly classified as a fake job posting (representing the minority class). Type I errors are generally less consequential for job seekers. Conversely, Type II errors arise when a fake job posting (minority class) is erroneously classified as a real job posting (majority class). Type II errors present a greater challenge for us, as considering any fake job posting as real can lead to numerous significant problems as discussed in Section I. Hence, our primary focus was on mitigating Type II errors.

Table 2 represents that the Type II error rate produced on BERT+actual data was 68.77% as it predicts 152 incorrect samples against 221 fake job samples. After implementing SMOTE variants, we observed that the Type II error is reduced greatly in almost all of the implemented combinations. The Type II error rate obtained by BERT+Polynom fit SMOTE is reduced to 25.79% as it predicts only 57 fake job postings as real ones out of 221 fake job postings. The Type II error rate obtained by BERT+Assembled SMOTE is reduced to 20.81% as it predicts only 46 fake job postings as real ones. The Type II error rate obtained by BERT+LVQ SMOTE is reduced to 51.58% as it predicts only 114 fake job postings as real ones. The Type II error rate obtained by BERT+CCR SMOTE is reduced to 52.94% as it predicts only 117 fake job postings as real ones. The Type II error rate obtained by BERT+SMOBD SMOTE is reduced to 17.64% as it predicts only 39 fake job postings as real ones. The Type II error rate obtained by BERT+ProWSyn SMOTE is reduced to 24.43% as it predicts only 54 fake job postings as real ones. The Type II error rate obtained by BERT+Lee SMOTE is reduced to 48.86% as it predicts only 108 fake job postings as real ones. The Type II error rate obtained by BERT+G SMOTE is reduced to 21.26% as it predicts only 47 fake job postings as real ones. The Type II error rate obtained by BERT+SMOTE TomekLinks is reduced to 20.81% as it predicts only 46 fake

TABLE 3. Confusion matrices of RoBERTa with various SMOTE variants.

Methodology	Actual Real vs. Predicted Real	Actual Real vs. Predicted Fake	Actual Fake vs. Predicted Real	Actual Fake vs. Predicted Fake
RoBERTa + Actual Data	10849 100%	0	**221** **100%**	0
RoBERTa + Polynom fit SMOTE	9892 91.17%	957 8.83%	75 33.93%	146 66.07%
RoBERTa + Assembled SMOTE	8546 78.77%	2303 21.23%	35 15.83%	186 84.17%
RoBERTa + LVQ SMOTE	10342 95.32%	507 4.68%	145 65.61%	76 34.39%
RoBERTa + CCR SMOTE	9021 83.15%	1828 16.85%	86 38.91%	135 61.09%
RoBERTa + SMOBD SMOTE	8974 82.71%	1875 17.29%	40 18.09%	181 81.91%
RoBERTa + ProWSyn SMOTE	9703 89.43%	1146 10.57%	63 28.50%	158 71.50%
RoBERTa + Lee SMOTE	9607 88.57%	1240 11.43%	57 25.79%	164 74.21%
RoBERTa + G SMOTE	8820 81.29%	2029 18.71%	**35** **15.83%**	186 84.17%
RoBERTa + SMOTE Tomeklinks	9160 84.43%	1689 15.57%	51 23.07%	170 76.93%
RoBERTa + SMOTE IPF	8550 78.80%	2299 21.20%	36 15.85%	187 84.62%

job postings as real ones. The Type II error rate obtained by BERT+SMOTE IPF is reduced to 35.74% as it predicts only 79 fake job postings as real ones.

Table 3 represents that the Type II error rate produced on RoBERTa+actual data was 100% as it predicts all the samples incorrectly. After implementing SMOTE variants, we observed that the Type II error is reduced greatly in almost all of the implemented combinations. The Type II error rate obtained by RoBERTa+Polynom fit SMOTE is reduced to 33.93% as it predicts only 75 fake job postings as real ones out of 221 fake job postings. The Type II error rate obtained by RoBERTa+Assembled SMOTE is reduced to 15.83% as it predicts only 35 fake job postings as real ones. The Type II error rate obtained by RoBERTa+LVQ SMOTE is reduced to 65.61% as it predicts only 145 fake job postings as real ones. The Type II error rate obtained by RoBERTa+CCR SMOTE is reduced to 38.91% as it predicts only 86 fake job postings as real. The Type II error rate obtained by RoBERTa+SMOBD SMOTE is reduced to 18.09% as it predicts only 40 fake job postings as real ones. The Type II error rate obtained by RoBERTa+ProWSyn SMOTE is reduced to 28.50% as it predicts only 63 fake job postings as real ones.

The Type II error rate obtained by RoBERTa+Lee SMOTE is reduced to 25.79% as it predicts 57 fake job postings as real ones. The Type II error rate obtained by RoBERTa+G SMOTE is reduced to 15.83% as it predicts only 35 fake job postings as real ones. The Type II error rate obtained by RoBERTa+SMOTE TomekLinks is reduced to 23.07% as it predicts only 51 fake job postings as real ones. The Type II error rate obtained by RoBERTa+SMOTE IPF is reduced to 15.85% as it predicts only 36 fake job postings as real ones.

