A project report on

"FAKE JOB POSTING PREDICTION USING MACHINE LEARNING"

Submitted to

VISVESVARAYA TECHNOLOGICAL UNIVERSITY (VTU), BELAGAVI

In partial fulfillment of the requirements for the award of the degree

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

NIRSITHA A P **1SB19CS063**

Under the valuable guidance of

Mrs. C.VALARMATHI
Assistant Professor,
Dept. of CSE.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRI SAIRAM COLLEGE OF ENGINEERING
Anekal, Bengaluru - 562106

SRI SAIRAM COLLEGE OF ENGINEERING,
Anekal, Bengaluru – 562106

Department of Computer Science and Engineering

CERTIFICATE

Certified that the project work entitled

"FAKE JOB POST PREDICTION USING MACHINE LEARNING" is a bonafide work carried out by

NIRSITHA A P **1SB19CS063**

In partial fulfillment for the award of the Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2022-2023. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report deposited in the department library. This project has been approved as it satisfies the academic requirements with respect to the project work prescribed for the Bachelor of Engineering Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
Mrs. C.Valarmathi	**Dr. Smitha J A**	**Dr. B Shadaksharappa**
Asst. Professor, Dept. of CSE	**HOD, Dept. of CSE**	**Principal**

ACKNOWLEDGEMENT

We take this opportunity to thank Our Chairman and Chief Executive Officer **Dr. Sai Prakash Leo Muthu**, and **Dr. Arun Kumar R**, Management Representative of Sri Sairam College of Engineering for providing us with excellent infrastructure that is required for the development of our project.

We take immense pleasure in thanking **Dr. B. Shadaksharappa , Principal, Sri Sairam College of Engineering**, Bengaluru for providing me all the facilities for successful completion of my project.

We are grateful to **Dr.Smitha J A, Professor & Head, Dept. of Computer Science and Engineering**, for his constant motivation, encouragement and guidance to make this project a success.

We would like to express my humble thanks to my project guide **Mrs.C.Valarmathi Associate Professor**, Dept. of Computer Science and Engineering, for guiding me and having facilitated meto complete my project work successfully.

We would like to thank our project Coordinator **Mrs. C.Valarmathi**, **Assisstant Professor**, **Dr. Vinola C, Associate Professor**, Department of Computer Science and Engineering, for their able assistance, timely suggestions and guidance throughout the duration of the project.

We would like to thank all the teaching and non-teaching staff of the Department of Computer Science and Engineering and parents who have contributed to this project directly or indirectly.

DECLARATION

I **NIRSITHA A P,** students of eighth Semester, Computer Science and Engineering, Sri Sairam College of Engineering, hereby declare theproject entitled "**FAKE JOB POST PREDICTION USING MACHINE LEARNING**" has been carried out by us under the guidance of Dr. B. Shadaksharappa, Principal, and Mrs. C.Valarmathi, Assistant Professor, Dept. of CSE,Sri Sairam College of Engineering, Bengaluru, for the partial fulfillment of requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering by Visvesvaraya Technological University, Belagavi during the academic year 2022-23.

Place: Bangalore

ABSTRACT

An automated application that utilizes categorization methods based on machine learning to stop fake job advertising online. The results of several classifiers are compared in order to choose the most effective model for detecting job scams. These classifiers are employed to validate fake online postings. It helps in spotting fake job listings amid many other ones. Single classifiers and ensemble classifiers are the two basic types of classifiers taken into account for the aim of detecting bogus job advertisements. However, experimental results demonstrate that ensemble classifiers are more effective at detecting fraud than single classifiers. In this project, We studied different methods that can be used to detect such fraud job postings as well as applications. This study shows that they can also be done using a convolutional algorithm known as Random-forest and K Nearest Neighbour. These fraudulent job post detections attract a lot of interest in developing an automated method for recognizing bogus jobs and alerting people to them so they won't apply for them. In order to do this, a machine learning technique is used, which makes use of a number of classification techniques. These days, a lot of businesses want to post their vacant positions online so that job hunters may easily find them. However, this could just be a ruse used by con artists to get others to labour for hem in exchange for money. This hoax deceives many individuals, who end up losing a lot of money. By performing an exploratory data analysis on the data and applying the insights obtained, we can distinguish between job ads that are fake and those that are not. A machine learning strategy that makes use of several classification algorithms is employed to detect fake postings

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

CHAPTER 1
INTRODUCTION

Employment scams are one of the serious issues that have lately been addressed in the field of online recruiting frauds (ORF). Many companies now decide to post their vacant positions online so that job seekers can easily and quickly find them. However, this can be a hoax as con artists promise work to job seekers in exchange for money. Fraudulent job advertising may target a respectable company in an effort to harm their standing. There is a significant amount of interest in creating an automated approach for identifying fake job postings and warning individuals about them in order they probably doesn't apply for them. This is accomplished by employing a machine learning approach that recognizes bogus postings by using a variety of classification algorithms. In this case, a classification tool alerts the user after separating bogus job listings from a sizable collection of job postings. Supervisory learning algorithms are first viewed as classification approaches to handle the issue of spotting fraudsters on job advertisements. A classifier converts input variables to target classes using training data. The classifiers employed in the study to separate fake job posts from the others are briefly described. Single classifier predictions and ensemble classifier predictions can be used to generally categorize these classifier-based predictions. The classification approach builds the necessary knowledge base from training data, which is then used to categorize future instances into predetermined categories. In recent years, the use of ML models in data science has exploded. Models are used to solve complicated organizational issues in different fields. It is more important than ever to find ways to detect manipulated fake job posting that are presented as real ones. In this paper, we will study method that perform great and can be used to detect such job postings. Smartphone culture and the incremental growth in social networking site have made the image and videos digitally popular or object in the last decades. Fake job postings make it difficult for job seekers to locate the positions they desire, which is a significant waste of their time. A new avenue for dealing with challenges in the field of human resource management is opened by an automated method to predict fake job posting.

1.1 LITERATURE SURVEY

1.1.1 S. Vidros, C. Kolias , G. Kambourakis ,and L. Akoglu,

"Automatic Detection of Online Recruitment Frauds: Characteristics, Methods, and a Public Dataset"

The critical process of hiring has relatively recently been ported to the cloud. Specifically, the automated systems responsible for completing the recruitment of new employees in an online fashion, aim to make the hiring process more immediate, accurate and cost-efficient. However, the online exposure of such traditional business procedures has introduced new points of failure that may lead to privacy loss for applicants and harm the reputation of organizations. So far, the most common case of Online Recruitment Frauds (ORF), is employment scam. Unlike relevant online fraud problems, the tackling of ORF has not yet received the proper attention, remaining largely unexplored until now. Responding to this need, the work at hand defines and describes the characteristics of this severe and timely novel cyber security research topic. At the same time, it contributes and evaluates the first to our knowledge publicly available dataset of 17,880 annotated job ads, retrieved from the use of a real-life system

1.1.2 B. Alghamdi, F. Alharby

"An Intelligent Model for Online Recruitment Fraud Detection"

This study research attempts to prohibit privacy and loss of money for individuals and organization by creating a reliable model which can detect the fraud exposure in the online recruitment environments. This research presents a major contribution represented in a reliable detection model using ensemble approach based on Random forest classifier to detect Online Recruitment Fraud (ORF). The detection of Online Recruitment Fraud is characterized by other types of electronic fraud detection by its modern and the scarcity of studies on this concept. The researcher proposed the detection model to achieve the objectives of this study. For feature selection, support vector machine method is used and for classification and detection, ensemble classifier using Random Forest is employed. A freely available dataset called Employment Scam Aegean Dataset (EMSCAD) is used to apply the model. Pre-processing step had been applied before the selection and classification adoptions. The results showed an obtained accuracy of 97.41%. Further, the findings presented the main features and important factors in detection purpose include having a company profile feature, having a company logo feature and an industry feature.

