FAKE JOB POST PREDICTION

MSc Research Project
MSc Data Analytics

Aishwarya Dinesh Rathudi
Student ID: X21222762

School of Computing
National College of Ireland

Supervisor: Abubakr Siddig

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Aishwarya Dinesh Rathudi ……………………………………………………………………………………………………

Student ID: x21222762 ……………………………………………………………………………………………………..……

Programme: MSc Data Analytics ……………………………………………………… **Year:** 2022-2023 …………………………..

Module: MSc Research Project …………………………………………………………………………………………….………

Supervisor: Abubakr Siddig …………………………………………………………………………………………….………

Submission Due Date: 18-09-2023 …………………………………………………………………………………………….………

Project Title: Fake Job Post Prediction …………………………………………………………………………………………….………

Word Count: 9024 ……………………………………… **Page Count** 25 …………………………………………….……..

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: Aishwarya Dinesh Rathudi ……………………………………………………………………………………………………

Date: 18-09-2023 ……………………………………………………………………………………………………

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):	

FAKE JOB POST PREDICTION

Aishwarya Dinesh Rathudi
X21222762

Abstract

Online job platforms have revolutionized the job search process in the modern digital age by providing a wide range employment option. The growing number of employment scams raises concerns about the authenticity of these platforms. Such fraudulent job postings take the advantage of job seekers, leading to money losses, identity theft, and lost opportunities for employment. Addressing this issue is crucial for both job seekers and platform owners to maintain trust and credibility in the online job market. This study's motivation lies in addressing the impact of these scams on job seekers trust and online platform credibility. The research aims to develop predictive models using advanced machine learning techniques to effectively identify and prevent fraudulent job listings. This study successfully employs advanced neural network architectures, Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM), and natural language processing (NLP) technique like word2vec, to tackle the challenge of detecting fake job postings. The proposed models achieve remarkable accuracy rates of 97.18% for LSTM and 96.86% for BiLSTM by rigorous data preparation, feature engineering, and hyperparameter tuning. This work enhances the safety of online job hunting, promoting user confidence while protecting against potential scams, by utilizing cutting-edge machine learning models.

Keywords: Fake Job postings, Natural Language Processing (NLP), machine learning, job scams, LSTM, Bidirectional LSTM.

1 Introduction

Job searching platforms have transformed the way job searchers contact with potential employers in the modern digital era. They have become an essential component of the job market. The job search procedure is more accessible and effective because to these platforms' wide range of employment prospects. Despite this ease, there is rising worry about the prevalence of fake job advertisements on these networks. False job postings, commonly referred to as job scams, are dishonest advertisements that prey on job searchers and result in money losses, identity theft, and ruined employment opportunities. Both job searchers and the owners of online job portals are increasingly concerned about the rise in fake job listings. Job seekers devote time, effort, and personal information to applying for positions they believe are legitimate in the hopes of obtaining suitable work possibilities. People who fall for a false job listing are at risk of victims of identity fraud or falling into monetary scams, both of which can have serious consequences. It is essential to identify and prevent such fraudulent activities if we want to guarantee the reliability and legality of online job marketplaces.

1.1 Motivation and Project Background

The significant impact that false job posting has on both job seekers and online a jobplatform serves as the driving force for this research. In exploring potential employment prospects, job searchers invest a lot of time and energy, thus it can be devastating to fall fake job postings. According to a FlexJobs survey from 2015, there are roughly 60–70 fraudulent job postings for every real job ad. just 48% of the applicants were aware of online recruitment fraud

(Tabassum, 2021), indicating that job searchers are not aware of how common false job advertising are. Scammers use sophisticated techniques to create false job postings that closely resemble real opportunities, taking advantage of the vulnerability that thisignorance causes. They take advantage of job seekers ambitions and desire to land a job, causing them to provide personal information and even pay money for fictitious job offers. Many job searchers consequently suffer monetary losses, have their identities compromised, and get discouraged with the online job-seeking process.

The project emphasizes the need of developing effective predictive models to detect and prevent fake job postings on online job platforms. It gets harder for job hunters to distinguish between legitimate job opportunities and fraudulent ones as technology develops and scammers grow more devious in their methods. This emphasizes the necessity for reliable machine learning algorithms, especially those that incorporate natural language processing, to analyse the content of job postings and extract insightful knowledge to identify fraudulent patterns. By creating precise predictive models for identifying fake job ads, this research's main objective is to help make the world of online job searching safer and more secure. This will protect job searchers from potential frauds and increase user trust in the site.

1.2 Research Question

How can machine learning algorithms be used to identify and detect fraudulent job postings effectively?

To accurately identify and detect fraudulent job listings, this research question explores the use of machine learning approaches. The goal of the study is to create an automated system that can effectively distinguish between genuine and false job listings, improving the entire job-seeking experience and protecting job seekers from potential scams.

1.3 Contribution to the Scientific Literature

This study contributes by offering a reliable and useful method for identifying fake job postings on online job boards. We intend to improve the safety and reliability of job-seeking experiences for users by utilizing machine learning techniques and NLP. This study will also throw light on the significance of being vigilant against fraud and add to the expanding body of knowledge on the detection of fraudulent activity in online platforms.

1.4 Research Objectives

1) literature review: Conduct a comprehensive review of existing literature on the detection of fake job postings.

2) Data Collection and Preprocessing: Gather a dataset of fake job postings, perform data preprocessing, including text cleaning, and feature engineering, to prepare the dataset for model training and evaluation.

3) model development and evaluation of Recurrent Neural Network (RNN) Model using Long Short-Term Memory (LSTM)

4) model optimization: Fine-tune the LSTM-based RNN model by experimenting with different hyperparameters.

4) Comparison of developed models with past research.

The rest of the report is structured as follows; chapter 2 presents a critical analysis of existingliterature on the fake job post detection, chapter 3 presents the methodology followed by feature extraction and chapter 4 presents the implementation, followed by evaluation, and results of machine learning model.

2. Related Work

In recent years, as job posting fraud has increased in prevalence, the identification of false job postings has received a lot of attention. An extensive analysis of the literature reveals that numerous strategies and techniques are employed in this sector. We will evaluate eachreviewed paper's detection methods, impact on job seekers and employers, strengths and weaknesses, Summary of Findings, and Research Justification. The outcomes of the literatureassessment will be the basis for justifying the necessity of our research question and the distinctive contributions our study intends to provide.

2.1 Detection Techniques for Fake Job Postings

Researchers have investigated several methods to identify and prevent fake job posts on online job boards in recent years. This section offers a critical analysis of the various approaches and detection techniques used in the literature.

(Tabassum, 2021) proposed a solution that involves creating a specialized dataset for the Bangladesh job market and applying various machine learning algorithms. They evaluate algorithms like Random Forest, LightGBM, Gradient Boosting, etc., to find the mosteffective one. The paper highlights the significance of features like job title, location, salary range, and others in detecting fraud. The results show that accuracy ranges from 94% to 96%.The paper's strength is its practical strategy for tackling online recruiting fraud, which takes into consideration the specifics of the local labour market. The study aims to increase accuracyby combining traditional and modern techniques. The paper does, however, have several limitations. It fails to fully clarify the difficulties specific to the employment market in Bangladesh and does not properly justify the selection of algorithms. The suggested model's evaluation is based just on accuracy, which may not be the most complete measure for fraud detection.

(Nasser, 2021) used Artificial Neural Network based model to detectfraud job posts using a Public Employment Scam Aegean dataset (EMSCAD). The authors evaluated the model and the results showed that it achieved recall and f-measure of 96.02% and 93.88% respectively. The model also achieved the NPV score of 95.67% and f-score of 93.36% respectively. The use of an Artificial Neural Network (ANN) model is a strong aspectof this research, as ANNs have shown success in capturing complex patterns in data. However,some weaknesses are apparent. The absence of specificity and precision metrics hinders a comprehensive evaluation.

Another paper (FHA. Shibly, 2021) presents a comparative analysis of two machine learning algorithms, the two-class boosted decision tree and the two-class decision forest algorithm, in detecting fake job postings. It compares two machine learning algorithms, using metrics like Accuracy, Precision, Recall, and F1 Score. The two-class boosted decision tree algorithm achieved an accuracy of 93.8% and two-class decision forest algorithm achieved an accuracy of 95.4%. The boosted decision tree algorithm outperforms the decision forest in identifying fake job postings. In this study, the boosted decision tree algorithm achieves higher recall, precision, and F1 Score, indicating its better ability to correctly classify both positive (fake job postings) and negative (legitimate job postings) instances, making it more suitable for identifying fake job postings.