After performing type error analysis, it has been noticed that Type II error rates have been reduced on all implemented combinations. However, the error rate obtained by BERT+SMOBD SMOTE is significantly reduced from 68.77% to 17.64% due to the working mechanism of SMOBD that it considers density and distribution of those data points that are so close to the actual distribution of the data, for the synthesis of new data points. This empowers the SMOBD SMOTE to better generate the synthesized data points to balance the class distribution. On the other hand, the error rate obtained by RoBERTa+G SMOTE is significantly reduced from 100% to 15.38% due to the reason that G SMOTE defines a safe area to ensure that no noisy data instance is synthesized. Its objective is to generate diverse minority class instances to prevent intra-cluster skewness. This mechanism empowers G SMOTE to balance class distribution well. Hence, the values obtained after conducting type error analysis confirm the effectiveness of the implemented approaches.

TABLE 4. Performance comparison chart of BERT+SMOTE variants.

Methodology	Accuracy	Bal. Acc.	Recall	Sensitivity	Specificity	F-score	G-mean
BERT + Actual Data	**98.42%**	65.50%	**65.50%**	99.79%	31.22%	67.47%	55.81%
BERT + Polynom fit SMOTE	97.10%	85.89%	85.89%	97.57%	74.20%	80.24%	85.09%
BERT + Assembled SMOTE	97.72%	88.64%	88.64%	98.10%	79.18%	83.78%	88.13%
BERT + LVQ SMOTE	98.54%	73.99%	73.99%	99.56%	48.41%	75.49%	69.43%
BERT + CCR SMOTE	98.57%	73.34%	73.34%	99.62%	47.05%	75%	68.46%
BERT + SMOBD SMOTE	**97.03%**	89.84%	**89.84%**	97.33%	82.35%	82.47%	89.53%
BERT + ProWSyn SMOTE	96.49%	86.24%	86.24%	96.92%	75.56%	78.85%	85.57%
BERT + Lee SMOTE	98.76%	75.43%	75.43%	99.73%	51.13%	77.32%	71.41%
BERT + G SMOTE	97.66%	88.38%	88.38%	98.04%	78.73%	83.42%	87.86%
BERT + TomekLinks SMOTE	97.24%	88.39%	88.39%	97.61%	79.18%	82.19%	87.91%
BERT + SMOTE IPF	98.31%	81.63%	81.63%	99.01%	64.25%	80.86%	79.76%

TABLE 5. Performance comparison chart of RoBERTa+SMOTE variants.

Methodology	Accuracy	Bal. Acc.	Recall	Sensitivity	Specificity	F-score	G-mean
RoBERTa + Actual Data	**98%**	50%	**50%**	100%	0%	49.79%	0%
RoBERTa + Polynom fit SMOTE	90.67%	78.62%	78.62%	91.17%	66.06%	64.71%	77.61%
RoBERTa + Assembled SMOTE	78.87%	81.46%	81.46%	78.77%	84.16%	54.89%	81.42%
RoBERTa + LVQ SMOTE	94.11%	64.85%	64.85%	95.32%	34.38%	60.93%	57.25%
RoBERTa + CCR SMOTE	82.71%	72.11%	72.11%	83.15%	61.08%	54.80%	71.26%
RoBERTa + SMOBD SMOTE	82.70%	82.30%	82.30%	82.71%	81.90%	58.19%	82.30%
RoBERTa + ProWSyn SMOTE	89.07%	80.46%	80.46%	89.43%	71.49%	63.68%	79.96%
RoBERTa + Lee SMOTE	88.28%	81.38%	81.38%	88.57%	74.20%	63.19%	81.07%
RoBERTa + G SMOTE	**81.35%**	82.73%	**82.73%**	81.29%	84.16%	57.20%	82.71%
RoBERTa + TomekLinks SMOTE	84.28%	80.67%	80.67%	84.43%	76.92%	59.02%	80.59%
RoBERTa + SMOTE IPF	78.92%	81.71%	81.71%	78.80%	84.61%	54.99%	81.66%

B. MODELS CLASSIFICATION RESULTS

This section presents the classification outcomes attained through the application of each implemented approach. Notably, when BERT and RoBERTa models were employed on actual data, they demonstrated remarkable accuracies of 98.42% and 98%, respectively. However, it is crucial to note the significantly low recall values associated with these accuracies, namely 65.50% and 50%, respectively. The occurrence of high accuracies alongside poor recall represents class biasness, rendering the accuracies misleading. In order to obtain genuine and unbiased results, the data was subsequently balanced using various SMOTE variants.

Table 4 and Table 5 shows that when data is balanced by using Polynom fit SMOTE, accuracies of BERT and RoBERTa are decreased to 97.10% and 90.67%, but recall is greatly increased up to 85.89% and 78.62%, respectively. When data is balanced by using Assembled SMOTE, accuracies of BERT and RoBERTa are decreased to 97.72% and 78.87%, but recall is greatly increased up to 88.64% and 81.46%, respectively. BERT and RoBERTa with LVQ SMOTE achieved accuracies of 98.54% and 94.11%, with recall values of 73.99% and 64.85%, respectively. BERT and RoBERTa with CCR SMOTE achieved accuracies of 98.57% and 82.71%, with recall values of 73.34% and 72.11%, respectively. When data is balanced by using SMOBD SMOTE, accuracies of BERT and RoBERTa are decreased to 97.03% and 82.70%, but recall is greatly increased up to 89.84% and 82.30%, respectively. When data is balanced by using ProWSyn SMOTE, accuracies of BERT and RoBERTa

TABLE 6. Performance comparison chart of CNN+SMOTE variants.