1.1.3 Tin Van Huynh1, Kiet Van Nguyen, Ngan Luu-Thuy Nguyen1, and Anh Gia-Tuan Nguyen, "Job Prediction: From Deep Neural Network Models to Applications"

Determining the job is suitable for a student or a person looking for work based on their job descriptions such as knowledge and skills that are difficult, as well as how employers must find ways to choose the candidates that match the job they require. In this paper, we focus on studying the job prediction using different deep neural network models including TextCNN, Bi-GRU-LSTM-CNN, and Bi-GRU-CNN with various pre-trained word embeddings on the IT job dataset. In addition, we proposed a simple and effective ensemble model combining different deep neural network models. Our experimental results illustrated that our proposed ensemble model achieved the highest result with an F1-score of 72.71%. Moreover, we analyze these experimental results to have insights about this problem to find better solutions in the future.

1.1.4 Bowen Dong, Philip S. Yu,

"FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network"

In recent years, due to the booming development of online social networks, fake news for various commercial and political purposes has been appearing in large numbers and widespread in the online world. With deceptive words, online social network users can get infected by these online fake news easily, which has brought about tremendous effects on the offline society already. An important goal in improving the trustworthiness of information in online social networks is to identify the fake news timely. This paper aims at investigating the principles, methodologies and algorithms for detecting fake news articles, creators and subjects from online social networks and evaluating the corresponding performance. This paper addresses the challenges introduced by the unknown characteristics of fake news and diverse connections among news articles, creators and subjects. This paper introduces a novel automatic fake news credibility inference model, namely FAKEDETECTOR. Based on a set of explicit and latent features extracted from the textual information, FAKEDETECTOR builds a deep diffusive network model to learn the representations of news articles, creators and subjects simultaneously. Extensive experiments have been done on a real-world fake news dataset to compare FAKEDETECTOR with several state-of-the-art models, and the experimental results have demonstrated the effectiveness of the proposed model.

1.1.5 Scanlon, J.R. and Gerber, M.S

"Automatic Detection of Cyber Recruitment by Violent Extremists"

Growing use of the Internet as a major means of communication has led to the formation of cyber-communities, which have become increasingly appealing to terrorist groups due to the unregulated nature of Internet communication. Online communities enable violent extremists to increase recruitment by allowing them to build personal relationships with a worldwide audience capable of accessing uncensored content. This article presents methods for identifying the recruitment activities of violent groups within extremist social media websites. Specifically, these methods apply known techniques within supervised learning and natural language processing to the untested task of automatically identifying forum posts intended to recruit new violent extremist members. We used data from the western jihadist website Ansar Al-Jihad Network, which was compiled by the University of Arizona's Dark Web Project. Multiple judges manually annotated a sample of these data, marking 192 randomly sampled posts as recruiting (YES) or non-recruiting (NO). We observed significant agreement between the judges' labels; Cohen's κ=(0.5,0.9) at p=0.01. We tested the feasibility of using naive Bayes models, logistic regression, classification trees, boosting, and support vector machines (SVM) to classify the forum posts. Evaluation with receiver operating characteristic (ROC) curves shows that our SVM classifier achieves an 89% area under the curve (AUC), a significant improvement over the 63% AUC performance achieved by our simplest naive Bayes model (Tukey's test at p=0.05). To our knowledge, this is the first result reported on this task, and our analysis indicates that automatic detection of online terrorist recruitment is a feasible task

1.2 PROBLEM STATEMENT

In recent times, the COVID-19 pandemic has created a perfect storm for employment fraud. With the surge in remote work, job loss, and economic uncertainty, people are more vulnerable to fraudulent job offers. Moreover, scammers are taking advantage of the increased reliance on digital communication channels, making it easier for them to reach a larger audience. To address this dangerous issue, machine learning approaches can be employed. One possible solution is to develop machine learning algorithms that can detect and prevent employment fraud. Such algorithms can analyze large amounts of data,

including job postings, employer information, and candidate profiles, to identify suspicious patterns and behaviors. For instance, machine learning models can be trained to identify job postings that contain certain keywords or phrases commonly used by scammers, such as "work from home" or "no experience necessary." These models can also analyze candidate profiles to identify discrepancies or anomalies in their education, work experience, or other personal information. Employment fraud is becoming more prevalent. Numerous people have experienced much less job loss and economic stress as a result of the coronavirus, Such a situation offers scammers the ideal opening. Due to a rare incidence, many individuals are becoming victims of fraudsters. Most scammers use this technique to obtain personal information from their]victims. Addresses, bank account information, social security numbers, and other personal information. Machine learning approaches can be used to deal with this dangerous issue.

1.3 AIM OF THE PROJECT

The goal of this project is to develop a classifier that can distinguish between phony and legitimate jobs. Two distinct models are used to evaluate the outcome. One model will be applied to the category data and another on the numeric data as the submitted data comprises both text and numeric properties. The two models will be combined to produce the final product. The final model will incorporate any pertinent information about job postings and generate a conclusion about whether the position is legitimate or not in that specific instance. The hardest aspect of our endeavor was figuring out which places were the poster children for fake jobs. For instance, the ratio of phony to actual jobs in major cities is 15:1. locations like this require some extra monitoring.

1.4 EXISTING SYSTEM

The traditional machine learning classifiers (Support Vector Machine Algorithm, or naive algorithms), deep neural networks, convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory (LSTM), and many other techniques can all be used to identify GAN-generated deepfake images.

The conventional machine learning classifiers (Support Vector Machine Algorithm, or naïve algorithms), deep neural networks, convolutional neural networks (CNN), decision trees, deep neural networks (ANN), and many more techniques are available to identify bogus job

postings.

Disadvantages

- The majority of strategies explored were unsuccessful in getting the system to support huge data analysis sets.
- Accuracy and F1 scores are generally lacking in Traditional Convolutional Neural Network Models.
- There are several blogs and websites that provide basic instructions on how to recognize a fraudulent job, but they are absolutely worthless.

1.5 PROPOSED SYSTEM

In this proposed system, We found our approach to be useful since it accurately detects fake job postings, which make it difficult for job seekers to locate the positions they desire and significantly squander their time analyze the best method for fake job detection.

ADVANTAGES OF PROPOSED SYSTEM:

- This approach reduces the number of trainable attribute effectively with less processing time.
- We have achieved approximately 98% classification accuracy (highest) for Random Forest classifier.
- We have analyzed performance analysis parameters also to check if the model works well at both false positive and false negative samples.
- Employment scam detection will guide job-seekers to get only legitimate offers from companies. For tackling employment scam detection, Random Forest Classifier has proved the best prediction results.

ORGANIZATION OF THE PROJECT

Chapter 1: **Introduction** talks about the issue explanation, existing and proposed

frameworks.

Chapter 2: **Software Requirement Specification** lists the equipment and programming determination for this task. It additionally portrays the general depictionof venture, item view point and particular necessities. Here diverse outline requirements, interface and execution necessities clarified.

Chapter 3: **System Design** manages the propelled programming building where the whole stream of the venture is spoken to by expert information stream charts and grouping graphs.

Chapter 4: **Implementation** area clarifies coding rules and framework upkeep for the venture.