The presented paper offers several strengths in its approach, the paper's use of well-established machine learning algorithms and evaluation metrics, such as Accuracy, Precision, Recall, and F1 Score, enhances the reliability of its findings. By directly comparing the two algorithms, the paper provides valuable insights into their respective strengths and weaknesses, aiding readers in understanding their relative performance. The paper evaluates the algorithms across multiple performance metrics, including precision, recall, and F1 Score. This comprehensive analysis provides a holistic understanding of their effectiveness. While the paper demonstrates several strengths, there are also some areas that could be improved, the two-class boosted decision tree and two-class decision forest algorithms are selected for comparison in the research. The reader might not fully understand the justification for this option because there is not enough information provided. The paper compares the metrics of the algorithms; however, it doesn't go into detail about the drawbacks of the selected algorithms or the possible causes of their performance gaps.

(Nessa, 2022) uses a one-layer Gated Recurrent Unit (GRU) model to address the critical challenge of identifying fake job postings. The study uses the Employment Scam Aegean Dataset (EMSCAD) to build and evaluate the model, and it achieves a remarkable AUC score of 93.51%. The strength of this paper includes it detailed approach, the technique,including data preparation, model construction, and evaluation using suitable metrics, is thoroughly described by the authors. The EMSCAD dataset is fully described, guaranteeing the study's accessibility and reliability. The research applies modern methods for text analysis and classification using NLP methodologies and the GRU model. However, the model's limitation lies in its inability to classify based on certain attributes such as location and employment type. The model's applicability to more complicated datasets may be questioned given that the study's research was limited to only five attributes.

(Naudé, 2023) addresses the growing problem of fake job postings, which can be harmful to job seekers. The Employment Scam Aegean Dataset (EMSCAD), which contains of 17,880 job openings gathered during 2012 and 2014, was used by researchers to examine different characteristics and classifiers. To categorize various sorts of fraudulent jobs based onspecified criteria, they introduced a categorical variable called "type" with four possible values. For different machine learning classifiers, they used features from four different classes: empirical rule set-based features, bag-of-word models, most recent state-of-the-art wordembeddings, and transformer models. Transformer models and word embeddings have been performed well. The model uses a Gradient Boosting classifier with a combination of rule-setfeatures, part-of-speech tags, and bag-of-words vectors to reach the best F1-score of 0.88. Thispaper's strength lies in its interpretability and range of features. In addition to exploring numerous feature classes and offering a thorough examination of various feature types, the study also highlights the significance of feature interpretability. Dataset homogeneity and lackof real-time data are some of its drawbacks. Due to its time-bound and platform-specific nature, the dataset utilized for model training and testing may lack diversity. The information collection period ends in 2014, therefore it may not accurately reflect how online employmentscams have changed over the past few years.

(Keerthana, 2021) addresses the use of several machine learning classification approaches to differentiate between fraudulent and legitimate job advertisements and provides a thorough analysis of false job listings. The authors make use of a dataset that is openly accessible and contains about 17,880 job descriptions, 800 of which are false. On thedataset, the study evaluates several machine learning methods, such as Logistic Regression (with TFIDF and Count Vectorizer), K-Nearest Neighbour (KNN), Support Vector Classifier(SVC), Random Forest, and Neural Networks (MLP Classifier with "lbfgs" and "adam" solvers). The study shows that when it comes to classifying fake job advertisements, Neural Networks with the "adam" solver had the highest accuracy (71%). This paper's notable strength is the use of various feature engineering approaches like one-hot encoding, TFIDF Vectorizer, and Count Vectorizer. These methods enable a thorough investigation of ways to improve model performance. Another important strength of the paper is its comparison of several categorization algorithms. The distinctive benefits and drawbacks of each algorithm are clarified by this research, making it simpler to select the ideal model.

(Lilapati Waikhoma, 2019) discusses the rising issue of fake news spreading online. Through an ensemble technique, it uses the LIAR dataset from POLITIFACT.COM to increase the accuracy of fake news detection. To identify fake news in political words, thestudy makes use of the LIAR dataset. The collection includes labelled data and URLs to source materials, and it covers 12.8k samples from 2007 to 2016. Precision, recall, and F1- score accuracy scores for the study, which employs a Bagging Classifier and XGBoost, were recorded at 39%. This result showed a moderate level of performance in identifying fake news. However, significant improvement was seen after the Bagging Classifier and AdaBoostwere used. Precision, F1-score, and recall accuracy levels increased to 70%. Over 150 model iterations, these improved results were reliably repeated. This paper's use of real-world data and the ensemble technique are its strong points. Implementation and the choice to reduce thecomplexity of various truth levels by classifying assertions as true or false into a binary classification is a practical and realistic method. Lack of Comparative Analysis is one of this paper's weaknesses. The research doesn't include a comparison with other cutting-edge methods or datasets for fake news identification, and Small-scale Model Exploration, the paper largely focuses on ensemble techniques. Expanding the use of hybrid methodologies ora wider range of machine learning algorithms may help to improve the detection of fake news.

(Lal, 2019) proposed an ensemble learning-based model called ORFDetector tohandle the issue of Online Recruitment Fraud (ORF) detection. A publicly accessible dataset of annotated job listings is used to test the model. Three base classifiers and three ensemble approaches are combined to generate the ORFDetector model. As standard classifiers, the paper uses the logistic regression (LR), J48 decision tree, and random forest (RF) methods.

To further improve prediction accuracy, ensemble approaches including Average Vote (AV), Majority Vote (MV), and Maximum Vote (MXV) are used. The suggested ORFDetector model has a 95.4% accuracy rate for detecting online recruiting fraud (ORF). This paper's key strength is its proficient use of ensemble approaches, which combine many base classifiers. This strategy is especially beneficial since it makes use of the many advantages that each classifier has, which improves predicted accuracy and performance. The proposed model's capacity to successfully handle imbalanced data is another important attribute. The model performs well on an imbalanced dataset, a situation that is frequently found in real- world applications, demonstrating its robustness. However, a potential weakness of the papercould be the lack of an in-depth exploration of the potential limitations of the proposed model. While the results are promising, a more thorough study of probable failure cases or situations where the model might have trouble could give a greater understanding of its limitations.

(V.Mahitha, 2023) addresses the challenge of minimizing fraudulent job postings by utilizing machine learning techniques to predict the likelihood of a job being fake. The proposed model employs Natural Language Processing (NLP) to analyze sentiment and patterns in job postings. The model is developed as a Sequential Neural Network trained withthe GloVe algorithm. The study evaluates the proposed system using five different machine learning algorithms: Decision Trees, Random Forests, Naive Bayes, K Nearest Neighbours, and Gradient Boost. The accuracy of each algorithm varies, with Decision Tree achieving 94.96%, Naive Bayes at 59.96%, K Nearest Neighbours reaching 94.59%, and the highest accuracy achieved by Random Forest at 96.84%. Comparison of Different Algorithm is this paper's strongest point. The study thoroughly assesses the proposed system by putting it to the test with various machine learning algorithms, offering a detailed analysis of each algorithm's performance individually and the model uses NLP approaches to examine the patterns and sentiment in job advertisements, improving its capacity to spot fake information.However, there are some notable drawbacks are the paper does not provide a comprehensive explanation of the feature selection process or the specific attributes used to predict fake job postings. This lack of detail raises questions about the relevance and effectiveness of the chosen features. The study briefly touches on the interpretability of Decision Trees, but for more complex models like the Sequential Neural Network, there is a lack of discussion on methods to explain the model's predictions to users. (Srivastava, 2022), the author proposes the use of several predictive models, includingSupport Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest, Nave Bayes, and Logistic Regression, to identify fraudulent job listings to combat Online Recruitment Fraud (ORF). To evaluate the performance of these models, the study uses a dataset of 17,780 job advertisements and 14 characteristics. Five predictive models are used to categorize job ads that are fake, including Logistic Regression, SVM, Naive Bayes, ANN, and Random Forest. With an accuracy of about 95.2%, Random Forest stands out as the top performer. The paper's diversity of algorithms and Robustness is two of its strong points. Thestudy used a variety of prediction models, increasing the likelihood that useful models for fraud detection would be discovered. The Random Forest model showed a strong capacity to distinguish between fake and legitimate job ads, highlighting its ability in dealing with complex and varied data patterns. However, the study has limitations including potential databiases, imbalanced data, and overfitting risks.

(Vidros, 2017) discusses the issue of online employment scams, which are on therise, and suggests a strategy to identify them using text mining and machine learning. They provide a brand-new dataset called EMSCAD that contains 17,880 job advertisements that have been classified as genuine or fraudulent. In the study, the dataset will be analysed usinga variety of methods, such as text analysis and machine learning models. Six classifiers— ZeroR, OneR, Naive Bayes, J48 decision trees, logistic regression, and random forest—are used by the authors to assess text fields in the dataset using a bag of words model. The Random Forest classifier served as the main algorithm for their machine learning analysis.