Methodology	Accuracy	Bal. Acc.	Recall	Sensitivity	Specificity	F-score	G-mean
CNN + Actual Data	95.62%	58.63%	58.70%	96.77%	48.38%	61.01%	59.17%
CNN + Polynom fit SMOTE	95.78%	83.17%	83.18%	96.81%	79.89%	83.76%	73.97%
CNN + Assembled SMOTE	96.22%	86.48%	86.49%	97.12%	83.34%	86.78%	89.21%
CNN + LVQ SMOTE	96.46%	84.89%	84.89%	98.43%	54.13%	80.19%	71.39%
CNN + CCR SMOTE	96.17%	69.94%	69.94%	99.53%	51.02%	73.67%	66.41%
CNN + SMOBD SMOTE	93.23%	86.63%	86.63%	98.13%	79.66%	80.16%	88.27%
CNN + ProWSyn SMOTE	95.29%	84.12%	84.12%	97.21%	78.26%	79.95%	88.15%
CNN + Lee SMOTE	91.86%	70.16%	70.16%	98.96%	53.49%	73.13%	76.21%
CNN + G SMOTE	95.47%	86.97%	87.13%	98.53%	79.83%	85.51%	86.16%
CNN + TomekLinks SMOTE	96.36%	86.53%	86.63%	96.31%	81.53%	84.19%	85.65%
CNN + SMOTE IPF	96.11%	84.53%	84.53%	98.45%	76.25%	86.37%	83.25%

TABLE 7. Performance comparison chart of RNN+SMOTE variants.

Methodology	Accuracy	Bal. Acc.	Recall	Sensitivity	Specificity	F-score	G-mean
RNN + Actual Data	89.21%	55.23%	55.23%	98.89%	36.41%	61.17%	53.37%
RNN + Polynom fit SMOTE	93.51%	81.25%	81.25%	95.23%	78.12%	81.01%	79.41%
RNN + Assembled SMOTE	95.23%	84.17%	84.23%	95.85%	71.87%	78.63%	80.15%
RNN + LVQ SMOTE	96.13%	80.09%	80.09%	99.56%	45.89%	76.27%	73.43%
RNN + CCR SMOTE	96.37%	78.35%	78.36%	98.92%	51.23%	78.01%	73.96%
RNN + SMOBD SMOTE	96.03%	89.54%	89.54%	98.43%	83.87%	80.17%	85.13%
RNN + ProWSyn SMOTE	96.59%	89.94%	89.01%	98.32%	79.85%	80.25%	84.97%
RNN + Lee SMOTE	96.27%	73.18%	73.20%	98.83%	56.23%	78.12%	74.24%
RNN + G SMOTE	96.26%	89.53%	89.53%	97.84%	79.93%	84.14%	88.76%
RNN + TomekLinks SMOTE	95.53%	89.49%	89.49%	98.67%	81.21%	84.11%	85.99%
RNN + SMOTE IPF	96.51%	85.13%	85.13%	99.31%	69.25%	81.93%	81.79%

are decreased to 96.49% and 89.07%, but recall is greatly increased up to 86.24% and 80.46%, respectively. BERT and RoBERTa with Lee SMOTE achieved accuracies of 98.76% and 88.28%, with recall values of 75.43% and 81.38%, respectively. BERT and RoBERTa with G SMOTE achieved accuracies of 97.66% and 81.35%, with recall values of 88.38% and 82.73%, respectively. When data is balanced by using SMOTE TomekLinks, accuracies of BERT and RoBERTa are decreased to 97.24% and 84.28%, but recall is greatly increased up to 88.39% and 80.67%, respectively. BERT and RoBERTa with SMOTE IPF achieved accuracies of 98.31% and 78.92%, with an increased recall of 81.63% and 81.71%, respectively. Fig. 9 and Fig. 10 show the graphical representation of combined evaluation metrics of BERT and RoBERTa with various implemented SMOTE variants.

Considering all observations mentioned earlier, it is stated that among BERT's combinations, BERT+SMOBD SMOTE achieved the highest recall of 90%, which was just 65.50% for BERT+actual data. Similarly, among RoBERTa's combinations, RoBERTa+G SMOTE achieved the highest recall of 82.73%, which was just 50% for RoBERTa+actual data. By looking at overall results, BERT was considered to achieve the optimal results with the data balanced by SMOBD SMOTE. This is due to the reason that BERT is pre-trained for the next sentence prediction objective; it captures the context and semantics of job postings bi-directionally to give the best results.