Chapter 5: **System Testing** manages the different sorts of experiments to demonstrate thelegitimacy of the venture.

Chapter 6: **Result And Analysis** clarifies in insights about the result of the test and contrasts it and the outcome acquired in existing framework.

Chapter 7: **Conclusion and Future work** this segment depicts the synopsis of the related work and future improvements of the proposed framework.

References: This segment basically highlights all the diaries and contextual analysis

CHAPTER 2

CHAPTER 2
SOFTWARE REQUIREMENT SPECIFICATIONS

A software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRSis developed based the agreement between customer and contractors. It includes the use casesof how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development.

2.1 Software Requirements

SL NO	ITEM	SPECIFICATION
1	Language	Python, HTML
2	Operating System	Windows 10
3	Web Framework	Flask

2.2 Hardware Requirements

1. **System : Pentium i3 Processor.**
2. **Hard Disk : 500 GB.**
3. **Monitor : 15'' LED**
4. **Input Devices : Keyboard, Mouse**
5. **Ram : 4 GB**

CHAPTER 3

CHAPTER 3

SYSTEM DESIGN

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. Systems design services firms plan and design computer systems that integrate computer hardware, software, and communications technologies. They help clients select the right hardware and software products for a particular project, and then develop, install, and implement the system.

HIGH LEVEL DESIGN

High level design is a more general architecture document, where you can find things like relationships between the modules and systems.

LOW LEVEL DESIGN

High level design is a more general architecture document, where you can find things like relationships between the modules and systems.

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.

- Trigger an action.
- Confirm an action.

3.1 SYSTEM ARCHITECTURE

The architectural configuration procedure is concerned with building up a fundamental basic system for a framework. It includes recognizing the real parts of the framework and interchanges between these segments. The beginning configuration procedure of recognizing these subsystems and building up a structure for subsystem control and correspondence is called construction modeling outline and the yield of this outline procedure is a portrayal of the product structural planning. The proposed architecture for this system is given below. It shows the way this system is designed and brief working of the system.

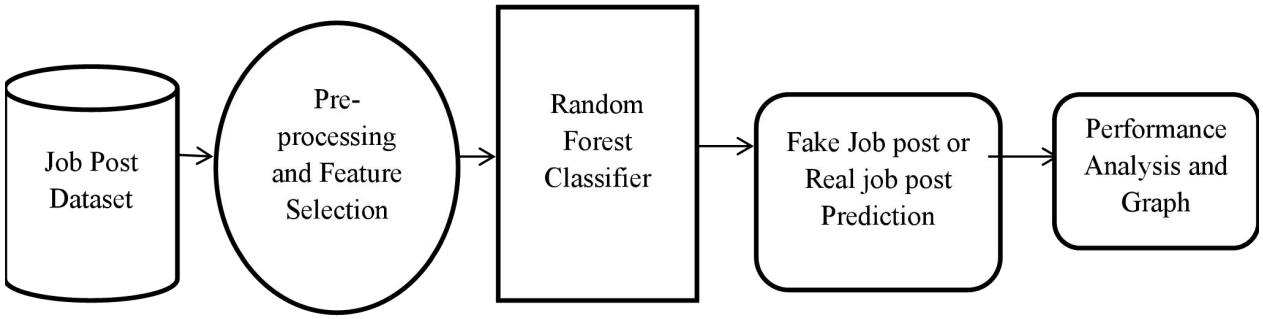

Figure 3.1: System Architecture

3.2 USE CASE DESIGN

A use case diagram at its simplest is a representation of a user's interaction with the systemthat shows the relationship between the user and the different use cases in which the user isinvolved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