They experimented with other algorithms as well, but Random Forest produced the greatest outcomes. The experiments in the study demonstrated that the model achieved an accuracy of around 89.5%. The paper's strength includes Empirical Analysis: The study employs a variety of methods, such as text analysis and machine learning models, to undertake a thorough analysis of the dataset. This empirical methodology offers useful insights into the traits offake job postings. Text analysis, metadata, and machine learning are all included in the paper's multifaceted detection approach to create a ruleset-based detection model. This comprehensive strategy makes use of numerous characteristics of job advertisements to increase the efficacy of fraud detection. The weaknesses of this paper are Limited Generalizability and Imbalanced Dataset.

2.2 Impact on Job Seekers and Employers

The impact of fake job postings goes beyond the limits of detection accuracy and influence both employers and job searchers. The rise of false advertising wastes the time and resources of job seekers who rely on job advertisements to find acceptable possibilities. Job searchers who are trying to land legitimate employment may become confused and frustrated because of these false postings (Sultana Umme Habiba, 2021). The 2030 vision for Saudi Arabia projects significant job growth (Alghamdi, 2019). But along with this expansion comes the problem of assuring cybercrime prevention during the hiring process. Online hiring is advantageous for both employers and candidates. However, fraudsters have just launched thisonline recruitment market, creating a new sort of fraud known as Online Recruitment Fraud (ORF). In ORF, spammers approach job prospects with enticing offers of employment while stealing their cash and personal data. ORF is not only bad for users; it's also bad for businesses. As a result, businesses suffer reputational damage, and job searchers have a poor perception of the organization (Lal, 2019). Online recruitment fraud (ORF) is a particular kind of cybercrime that has evolved as a threat to people's privacy and financial security.

ORF entails taking advantage of web services and internet technologies to deceive job seekers and damage the reputation of companies. By enabling the creation of effective fraud detection models, data mining techniques have been essential in the fight against cybercrime.Online Employment Fraud violates job seekers privacy, damages businesses reputations, andcauses monetary losses for individuals. It happens when malicious people create false job adverts to control and dupe job searchers. Nearly 700,000 job searchers in the UK reported losing more than $500 000 because of work scams (B. Snidhuja, 2023). The survey indicatedan almost 300% growth in the UK during the previous two years. Students and recent graduates are the main targets of fraudsters since they frequently strive to acquire a stable jobfor which they are prepared to pay more money. Because job scammers continuously adapt their methods, efforts to prevent or deter cybercrime are ineffective. The increase of fake job advertising makes it difficult for businesses and organizations to draw in qualified candidates(K. Swetha, 2023). Because there are so many fraudulent job listings, the hiring process is disrupted by the abundance of opportunities that aren't real. In addition to wasting job searchers' time, this flood of fraudulent posts makes it more difficult for them to locate jobs that truly match their qualifications and career goals. Consequently, a smaller pool of qualified candidates is available to employers because of the proliferation of false job postings.

2.3 Conclusion and Justification

The literature review thoroughly investigates several methods of detecting fake job postings. To address the issue of online recruitment fraud, many studies use a variety of machine learning algorithms, feature engineering techniques, and evaluation criteria. The existing solutions fall short of a comprehensive strategy that can successfully handle the complex problems caused by online recruitment fraud. While some research has excellent model accuracy, others don't go into detail about model flaws or comparative studies. Several papers also deal with incomplete data or apparent biases. To develop a strong and efficient solution for recognizing and detecting fraudulent job postings, a holistic approach that blends the advantages of various strategies while resolving their disadvantages is still urgently needed. This demonstrates the significance of the research question. The goal of the proposed research is to advance the area by creating a more reliable and accurate method for identifying fake job advertisements by developing on the strengths of previous research and resolving their drawbacks. The thorough study supports the necessity for a comprehensive strategy that integrates the benefits of various approaches to improve accuracy, interpretability, and generalizability. The proposed research aims to improve the efficiency ofusing advanced machine learning-based detection strategies to tackle the ongoing difficulty of false job posts in online platforms through a combination of advanced machine learning and NLP techniques.

3 Research Methodology

3.1 Introduction

The Knowledge Discovery in Databases (KDD) methodology is used in this research to extract useful knowledge and insights from data. Starting with data collection, Data Preprocessing (analysis of the raw data, text preprocessing using NLP techniques), transformation of the data, Data Modelling and results analysis using multiple evaluation metrics are all steps in the process. The following Diagram explain each phase reflect the application of the KDD approach.

3.2 KDD Methodology

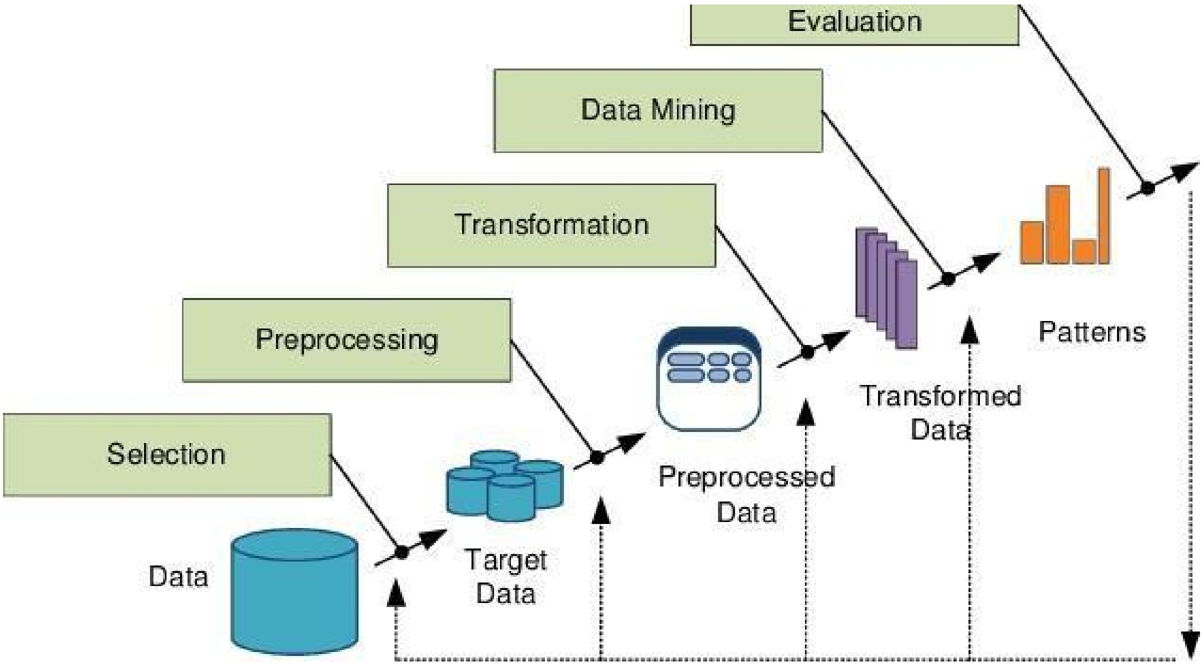


Fig 1. KDD (Costagliola, 2009)

1. Data Collection:
   The data for this research is obtained from Kaggle, a well-known platform for open datasets. The dataset comprises 17,880 rows and 18 columns, includes a diverse collection of job postings from various source. The data consists of both textual information and meta-information about the jobs.
2. Data Preprocessing: Preprocessing data involves a series of essential steps. This process starts with identifying and fixing missing or inconsistent data, which frequently requires imputation of missing values. Since the data is text-based, the firststage involves identifying the word frequencies using tools like word clouds. Stop words are removed, text is changed to lowercase for uniformity, sentences are tokenized into individual words, and lemmatization is done to break down words to their simplest forms to improve textual analysis.
3. Data Transformation: Using word embedding or NLP approaches, data is transformed into numerical vectors that can be used by machine learning algorithms. These numerical vectors are then converted into a matrix-like format for neural networks.

Padding is used to ensure that all sequences are the same length because sequences can vary in length.

4. Data Modelling: After the data has been transformed and prepared, the next step is data modelling. In this phase, the transformed data is divided into separate training and testing sets. Machine learning or deep learning models is used the training data asinput so they may discover patterns and relationships in the data. The model is trainedon the training set, and its performance on new, untested data is evaluated using the testing set.
5. Model Evaluation: It's essential to evaluate the model's performance after training to see how well it generalizes to new, untested data. This stage enables us to determine if the model is overfitting or accurately capturing patterns. Based on accuracy, confusion matrix and classification report, the models are evaluated.

3.4 Choice of models

In this research, Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) Recurrent Neural Networks (RNN) are the primary models of choice. These models were chosen because they can capture sequential dependencies in textual data, which is essential in analysing job descriptions and identifying misleading language.