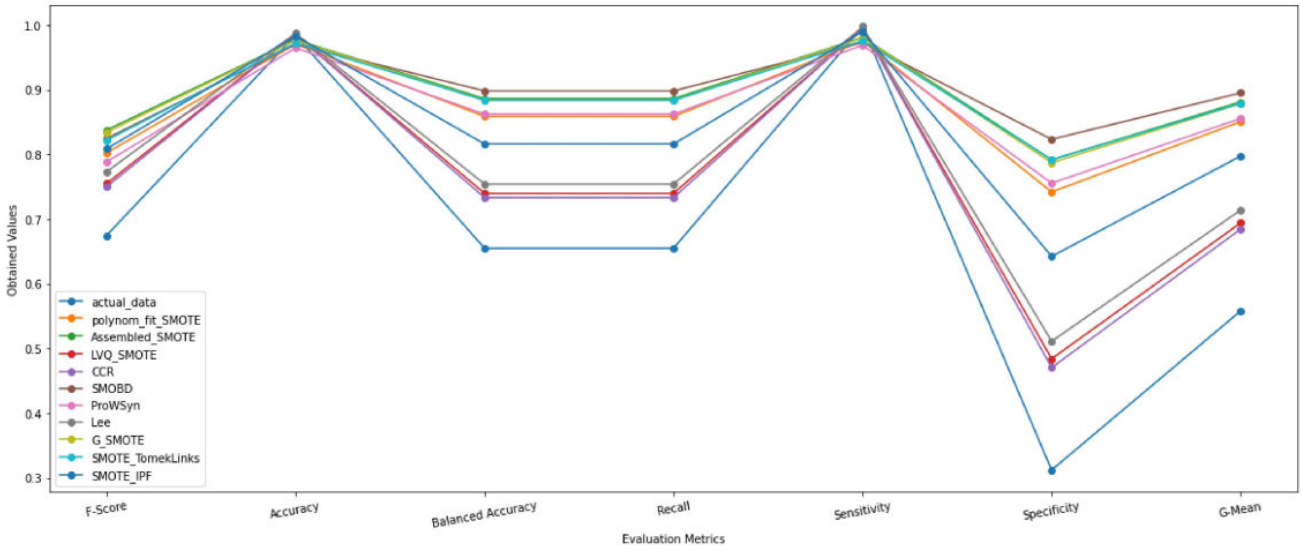


FIGURE 9. Combined evaluation metrics graph of BERT+SMOTE variants.

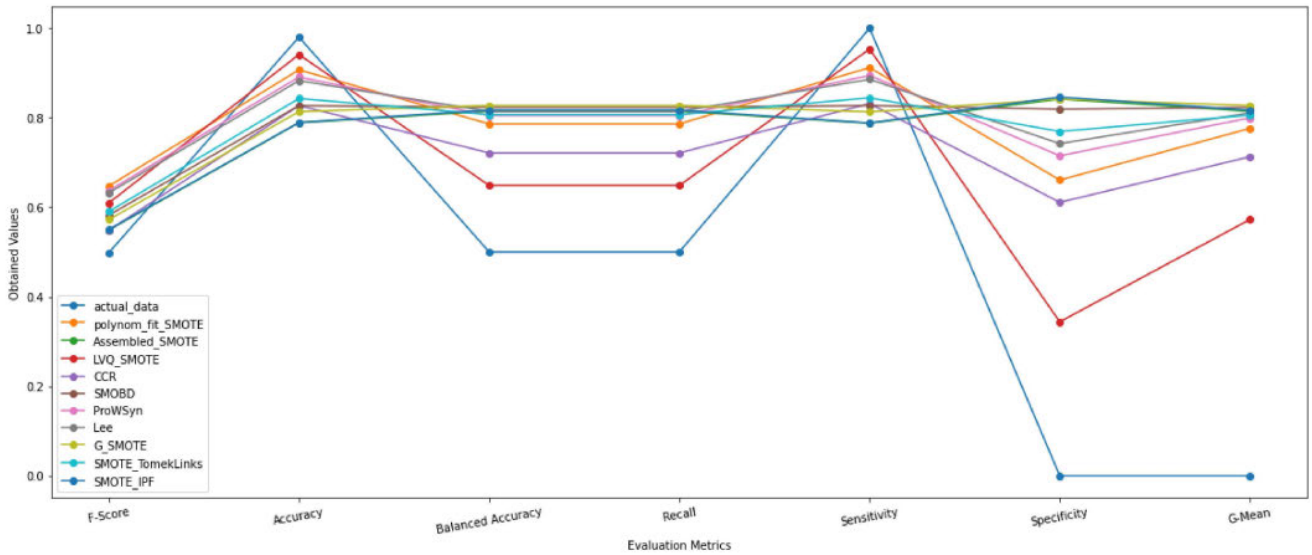


FIGURE 10. Combined evaluation metrics graph of RoBERTa+SMOTE variants.

C. CRITICAL REVIEW DISCUSSION

While the initial results showcased high accuracies, they were paired with poor recall values, indicating significant class bias. This makes the initial high accuracy rates deceptive, as they do not truly reflect the models' performance across all classes. The use of different SMOTE variants substantially improved recall values, showcasing a more balanced and fair performance of the models. The application of SMOTE variants significantly improved recall values, indicating better handling of class imbalance. BERT and RoBERTa, when combined with specific SMOTE techniques, managed to maintain relatively high accuracies while also improving recall, striking a better balance between precision and sensitivity. Despite improvements, some SMOTE variants led to a notable decrease in accuracy, particularly for RoBERTa. This trade-off highlights the challenge of achieving both high accuracy and high recall. Certain SMOTE techniques still resulted in recall values that, while improved, were not as high as desired, indicating room for further optimization.

The results highlight the importance of addressing the class imbalance issue in predictive modeling. High accuracy alone is insufficient if recall is low, as it suggests the model is biased towards the majority class. Balancing techniques like SMOTE are essential to ensure that models perform

well across all classes, providing more reliable and unbiased results.