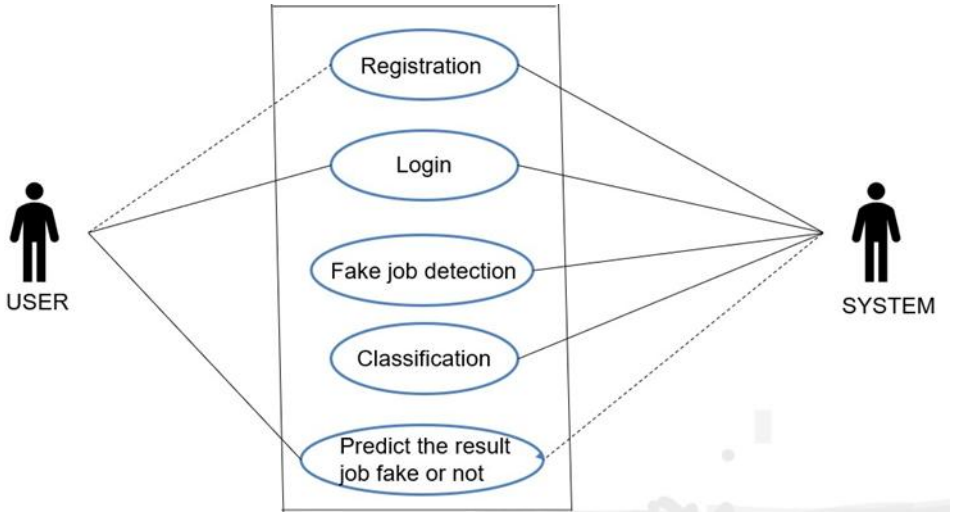

Figure 3.2: Use Case Design

3.3 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenario

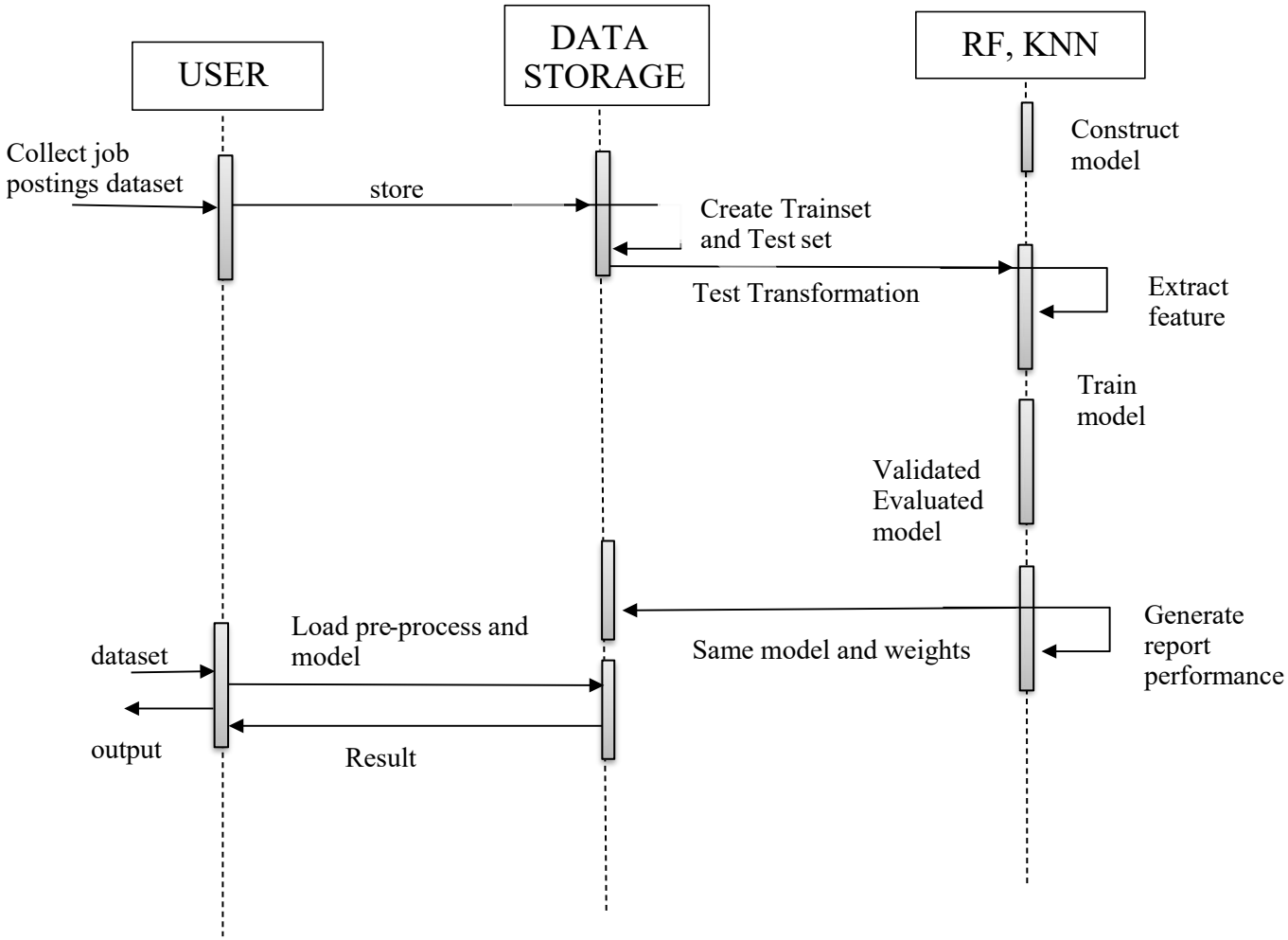

Figure 3.3: Sequence Diagram

CHAPTER 4

CHAPTER 4
IMPLEMENTATION

MODULES:

- Data Collection
- Data Preparation
- Model Selection
- Analyze and Prediction

4.1 MODULES DESCSRIPTION:

4.1.1 Data Collection:

The collection of data is the first significant stage in the construction of a machine learning model. This is a crucial stage that will determine how effective the model is; the more and better data we collect, the more effectively our model will function.There are numerous ways to gather the data, including online scraping, manual interventions and etc. A Comparative Study on Fake Job Post Prediction Using Different Data mining Techniques.

Data set link: https://www.kaggle.com/shivamb/real-or-fake-fake-jobposting-prediction

Dataset:

The dataset consists of 17880 individual data. There are 18 columns in the dataset, which are described below.

1. job_id - unique vacancy identifier
2. title - headline
3. location - the geographical location of the job advertisement
4. department - corporate department (for example, sales)
5. salary_range - indicative salary range (eg 50,000-60,000)
6. company_profile - a short description of the company
7. description - detailed description of the job advertisement
8. requirements - the requirements for the vacancy are listed
9. benefits - the proposed benefits are listed;
10. telecommuting - true for remote posts
11. has_company_logo - true if the company logo is present;
12. has_questions - true if test questions are present
13. employment_type - type of employment;
14. required_experience - necessary experience
15. required_education - necessary education

16. industry - industry
17. function - function to be performed
18. fraudulent - indicates whether the job is fraudulent

4.1.2 Data Preparation:

To prepare data for training, gather it. Remove duplicates, fix problems, deal with missing values, normalise data, convert data types, and other necessary cleaning. Data can be made random, removing the impact of the specific sequence in which it was collected and/or otherwise prepared.

Use visualisation to identify pertinent correlations between variables or class imbalances (bias alert!), or carry out additional exploratory investigation. Create separate training and evaluation sets

4.1.3 Model Selection:

We used Random Forest Classifier algorithm, We got a accuracy of 94.7% on test set so we implemented this algorithm.

Random Forests Algorithm

Random Forest is a popular machine learning algorithm that is used for both classification and regression tasks. It is an ensemble learning method that combines the predictions of multiple decision trees to make more accurate and robust predictions. Here's an overview of how Random Forest works. Random Forest builds multiple decision trees using different subsets of the training data and a random selection of features. Each decision tree is trained independently on a bootstrap sample (randomly selected data points with replacement) from the original training data. During the tree construction process, at each split, only a random subset of features is considered. The final predictions of the Random Forest are obtained by averaging the predictions of all individual decision trees (for regression) or by majority voting (for classification)

How does the algorithm work?

- Four steps make it work:
- Pick samples at random from the dataset provided.
- Create a decision tree for each sample, then analyse the predictions it produces.
- Cast a vote for each expected outcome.

- As the final forecast, choose the outcome that received the most votes.

Advantages:

Random Forest reduces the risk of overfitting by averaging the predictions of multiple decision trees. It can handle noisy and missing data effectively. Random Forest generally provides better accuracy compared to a single decision tree. It can capture complex relationships between variables and make accurate predictions. Random Forest can measure the importance of each feature, enabling feature selection and understanding the relative impact of variables on the prediction. Random Forest is less sensitive to outliers in the data due to the averaging of multiple tree.Random Forest can efficiently handle large datasets with high dimensionality. It can be parallelized to speed up training on multicore processors.

Disadvantages:

Random Forest models are not as easily interpretable as individual decision trees. Understanding the specific decision-making process of a Random Forest can be challenging. Training a Random Forest can be computationally expensive, especially when dealing with a large number of trees and high-dimensional data. Random Forest requires storing multiple decision trees in memory, which can be memory-intensive for large Random forest. Random Forest tends to be biased towards features with a large number of levels or categories since such features tend to have more splits and are more likely to be selected in each tree.

Finding important features

- Random Forest provides a way to measure the importance of features in a prediction task. The importance of a feature is based on how much it contributes to the overall performance of the model.
- One commonly used method to measure feature importance in Random Forest is called Mean Decrease Impurity. For each decision tree in the forest, this method calculates the total reduction in impurity (e.g., Gini impurity) achieved by splitting on a particular feature. The importance of a feature is then computed by averaging the impurity reductions over all trees.
- Another method is Mean Decrease Accuracy. This approach measures the decrease in accuracy when a particular feature is randomly shuffled or permuted. By permuting the values of a feature while keeping the other features unchanged, the importance of the feature is evaluated based on the drop in model accuracy. Features that are more important will result in a greater decrease in accuracy when shuffled.

- After computing the feature importances using either of the above methods, you can create a

feature importance plot to visualize the results. This plot ranks the features based on their importance scores. You can use various visualization libraries (e.g., Matplotlib or Seaborn in Python) to create a bar plot or a heatmap to represent the feature importances.

4.1.4 Analyze and Prediction:

In the actual dataset, we chose only 8 features

1. telecommuting - true for remote posts
2. has_company_logo - true if the company logo is present;
3. has_questions - true if test questions are present
4. employment_type - type of employment;
5. required_experience - necessary experience
6. required_education - necessary education
7. industry - industry
8. function - function to be performed

Result : indicates whether the job is fraudulent

Accuracy on test set:

We got an accuracy of 97.80% on test set.

Saving the Trained Model:

Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or pkl file using a library like pickle.

Make sure you have pickle installed in your environment.

Next, let's import the module and dump the model into.pkl file

Fake Job Description Prediction Dataset:

The dataset consists of 17880 individual data. There are 18 columns in the dataset, which are described below.