3.4.1 Long Short-Term Memory (LSTM)

A kind of RNN called the Long Short-Term Memory (LSTM) model was created to address the vanishing gradient issue, enabling it to successfully record long-distance connections in sequences. LSTM cells feature memory units that can retain data over extended periods of time, making them appropriate for analysing job descriptions that could include complex language clues. The kind of textual data present in job postings fits with the LSTM's capacity to learn from sequences and store context information (Waqas Haider Bangyal, 2021).

3.4.2 Bidirectional LSTM (BiLSTM)

The Bidirectional LSTM (BiLSTM) expands the LSTM's capabilities by processing sequences both forward and backward. Using this feature, the model may successfully capture context in both directions by considering both the words that come before and those that come after. The ability to comprehend a job description's whole context is crucial for performing tasks like predicting fake job posts, hence this characteristic is especially helpful in those situations.

3.4.3 Justification for Choice

The sequential pattern of text data in job postings naturally leads to the selection of LSTM and BiLSTM models. A model that is excellent at capturing long-range dependencies is required since the linguistic cues and false patterns may appear in different regions of the text. The difficulties provided by predicting fake job posts can be met by LSTMs and BiLSTMs, which have proven effective in a variety of natural language processing applications. The research objective of identifying misleading content is consistent with theircapacity for processing and learning from sequences.

3.3 Conclusion

The research utilizes the KDD methodology, which offers a structured and systematic strategy for gaining significant insights from data. By doing all the necessary actions, this study aims to solve the limitations mentioned in the existing literature and provide a comprehensive solution to the problem.

4 Design Specification

Several steps are carried out in the prediction of a fake job posting with the goal of efficiently processing and analysing the data, creating, and improving the models, and eventually evaluating the performance. The following figure shows the thorough step-by-step procedure.

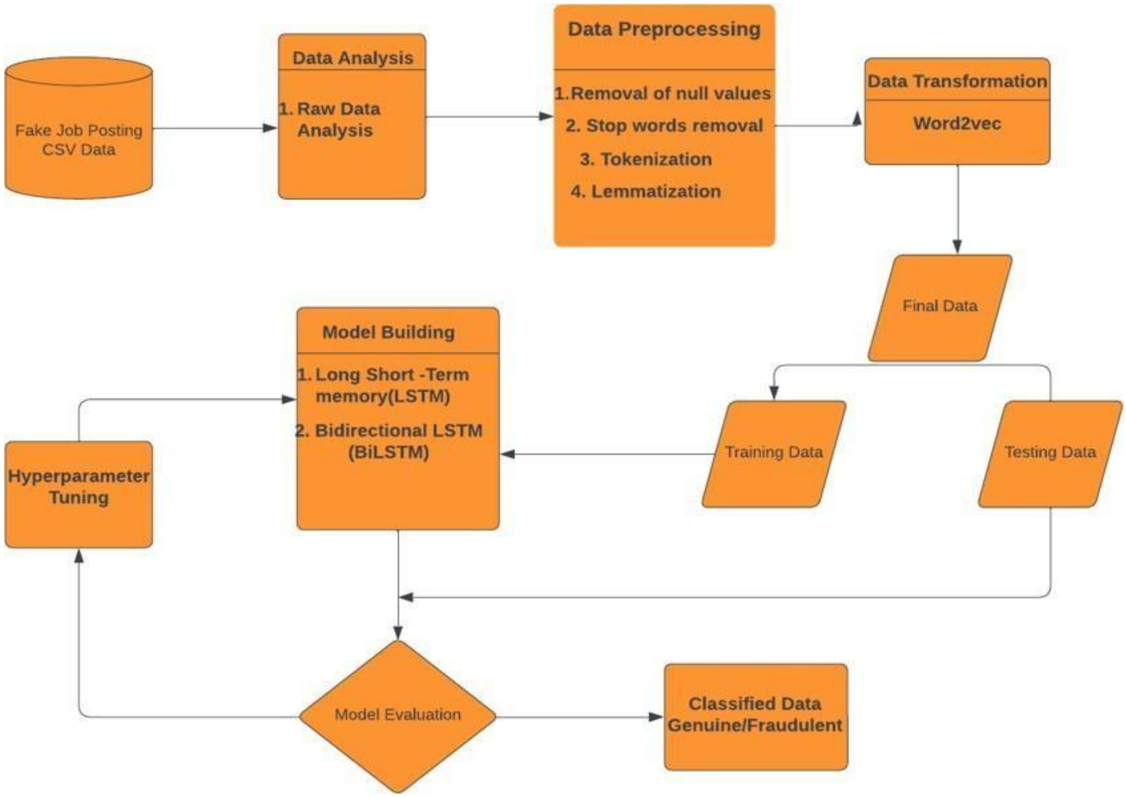


Fig 2. Design Flow of the Project

The project's initial step involved gathering data, which was followed by an initial analysis to learn more about the features of the dataset, such as the presence of null values and irrelevant columns. Exploratory data analysis (EDA) was then carried out to comprehend the distribution of fake job advertisements across different columns. This technique gave significant details about the data's structure and offered direction for the following data processing choices.

After EDA, a data cleaning procedure started, which involved removing unnecessary columns and null values to make the dataset more manageable for analysis. The textual data was processed using the following Natural Language Processing (NLP) steps to get it ready for model construction. The text data was effectively cleaned up because of these procedures, which also included tokenization, lemmatization, and stop word removal.

The data was transformed by using feature extraction techniques, and techniques like word2vec were used to increase the data's representational capability for model training. To make the model construction and model validation process easier, the converted data were divided into separate training and testing sets. Both Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) models were used for the core model implementation. These models were chosen for their relative abilities to comprehend contextual relationships and sequential data. The LSTM model was specifically subjected to hyperparameter tuning, which fine-tunes model parameters to enhance performance.

This stage was finished by applying a variety of evaluation measures to the test dataset. With the help of these measurements, a thorough evaluation of model performance for various classifiers was possible, illuminating how well the models identified fake job listings. The final stage of the process involves applying the trained models to job postings to classify them as fake or legitimate.

Fundamentally, the design specification is a precisely planned process that includes data collection, analysis, cleaning, NLP preprocessing, feature extraction, model selection, hyperparameter tuning, and rigorous performance evaluation. The LSTM and BiLSTM models serve as the system's core, and each step helps to build a strong and trustworthy fakejob posting prediction system.

5 Exploratory Data Analysis (EDA)

EDA is a necessary phase of analysis. It was performed to analyse the dataset structure, relationships, patterns, and attributes. Importing the required libraries and loading the dataset were part of the analysis's initial stage. The dataset is collected from Kaggle and Successfully loaded into Google Colab. The dataset has 18 columns and 17880 rows, of which 866 are fake. Both textual and job-related meta-information make up the data. The dataset is eventually subjected to initial analysis and preprocessing.

Several commands are used to study the dataset's structure and contents. The first five rows of the dataset are shown using head command, providing an overview of the columns and values of the data. The dimensions of the dataset are provided by shape, and info () offers details on the data types and the presence of any missing values in each column. Checking formissing values is crucial, and isna().sum() counts the number of missing values in each column. The results show whether any columns have missing data and help in determining the necessary preprocessing steps.

Using drop(columns=[]), unnecessary columns are eliminated from the dataset. The columns "job_id," "salary_range," "department," "telecommuting," "has_company_logo," and "has_questions" are deleted. To make sure that the dataset has no null values, the remaining columns are then filled with empty strings using fillna(' ').

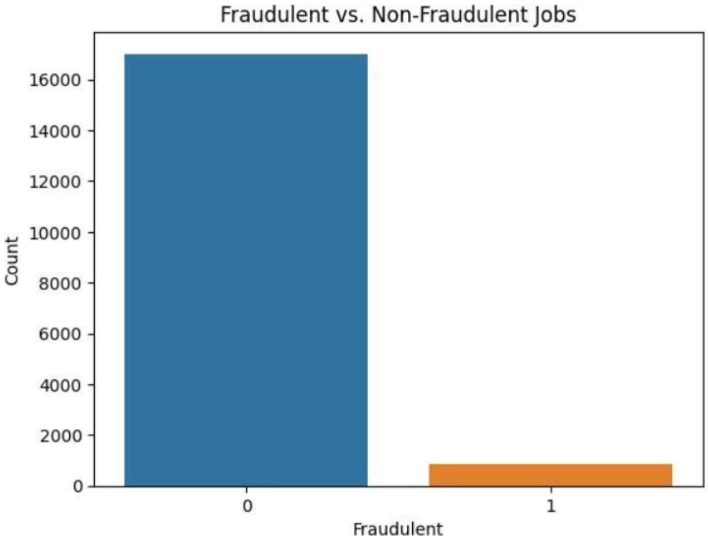


Fig 3.

The distribution of fraudulent and non-fraudulent job postings is visualized using a bar plot. sb.countplot() from the Seaborn library creates a plot that displays the count of each class. This helps in understanding the balance between the classes in the dataset.

