National College of Ireland

Online Job Posting Authenticity Prediction using Machine and Deep Learning Techniques

MSc Research Project
Data Analytics 2022-2023

Gayathri Malaichamy
Student ID: X21117683

School of Computing
National College of Ireland

Supervisor: Anderson Simiscuka

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Gayathri Malaichamy
Student ID:	X21117683
Programme:	Data Analytics
Year:	2022-2023
Module:	Research Project
Supervisor:	Anderson Simiscuka
Submission Due Date:	15/12/2022
Project Title:	Online Job Posting Authenticity Prediction using Machine and Deep Learning Techniques
Word Count:	6249
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Gayathri Malaichamy
Date:	30th January 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Online Job Posting Authenticity Prediction using Machine and Deep Learning Techniques

Gayathri Malaichamy
X21117683

Abstract

Today's world is all about virtually managing every aspect of human existence, including banking online, education, security, and employment. A fraudster can easily swindle people and gain fast profits due to this rise in technology use. Nowadays, fraudulent job postings are a widespread scam. People give their personal details as well as processing fees to scammers when they submit an application for these fake job postings, and they are then scammed out of their funds. Therefore, every individual will be quite concerned about the problem of predicting bogus job postings. To accomplish this, a methodology has been suggested that utilizes Machine Learning, Deep Learning and Natural Language Processing (NLP) techniques. For feature extraction, the N-Gram model (Unigram, Bigram and Trigram) and TF-IDF (Term Frequency-Inverse Document Frequency) techniques were used. In this research, the impact of N-gram and TF-IDF feature techniques on fake job data classification has been analysed and it is found that the TF-IDF features performed better than N-Gram feature models. The analysis was done by using all five classifiers such as Naive Bayes, Random Forest, LightGBM, XGBoost and Multi Layer Perceptron (MLP) classifier. It is observed that the MLP classifier with ADAM optimizer outperformed all other classifiers with an accuracy of 95.68% and a prediction time of 13s. The second highest performer is the Naive Bayes classifier which attained an accuracy of 95.38% and a prediction time of 0.2s.

1 Introduction

The advancement of modern technology has given job seekers today a greatly increased chance of finding new jobs in large numbers. Business organisations use the internet to post online job advertisements in order to fill open positions. Job searchers can determine their preferences based on their time, qualifications, experience, suitability and other factors that are highly beneficial for finding a suitable job quickly.

Despite these benefits, it has some downsides, one of which is recruitment fraud that harasses job applicants. An online recruiting swindle is a kind of spam mail that is distributed by hackers who are after victims' sensitive information. Such fraudulent advertisements are sent out both domestically and internationally. Additionally, some ignorant job seekers share their personal data as part of application process. The hacker then confronts those victims using that information.

According to reports, there are 188 million unemployed individuals in the world. Hence, internet job advertising sites have grown to be very significant in helping people find employment. This has resulted in a large number of false job posts on these online employment portals where scammers utilize a corporation's logo as well as the name for fraudulent operations, damaging the company's reputation as well as the applicant's financial situation and personal data. Therefore, both job seekers and corporate businesses are concerned about the duplicity of job postings published on online job portals. It is a difficult task to distinguish between genuine and fraudulent employment offers. Also, the research community has not yet paid much attention to it, hence the topic is still mostly undiscovered.

In order to comprehend the fake job prediction more thoroughly, the research question that follows is addressed: *"How precisely do Machine Learning and Deep Learning models identify fraudulent online job postings to prevent job searchers from falling for scams, and how well do these algorithms stack up in terms of performance?"*

On the Fake job advertisement dataset, the current study analyses various classification algorithms such as Random Forest, Naive Bayes, XGBoost, LightGBM and MLP by using N-Gram and TF-IDF Feature extraction techniques. The primary contributions of this study are detailed as follows.

- To identify the fake job advertisements distribution on Industry, Country, Employment, Job experience level and Job Titles through various visualizations.
- To perform different NLP(Natural Language Processing) preprocessing techniques for Job description data cleaning that would remove unwanted information and build an accurate model.
- To perform word cloud to showcase the most frequent keywords used in fake and real job advertisements.
- To balance the target class of imbalanced fake job data using Random under sampler method.
- To extract features using N-Gram models(Unigram, Bigram and Trigram) and TF_IDF before applying Machine learning and Deep Learning algorithms in order to extract features and to predict the most frequent words.
- To implement the classification models and compare the model performance in terms of various metrics for both Machine Learning and Deep Learning models.

Several research studies have been carried out to identify fake job advertisements using several techniques. There was an ensemble model-based case study conducted by (Lal et al.; 2019) to predict fake job postings using Logistic Regression, J48 and Random Forest algorithms as baseline classifiers. This framework was able to predict fake jobs with an accuracy of 94% and F1 score of 95.4. However, there were no specific approaches followed to conduct the feature extraction to identify the specific Semantic and word sentences to predict the fraudulent job advertisements and to improve the model performance. This was addressed in this research by applying the N-Gram model and TF-IDF on the classifiers to extract the specific features with the most frequent words used. In addition,

as a comparative study, Both Machine Learning and Deep Learning models' performance was compared.

The research paper is divided into following segments. The related research that was done earlier is described in section 2 and the Methodology is discussed in section 3. The section 4 highlights the implementation steps while section 5 illustrates the specification of design of the research. The section 6 explains about the Results and Evaluation metrics. The project report is concluded with section 7, which is followed by References.

2 Related Work

The results and outcome of previous research projects have been carefully analysed in order to ensure that a distinctive, unique strategy is proposed for this case study. This section highlights the recent proposed and adopted changes to influential earlier works. Depending on the techniques employed, the related work review is classified into three major categories.

2.1 Machine Learning Models

One of today's most important issues on the internet is online recruiting fraud. Although the issue poses major risks to one's personal, social, and financial security as well as their right to privacy, the scientific community has not yet taken adequate steps to solve it. Research work was carried out to predict fake jobs by comparing two different datasets. Traditional classifier and boosting algorithms have been used to find out the highest accurate model. LightGBM(Gradient Boosting Machine) and Gradient Boosting algorithms obtained higher accuracy of 95.17%. The researchers (Tabassum et al.; 2021) identified the highly influential features which improve the model performance. However, the important features were not used in the experiments to improve the performance further, which can be considered as the flaw of this study.

Another case study was done by (Shree et al.; 2021) on job scam prediction by using the ensemble model approach. The authors used the Employment Scam Aegean Dataset. They removed outliers as part of data preprocessing which is a significant point to be noted for this study. The classifiers such as Logistic Regression, K Nearest Neighbors (KNN) and Random Forest were used for assessing the model performance and the Random Forest algorithm brought a higher accuracy of 99.8%. There are a few more improvements needed for this approach like comparing the real-time job portal data such as LinkedIn, Twitter and Indeed. Also, hybrid algorithms could have been implemented for the comparative study approach.

A scam detection framework was implemented by (Prashanth et al.; 2022) by using Machine Learning techniques. This is a browser tool that accepts URLs from various job recruitment sites in order to use Machine Learning algorithms to verify legitimacy. The exploratory data analysis that classifies the raw data based on Academic, Job Role, Employment Type, and Industry, was the highlight of this study. Additionally, the information about the results that were obtained is kept in a special database for later study. This tool's apparent restriction is that it only supports English-language tools. Thus, the framework could only be used in specific parts of the world.

A paper was presented by (Sundaram et al.; 2021) to analyse the emotions using the TF-IDF method. By using TF-IDF, the authors predict the words that influence more on each emotion. Data representation relies on semantic structure, and emotion is

taken out of various texts. The classifiers namely Random Forest and Support Vector Machine(SVM) were utilized for prediction. The significant part of this research work is, it outperformed the existing approaches on emotion analysis with the help of TF-IDF. This research study brought better results than the existing approaches and increased the accuracy by 5%.