1. job_id - unique vacancy identifier
2. title - headline
3. location - the geographical location of the job advertisement
4. department - corporate department (for example, sales)
5. salary_range - indicative salary range (eg 50,000-60,000)
6. company_profile - a short description of the company
7. description - detailed description of the job advertisement
8. requirements - the requirements for the vacancy are listed

9. benefits - the proposed benefits are listed;
10. telecommuting - true for remote posts
11. has_company_logo - true if the company logo is present;
12. has_questions - true if test questions are present
13. employment_type - type of employment;
14. required_experience - necessary experience

15. required_education - necessary education
16. industry - industry
17. function - function to be performed
18. fraudulent - indicates whether the job is fraudulent

Data Preparation:

We will transform the data. by getting rid of missing data and removing some columns. We will create a list of column names that we want to keep or retain. We drop or remove all columns except for the columns that we want to retain. We drop or remove the rows that have missing values from the data set.

Steps to follow:

- Removing extra symbols
- Removing punctuations
- Removing the Stopwords
- Stemming
- Tokenization
- Feature extractions
- TF-IDF vectorizer
- Counter vectorizer with TF-IDF transformer

Model Selection:

We used Decision Tree Classifier algorithms

Decision Tree Classifier:

A decision tree is a tool for decision-making that employs a tree-like structure resembling a flowchart. It serves as a model representing decisions and their potential outcomes, encompassing factors such as input costs, utility, and various results. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

The branches/edges represent the result of the node and the nodes have either:

Conditions [Decision Nodes]

Result [End Nodes]

The branches/edges represent the truth/falsity of the statement and take makes a decision based on that in the example below which shows a decision tree that evaluates the smallest of three numbers:

Decision Tree Regression:

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

Analyze and Prediction:

In the actual dataset, we chose only 2 features

1. Description - detailed description of the job advertisement
2. Fraudulent - indicates whether the job is fraudulent

Accuracy on test set:

We got an accuracy of 95.02% on test set.

Saving the Trained Model:

Once you're confident enough to take your trained and tested model into the production-ready environment, the first step is to save it into a .h5 or . pkl file using a library like pickle . Make sure you have pickle installed in your environment. Next, let's import the module and dump the model into . pkl file

CHAPTER 5

CHAPTER 5

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.2 Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised

Output : identified classes of application outputs must be exercised.

Systems/ Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

5.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. Itis used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

5.5 Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

▪ All field entries must work properly.

▪ Pages must be activated from the identified link.

▪ The entry screen, messages and responses must not be delayed

. **Features to be tested**

▪ Verify that the entries are of the correct format

▪ No duplicate entries should be allowed

▪ All links should take the user to the correct page

5.6 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Test Environment Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested. In our project we aim to predict the plant diseases using their leaves. And we will classify the leaf with their diseases, details of that disease and solution.

Unit Testing Of Modules Unit testing involves the design of test cases that validate that the Internal program logic is functioning properly and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software unit of the application; it is done after the completion of an individual unit before integration. This is a structural testing that relies on knowledge of its construction and its invasive. Unit tests perform basic test at component level and test the specific business process application and/or system configuration. Unit tests ensured that each unit path of a business process performs accurately to the documented specification and contains clearly defined input and expected results.

Module	Input	Expected output	Actual Output	Results
Data preprocessing	Preprocessed Data	Legit/Fake job post	Legit/Fake job post	Pass

5.1 Table Test Case Design

CHAPTER 6

CHAPTER 6
RESULT AND ANALYSIS

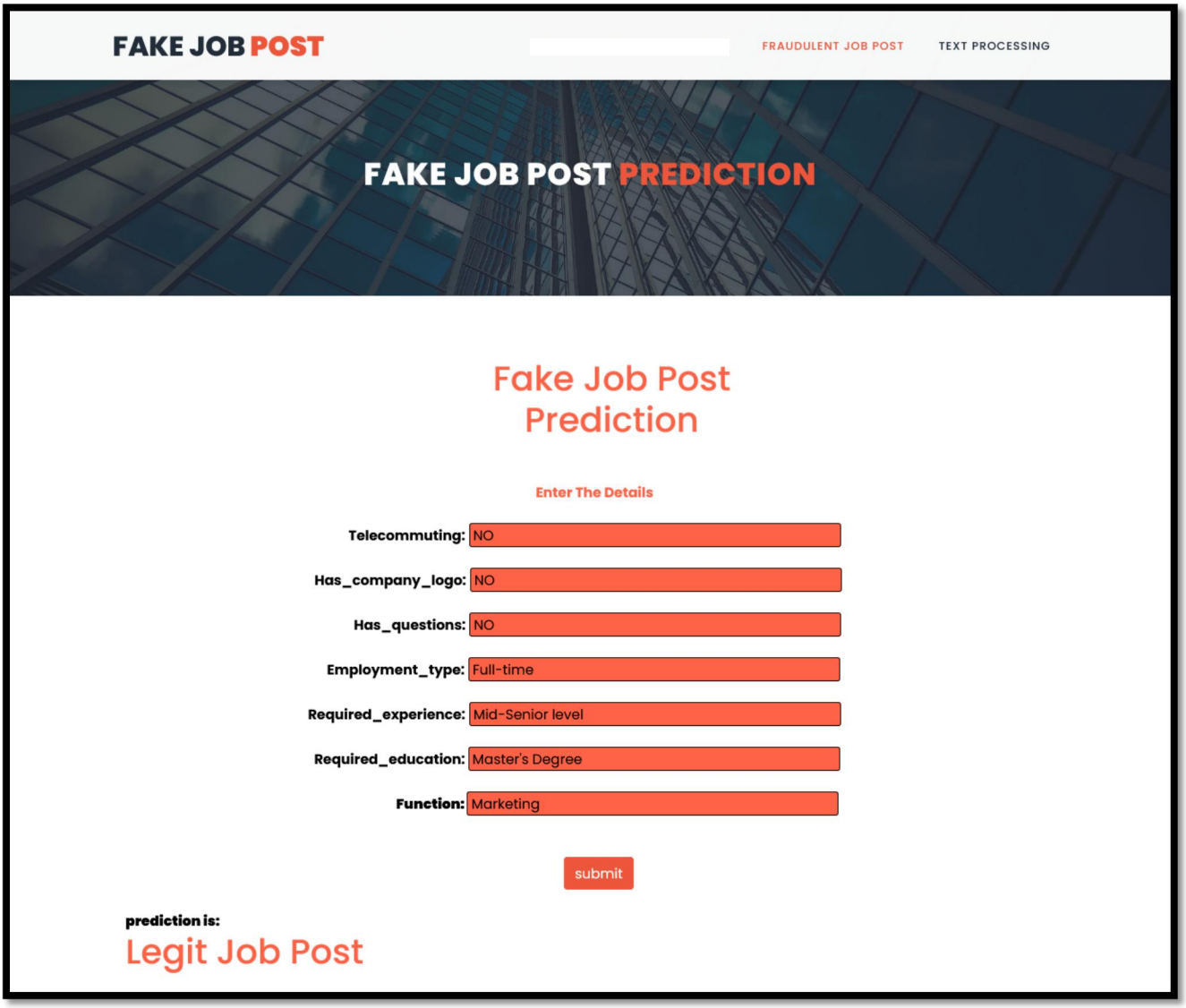

Fig 6.1 Legit job post prediction Page

The Fig 6.1 provides users with insights and predictions about potential job opportunities and career paths based on current trends in the job market. The page may use various data sources and analysis techniques to gather information and generate predictions about which industries and occupations are likely to experience growth or decline in the near future. Users of a legit job prediction page can typically input their skills, interests, and qualifications to receive personalized recommendations for jobs and industries that match their profile. The page may also provide information on the educational requirements, job duties, and salary expectations for various occupations.