The bar graph in above Fig 3. displays the counts, which are 0 is non-fraudulent and 1 is fraudulent. The results show that there are 17014 legitimate job postings (class '0') and 866 instances of fraudulent job advertisements (class '1'). It shows the dataset is highly imbalanced.

To address the class imbalanced in the dataset, I have used class weights during model training.

The country names are extracted from the 'location' column. A lambda function is used to transform the "location" column into a new "country" column. A bar plot is created to visualize job postings by country, providing insights into geographic distribution.

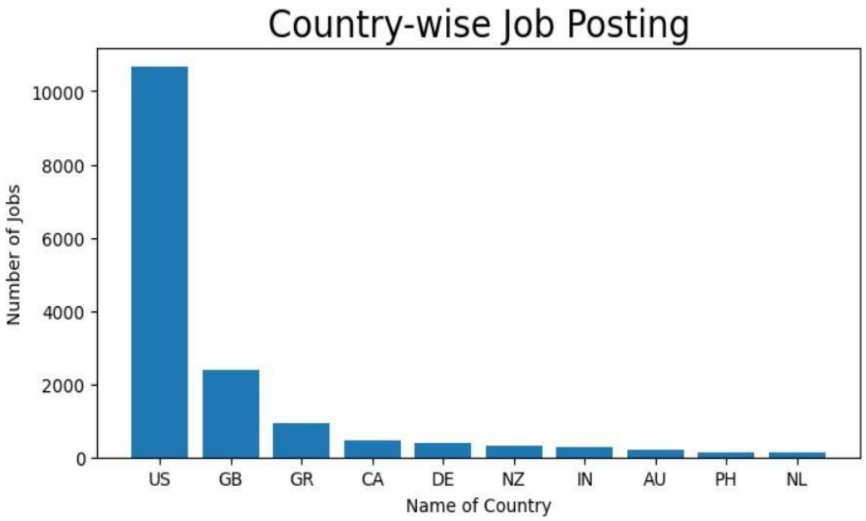


Fig 4.

From this figure 4, it shows that US has the highest number of Job postings, followed by GB(Great Britain).

The visualization of job postings by required experience level follows a similar methodology. To better understand the distribution of experience requirements in the dataset, a bar plot showing the number of job listings for each level of experience is presented.

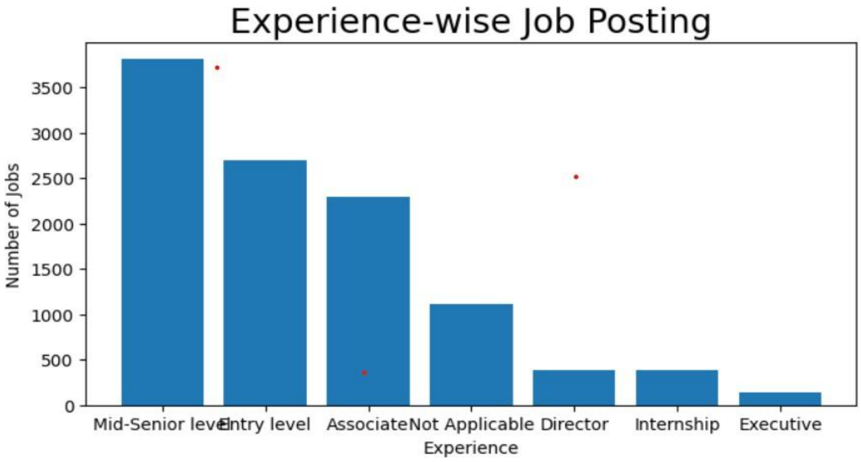


Fig 5.

This Figure 5. illustrates that the Mid-Senior level Experience has the highest number of Jobpostings, followed by Entry level, and then Associate.

To show the most common words throughout the full dataset, genuine job listings, and fake job postings separately, word clouds are generated. These illustrations give a quick overview of common terms.

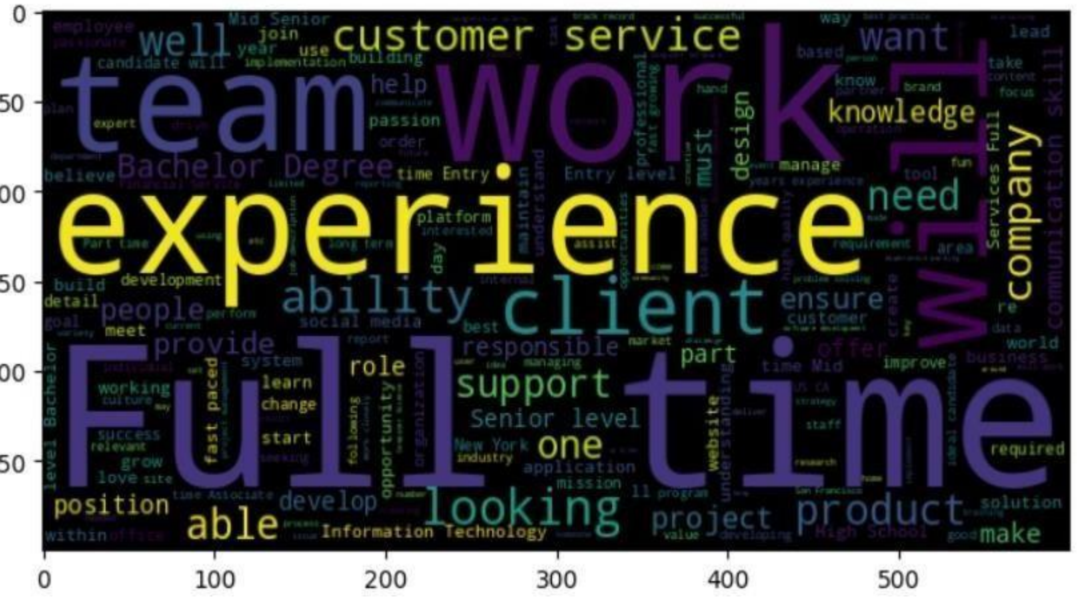


Fig 6.

We can see from the figure 6, that most frequent words are teamwork, experience, full time, customer service, company, client, support.

The most Frequent words in real Job postings

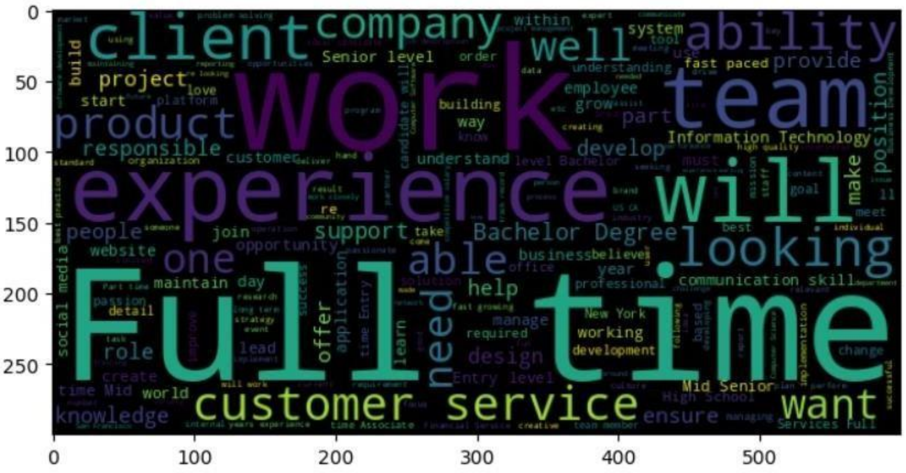


Fig 7.

The most frequent words in Fake Job postings

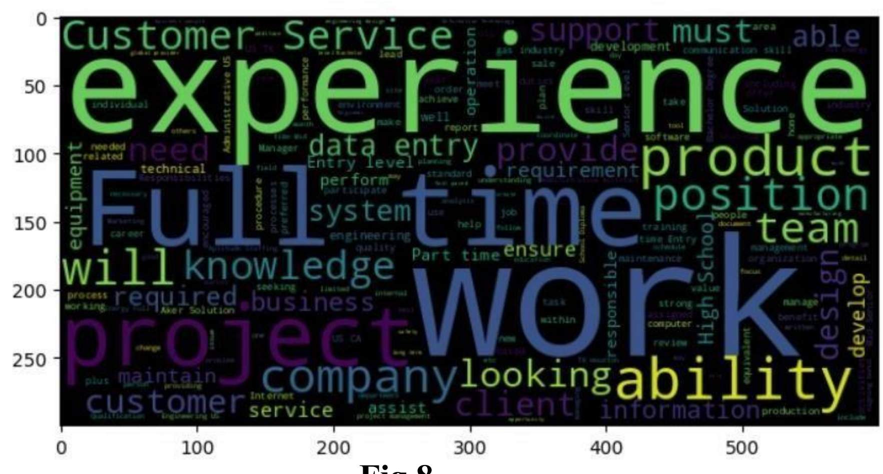


Fig 8.