D. COMPARATIVE ANALYSIS

The comparative analysis of different deep learning models (RNN, CNN, RoBERTa, and BERT) using various SMOTE variants for ORF detection reveals notable differences across performance metrics. Table 6 displays the performance of various CNN models using different SMOTE variants for Online Recruitment Fraud detection. The results indicate that all SMOTE variants enhance model performance compared to using actual data alone, which shows the lowest balanced accuracy and specificity. Among the SMOTE variants, CNN + Assembled SMOTE and CNN + G SMOTE stand out, with CNN + Assembled SMOTE achieving the highest G-mean of 89.21% and CNN + G SMOTE showing impressive improvements across balanced accuracy, recall, and F-score, notably the highest F-score at 85.51%. These enhancements suggest that Assembled SMOTE and G SMOTE are particularly effective for improving the predictive performance of CNN models in this context.

Table 7 represents the performance of various RNN models utilizing different SMOTE variants. All SMOTE variants enhanced the RNN model performance compared to the baseline (actual data), which has the lowest scores in several metrics. The RNN + ProWsyn SMOTE variant stands out with the highest F-score 80.25% and a very high recall of 89.01%, suggesting its strength in minimizing false negatives. Additionally, RNN + SMODB SMOTE shows the best overall balance with the highest G-mean 85.13% and impressive balanced accuracy 89.54%, indicating robust detection capabilities across various aspects of the model's performance. These results highlight SMODB SMOTE and ProWsyn SMOTE as particularly effective for improving the accuracy and balanced performance of RNN models in fraud detection tasks.

RNN + ProWsyn SMOTE and RNN + SMODB SMOTE excel in balancing detection capabilities, showing high G-mean and F-score. CNN models, particularly with Tomek-Links SMOTE and Assembled SMOTE, achieve significant improvements in accuracy and G-mean. RoBERTa and BERT models display diverse effectiveness; RoBERTa + Polynom fit SMOTE and BERT + CCR SMOTE offer high accuracy, though with trade-offs in sensitivity and specificity. BERT + ProWsyn SMOTE, in particular, demonstrates robust performance across balanced accuracy and F-score. Overall, these results indicate that specific SMOTE variants can greatly enhance the performance of deep learning models in fraud detection tasks, with some combinations like BERT + ProWsyn SMOTE providing particularly balanced and effective outcomes.

In comparison with other deep learning architectures for ORF detection, CNNs are adept at image classification but may miss temporal dependencies. The RNNs, especially Long Short-Term Memory (LSTM) networks, capture sequential patterns but struggle with longer contexts. Transformer-based models excel in understanding long-range dependencies, crucial for ORF detection. However, CNN and RNN models are implemented for performance comparison with BERT/RoBERTa models.

V. CONCLUSION AND FUTURE WORK

In this research, the problem of ORF detection is analyzed thoroughly. This paper presented a novel dataset of fake job postings. The proposed data is a combination of job postings from three different sources. Upon conducting EDA, it was discovered that the class distribution within the collected dataset was highly imbalanced. To rectify this class distribution imbalance, the top ten highly effective SMOTE variants were implemented on the imbalanced data. Subsequently, a type error analysis was conducted to investigate the impact of employing SMOTE variants on predictive models. Transformer-based classification models, BERT and RoBERTa, were implemented on both the imbalanced and balanced data, and the results were compared to derive more comprehensive insights from the experiments. Diverse evaluation metrics were employed to compare the performance of the implemented techniques. Due to the class imbalance issue, only accuracy as an evaluation metric failed to provide an accurate representation of the overall performance. Because high predictive accuracy for the majority class can be misleading, as it may overshadow the minority class, leading to incomplete assessment. Thus, this study prioritized enhancing balanced accuracy and recall as evaluation metrics. All implemented approaches exhibited commendable performance. However, based on the type error and classification results, it was observed that BERT, in conjunction with the SMOBD SMOTE technique, demonstrated exceptional performance on our data and achieved optimal outcomes.

The experiments performed in this research can provide valuable directions to job-seekers and reputed organizations to better understand fact-based insights about employment scam and their effects on society. Consequently, people would not fall into the trap of employment scams anymore. By distinguishing ORF, the people who were wasting their time and money on those fraudulent activities can be vigilant now. Conventional fraud detection without considering class imbalance problems can lead to misleading conclusions for both job-seekers and organizations. To get a true set of results, it is necessary to handle this problem as well. In this research, we extensively improved the system's performance and gained valuable results based on balanced data; still, it has many gaps that can be covered in the future. All sets of analyses are performed on the job postings advertised in the English language only.

For a more comprehensive examination, it is possible to conduct a similar analysis on job postings published in languages other than the current dataset. Given the rising popularity of online recruitment, the dataset can be enriched by incorporating the latest job postings. This study encompasses

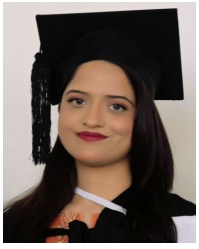
job postings from diverse regions worldwide. A comparable analysis can be conducted specifically for job postings within a particular region to ascertain the rate of fraudulent postings, thereby serving the public's best interests. Furthermore, the inclusion of job postings that pertain to remote work opportunities through online platforms can be deemed crucial for the creation of a novel dataset, given the significant prevalence of fraudulent activities associated with online jobs from home.