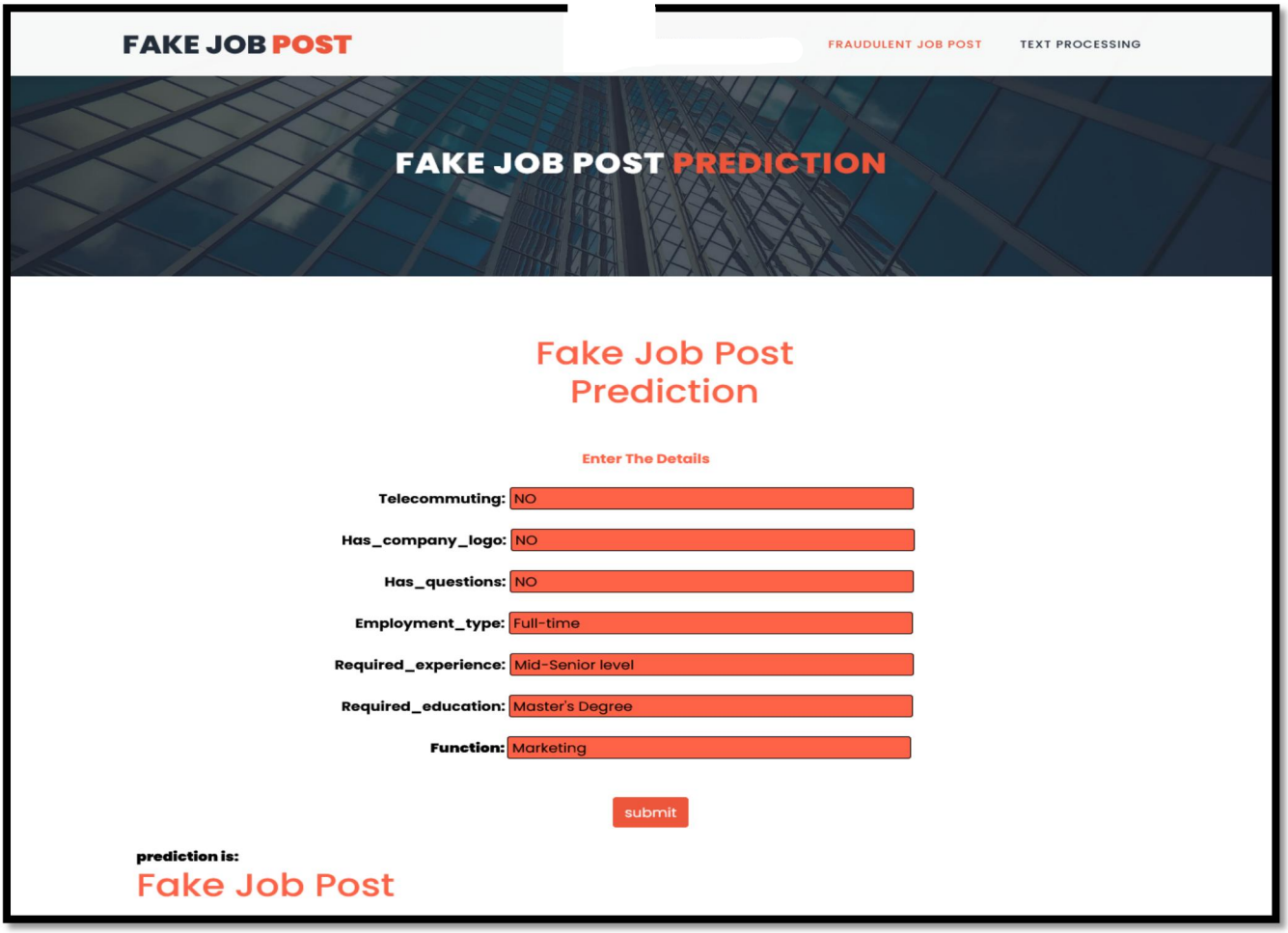

Fig 6.2: Fake job post prediction Page

The Fig 6.2 page may use various data sources and analysis techniques to identify patterns or red flags in job postings that suggest they may be fake or scams. Users of a fake job post prediction page can typically input the details of a job posting, such as the company name, job title, and job description, to receive an analysis of whether the posting is likely to be legitimate or fake. The page may also provide tips and advice for avoiding job scams and protecting personal information.

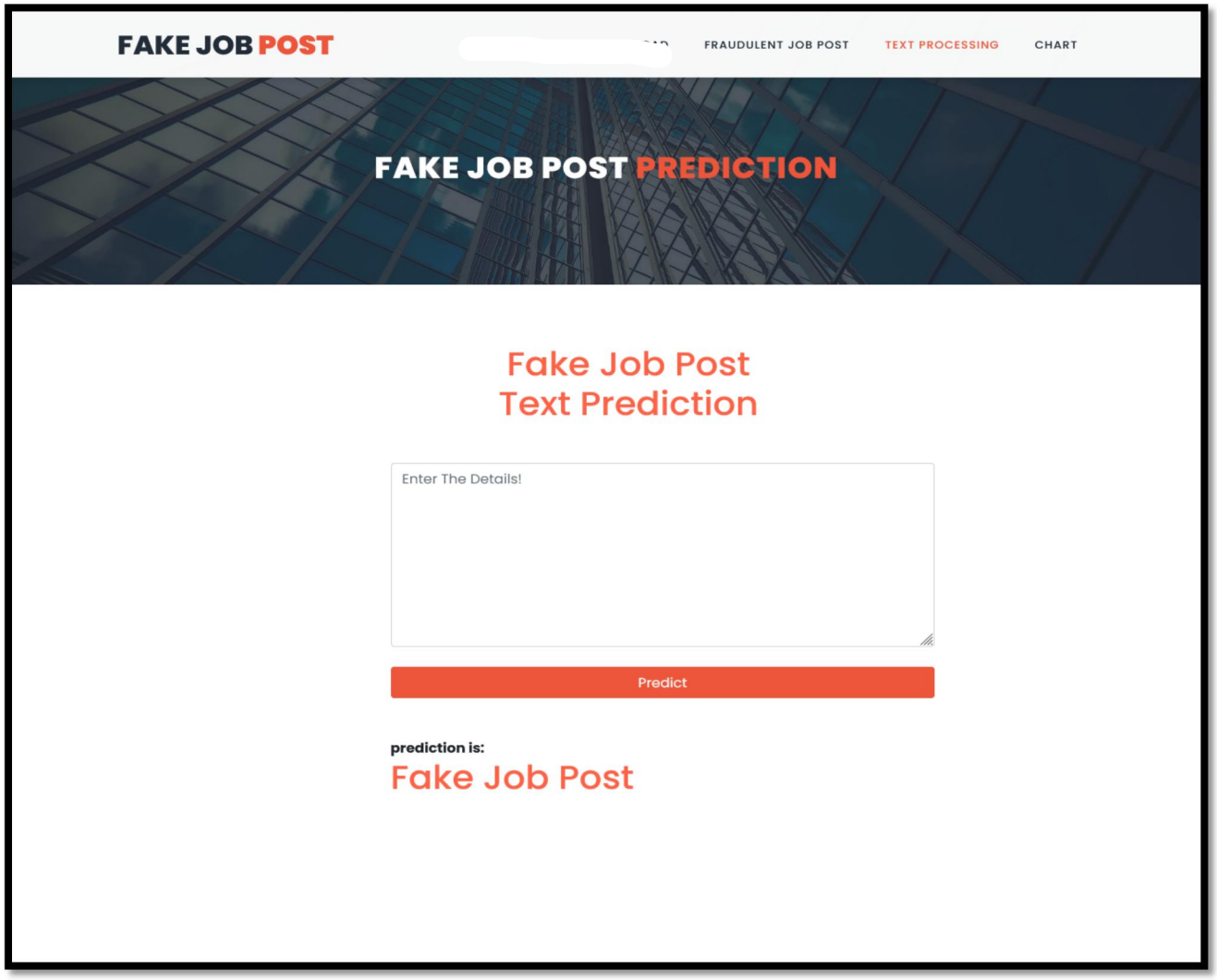

Fig 6.3: Fake job post text prediction Page

The Fig 6.3 generates fraudulent job postings designed to scam job seekers out of money or personal information. The page may use algorithms or templates to create convincing job descriptions, requirements, and qualifications for non-existent or fraudulent job opportunities. Users of a fake job post text prediction page may be asked to input their personal information or pay a fee in order to apply for the fake job. The page may also use tactics such as fake job interviews or background checks to further deceive job seekers and extract more personal information.