The advantages of social media are exploited by fake news to propagate swiftly. There was a case study proposed by (Tian and Baskiyar; 2021) to detect fake news with the help of two different feature extraction techniques namely Evolutionary and Genetic features. The highlight of the study is the feature extraction techniques that made KNN classifier to yield a better accuracy of 91.3%. However, there were no model optimization techniques used for Quantum Machine model optimization.

2.2 N-Gram Feature Extraction Technique

People in academia and the publishing industry are especially interested in plagiarism prevention. Plagiarism is the act of using another author's words and ideas and presenting them as their own work. The authors (Khan et al.; 2011) proposed an N-Gram model that detects plagiarism in students' Urdu language assignment work. In terms of both complexity and false alarms, they found that the trigram model offers an average performance that is satisfactory while being cost-effective. They compared trigram model selection to Bigram as well as Fourgram models to evaluate its validity. Even though the constructed model produced good results, Urdu Text categorization performance might have been improved by implementing Machine Learning methods.

Another research was conducted by (Yang et al.; 2007) to identify OLE file types using the combined model of N-Gram and vector space models. Different file types with various sizes were used to train the samples. The top M common n-grams in the CNG technique, along with their class model, and normalized frequencies are used. In their research, the size of profile, truncation and file size are main variables. The best accuracy is greater than 99%, whereas the lowest accuracy is just no less than 91%. By using this methodology, researchers were able to distinguish between four major OLE files and outperform the CNG method. However, the same approach could have been implemented in the malware code detection field for comparative analysis which is the improvement area in this study.

Spam has become a major issue across several media types since the rise of the internet and the decline in the cost of digital communication. The research was proposed by (Bozkir et al.; 2017) to classify spam emails. In this case study, a pure usage of linked texts combined with a word-level N-Gram model was given as a substitute for the subject or body parts of emails to produce features for a phishing email classifier. The authors built an N-Gram model with SVM, SVM-Pegasos and Naive Bayes classifiers for classification. The trigram configuration of SVM Pegasos produced the greatest results (i.e., the accuracy of 98.75%) on the 50 threshold dataset. However, more efficient Machine Learning techniques such as Deep Random Forest as well as Twin SVM could have been used to produce more accurate categorization.

A case study was suggested by (Ahuja et al.; 2019) to classify the sentiment comments posted on Twitter. The authors compare the N-gram and TF-IDF feature extraction techniques by applying Machine Learning algorithms such as Naive Bayes, Decision Tree, Logistic Regression, KNN, SVM and Random Forest. Logistic Regression performed well with an accuracy of 57% and the TF-IDF model category brought better results

compared to the N-Gram model. Several evaluation metrics were considered for model performance validation. The main drawback of this study was the less accuracy. Different hyperparameter tuning methods as well as feature selection could have been used to improve the model performance.

2.3 Deep Learning Models

A Research was conducted for fake job prediction by utilizing single as well as ensemble classifier models. The authors (Dutta and Bandyopadhyay; 2020) used classifiers such as Naïve Bayes, KNN and Decision Tree and found that the Multilayer Perceptron(MLP) classifier produced good accuracy of 96.14% in the single classifier class, whereas the Random Forest Model outperformed with the accuracy of 98.27% in ensemble classifier category. In addition, the model performance of both single and ensemble classifiers was compared with other evaluation metrics namely F1-Score, Cohen-Kappa Score and MSE. Although the framework brought satisfactory results there was no Exploratory Data Analysis (EDA) performed on the raw data to analyse the data behaviour before the model build.

In recent years, false news has become a major issue, especially on sites such as Facebook and Twitter and other internet publications like websites and blogs. To predict the fake news, the authors (Jehad and Yousif; 2021) presented a framework by using MLP and TF-IDF as feature extraction methods. In addition, the classifiers such as Support Vector Machine (SVM), Random ForestStochastic Gradient Descent (SGD), Gradient Boosting were used for comparative analysis. MLP brought a higher accuracy of 95.47%. However, different hyperparameter tuning techniques could have been applied for traditional classifiers as the average accuracy obtained was 77.6% which is a way more lesser than MLP's accuracy.

Another fake job prediction framework was invented by (Nasser et al.; 2021)to detect fake job postings using Artificial Neural Network (ANN) on the EMSCAD dataset. The down-sampling method was implemented to balance the dataset. The data was preprocessed using NLP functions such as stop words removal and tokenization. Also, the count vectorization – Bag of words(BOW) method was implemented for feature extraction. Although the ANN model obtained 93.88 of F1 score, different feature extraction techniques could have been implemented to study the semantic relations of the text data in order to acquire better performance that leads to better performance for the classification model.

The problem of fraudulent recruitment information is addressed by identifying online fraud that makes use of crowdsourcing as well as multi-feature fusion methods. The authors (Wang and Liu; 2022) used Bidirectional Encoder Representation from Transformers(BERT), Bidirectional Long Short Term Memory(BiLSTM) and CNN algorithms. The significance of the keywords with in text classification is emphasized by this model. The experimental results demonstrates that the suggested model has increased it's classification accuracy by 5.3% when compared to existing network models. However, real-time data was not used in the investigation. Moreover, the authors could have offered a lot more information if they had included the confusion matrix that was generated.

In this research, The exploratory data analysis was performed to analyse the fake and real job distribution on various factors. Different hyperparameter tuning techniques were applied to improve the model performance. None of the previous studies mentioned in this section, calculated the performance time. Hence, the time factor was also used as one

of the measure for assessing the performance along with other measures namely F1-Score, Precision, Recall, Accuracy, ROC_AOC curve and Confusion Matrix. In addition, MLP is used with two different optimizers, which is simple and there is no need of pre-trained models that increases the complexity in computations. As a novel approach, the fake job prediction framework was implemented by utilizing both Machine Learning classifiers and Deep Learning algorithms on the features extracted using N-Gram models like Unigram, Bigram, and Trigram as well as the TF-IDF vectorization technique.

In Table 1, the summary results of previous research studies are illustrated.

Table 1: Summary of Related Research Results

Authors	**PredictionTopic**	**Approach**	**Accuracy(%)**
Hridita et al. 2021	Fake Job	MachineLearning	95.17%
Asmitha et al.2021	Fake Job	Machine Learning	99.8%
Cprashanth et al.2022	Fake Job	Machine Learning	90%
Muhammad et al. 2011	Plagiarism	N-Gram model	Good Results
Hong-Rong et al. 2007	OLE files	N-Gram model	99%
Selman et al. 2017	Spam Emails	Machine Learning	98.75%
Shawni 2020	Fake Job	Machine Learning,MLP	98.27%
Reham 2021	Fake News	Machine Learning,MLP	95.47%
Ziyan 2021	Fake News	Machine Learning	91.3%
Ibrahim et al. 2021	Fake Job	ANN	93.88%
JunlingBo 2022	Fake Job	BERT,BiLSTM	91.68%
Ahuja et al. 2019	Sentiment Analysis	Machine Learning	57%
Sundaram et al.2021	Emotion Analysis	MachineLearning,TF-IDF	85%

3 Methodology

The methodology implemented in this research work is Knowledge Discovery in Databases(KDD). It starts from collecting the fake job advertisement data, analysis of the raw data, preprocessing the text data using NLP techniques and transforming the data to train the classification models, and analysing the performance by applying various evaluation metrics. Figure 1 illustrates the methodology in stages implemented in this research.

3.1 Collection of SourceData

The dataset was sourced from a public repository called "Kaggle". This is an Employment Scam Aegean Dataset (EMSCAD) which consists of over 17000 records and 18 attributes of job advertisement information. The information is made up of both textual and job-related meta data[1].