6 Implementation

6.1 Tools and Platforms used:

The platforms and tools used for fake job post prediction are described in this section. The Google Colab environment, which uses Python to its fullest potential, makes the implementation easier. For data manipulation, visualization, natural language processing, and the development of deep learning models, a wide range of libraries, including Pandas, NumPy, Seaborn, Matplotlib, NLTK, Gensim, TensorFlow, and Keras, are seamlessly integrated. The collaborative and cloud-based features of Google Colab allow for seamless code execution, simple resource management, and cooperative analysis.

6.2 Data Transformation:

These improvements include the creation of an entirely new feature called text, which merges information from several columns. The code concatenates information from many columns, including title, location, benefits, company_profile, description, requirements, industry, employment_type, required_experience, required_education, function, and country. 'Text' is a single textual feature made up of these several kinds of information. This feature's purpose is probably to gather full textual context from various job posting elements.

After creating the "text" feature, the several specific columns have removed. Title, location, company profile, description, requirements, benefits, employment type, necessary education and experience, industry, function, and country are some of these. The aggregated 'text' feature probably contains most of the relevant textual data needed for further analysis and model development; thus, the removals are made to simplify the dataset and reduce complexity. The textual data is prepared for analysis and modelling using a thorough text pretreatment process. Tokenization, lemmatization, and the creation of word embeddings using Gensim's Word2Vec model are a few of the preparation procedures. The following actions are taken by the code.

- Libraries and Downloads are Imported: The required libraries are imported, including re for regular expressions, nltk for natural language processing, and gensim for Word2Vec. Using nltk.download('stopwords'), the NLTK stopwords corpus and other necessary NLTK resources are obtained.

- Stop words: Stop words are frequent words that are often filtered out or eliminated during the preprocessing of text data. Without sacrificing much valuable information, removing stop words can help reduce the complexity of text data and increase the effectiveness of subsequent NLP activities.

- Tokenization and Sentence Preprocessing: Tokenization is the process of breaking a text into small, discrete parts, or tokens, like words or subwords. Tokenization involves splitting a text into smaller pieces, which, depending on the context, may be words, phrases, or even characters. In the code, created an empty list called tokenized_sentences and initializes the 'text' column of the DataFrame to contain the tokenized words for each sentence. Using word_tokenize() from the NLTK package, a loop iterates through each sentence, tokenizes it, and adds the tokenized words to the list.

- Lemmatization and Lowercasing: Lemmatization and lowercase conversion take place on the tokenized words. Lemmatization, which involves reducing words down to their root or basic form, frequently makes use of a word's vocabulary and grammar. This reduces word variants and makes text analysis easier. It Once again combined intocomplete sentences, the processed words are then added to the corpus list.
  Gensim's simple_preprocess() function is used to further process the corpus list. This function uniformly tokenizes and preprocesses the text data.

- Word Embedding using Word2Vec: In natural language processing (NLP), the word embedding approach is used to turn words or tokens into dense vector representations in a continuous vector space. A popular technique for creating word embeddings called Word2Vec is frequently used to record the semantic connections between words. By analysing the context in which words appear inside the text data, Word2Vec learns these embeddings.
  Gensim() library is used to create word Embedding using Word2Vec. Sentences are tokenized and cleaned for use with analysis by first preparing the text input with the gensim.utils.simple_preprocess() function. The Word2Vec model is then trained using the pre-processed word data. To narrow the range of word connections, this training method uses parameters like the window size and minimum word count. After the model has been trained, vocabulary data is extracted. After the model has been trained, vocabulary data is extracted. It is used to get the list of words that are part of the vocabulary for the model. And, model.vector_size is used to determines the dimensionality of the word embeddings.

- Sentence Embeddings:
  The steps deal with creating sentence embeddings using the trained word embeddings. Each sentence pre-processed words are iterated through by the code. It attempts to obtain each word's embedding from the Word2Vec model. The word is skipped if it cannot be found in the model's vocabulary. The code generates a sentence embedding by calculating the mean of all word embeddings in a sentence. An empty list is added to the X list if the sentence is empty. If not, the predicted sentence embedding is added. These sentence embeddings are stored in the list X.

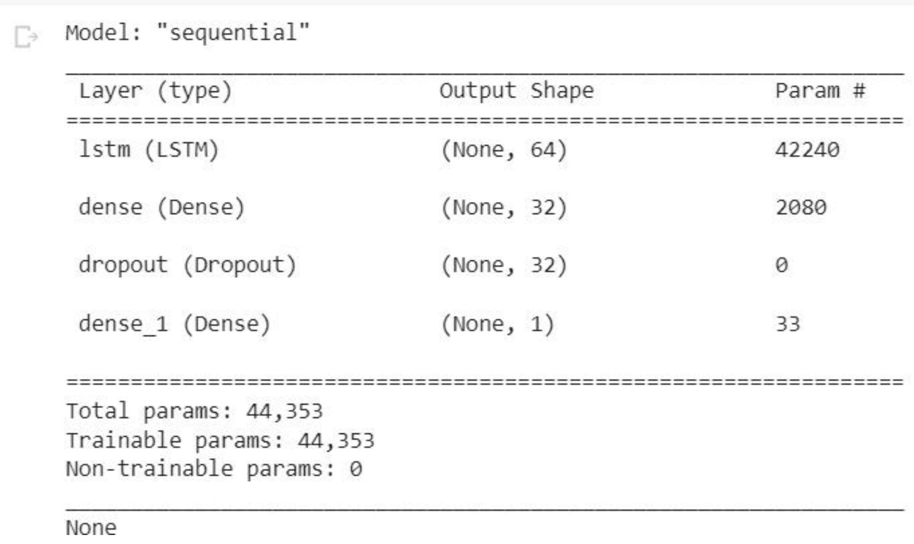
- Padding Sequences:
  The y list contains the target variable "fraudulent." Then, to guarantee that each sentence is the same length, the sentence embeddings are padded using Keras' pad_sequences() method. To feed the data into machine learning models, this is crucial. Sentences that are shorter than the specified length can be "padded" by adding zeros or padding tokens.

6.3 Model Building and Training:

Model 1 – Long Short-Term Memory (LSTM)

TensorFlow Keras is used to build a sequential model. Sequential () from Keras is used to initialize model1. A linear stack of layers is suited for the sequential model. A 64-unit LSTM layer is added, there are 64 LSTM units in this layer. The batch_size, timesteps, and features input dimensions are specified by the input_shape parameters. Followed by a 32-unit Dense layer, A Dense layer with 32 units and ReLU activation is added. To prevent overfitting, a Dropout layer with a dropout rate of 0.3 is included, the output layer with a single neuron and sigmoid activation is added. The model is compiled using binary cross-entropy loss and the Adam optimizer.

The summary of model1 is printed using print (model1.summary()). This summary includes layer names, output shapes, and the number of parameters in each layer. The below figure 9, shows the summary of the model1.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 64)                42240