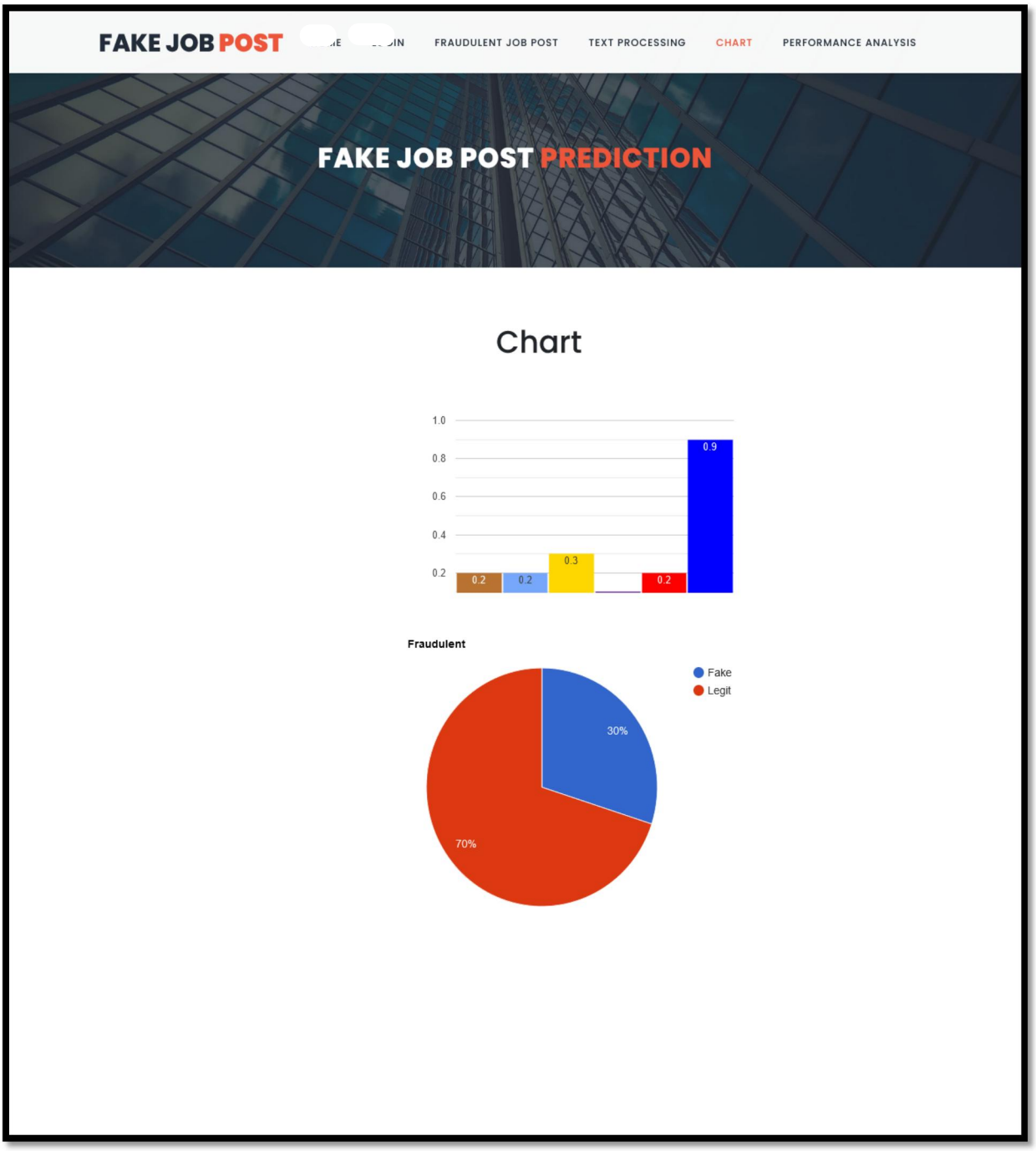

Fig 6.4: Analysis of the fake/real job page.

The Fig 6.4 pie chart can be used to visually represent the breakdown of fake and real job opportunities in a particular market or industry. The chart would show the percentage of job listings that are confirmed to be legitimate versus those that are known to be fraudulent or suspicious. It's important to note that determining the legitimacy of job listings can be a complex process, and the accuracy of the pie chart will depend on the quality and reliability of the data used to generate it. Job seekers should always exercise caution and conduct thorough research before applying for any job opportunity.

CHAPTER 7

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

Job scam detection has become a great concern all over the world at present. In this paper, we have analyzed the impacts of job scam which can be a very prosperous area in research filed creating a lot of challenges to detect fraudulent job posts. We have experimented with EMSCAD dataset which contains real life fake job posts. In this paper we have experimented both machine learning algorithms (SVM, KNN, Naive Bayes, Random Forest and MLP) and deep learning model (Deep Neural Network). This work shows a comparative study on the evaluation of traditional machine learning and deep learning based classifiers. We have found highest classification accuracy for Random Forest Classifier among traditional machine learning algorithms and 99 % accuracy for DNN (fold 9) and 97.7% classification accuracy on average for Deep Neural Network.

Our future work focuses on incorporating more advanced natural language processing (NLP) techniques can improve the accuracy of fake job prediction models. These techniques can be used to extract more meaningful features from job postings, such as sentiment analysis and topic modeling. Additionally, NLP models can be trained to distinguish between authentic and fake job postings based on the language used in them, developing a real-time system that can detect fake job postings as they are posted online can help prevent job seekers from falling victim to these scams. Such a system can use machine learning models to analyze job postings in real-time and alert job seekers to potential scams.