3.2 Raw Data Analysis

The fake job dataset is a binary classification data that has the target column "Fraudulent" with values 0 (Non Fraud) and 1 (Fraud). The columns Company Logo, Questions,

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

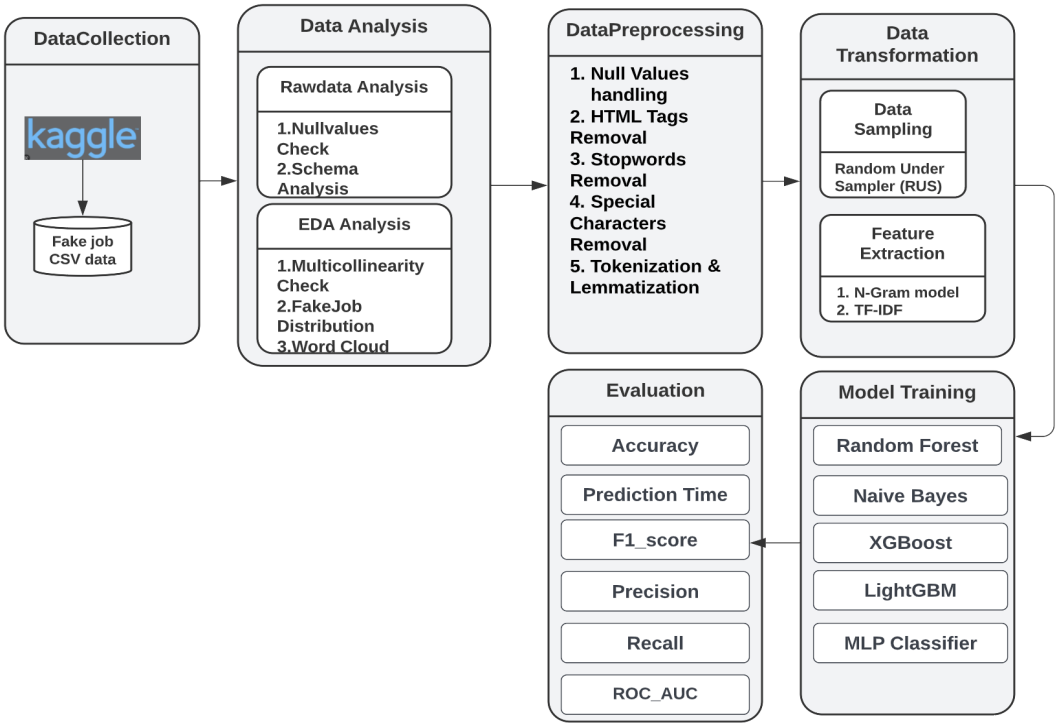

Figure 1: Research Methodology

and Telecommunicating are also binary columns. The other columns such as Required Experience, Employment Type, Industry, Required Education, and Function are nominal columns. Requirements, Description, Company Profile, and Benefits are HTML fragment columns which were used mainly for text processing. Null values were handled as part of preprocessing.

3.3 Exploratory Data Analysis

Exploratory analysis was performed to analyse the relationship between the independent variables. The collinearity was checked by plotting a heat map for the independent variables. To analyse the fake job advertisements distribution, Employment_type and Required_experience columns were used. Figure 2 illustrates that more fake advertisements are present for Full-time jobs.

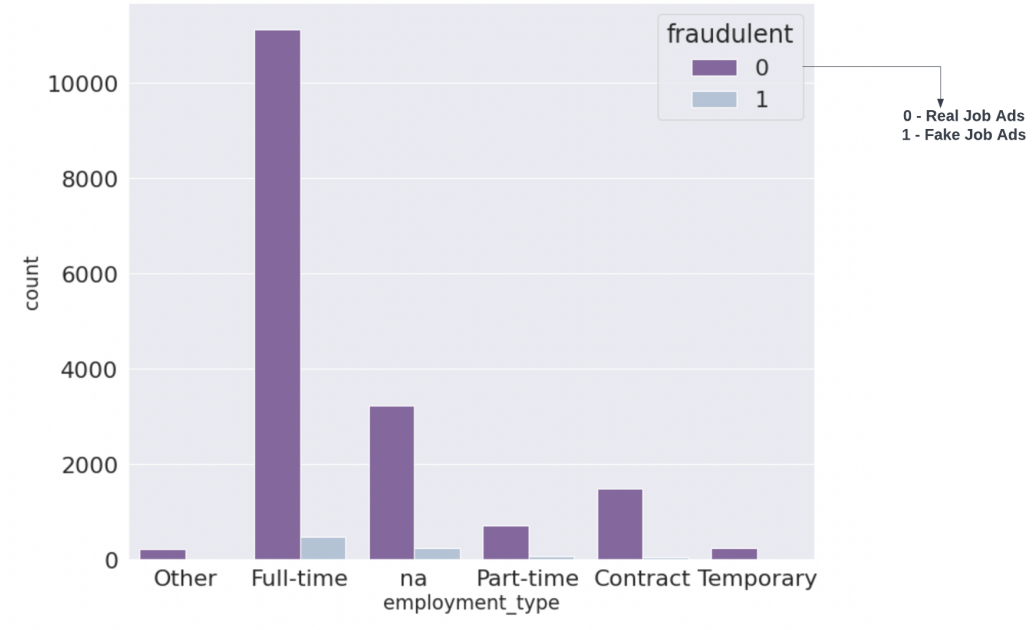

Figure 2: EmploymentType

Figure 3 illustrates that more fake advertisements are present for Entry-Level Employees.

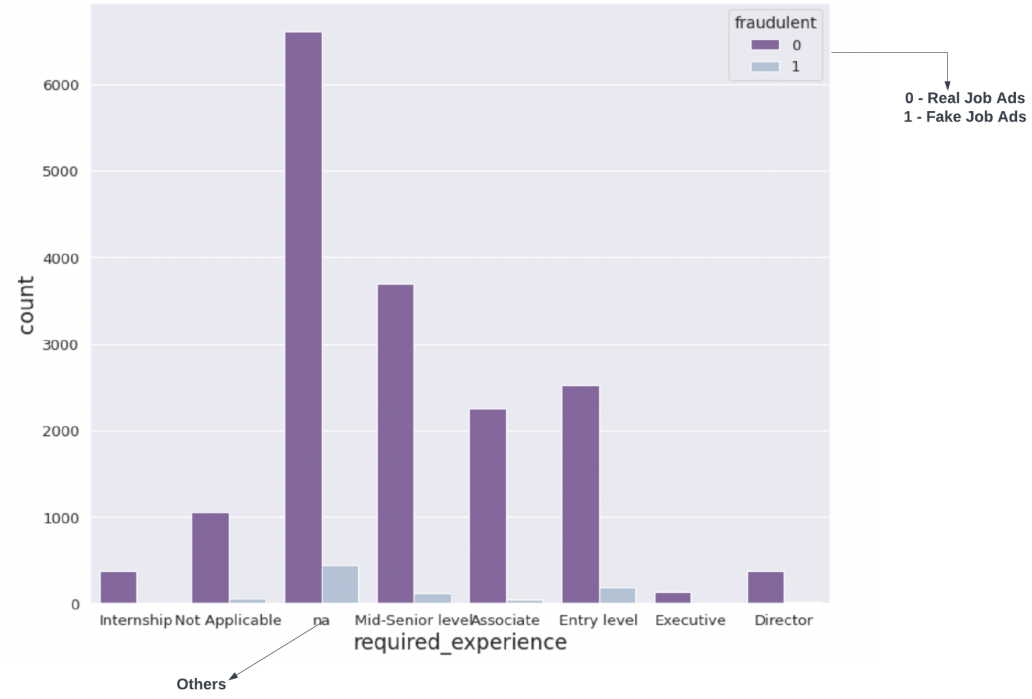

Figure 3: Required_Experience

Also, based on the location column the distribution of Fake job postings was analysed

using the world map feature as shown in figure4. It is clearly evident that Malaysia has the higher number of fake job advertisements at 57.14% and the second highest is in Australia with 18.69%.