 dense (Dense)               (None, 32)                2080

 dropout (Dropout)           (None, 32)                0

 dense_1 (Dense)             (None, 1)                 33

=================================================================
Total params: 44,353
Trainable params: 44,353
Non-trainable params: 0
_________________________________________________________________
None
```

Fig 9.

Converting to NumPy Arrays: The lists X and y are converted to NumPy arrays using the functions np.array(X) and np.array(y).

Train-Test Split: Using train_test_split() from scikit-learn, the dataset is divided into training and testing sets. 25% is testing, and 75% is training. X_train, X_test, y_train, and y_test are the resulting sets.

Reshaping Data: The input data is reshaped to have three dimensions (batch_size, timesteps, features). This is done using np.expand_dims() to add an extra dimension to X_train and X_test.

Class Weights: Class weights are defined to handle class imbalance: {0: 1, 1: 18.64}. We used class weights for training the models. Due to the uneven representation of classes, unbalanced datasets can lead to prediction bias. To address this, we trained with lower weights for the majority class and heavier weights for the minority class. As a result, the effect of both classes on the loss function was balanced. Our algorithm was able to learn fromthe minority class more efficiently as a result, which improved forecasts for both classes.

Training the LSTM model:
Using the training set of data, fit() is used to train the model. The number of epochs is set to 10, and the batch size is 32. A validation split of 20% is applied, and class weights are used. After training, the model makes predictions on the test data. The predicted probabilities are stored in y_pred.

Thresholding and Binary Predictions:
After training the model, the model is used to predict the test data, predict method returns an array of predicted probabilities. The Predicted probabilities are converted to binary predictions. Each sample in the test set is given a set of probabilities representing the

predictions. The next step is to use a threshold of 0.5 to translate these probabilities into binary predictions. For classification tasks, where predictions above the threshold are classified as class 1 and those below as class 0, this is essential. The binary predictions arestored in the binary_predictions array.

Model 2 – Bidirectional Long Short-Term Memory (BiLSTM)

The sequential model (model2) is created using the Sequential class, which allows for a linear stack of layers. A bidirectional LSTM layer is added to the model using the Bidirectional wrapper around the LSTM layer. The LSTM layer contains 128 units, which denotes the dimensionality of the output space. This layer captures sequential information from both forward and backward directions. A dense layer with 64 units and ReLU activation follows, introducing non-linearity. A dropout layer with a dropout rate of 0.3 is included to prevent overfitting. The final layer is a dense layer with sigmoid activation, suitable for binary classification tasks. The model is compiled with the binary cross-entropy loss function and the Adam optimizer.

Training the BiLSTM Model:
The BiLSTM model (model 2) is then trained. Training data (X_train_reshaped and y_train), the batch size (32), the number of epochs (10), and the validation split (20%) are all inputs to the fit function, which uses them. To reduce the loss function, the model's weights are adjusted as it learns from the training data. After training, the model is utilized to make predictions on the test data (X_test_reshaped). The predict method returns an array of predicted probabilities for each sample in the test set.

Probabilities to Binary Predictions Conversion:
Each sample in the test set is given a predicted probability array using the predict technique. The probability predictions are then converted into binary predictions using a threshold of 0.5. Predictions above or below the threshold are categorized as class 1 (fraudulent), while those in the range of the threshold are categorized as class 0 (genuine).

7 Evaluation

The model evaluation is an important step to determine the performance of a model. The accuracy of the model gets evaluated using a variety of measures and methods. This research used a combination of techniques for model evaluation, including classification report creation, confusion matrix analysis, and accuracy computation. Let's go through each step of the evaluation process in detail:

7.1 Evaluation of Model 1-(LSTM)

Confusion Matrix: Using the confusion_matrix function, the code generates a confusion matrix. This matrix compares the model's predictions to the test set's actual labels. The number of true positives, true negatives, false positives, and false negatives are all shown below the given figure.

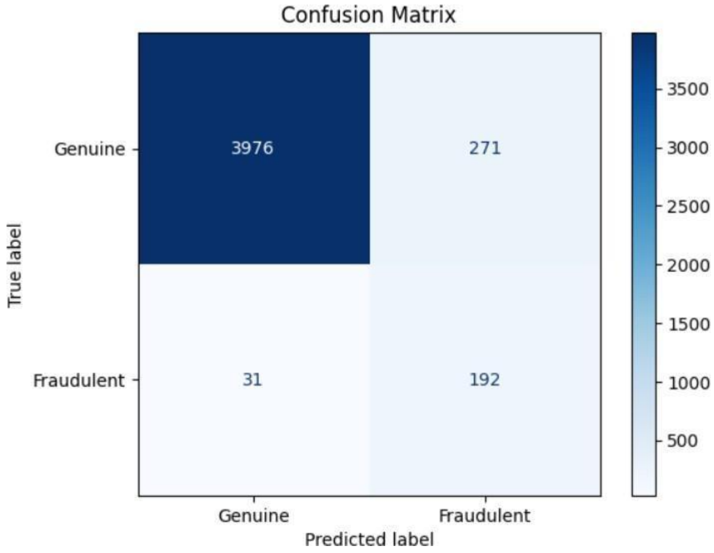
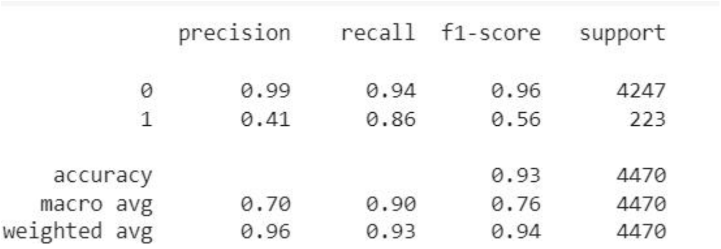


Fig 10.

In a confusion matrix, the columns belong to the predicted classes (predicted labels), whereas the rows represent the actual classes (true labels).
True Negatives (TN): 3976
False Positives (FP): 271
False Negatives (FN): 31
True Positives (TP): 192

Classification Report:
To produce a thorough report on various metrics related to precision, recall, and F1-score for each class (genuine and fraudulent), the classification-report function is used. The above figure offers insightful information about the model's performance according to various evaluation criteria.



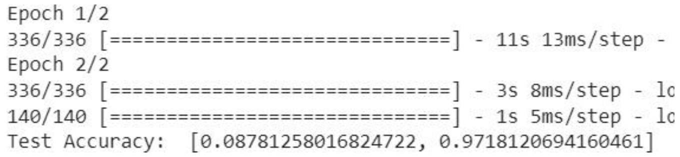
```
              precision    recall  f1-score   support

           0       0.99      0.94      0.96      4247
           1       0.41      0.86      0.56       223

    accuracy                           0.93      4470
   macro avg       0.70      0.90      0.76      4470
weighted avg       0.96      0.93      0.94      4470
```

Fig 11.

Accuracy Calculation: The accuracy of the model is calculated using the accuracy_score function, which measures the ratio of correct predictions to the total number of predictions. The calculated accuracy is displayed as a percentage. In LSTM model, the accuracy achieved is 93.2%.
After training and predicting using LSTM model, an initial accuracy of 93.2% was achieved on the test data. Subsequently, a hyperparameter tuning process was conducted using the Keras Tuner framework to further optimize the model's performance. The hyperparameter search involved tuning key parameters such as the number of units in the dense layer, activation function, and optimizer choice.

Hyperparameter Tuning: The Keras Tuner library is then used to tune hyperparameters. This entails utilizing the HyperModel class to define a unique hypermodel. The hypermodel has hyperparameters such the activation function, optimizer, and number of units. For

hyperparameter search, the Hyperband algorithm is employed.
Using the Best Hyperparameters for Model Training: The model1 is recompiled with the best hyperparameters after hyperparameter adjustment, and then trained once more using the fit method. Finally, the model is evaluated on the test data.



```
Epoch 1/2
336/336 [==============================] - 11s 13ms/step -
Epoch 2/2
336/336 [==============================] - 3s 8ms/step - lc
140/140 [==============================] - 1s 5ms/step - lc
Test Accuracy:  [0.08781258016824722, 0.9718120694160461]
```

Fig 12.

The model accuracy increased to 97.18% with the modified hyperparameters. This denotes a significant improvement in prediction performance, demonstrating the value of hyperparameter tuning in optimizing the model's design and increasing accuracy.

7.2 Evaluation of Model 2-(BiLSTM)

Confusion Matrix:
The confusion_matrix function is used to calculate and display a confusion matrix. The true positive, true negative, false positive, and false negative predictions provided by the model are explained by the confusion matrix figure below.

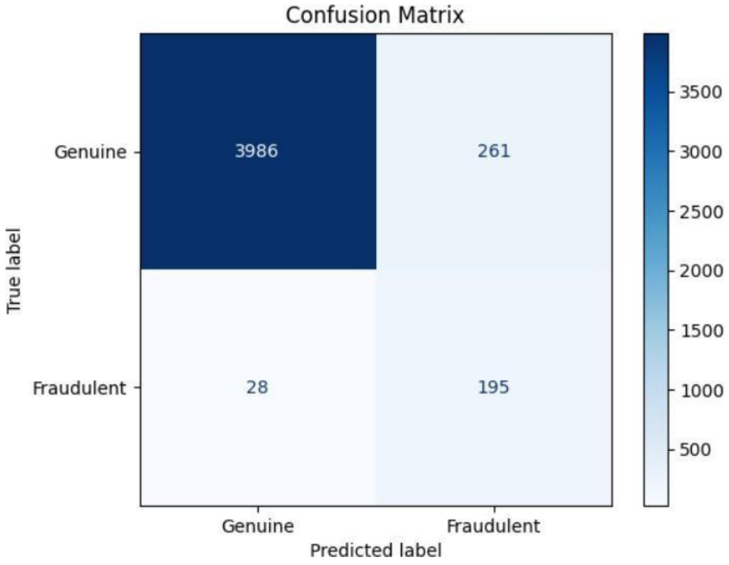


Fig 13.

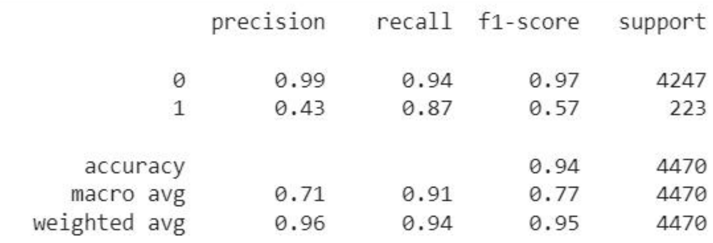
TN: 3986 True Negative
This is the number of cases where the model prediction of the negative class (genuine) was accurate.
FP: 261 False Positive
This shows the quantity of cases that the model expected to be from the positive class (fraudulent) but were really from the negative class (true).

28 False Negatives (FN)

The number of cases that the model expected to be from the negative class but were from the positive class is displayed here.
195 are true positives.
The number of times for which the model positive class prediction was accurate is

represented here.

Classification Report: The classification_report is used to produce a thorough classification report. This report provides several metrics for both classes (genuine and fraudulent), including precision, recall, and F1-score. It offers a comprehensive understanding of the model's effectiveness when measured against several evaluation criteria.

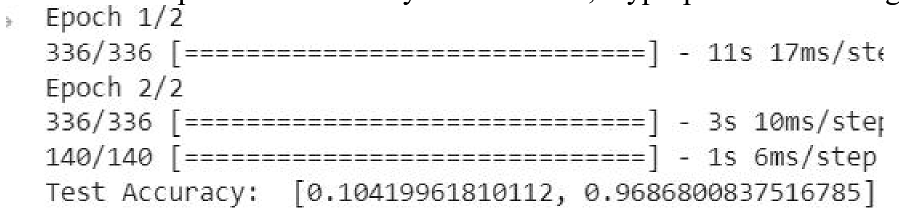


