ONLINE RECRUITMENT FRAUD (ORF) DETECTION USING DEEP LEARNING APPROACHES

A PROJECT REPORT

Submitted by

RAUSHAN KUMAR	**810421104139**
SANNI K RANJAN KUMAR	**810421104148**
SANSHAY KUMAR TIWARY	**810421104149**
SURAJ KUMAR	**810421104178**

In partial fulfillment for the award of degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE

(AUTONOMOUS)

PERAMBALUR-621 212

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2025

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE
(AUTONOMOUS)
PERAMBALUR – 621 212

BONAFIDE CERTIFICATE

Certified that this mini project report **"Online Recruitment Fraud (ORF) Detection Using Deep Learning Approaches"** is the bonafide work of **"RAUSHAN KUMAR (810421104139), SANNI K RANJAN KUMAR (810421104149), SANSHAY KUMAR TIWARY (810421104149), SURAJ KUMAR (810421104172),"** who carried out the project work under my supervision.

SIGNATURE

Dr. R. GOPI, M.Tech., Ph.D., (PDF).,
PROFESSOR And HEAD,
Department of Computer Science and Engineering,
Dhanalakshmi Srinivasan
Engineering College (Autonomous),
Perambalur – 621 212.

SIGNATURE

Dr. R. GOPI, M.Tech., Ph.D., (PDF).,
SUPERVISOR,
Department of Computer Science and Engineering,
Dhanalakshmi Srinivasan
Engineering College (Autonomous),
Perambalur – 621 212.

Submitted for Main-Project Viva-Voce Examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our gratitude and thanks to **Our Parents** first for giving health and a sound mind for completing this project. We give all the glory and thanks to our almighty **GOD** for showering upon, the necessary wisdom and grace for accomplishing this project.

It is our pleasant duty to express deep sense of gratitude to our honorable Chancellor **Shri. A. Srinivasan**, for his kind encouragement. We have unique pleasure in thanking our Principal **Dr. D. Shanmugasundram, M.E., Ph.D., F.I.E., C.Eng.,** our Dean **Dr. K. Anbarasan, M.E., Ph.D.,** and our COE **Dr. M.Chellapan, M.E., Ph.D.,** for their unflinching devotion, which leads us to complete this project.

We express our faithful and sincere gratitude to our Professor and Head **Dr. R. Gopi, M.Tech.,Ph.D., (PDF)** for his valuable guidance and support that he gaveus during the project time.

We express our faithful and sincere gratitude to our Project Coordinator **Mrs.B.Deepika., M.E.,(Ph.D).,** of Department of Computer Science and Engineering for giving and support throughout our project.

We are also thankful to our internal guide **Dr. R. Gopi, M.Tech., Ph.D.,(PDF)** of Department of Computer Science and Engineering for his valuable guidance and precious suggestion to complete this project work successfully.

We render our thanks to all **Faculty members** and **Programmers** of Department of **Computer Science and Engineering** for their timely assistance.

DHANALAKSHMI SRINIVASAN ENGINEERIN COLLEGE (AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision and Mission of the Department:

Vision

To produce globally competent, socially responsible professionals in the field of Computer Science and Engineering.

Mission

M1: Impart high quality experiential learning to get expertise in modern software tools

M2: Inculcate industry exposure and build inter disciplinary research skills.

M3: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M4: Acquire Innovative skills and promote lifelong learning with a sense of societal and ethical responsibilities

Program Educational Objectives (PEOs)

PEO 1: Graduates of the programme will develop proficiency in identifying, formulating, and resolving complex computing problems.

PEO 2: Graduates of the programme will achieve successful careers in the field of computer science and engineering, pursue advanced degrees, or demonstrate entrepreneurial success.

PEO 3: Graduates of the programme will cultivate effective communication skills, teamwork abilities, ethical values, and leadership qualities for professional engagement in industry and research organizations.