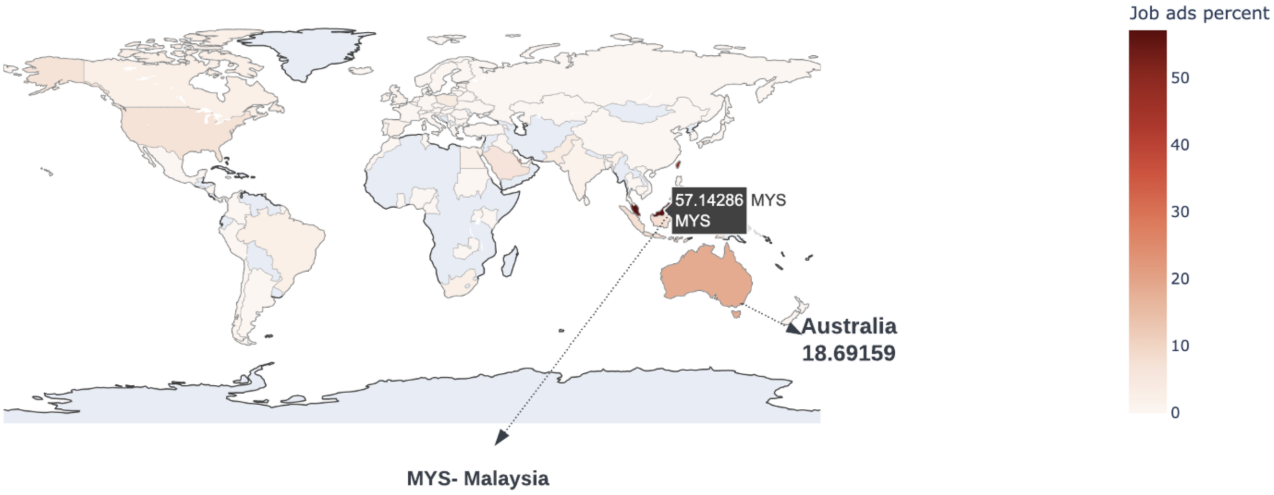

Figure 4: Fakejob Postings distribution

In addition, the word cloud was plotted for fake and real job advertisement records as shown in word cloud figures. From figure 5, it is quite evident that the common words mostly occurred and it does not include the job requirements in fake job advertisements.

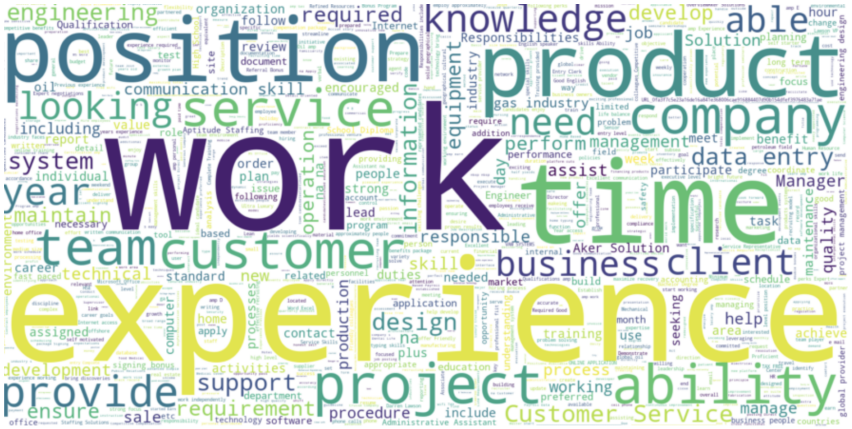

Figure 5: Frequentwords in Fakejob Postings

On the other hand, figure 6 illustrates the real job postings word cloud which comprises of job-related information such as team, software, customer, client, service, business and design.

Figure 6: Frequentwords in Realjob Postings

3.4 Data Preprocessing

Firstly, fake job data was analysed for null values and replaced with 'na' values. Then, irrelevant features such as 'department' and 'salary_range' were removed from the data. On the other hand, the HTML tags were removed from the columns namely 'company_profile', 'description', 'requirements', and 'benefits' as they do not add any value to the prediction. The special characters were removed using regular expressions. Following NLP operations were performed on the processed text to make the text back to it is original form (Nasser et al.; 2021).

- Stopword Removal: Many words in text data, such as "a," "about," "after," "again," etc., are not given any meaningful value in classification tasks. The preferred approach to treat them is to completely remove them from tokens (words).
- Tokenization: Tokenization refers to the process of decomposing a text into appropriate elements known as tokens. In this method, the words in specified sentences are explored.
- Lemmatization: Lemmatization was performed to extract the fundamental form of the words for Text.

3.5 Data Balancing using RandomUnderSampler

Since the data is highly imbalanced, the RandomUnderSampler(RUS) was used to balance the data. According to studies, the sample distribution is used to infer the overall population from where the sample was obtained as long as it was chosen at random. As a result, by understanding the sample distribution, the target distribution can be predicted as done by (Bagui and Li; 2021). Also, RUS is preferred for large data analytics since it has a comparatively reduced computational requirement and faster training times as presented by (Hasanin et al.; 2019).

3.6 Feature Extraction using N-Gram and TF-IDF

Feature extraction is a method of turning the raw data into various features that can be maintained while keeping the data from its earlier set. The feature extraction technique

achieves good result rather than applying Machine Learning algorithms directly on the data. N-Gram (Unigram, Bigram and Trigram) and TF-IDF are prominent as well as effective feature extraction methods for representing text data. A contiguous group of elements of the length n is called an N-Gram. It might consist of a string of letters, numbers, bytes, or other characters. Word-based, as well as character-based N-Gram models, are the most frequently utilized N-Gram algorithms in text categorization. In this research, the text context is described to create attributes and to classify the text using word-based N-Gram (Unigram, Bigram and Trigram).

- Unigram: It considers only the frequency of a word not the previous occurred words.
- Bigram: Bigram uses the previous word to predict the next occurrence of the word.
- Trigram: Two preceding words are taken in to consideration while predicting the next word.

A common weighting metric in information extraction and natural language processing, TF-IDF is a statistical metric that examines the significance of a word to a text in a dataset. The occurrence of the word in a phrase balances the effect of total no of occurrences of a word in a sentence. The baseline N-Gram (Unigram, Bigram and Trigram) as well as the TF-IDF were used to examine the implication of N-Gram length on the performance of various classifier algorithms as used by (Ahmed et al.; 2017) in online fake news prediction.

3.7 Modelling Approach

After feature extraction, the dataset was divided into train and test sets in the ratio of 80:20 for training as well as validation purposes. Then, the models were built on the training data set to assess the model performance. For this research, both Machine Learning as well as Deep Learning algorithms were used in order to perform a comparative study.

3.7.1 Machine Learning Classifiers

The dataset used for this research is related to classification issue where the target variable has a binary class. Hence, traditional classifiers such as Naive Bayes, Random Forest, XGBoost (Extreme Gradient Boosting) and LightGBM were used for training the models. The Naive Bayes is a widely used algorithm for the application of text classification. Besides being simple, the Nave Bayes Classifier is well-known to perform better than even more complex classification techniques. The size of the model is one significant difference between Random Forest and Naive Bayes. Unlike Naive Bayes, Random Forest may experience overfitting problems due to the high model size as presented by (Fayoumi et al.; 2022).