APPENDIX

App.py

```
import numpy as np
import pandas as pd
from flask import Flask, request, jsonify, render_template, redirect, flash, send_file
from sklearn.preprocessing import MinMaxScaler
from werkzeug.utils import secure_filename
import pickle

app = Flask(__name__) #Initialize the flask App

model = pickle.load( open('random.pickle', 'rb') )

vecs = pickle.load( open('vectorizers.pickle', 'rb') )
classifiers = pickle.load( open('classifiers.pickle', 'rb') )

@app.route('/')
@app.route('/index')
def index():
        return render_template('index.html')

@app.route('/chart')
def chart():
        return render_template('chart.html')

@app.route('/performance')
def performance():
        return render_template('performance.html')


@app.route('/login')
def login():
        return render_template('login.html')
@app.route('/upload')
def upload():
```

```
return render_template('upload.html')
@app.route('/preview',methods=["POST"])
def preview():
    if request.method == 'POST':
        dataset = request.files['datasetfile']
        df = pd.read_csv(dataset,encoding = 'unicode_escape')
        df.set_index('Id', inplace=True)
        return render_template("preview.html",df_view = df)

@app.route('/fake_prediction')
def fake_prediction():
    return render_template('fake_prediction.html')

@app.route('/predict',methods=['POST'])
def predict():

    features = [float(x) for x in request.form.values()]
    final_features = [np.array(features)]
    y_pred = model.predict(final_features)
    if y_pred[0] == 1:
      label="Fake Job Post"
    elif y_pred[0] == 0:
      label="Legit Job Post"
    return render_template('fake_prediction.html', prediction_texts=label)
@app.route('/text_prediction')
def text_prediction():
        return render_template("text_prediction.html")

@app.route('/job')
def job():
        abc = request.args.get('news')
        input_data = [abc.rstrip()]
        # transforming input
        tfidf_test = vecs.transform(input_data)
        # predicting the input
        y_preds = classifiers.predict(tfidf_test)
```

```
if y_preds[0] == 1:
                    labels="Fake Job Post"
          elif y_preds[0] == 0:
                    labels="Legit Job Post"


return render_template('text_prediction.html', prediction_text=labels)


if __name__ == "__main__":
    app.run(debug=True)
```

INDEX

```
<!DOCTYPE html>
<html lang="en">

  <head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link
href="https://fonts.googleapis.com/css?family=Poppins:100,100i,200,200i,300,300i,400,400i,500,50
0i,600,600i,700,700i,800,800i,900,900i&display=swap" rel="stylesheet">


 <link rel="stylesheet" type="text/css" href="../static/assets/css/bootstrap.min.css">

    <link rel="stylesheet" type="text/css" href="../static/assets/css/font-awesome.css">

    <link rel="stylesheet" href="../static/assets/css/style.css">

    </head>
```

```
<body>

    <!-- ***** Preloader Start ***** -->
    <div id="js-preloader" class="js-preloader">
      <div class="preloader-inner">
        <span class="dot"></span>
        <div class="dots">
          <span></span>


<span></span>
          <span></span>
        </div>
      </div>
    </div>
    <!-- ***** Preloader End ***** -->


    <!-- ***** Header Area Start ***** -->
    <header class="header-area header-sticky">
        <div class="container">
            <div class="row">
                <div class="col-12">
                    <nav class="main-nav">
                        <!-- ***** Logo Start ***** -->
                        <a href="{{url_for('index')}}" class="logo">Fake Job<em> Post</em></a>
                        <!-- ***** Logo End ***** -->
                        <!-- ***** Menu Start ***** -->
                        <ul class="nav">
                            <li><a href="{{url_for('index')}}" class="active">Home</a></li>
                            <li><a href="{{url_for('login')}}">Login</a></li>


          </ul>
                        <a class='menu-trigger'>
                            <span>Menu</span>
                        </a>
```

```
 <!-- ***** Menu End ***** -->
                </nav>
    </header>
    <!-- ***** Header Area End ***** -->
    <!-- jQuery -->
    <script src="../static/assets/js/jquery-2.1.0.min.js"></script>

    <!-- Bootstrap -->
    <script src="../static/assets/js/popper.js"></script>
    <script src="../static/assets/js/bootstrap.min.js"></script>

    <!-- Plugins -->


 <script src="../static/assets/js/scrollreveal.min.js"></script>
    <script src="../static/assets/js/waypoints.min.js"></script>
    <script src="../static/assets/js/jquery.counterup.min.js"></script>
    <script src="../static/assets/js/imgfix.min.js"></script>
    <script src="../static/assets/js/mixitup.js"></script>
    <script src="../static/assets/js/accordions.js"></script>

    <!-- Global Init -->
    <script src="../static/assets/js/custom.js"></script>

  </body>
</html>
```

REFERENCES

[1] S. Vidros, C. Kolias , G. Kambourakis ,and L. Akoglu, “Automatic Detection of Online Recruitment Frauds: Characteristics, Methods, and a Public Dataset”, Future Internet 2017, 9, 6; doi:10.3390/fi9010006.

[2] B. Alghamdi, F. Alharby, “An Intelligent Model for Online Recruitment Fraud Detection”, Journal of Information Security, 2019, Vol 10, pp. 155176, https://doi.org/10.4236/iis.2019.103009 .

[3] Tin Van Huynh1, Kiet Van Nguyen, Ngan Luu-Thuy Nguyen1, and Anh Gia-Tuan Nguyen, “Job Prediction: From Deep Neural Network Models to Applications”, RIVF International Conference on Computing and Communication Technologies (RIVF), 2020.

[4] Jiawei Zhang, Bowen Dong, Philip S. Yu, “FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network”, IEEE 36th International Conference on Data Engineering (ICDE), 2020.

[5] Scanlon, J.R. and Gerber, M.S., “Automatic Detection of Cyber Recruitment by Violent Extremists”, Security Informatics, 3, 5, 2014, https://doi.org/10.1186/s13388-014-0005-5

[6] Y. Kim, “Convolutional neural networks for sentence classification,” arXiv Prepr. arXiv1408.5882, 2014.

[7] T. Van Huynh, V. D. Nguyen, K. Van Nguyen, N. L.-T. Nguyen, and A.G.-T. Nguyen, “Hate Speech Detection on Vietnamese Social Media Text using the Bi-GRU-LSTM-CNN Model,” arXiv Prepr. arXiv1911.03644, 2019.

[8] P. Wang, B. Xu, J. Xu, G. Tian, C.-L. Liu, and H. Hao, “Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification,” Neurocomputing, vol. 174, pp. 806814, 2016.

[9] C. Li, G. Zhan, and Z. Li, “News Text Classification Based on Improved BiLSTM-CNN,” in 2018 9th International Conference on Information Technology in Medicine and Education (ITME), 2018, pp. 890-893.

[10] K. R. Remya and J. S. Ramya, “Using weighted majority voting classifier combination for relation classification in biomedical texts,” International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014, pp. 1205-1209.

[11] Yasin, A. and Abuhasan, A. (2016) An Intelligent Classification Model for Phishing Email Detection. International Journal of Network Security & Its Applications, 8, 55-72. https://doi.org/10.5121/imsa.2016.8405

[12] Vong Anh Ho, Duong Huynh-Cong Nguyen, Danh Hoang Nguyen, Linh Thi-Van Pham, Duc-Vu Nguyen, Kiet Van Nguyen, and Ngan Luu- Thuy Nguyen."Emotion Recognition for Vietnamese Social Media Text", arXiv Prepr. arXiv:1911.09339, 2019.

[13] Thin Van Dang, Vu Duc Nguyen, Kiet Van Nguyen and Ngan Luu- Thuy Nguyen, “Deep learning for aspect detection on vietnamese reviews” in In Proceeding of the 2018 5th NAFOSTED Conference on Information and Computer Science (NICS), 2018, pp. 104-109.

[14] Li, H.; Chen, Z.; Liu, B.; Wei, X.; Shao, J. Spotting fake reviews via collective positive-unlabeled learning. In Proceedings of the 2014 IEEE International Conference on Data Mining (ICDM), Shenzhen, China, 14-17 December 2014; pp. 899-904.

[15] Ott, M.; Cardie, C.; Hancock, J. Estimating the prevalence of deception in online review communities. In Proceedings of the 21st international conference on World Wide Web, Lyon, France, 16-20 April 2012; ACM: New York, NY, USA, 2012; pp. 201-210.