ABSTRACT

Most companies nowadays are using digital platforms for the recruitment of new employees to make the hiring process easier, faster, and more accessible. The rapid increase in the use of online platforms for job posting has resulted in a significant rise in fraudulent advertising and malicious job scams. The scammers are making money through fraudulent job postings by luring unsuspecting job seekers with attractive, yet fake, employment offers. As a result, online recruitment fraud has emerged as an important issue in the domain of cybercrime, causing financial and emotional harm to job seekers. Therefore, it is necessary to detect fake job postings to protect users and get rid of online job scams. In recent studies, both traditional machine learning and advanced deep learning algorithms have been implemented to detect fake job postings. This research aims to use two state-of-the-art transformer-based deep learning models, i.e., Bidirectional Encoder Representations from Transformers (BERT) and Robustly Optimized BERT-Pretraining Approach (RoBERTa), to detect fake job postings more precisely and efficiently. In this research, a novel dataset of fake job postings is proposed, formed by the combination of job postings from three different sources to ensure diversity and relevance. Existing benchmark datasets are outdated and limited due to knowledge of specific job postings, which restricts the existing models' capability in detecting newly evolving fraudulent job advertisements. Hence, we extend it with the latest job postings collected from trusted sources. Exploratory Data Analysis (EDA) highlights a noticeable class imbalance problem in detecting fake jobs, which tends to make the model biased and act aggressively toward the minority class. In response to overcome this issue, the work at hand implements ten top-performing variants of the Synthetic Minority Oversampling Technique (SMOTE), a popular oversampling method for handling imbalanced datasets. The models' performances balanced by each SMOTE variant are thoroughly analyzed and compared through multiple evaluation metrics. All implemented approaches perform competitively in identifying fraudulent jobs. However, the combination of BERT and SMOBD SMOTE achieved the highest balanced accuracy and recall of about 90%, showcasing its effectiveness in fake job detection tasks.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	i
	LIST OF FIGURES	vii
	LIST OF ABBREVIATIONS	viii
1	**INTRODUCTION**	**1**
	1.1 OVERVIEW	1
	1.2 ARTIFICIAL INTELLIGENCE	2
	1.3 DEEP LEARNING	3
	1.4 MACHINE LEARNING	4
	1.5 PREPARING THE DATASET	4
	1.5.1 Proposed system	4
	1.5.2 Exploratory data analysis	5
	1.5.3 Data wrangling	5
	1.5.4 Data collection	5
	1.5.4 Building the classification model	5
2	**LITERATURE SURVEY**	**7**
	2.1 Online Recruitment Fraud Detection: A Study on Contextual Features in Australian Job Industries.	7
	2.2 A Credit Card Fraud Detection Algorithm Based on SDT and Federated Learning.	8
	2.3 A Novel Framework for Credit Card Fraud Detection.	9
	2.4 A Systematic Literature Review of Fraud Detection Metrics in Business Processes.	10
	2.5 Advanced Credit Card Fraud Detection: An Ensemble Learning Using Random Under sampling and Two-Stage.	11

2.6 An Adversary Model of Fraudsters' Behavior to Improve Oversampling in Credit Card Fraud Detection. 12

2.7 Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms. 13

2.8 Enhancing Fraud Detection in Banking with Deep Learning: Graph Neural Networks and Autoencoders for Real-Time Credit Card Fraud Prevention. 14

2.9 Enhancing Medicare Fraud Detection Through Machine Learning: Addressing Class Imbalance with SMOTE-ENN. 15

2.10 Ensemble Synthesized Minority Oversampling-Based Generative Adversarial Networks and Random Forest Algorithm for Credit Card Fraud Detection. 16

2.11 Evaluating the Computational Advantages of the Variational Quantum Circuit Model in Financial Fraud Detection. 17

2.12 Fed Fusion: Adaptive Model Fusion for Addressing Feature Discrepancies in Federated Credit Card Fraud Detection. 18

2.13 Financial Fraud Detection Using Value at Risk with Machine Learning in Skewed Data. 19

2.14 Generative Adversarial Networks-Based Novel Approach for Fraud Detection for the European Cardholders 2013 Dataset 20