Generally, boosting enables Machine Learning models to improve the accuracy of their predictions. Additionally, it lowers the ensemble model's bias and variance. XGBoost is a powerful prediction algorithm that separates level-wise instead of leaf-wise in comparison to LightGBM. LightGBM is a gradient lifting framework that is rapid, distributed, and high-performing and is based on the well-known Machine Learning technique, Decision Tree. Also, it is seven times faster than the XGBoost algorithm. Hence, in addition to other classifiers, XGBoost, as well as LightGBM, were utilized to compare the model performance as presented by (Tabassum et al.; 2021).

3.7.2 MLP Classifier

On the other hand, The MLP classifier was used for comparative analysis. A fully connected type of feedforward-artificial neural network (ANN) is known as a Multilayer Perceptron or MLP. A completely connected input layer as well as an output layer make up the Perceptron as shown in Figure 7 [2]. MLP was chosen to predict fake job postings as it has fewer hidden layers than Deep Neural Networks, which means it trains the model quickly. Moreover, the MLP classifier makes accurate predictions quickly and responds well to large datasets.

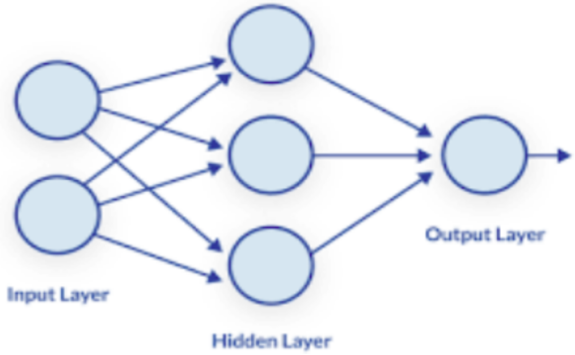

Figure 7: Multi Layers Perceptron (MLP classifier)

MLP's hidden layer and the number of nodes in each layer might differ depending on the circumstances. The training set and network architecture are taken into consideration while determining the parameters as implemented by (Shawni and Samir 2020). Adam is the default optimizer that was chosen for model optimization as it has few parameters to tune the model and less computation time for large datasets. Both in terms of training duration and validation score, Adam performs well. On the other hand, LBFGS is more stable and has fewer parameters to tune the mode. Also, It does not use a learning rate. Hence, these two optimizers were used with MLP classifier as implemented by (Nwaogu and Dimililer; 2021).

4 Design Specification

There are several steps involved in fake job posting prediction. The overall step-by-step process is depicted as in figure 8. Firstly, the data was collected and performed an initial analysis to understand the data behaviour such as null values and multicollinearity. Then EDA was performed in order to analyse the fake job postings distribution on multiple columns.
Then, irrelevant columns and HTML tags from the text columns were removed. As part of NLP processing, the operations such as stop word removal, tokenization and lemmatization were performed to clean the text data before model building. As the data was imbalanced, Random Under Sampler was applied to make the data balance.

[2] https://machinelearninggeek.com/multi-layer-perceptron-neural-network-using-python/

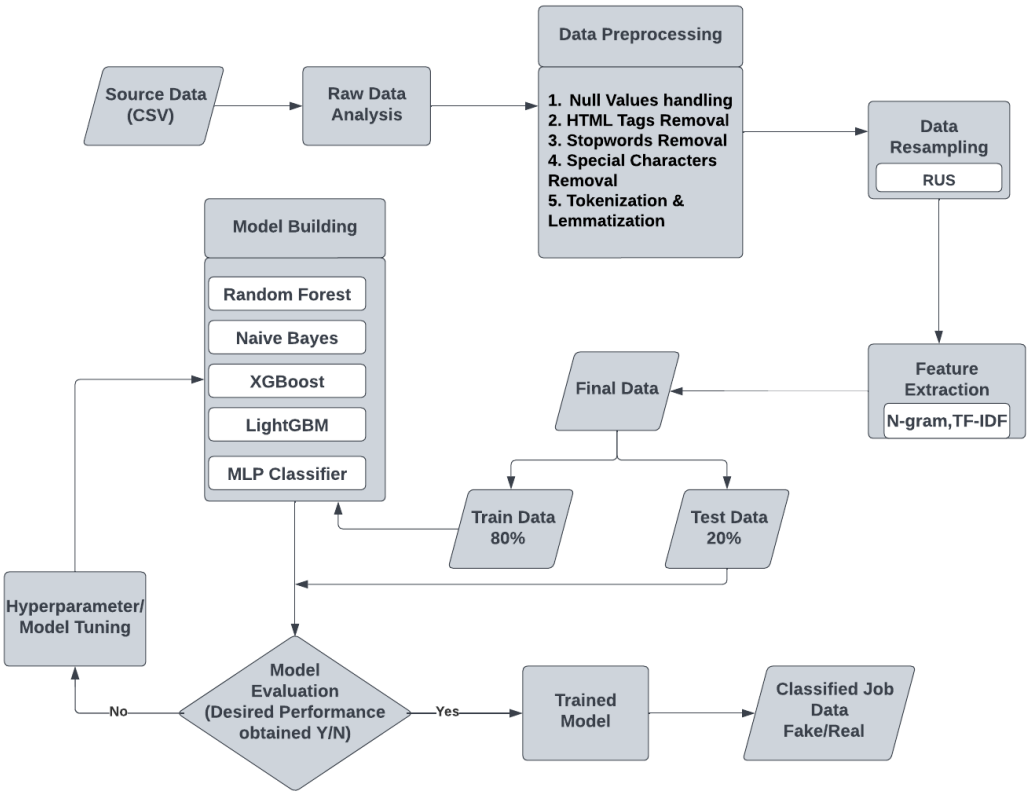

Figure 8: Data Flow Diagram

Also, the feature extraction techniques such as N-Gram and TF-IDF were applied to the resultant data before model training. The categorical variables were encoded and the transformed data was divided into 80:20 ratio for Train and Test sets. Finally, models were implemented by applying Machine Learning as well as MLP algorithms. Hyperparameter tuning technique was also applied for Random Forest and Light GBM models to improve the performance. Finally, several evaluation metrics were applied to the test data to validate how well the model performed for different classifiers.

On a high level, the fake job prediction architecture consists of three layers as illustrated in figure 9. The data selection and preprocessing are performed in the "Database Layer". The pre-processed data is sent to the middle layer which is known as the "Application Layer" where the data is undergone multiple transformations makes the data suitable for model training.