In the present research, a range of SMOTE variants were employed to address class distribution imbalance. To attain even more precise results, the utilization of hybrid oversampling techniques can be considered. For future research, explainable AI and novel transformer based hybrid models need to be explored.

REFERENCES

[1] P. Kaur, "E-recruitment: A conceptual study," *Int. J. Appl. Res.*, vol. 1, no. 8, pp. 78–82, 2015.
[2] C. S. Anita, P. Nagarajan, G. A. Sairam, P. Ganesh, and G. Deepakkumar, "Fake job detection and analysis using machine learning and deep learning algorithms," *Revista Gestão Inovação e Tecnologias*, vol. 11, no. 2, pp. 642–650, Jun. 2021.
[3] A. Raza, S. Ubaid, F. Younas, and F. Akhtar, "Fake e job posting prediction based on advance machine learning approachs," *Int. J. Res. Publication Rev.*, vol. 3, no. 2, pp. 689–695, Feb. 2022.
[4] *Online Fraud*. Accessed: Jun. 19, 2022. [Online]. Available: https://www.cyber.gov.au/acsc/report
[5] J. Howington, "Survey: More millennials than seniors victims of job scams," Flexjobs, CO, USA, Sep. 2015. Accessed: Jan. 2024 [Online]. Available: www.flexjobs.com/blog/post/survey-results-millennials-seniors-victims-job-scams
[6] *Report Cyber*. Accessed: Jun. 25, 2022. [Online]. Available: https://www.actionfraud.police.uk/
[7] S. Vidros, C. Kolias, G. Kambourakis, and L. Akoglu, "Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset," *Future Internet*, vol. 9, no. 1, p. 6, Mar. 2017.
[8] S. Dutta and S. K. Bandyopadhyay, "Fake job recruitment detection using machine learning approach," *Int. J. Eng. Trends Technol.*, vol. 68, no. 4, pp. 48–53, Apr. 2020.
[9] B. Alghamdi and F. Alharby, "An intelligent model for online recruitment fraud detection," *J. Inf. Secur.*, vol. 10, no. 3, pp. 155–176, 2019.
[10] S. Lal, R. Jiaswal, N. Sardana, A. Verma, A. Kaur, and R. Mourya, "ORFDetector: Ensemble learning based online recruitment fraud detection," in *Proc. 12th Int. Conf. Contemp. Comput. (IC3)*, Noida, India, Aug. 2019, pp. 1–5.
[11] I. M. Nasser, A. H. Alzaanin, and A. Y. Maghari, "Online recruitment fraud detection using ANN," in *Proc. Palestinian Int. Conf. Inf. Commun. Technol. (PICICT)*, Sep. 2021, pp. 13–17.
[12] C. Lokku, "Classification of genuinity in job posting using machine learning," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. 12, pp. 1569–1575, Dec. 2021.
[13] O. Nindyati and I. G. Bagus Baskara Nugraha, "Detecting scam in online job vacancy using behavioral features extraction," in *Proc. Int. Conf. ICT Smart Soc. (ICISS)*, vol. 7, Bandung, Indonesia, Nov. 2019, pp. 1–4.
[14] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS Int. Trans. Comput. Sci. Eng.*, vol. 30, no. 1, pp. 25–36, 2006.
[15] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani, "Toward credible evaluation of anomaly-based intrusion-detection methods," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 5, pp. 516–524, Sep. 2010.
[16] Y.-H. Liu and Y.-T. Chen, "Total margin based adaptive fuzzy support vector machines for multiview face recognition," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Waikoloa, HI, USA, Oct. 2005, pp. 1704–1711.
[17] M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi, "Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance," *Neural Netw.*, vol. 21, nos. 2–3, pp. 427–436, Mar. 2008.
[18] Y. Li, G. Sun, and Y. Zhu, "Data imbalance problem in text classification," in *Proc. 3rd Int. Symp. Inf. Process.*, Luxor, Egypt, Oct. 2010, pp. 301–305.
[19] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[20] S. U. Habiba, Md. K. Islam, and F. Tasnim, "A comparative study on fake job post prediction using different data mining techniques," in *Proc. 2nd Int. Conf. Robot., Electr. Signal Process. Techn. (ICREST)*, Dhaka, Bangladesh, Jan. 2021, pp. 543–546.
[21] G. Othman Alandjani, "Online fake job advertisement recognition and classification using machine learning," *3C TIC, Cuadernos de Desarrollo Aplicados a las TIC*, vol. 11, no. 1, pp. 251–267, Jun. 2022.
[22] A. Gosain and S. Sardana, "Handling class imbalance problem using oversampling techniques: A review," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Delhi, India, Sep. 2017, pp. 79–85.
[23] F. Akhbardeh, C. O. Alm, M. Zampieri, and T. Desell, "Handling extreme class imbalance in technical logbook datasets," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4034–4045.
[24] J. Ah-Pine and E.-P. Soriano-Morales, "A study of synthetic oversampling for Twitter imbalanced sentiment analysis," in *Proc. Workshop Interact. Between Data Min. Nat. Lang. Process. (DMNLP)*, Riva del Garda, Italy, Sep. 2016, pp. 17–24.
[25] J. David, J. Cui, and F. Rahimi, "Classification of imbalanced dataset using BERT embeddings," Dalhousie Univ., Halifax, Canada, Jan. 2020. Accessed: Jan. 2024. [Online]. Available: https://fatemerhmi.github.io/files/Classification_of_imbalanced_dataset_using_BERT_embedding.pdf
[26] Kanika, J. Singla, A. Kashif Bashir, Y. Nam, N. UI Hasan, and U. Tariq, "Handling class imbalance in online transaction fraud detection," *Comput., Mater. Continua*, vol. 70, no. 2, pp. 2861–2877, 2022.
[27] S. Bansal. (2020). *[Real or Fake] Fake Jobposting Prediction*. [Online]. Available: https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction
[28] *Indeed Job Posting Dataset*. Accessed: Feb. 16, 2023. [Online]. Available: https://www.kaggle.com/datasets/promptcloud/indeed-job-posting-dataset
[29] *Pakistan's Job Market*. Accessed: Feb. 16, 2023. [Online]. Available: https://www.kaggle.com/datasets/zusmani/pakistans-job-market
[30] C. Nunes, D. Silva, M. Guerreiro, A. de Mendonça, A. M. Carvalho, and S. C. Madeira, "Class imbalance in the prediction of dementia from neuropsychological data," in *Proc. 16th Portug. Conf. Artif. Intell. (EPIA)*, Angra do Heroísmo, Portugal, Sep. 2013, pp. 138–151.
[31] J. Stefanowski and S. Wilk, "Selective pre-processing of imbalanced data for improving classification performance," in *Proc. 10th Int. Conf. Data Warehousing Knowl. Disc. (DaWaK)*, Turin, Italy, Sep. 2008, pp. 283–292.
[32] A. Adhikari, A. Ram, R. Tang, and J. Lin, "DocBERT: BERT for document classification," 2019, *arXiv:1904.08398*.
[33] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105662.
[34] S. Gazzah and N. E. B. Amara, "New oversampling approaches based on polynomial fitting for imbalanced data sets," in *Proc. 8th IAPR Int. Workshop Document Anal. Syst.*, Nara, Japan, Sep. 2008, pp. 677–684.
[35] S. Barua, M. M. Islam, and K. Murase, "ProWSyn: Proximity weighted synthetic oversampling technique for imbalanced data set learning," in *Proc. Pacific–Asia Conf. Knowl. Disc. Data Min. II*, Gold Coast, QLD, Australia, Apr. 2013, pp. 317–328.
[36] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, "SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering," *Inf. Sci.*, vol. 291, pp. 184–203, Jan. 2015.
[37] J. Lee, N.-R. Kim, and J.-H. Lee, "An over-sampling technique with rejection for imbalanced class learning," in *Proc. 9th Int. Conf. Ubiquitous Inf. Manag. Commun.*, New York, NY, USA, Jan. 2015, pp. 1–6.
[38] Q. Cao and S. Wang, "Applying over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning," in *Proc. Int. Conf. Inf. Manage., Innov. Manage. Ind. Eng.*, vol. 2, Shenzhen, China, Nov. 2011, pp. 543–548.
[39] T. Sandhan and J. Y. Choi, "Handling imbalanced datasets by partially guided hybrid sampling for pattern recognition," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Stockholm, Sweden, Aug. 2014, pp. 1449–1453.