```
              precision    recall  f1-score   support

           0       0.99      0.94      0.97      4247
           1       0.43      0.87      0.57       223

    accuracy                           0.94      4470
   macro avg       0.71      0.91      0.77      4470
weighted avg       0.96      0.94      0.95      4470
```

Fig 14.

Accuracy: the accuracy of the model is calculated using the accuracy_score function. which accepts the true labels and the predicted labels as inputs. The calculated accuracy for this BiLSTM model was 93.5%.

To further improve the accuracy of the model, Hyperparameter tuning was done.



```
Epoch 1/2
336/336 [==============================] - 11s 17ms/ste
Epoch 2/2
336/336 [==============================] - 3s 10ms/ste
140/140 [==============================] - 1s 6ms/step
Test Accuracy:  [0.10419961810112, 0.9686800837516785]
```

After hyperparameter adjustment, it was found that the test accuracy was 96.87%. The optimized model's ability to identify between genuine and false cases can be improved by boosting accuracy through hyperparameter adjustment, which will increase its value in practical applications.

7.3 Discussion

The proposed models, such as Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM), were successfully implemented for the task of predicting fake jobs.

After hyperparameter adjustment, the initial LSTM model's accuracy increased to 97.1% from a promising 93.2%. Like this, the accuracy of the BiLSTM model was 93.5%, which was improved to 96.86% after hyperparameter adjustment. These results highlight how well these algorithms handle the problem of predicting fake jobs. This study highlights the importance of ongoing model optimization and modification, which improves job market transparency and improves the process of identifying fake jobs. These results highlight the potential of cutting-edge neural network designs to overcome difficulties in correctly predicting fake job listings.

7.3.1 Comparison with Previous Research:

Paper	Model	Accuracy	Remarks
(Nessa, 2022)	Gated Recurrent Unit (GRU)	93.51%	Accuracy less than the proposed model
(Tabassum, 2021)	Various ML Algorithms	94% - 96%	Proposed models have a better accuracy
(Keerthana, 2021)	Various ML Algorithms	71%	Low accuracy Focus on ensemble methods.
(FHA. Shibly, 2021)	Decision Trees and Forests	93.8% - 95.4%	Focus on decision tree's strengths, insufficient insight into gaps.
(Lilapati Waikhoma, 2019)	Ensemble techniques	70%	Low accuracy Ensembling led to improved performance.
(Vidros, 2017)	Text mining and ML models	89.5%	Proposed models show better accuracy
(Lal, 2019)	Ensemble learning-based model	95.4%	Limited exploration of model limitations

The proposed model in this study, utilizing advanced NLP techniques, and LSTM and BiLSTM architectures with hyperparameter tuning, outperforms several existing models in terms of accuracy. The approach showcases the significance of optimization and advanced neural network models in achieving high accuracy rates for fraud detection in job postings. The comparison table highlighting the uniqueness of our approach in achieving exceptional accuracy rates.

8 Conclusion and Future Work

In this research, our focus was on addressing the growing concern of fake job postings on online platforms. Leveraging advanced Natural Language Processing (NLP) technique like word2vec and utilizing LSTM and BiLSTM architectures, we achieved remarkable accuracy rates of 97.1% and 96.86% after tuning. These outcomes clearly exceeded the performance of existing approaches, highlighting the effectiveness of combining cutting-edge neural network models with NLP preprocessing. This research not only provides a solid method for identifying fake job posts, but it also highlights the importance of ongoing model optimization. This research contributes significantly to enhancing user trust and security in online job-seeking processes by accurately detecting fake job listings. Even though the research shows encouraging findings for the detection of fake job postings, there are several limitations to be aware of. First off, the models rely a lot on the quality and diversity of the training data. The model's ability to generalize to situations in real life may be impacted by errors or biases in the dataset. Additionally, it is difficult to keep the models current and useful over time due to the dynamic nature of language and the always changing scammers techniques. Despite these drawbacks, our research offers a solid starting point for developingthis field of fraud detection in online job marketplaces.

In the future, further enhancements and extensions to our research can be explored. As the dynamics of work scams change over time, one approach is to investigate the model's potential to be applied in real-time. Maintaining the effectiveness of the models will require ongoing training and monitoring. Further research into ensemble approaches, which combine the strength of advanced machine learning or deep learning algorithms, may result in detection systems that are even more reliable. Additionally, including outside data sources like business profiles and customer reviews may offer richer contextual data for more accurate forecasts.

9 References

Alghamdi, B. a. (2019). An intelligent model for online recruitment fraud detection. *Journal of Information Security*, 155-176.

B. Snidhuja, B. A. (2023). Prediction of Fake Job Ad using NLP-based Multilayer Perceptron. *Turkish Journal of Computer and Mathematics Education*.

Costagliola, G. a. (2009). Monitoring Online Tests through Data Visualization. *IEEE Trans. Knowl. Data Eng.*, 773-784.

FHA. Shibly, U. S. (2021). Performance Comparison of Two Class Boosted Decision Tree snd Two Class Decision Forest Algorithms in Predicting Fake Job Postings. 2462 – 2472.

K. Swetha, M. T. (2023). FAKE JOB DETECTION USING MACHINE LEARNING APPROACH . *Journal of Engineering Sciences*.

Keerthana, B. R. (2021). Accurate Prediction of Fake Job Offers Using Machine Learning. *Machine Intelligence and Soft Computing. Advances in Intelligent Systems and Computing*, 101-112.

Lal, S. a. (2019). ORFDetector: Ensemble Learning Based Online Recruitment Fraud Detection. *2019 Twelfth International Conference on Contemporary Computing (IC3)* (pp. 1-5). Noida, India: IEEE.

Lilapati Waikhoma, R. S. (2019). Fake News Detection Using Machine Learning. *International Conference on Advancements in Computing & Management (ICACM-2019)*, 680-685.

Nasser, I. M. (2021). Online Recruitment Fraud Detection using ANN. *2021 Palestinian International Conference on Information and Communication Technology (PICICT)* (pp. 13-17). Gaza, Palestine: IEEE.

Naudé, M. A. (2023). A machine learning approach to detecting fraudulent job types. *AI & SOCIETY*, 1013-1024.

Nessa, I. a. (2022). Recruitment Scam Detection Using Gated Recurrent Unit. *2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC)* (pp. 445-449). Hyderabad, India: IEEE.

Srivastava, R. (2022). Identification of Online Recruitment Fraud (ORF). *Emirati Journal of Business, Economics and Social Studies*, 42 - 54.

Sultana Umme Habiba, M. K. (2021). A Comparative Study on Fake Job Post Prediction Using Different Data mining Techniques. *Research Gate*.

Tabassum, H. a. (2021). Detecting Online Recruitment Fraud Using Machine Learning. (pp. 472-477). Yogyakarta, Indonesia, 2021: IEEE.

V.Mahitha, T. A. (2023). Fake Job Prediction using Machine Learning. *International Journal of Innovative Research in Science and Engineering*, 91-94.

Vidros, S. a. (2017). Automatic Detection of Online Recruitment Frauds: Characteristics, Methods, and a Public Dataset. *Future Internet*.

Waqas Haider Bangyal, R. Q. (2021). Detection of Fake News Text Classification on COVID-19 Using. *Hindawi Computational and Mathematical Methods in Medicine*.