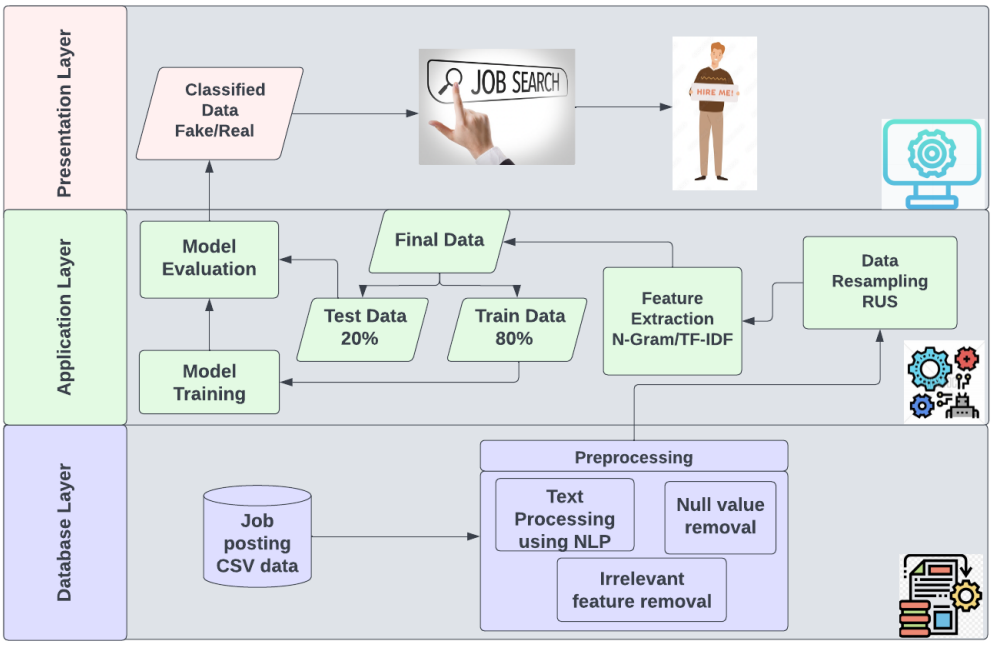

Figure 9: System Architecture

For accurate prediction, the data is resampled using RandomUnderSampler(RUS) method and the N-Gram as well as TF-IDF feature extraction techniques, are applied. The final data is divided into 80:20 ratio for training and test datasets. Thereafter, model building starts with Machine Learning classifiers and then with an MLP classifier. The final classification results are sent to the final level "Presentation Layer" where the Job Seekers will have access to identify whether the searched job advertisement is valid or not.

Fake Job Advertisement	**Real Job Advertisement**
We are a full-service marketing and staffing firm, serving companies ranging from Fortune 100 to new start-up organizations. We work with job seekers in an equally broad range, from light industrial temporary workers to executive-level candidates. Are you looking for a Work from Home Opportunity where you can earn up to $2500 and more per week? Our Online Service Representative position would be perfect for you! - Set your own hours - Make money every time you decide to work - Work remotely from home - Get paid weekly - If you have a computer with internet, this is for you.	The Administrative Assistant will be based in San Francisco, CA. The right candidate will be an integral part of our talented team, supporting our continued growth. Responsibilities: Attend meetings to record minutes. Compile, transcribe, and distribute minutes of meetings. Conduct research, compile data, and prepare papers for consideration and presentation by executives, committees, and boards of directors. Provide secretarial support for a professional, supervisor or manager, and in some cases, subordinate staff of that individual. Provide highly skilled keyboarding support in the transcribing and recording of information.

Figure 10: Fake Vs Real JobAdvertisement

Figure 10 shows the classified data sample for Fake and Real job advertisement.

5 Implementation

This section explains the procedures and tools that were used during the fake job posting prediction. Python 3 was used to create the classification model as it has a plethora of libraries and methods for creating Machine Learning models. For the execution environment, Google Colaboratory was used as it is faster and suitable for executing Deep Learning models. Also, the Deep Learning libraries such as PyTorch, TensorFlow, and Keras are already installed in Google Colaboratory.

The fake job data has been sourced from the Kaggle website. The data was loaded into Dataframe using the pandas library. Initially, the data was analysed for the schema structure and Null value presence. Null values were replaced by the 'na' value. Using a heatmap plot from the Seaborn library, collinearity was checked and found that there was no collinearity among features. Various Count plots were plotted using the matplotlib package to analyse the fake job distribution for various features.

The Location column has both city and country. The country value was extracted from the location column and geographical distribution was plotted for both total and fake job advertisements using Plotly express. The most frequent words were obtained with the help of a word cloud package for both fake and real job advertisement records.

Upon EDA completion, the data preprocessing started with irrelevant columns removal from the input data frame. Thereafter, categorical columns were moved to one data frame for further processing. The text columns such as company_profile, description, requirements and benefits have HTML tags that were removed from the data. Stop words and special characters were also removed for extracting clean text data using the nltk package. WordNetLemmatizer() was used for the lemmatization process to bring the words back to their original form.

The categorical variables were encoded using Label Encoder. Then, N-Gram models such as Unigram, Bigram and Trigram were built for feature extraction using the Count Vectorizer method from the Scikit library. In addition, TF-IDF is also used for feature extraction by applying TfidfVectorizer method. After feature extraction, the data was split into the raio of 80:20 for train and test dataset. Machine Learning models such as Naïve Bayes, Random Forest, XGBoost and LightGBM were trained for all three N-Gram models as well as TF-IDF. Hyperparameter parameter tuning was implemented for Random Forest and LightGBM in order to increase the model performance.

For Deep Learning models, the ANN model was built as a base model with optimizer 'Adam' and activation parameter with 'relu' and 'sigmoid'. The epochs value 65 was used. Secondly, the MLP classifier model which is a feed-forward ANN was trained using two optimizers 'Adam' and 'LBFGS'. The activation parameter value 'relu' was applied. Also, the hidden layers with the size of (100,50,30) and Max_iter 1000 were passed. After successful model training, several evaluation metrics were applied to the test dataset in order to validate the model performance.

The evaluation metrics such as F1 score, Accuracy, Recall and Precision were calculated for each model. Moreover, the time was measured in order to assess how fast the models were able to predict the outcome. The Confusion matrix was also plotted for each model to compare the percentage of predicted groups to expected ones. Finally, the ROC_AUC curve was plotted to compare the model performance.

6 Evaluation

This section summarizes the outcome of each model as well as the evaluation metrics that assess the performance of the model. The key objective of Machine Learning prediction is to assess the model performance. Hence, in this research, several metrics such as "Precision, "Accuracy", "F1-Score", "Recall" were used. In addition to these measures, the time factor was also considered for the duration of the model prediction. All these metrics are efficient measures to evaluate any binary classification problem as per the previous study referred to in section 2. In addition, the ROC_AUC (Receiver Operating Characteristic-the Area Under the Curve) was plotted in order to compare the model efficiency. Also, the Confusion Matrix was plotted for each model to evaluate the frequency of projected and expected classes. This section has four key subsections based on feature extraction approaches namely N-Gram (Unigram, Bigram, and Trigram) and TF-IDF. For all these four categories the Machine Learning classifiers namely LightGBM, Random Forest, XGBoost and Naive Bayes were trained and validated. Moreover, the MLP classifier which is a feed-forward ANN model was also trained using two optimizers for model comparison.

6.1 Unigram Model / Experiment 1

In the first experiment the N-Gram range was set to (1,1) for the Unigram model and all five classifiers namely Random Forest, Naïve Bayes, XGBoost, LightGBM and MLP classifier were trained and validated. For Unigram Model, the MLP classifier with ADAM optimizer outperformed other classifiers with an accuracy of 94.52%, F1-Score of 94.79, Precision of 94.02 and Recall of 95.58. The time duration to predict the model was 13s. The underperformer in this category is the XGBoost classifier, with an accuracy of 91.07%, F1-Score 91.46, Precision of 91.21 and Recall of 91.77. It took 21.4s for model prediction. Hyperparameter parameter tuning was also applied for both Random Forest and LightGBM classifiers. For Random Forest, tuning provided a slight improvement in the model accuracy but not for the LightGBM classifier.

6.2 Bigram Model / Experiment 2

In the second trial, the N-Gram range was set to (2,2) for the Bigram model and repeated the same steps as implemented in section 6.1. For Bigram Model, the MLP classifier with ADAM optimizer outperformed other classifiers with an accuracy of 95.39%, F1-Score of 95.65, Precision of 94.12 and Recall of 97.24. The time duration to predict the model was 14.1s. The least performer in this category is the XGBoost classifier, with an accuracy of 83%, F1-Score 82.18, Precision of 90.67 and Recall of 75.14. It took 21.4s for model prediction. Hyperparameter tuning did not improve the applied model accuracy for the Bigram model category.