[40] M. Koziarski and M. Woźniak, "CCR: A combined cleaning and resampling algorithm for imbalanced data classification," *Int. J. Appl. Math. Comput. Sci.*, vol. 27, no. 4, pp. 727–736, Dec. 2017.
[41] M. Nakamura, Y. Kajiwara, A. Otsuka, and H. Kimura, "LVQ-SMOTE—Learning vector quantization based synthetic minority over–sampling technique for biomedical data," *BioData Mining*, vol. 6, no. 1, pp. 1–10, Dec. 2013.
[42] B. Zhou, C. Yang, H. Guo, and J. Hu, "A quasi-linear SVM combined with assembled SMOTE for imbalanced data classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Anchorage, AK, USA, Jul. 2013, pp. 1–7.
[43] M. Zeng, B. Zou, F. Wei, X. Liu, and L. Wang, "Effective prediction of three common diseases by combining SMOTE with tomek links technique for imbalanced medical data," in *Proc. IEEE Int. Conf. Online Anal. Comput. Sci. (ICOACS)*, Chongqing, China, May 2016, pp. 225–228.
[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[45] Z. Xinxi, "Single task fine-tune BERT for text classification," in *Proc. 2nd Int. Conf. Comput. Vis., Image, Deep Learn.*, Kunming, China, Oct. 2021, pp. 194–206.
[46] S. Zheng and M. Yang, "A new method of improving BERT for text classification," in *Proc. 9th Int. Conf. Intell. Sci. Big Data Eng. (IScIDE)*, Nanjing, China, Oct. 2019, pp. 442–452.
[47] S. González-Carvajal and E. C. Garrido-Merchán, "Comparing BERT against traditional machine learning text classification," 2020, *arXiv:2005.13012*.
[48] Y. Hu, J. Ding, Z. Dou, and H. Chang, "Short-text classification detector: A BERT-based mental approach," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–11, Mar. 2022.
[49] H. Xu, B. Liu, L. Shu, and P. S. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis," 2019, *arXiv:1904.02232*.
[50] M. Hoang, O. A. Bihorac, and J. Rouces, "Aspect-based sentiment analysis using BERT," in *Proc. 22nd Nordic Conf. Comput. Linguistics*, Turku, Finland, Sep. 2019, pp. 187–196.
[51] D. Miller, "Leveraging BERT for extractive text summarization on lectures," 2019, *arXiv:1906.04165*.
[52] Z. Yang, N. Garcia, C. Chu, M. Otani, Y. Nakashima, and H. Takemura, "BERT representations for video question answering," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1545–1554.
[53] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," 2019, *arXiv:1904.09675*.
[54] Y.-C. Chen, Z. Gan, Y. Cheng, J. Liu, and J. Liu, "Distilling knowledge learned in BERT for text generation," 2019, *arXiv:1911.03829*.
[55] R. Guan, H. Zhang, Y. Liang, F. Giunchiglia, L. Huang, and X. Feng, "Deep feature-based text clustering and its explanation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3669–3680, Aug. 2022.
[56] N. Peinelt, D. Nguyen, and M. Liakata, "TBERT: Topic models and BERT joining forces for semantic similarity detection," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7047–7055.
[57] J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, and T.-Y. Liu, "Incorporating BERT into neural machine translation," 2020, *arXiv:2002.06823*.
[58] H. Tsai, J. Riesa, M. Johnson, N. Arivazhagan, X. Li, and A. Archer, "Small and practical BERT models for sequence labeling," 2019, *arXiv:1909.00100*.
[59] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.