6.3 Trigram Model / Experiment 3

In the third trial, the N-Gram range was set to (3,3) for the Trigram model and repeated the same steps as implemented in section 6.1. For Trigram Model, the Naive Bayes classifier outperformed other classifiers with an accuracy of 94.24%, F1-Score of 94.32, Precision of 97.08 and Recall of 91.71. The time duration to predict the model was 0.4s.

The last performer in this category is the Random Forest algorithm, with an accuracy of 89.05%, F1-Score 89.95, Precision of 86.29 and Recall of 93.92. It took 12.1s for model prediction. Hyperparameter tuning provided a noticeable improvement of 17% increment from the base model accuracy for the LightGBM classifier.

6.4 TF-IDF Model / Experiment 4

In the last trial, the features were extracted using TF-IDF and repeated the same steps as implemented in section 6.1. For TF-IDF Model, the MLP classifier with ADAM optimizer outperformed other classifiers with an accuracy of 95.68%, F1-Score of 95.82, Precision of 96.63 and Recall of 95.03. The time duration to predict the model was 13s. The last performer in this category is the XGBoost classifier, with an accuracy of 90.49%, F1-Score 90.76, Precision of 92.05 and Recall of 89.50. It took 22s for prediction. In the TF-IDF model, hyperparameter tuning slightly increased the accuracy only for the Random Forest classifier.

6.5 Model Comparison

The Machine learning as well as Deep Learning classifiers, were employed in order to compare the model performance under each category of feature extraction. Some of these models outperformed with better accuracy while others with moderate results. The time factor was also considered one of the key factors for model comparison. In terms of accuracy, the MLP classifier with ADAM optimizer from the TF-IDF category outperformed all four feature extraction models with the highest accuracy of 95.68% and a time duration of 13s. Also, the Naive Bayes classifier performed well with an accuracy of 95.38% and a time duration of 0.2s. However, some applications consider performance time as the most critical factor. In that case, MLP with LBFGS classifier is the outperformer with a time duration of 0s. The second fastest algorithm is the Naive Bayes with a time duration of 0.2s. Tables 2 and 3 illustrate the model comparison of all four categories with all important measures. In both the tables 2 and 3, the accuracy and time values are written in bold and italicized (Bold denotes the highest and Italicized denotes the second highest value) under each category.

In addition to these measures, The ROC-AUC was plotted to compare the models based on the ROC-AUC scores. Figure 11 depicts the comparison of all four categories and it is evident that the TF-IDF model achieved better performance in terms of ROC_AUC curve for all classifiers. Because for TF-IDF features, all five classifiers (Random Forest, Naive Bayes, XGBoost, LightGBM and MLP) performed well with the ROC_AUC value ranging from 0.97 to 0.99.

Table 2: Summary Results of Unigram & Bigram models

Models	Unigram					Bigram				
	Accuracy (%)	**Time (s)**	**F1_score**	**Precision**	**Recall**	**Accuracy (%)**	**Time (s)**	**F1_score**	**Precision**	**Recall**
RandomForest	93.08	*1.9*	93.26	94.86	91.71	91.93	3.4	92.43	90.48	94.48
RandomForest hyperparamer	93.37	25.4	93.59	94.38	92.82	92.22	49	92.13	97.53	87.30
NaiveBayes	*94.24*	**0.3**	94.32	97.08	91.71	92.80	22	92.80	96.99	88.95
XGBoost	91.07	21.4	91.46	91.21	91.71	83	21.4	82.18	90.67	75.14
LightGBM	93.95	2.3	94.15	94.94	93.37	89.63	**0.6**	89.89	91.4	88.40
LightGBM hyperparamer	93.08	*1.9*	93.37	93.37	93.37	87.61	*1.3*	87.24	94.23	81.22
MLP-'LBFGS'	91.64	11.5	91.97	92.22	91.71	*94.81*	9.6	95.08	94.05	96.13
MLP-'ADAM'	**94.52**	13	94.79	94.02	95.58	**95.39**	14.1	95.65	94.12	97.24

Table 3: Summary Results of Trigram & TF-IDF models

Models	Trigram					TF-IDF				
	Accuracy (%)	**Time (s)**	**F1_score**	**Precision**	**Recall**	**Accuracy (%)**	**Time (s)**	**F1_score**	**Precision**	**Recall**
RandomForest	89.05	12.1	89.95	86.29	93.92	93.08	2	93.22	95.38	91.16
RandomForest hyperparamer	89.91	140.9	89.30	1	80.66	93.37	30.9	93.52	95.40	91.71
NaiveBayes	**94.24**	**0.4**	94.32	97.08	91.71	*95.38*	*0.2*	95.45	98.25	92.81
XGBoost	77.52	41.9	74.34	91.87	62.43	90.49	22	90.76	92.05	89.50
LightGBM	71.76	*0.5*	67.11	85.47	55.25	92.51	3.8	92.61	95.32	90.06
LightGBM hyperparamer	89.05	2.3	88.55	97.35	81.22	92.22	2.6	92.35	94.77	90.06
MLP-'LBFGS'	89.91	26	90.81	86.5	95.58	92.22	**0**	92.56	92.31	92.82
MLP-'ADAM'	*92.21*	33.2	92.72	90.53	95.03	**95.68**	13	95.82	96.63	95.03

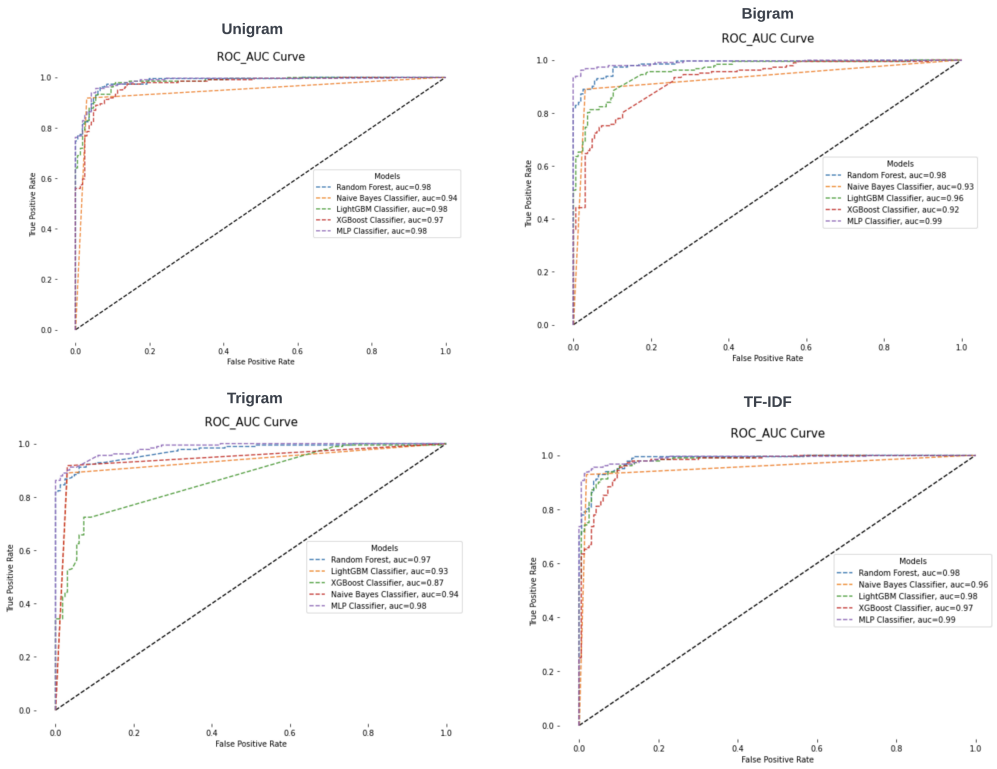

Figure 11: Comparison of ROC_AUC Curve

The confusion Matrix was plotted for all classifiers. Figure 12 shows the confusion matrix of the MLP classifier with ADAM optimizer. As per the matrix, it is inferred that the MLP classifier is able to predict 160 True positives of Non-Fraudulent cases and 172 True Negatives of Fraudulent cases accurately.

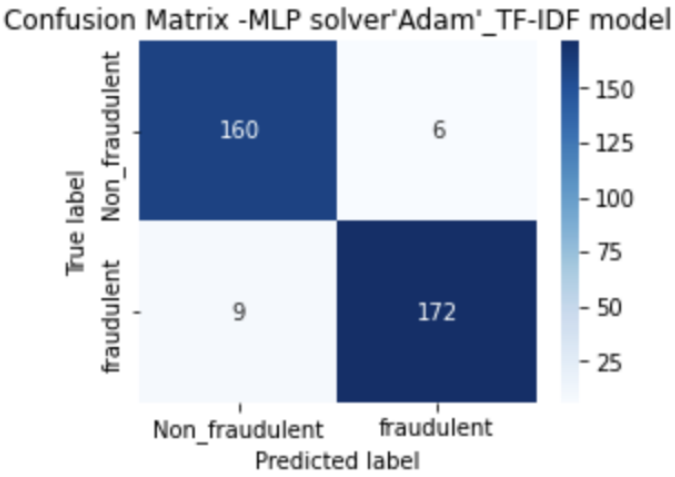

Figure 12: Confusion Matrix

6.6 Discussion

In this comparative study, different feature extraction techniques were implemented along with Machine learning and Deep Learning classifiers for prediction to prove the novelty. The N-Gram (Unigram, Bigram and Trigram) and TF-IDF feature extraction techniques play a critical role in text classification in various fields as mentioned in section 2. As the fake job data prediction is based on the text description, the N-Gram and TF-IDF techniques were implemented. It is observed that the TF-IDF feature models provided better performance results (1-2%) higher accuracy than N-Gram feature models. On comparison of classifiers, the MLP classifier with ADAM optimizer outperformed other classifiers in terms of accuracy as shown in the comparison tables 2 and 3. The second highest performer is the Naive Bayes classifier with less prediction time. In addition to accuracy and time, other evaluation metrics such as F1_Score, Precision, Recall, Confusion Matrix and ROC-AUC curve were also used for model performance validation.

For model optimization, hyperparameter tuning was applied for Random Forest and LightGBM classifiers. It was observed that the hyperparameter tuning improved the accuracy by 17% for the LightGBM classifier of the Trigram model category. Different parameters could have been used for tuning to achieve a better performance for the other three categories. Moreover, some classifiers took more time for model prediction. For instance, the Random Forest algorithm with hyperparameters took 141 seconds under Trigram category. This performance time could have been improved by reducing the parameter space and by implementing parallel computation.

7 Conclusion and Future Work

In this digital era, recruitment scam become a serious issue as it leads to personal data and financial loss for job applicants. Therefore, it is crucial for both job seekers and recruiters to identify bogus job postings. In this research, both Machine Learning classifiers, as well as Deep Learning classifiers, were used to predict fake job postings. Two different feature extraction techniques such as N-Gram and TF-IDF were used. It is observed that the TF-IDF feature models performed better than the N-Gram feature models. By comparing machine learning algorithms generally, the MLP classifier with ADAM optimizer outperformed all other classifiers with an accuracy of 95.68% and a prediction time of 13s. On the other hand, the Naïve Bayes attained the second highest accuracy of 95.38% and a prediction time of 0.2s. Overall, MLP classifier with ADAM and Naive Bayes are the top performers and are highly recommended for online fake job advertisement prediction. Also, both feature extraction methods are effective enough for text classification.

In future, different feature extraction techniques such as Word2Vec and Bag of word will be used for the job advertisements' text extraction. Moreover, the different job portals such as Indeed, LinkedIn and Amazon Jobs data will be used for real-time prediction. Also, different hyperparameter tuning techniques will be used for better model performance with less prediction time.

References

Ahmed, H., Traore, I. and Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques, *International conference on intelligent, secure, and dependable systems in distributed and cloud environments*, Springer, pp. 127–138.

Ahuja, R., Chug, A., Kohli, S., Gupta, S. and Ahuja, P. (2019). The impact of features extraction on the sentiment analysis, *Procedia Computer Science* **152**: 341–348.

Bagui, S. and Li, K. (2021). Resampling imbalanced data for network intrusion detection datasets, *Journal of Big Data* **8**(1): 1–41.

Bozkir, A. S., Sahin, E., Aydos, M., Sezer, E. A. and Orhan, F. (2017). Spam e-mail classification by utilizing n-gram features of hyperlink texts, *2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1–5.

Dutta, S. and Bandyopadhyay, S. K. (2020). Fake job recruitment detection using machine learning approach, *International Journal of Engineering Trends and Technology* **68**(4): 48–53.

Fayoumi, M. A., Odeh, A., Keshta, I., Aboshgifa, A., AlHajahjeh, T. and Abdulraheem, R. (2022). Email phishing detection based on naïve bayes, random forests, and svm classifications: A comparative study, *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0007–0011.

Hasanin, T., Khoshgoftaar, T. M., Leevy, J. and Seliya, N. (2019). Investigating random undersampling and feature selection on bioinformatics big data, *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 346–356.

Jehad, R. and Yousif, S. A. (2021). Classification of fake news using multi-layer perceptron, *AIP Conference Proceedings*, Vol. 2334, AIP Publishing LLC, p. 070004.

Khan, M. A., Aleem, A., Wahab, A. and Khan, M. N. (2011). Copy detection in urdu language documents using n-grams model, *International Conference on Computer Networks and Information Technology*, pp. 263–266.

Lal, S., Jiaswal, R., Sardana, N., Verma, A., Kaur, A. and Mourya, R. (2019). Orfdetector: Ensemble learning based online recruitment fraud detection, pp. 1–5.

Nasser, I. M., Alzaanin, A. H. and Maghari, A. Y. (2021). Online recruitment fraud detection using ann, *2021 Palestinian International Conference on Information and Communication Technology (PICICT)*, pp. 13–17.

Nwaogu, V. C. and Dimililer, K. (2021). Customer churn prediction for business intelligence using machine learning, *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–7.

Prashanth, C., Chandrasekaran, D., Pandian, B., Duraipandian, K., Chen, T. and Sathiyanarayanan, M. (2022). Reveal: Online fake job advert detection application using machine learning, pp. 1–6.

Shree, R. A., Nirmala, D., Sweatha, S. and Sneha, S. (2021). Ensemble modeling on job scam detection, **1916**(1): 012167.

Sundaram, V., Ahmed, S., Muqtadeer, S. A. and Ravinder Reddy, R. (2021). Emotion analysis in text using tf-idf, *2021 11th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pp. 292–297.

Tabassum, H., Ghosh, G., Atika, A. and Chakrabarty, A. (2021). Detecting online recruitment fraud using machine learning, pp. 472–477.

Tian, Z. and Baskiyar, S. (2021). Fake news detection using machine learning with feature selection, *2021 6th International Conference on Computing, Communication and Security (ICCCS)*, pp. 1–6.

Wang, J. and Liu, B. (2022). Recruitment fraud detection method based on crowdsourcing and multi-feature fusion, *2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 267–273.

Yang, H.-R., Xu, M. and Zheng, N. (2007). An improved classification method for the common ole file by n-gram analysis and vector space model, *2007 IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN07)*, pp. 983–986.