NATASHA AKRAM received the bachelor's degree in computer science from GCU, Lahore, and the master's degree in computer science from NUST, Islamabad, in 2023. Her current research interests include natural language processing, big data analysis, machine learning, and distributed systems.

RABIA IRFAN received the bachelor's degree in computer and information systems engineering from the NED University of Engineering and Technology, Karachi, and the Ph.D. degree in information technology from NUST, in 2018. She is currently an Assistant Professor with NUST. She is heading the Knowledge-Based System (KBS) Laboratory, SEECS, NUST. With the foundation of the engineering domain and the applied side of information technology, her research interests include incorporating analytics and decision-making capabilities in the solution of real-world problems, such as the domains of health, energy, and power. Being an Active Researcher, she believes that the identification of solutions to real-world problems involves interdisciplinary domains and techniques and she is always open to exploring new areas and domains. She also has more than five years of industrial experience. Her specialization is in the area of data science and machine learning. She was awarded the Presidential Gold Medal for the master's degree in information technology from NUST.

AHMAD SAMI AL-SHAMAYLEH received the master's degree in information systems from The University of Jordan, Jordan, in 2014, and the Ph.D. degree in artificial intelligence from the University of Malaya, Malaysia, in 2020. He is currently an Assistant Professor with the Faculty of Information Technology, Al-Ahliyya Amman University, Jordan. His research interests include artificial intelligence, human–computer interaction, the IoT, Arabic NLP, Arabic sign language recognition, language resource production, the design and evaluation of interactive applications for handicapped people, multimodality, and software engineering.

ADILA KOUSAR received the bachelor's degree in computer science from GCU, Faisalabad, and the master's degree in computer science from NUST, Islamabad, in 2023. Her current research interests include natural language processing, big data analysis, machine learning, and distributed systems.

ABDUL QADDOS received the degree in computer science from COMSATS University Islamabad, Pakistan, in 2023. He was a Web Developer with Global Gateway Services, Islamabad, from 2018 to 2020. His research interests include big data processing, machine learning, medical imaging, and computer vision.

MUHAMMAD IMRAN received the degree in software engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2006, and the master's degree in software engineering and the Ph.D. degree in computer science from the University of Southampton, U.K., in 2009 and 2015, respectively. He was a Lecturer with COMSATS University Islamabad (CUI), Islamabad, Pakistan, from 2007 to 2008. He is currently an Assistant Professor with the Department of Computer Science, CUI. His research interests include artificial intelligence, machine learning, natural language processing, and semantic web.

ADNAN AKHUNZADA (Senior Member, IEEE) has a proven track record of high-impact published research (i.e., U.S. patents, journals, transactions, reputable magazines, book chapters, conferences, and conference proceedings) and commercial products. His experience as an Educator and a Researcher is diverse that includes work as a Lecturer, a Senior Lecturer, a Year Tutor, and an Occasional Lecturer with other engineering departments; as an Assistant Professor with COMSATS University Islamabad (CUI); a Senior Researcher with RISE SICS Västerås AB, Sweden; a Research Fellow; the Scientific Lead of DTU Compute, Technical University of Denmark (DTU); the Course Director of Ethical Hacking of The Knowledge Hub Universities (TKH), Coventry University, U.K.; an Associate Professor with USM; and a Visiting Professor having mentorship of graduate students, and supervision of academic and research and development projects both at UG and PG levels. His main research interests include cyber security, secure future internet, artificial intelligence (i.e., machine learning, deep learning, and reinforcement learning), large-scale distributed systems (i.e., edge, fog, cloud, and SDNs), the IoT, IIoT, QoS/QoE, affective computing, industry 4.0, and the Internet of Everything (IoE). He is a member of the technical program committee of varied reputable conferences, journals, and editorial boards. He is a Professional Member ACM, with 15 years of research and development experience both in the ICT industry and academia.