2.15 Intelligent Fraud Detection in Financial Statements Using Machine Learning and Data Mining: A Systematic Literature Review. 21

2.16 Machine Learning Methods for Credit Card Fraud Detection: A Survey . 22

2.17 Medicare Fraud Detection Using Graph Analysis: A Comparative Study of Machine Learning and Graph Neural Networks. 23

2.18 Online Payment Fraud Detection Model Using Machine Learning Techniques. 24

2.19 OPT Dev Net: An Optimized Deep Event-Based Network Framework for Card Fraud Detection. 25

2.20 Quantum Autoencoder for Enhanced Fraud Detection in Imbalanced Credit Card Dataset. 26

3 METHODOLOGY 27

3.1 EXISTING SYSTEM 28

3.2 PROPOSED SYSTEM 28

3.3 FEASIBILITY STUDY 28

3.3.1 Data Wrangling 28

3.3.2 Data collection 29

3.3.3 Preprocessing 29

3.3.4 Building the classification model 29

3.3.5 Construction of a Predictive Model 29

3.4 LIST OF MODULES 30

3.5 PROJECT REQUIREMENTS 31

3.5.1 General 31

3.5.2 Functional requirements 31

3.5.3 Non-Functional Requirements 31

3.6 ENVIRONMENTAL REQUIREMENTS 31

3.7 SOFTWARE DESCRIPTION 32

3.8 ANACONDA NAVIGATOR 32

3.9 JUPYTER NOTEBOOK 34

	3.10 PYTHON	34
	3.11 SYSTEM ARCHITECTURE	35
	3.12 WORK FLOW DIAGRAM	35
	3.13 USE CASE DIAGRAM	36
	3.14 CLASS DIAGRAM	37
	3.15 ACTIVITY DIAGRAM	38
	3.16 SEQUENCE DIAGRAM	39
	3.17 ENTITY RELATIONSHIP DIAGRAM (ERD)	40
4	**RESULTS AND DISCUSSION**	**41**
	4.1 DISCUSSION	41
	4.1.1 Data Pre-processing	41
	4.1.2 Data Validation/ Cleaning/Preparing Process	42
	4.1.3 Exploration data analysis of visualization	44
	4.1.4 Comparing Algorithm with prediction in the form of best accuracy result	45
	4.2 ALGORITHM AND TECHNIQUES	47
	4.2.1 Algorithm Explanation	47
	4.2.2 Used Python Packages	48
	4.2.3 Natural Language Processing (NLP)	48
	4.2.4 Vectorization/Word Embedding	49
	4.2.5 Logistic Regression	49
	4.2.6 Random Forest Classifier	51
5	**CONCLUSION AND FUTURE WORK**	**53**
	5.1 CONCLUSION	53
	5.2 FUTURE WORK	53
6	**APPENDICES**	**54**
	6.1 SOURCE CODE	54
	6.2 SCREENSHOT	64

7 REFERENCE 66

CERTIFICATES

LIST OF FIGURES

FIGURE NO	TITLE	PAGE.NO
3.11.1	SYSTEM ARCHITECTURE	37
3.12.1	WORKFLOW DIAGRAM	37
3.13.1	USECASE DIAGRAM	38
3.14.1	CLASS DIAGRAM	39
3.15.1	ACTIVITY DIAGRAM	40
3.16.1	SEQUENCE DIAGRAM	41
3.17.1	ER – DIAGRAM	42
4.1.2	MODULE DIAGRAM	46

LIST OF ABBREVIATIONS

ORF	—	Online Recruitment Fraud
ML	—	Machine Learning
DL	—	Deep Learning
CNN	—	Convolutional Neural Network
RNN	—	Recurrent Neural Network
BERT	—	Bidirectional Encoder Representations from Transformers

CHAPTER 1
INTRODUCTION

1.1. OVERVIEW

The existing data-driven approaches typically capture credibility-indicative representations from relevant articles for fake news detection, such as skeptical and conflicting opinions. However, these methods still have several drawbacks: Due to the difficulty of collecting fake news, the capacity of the existing datasets is relatively small and there is considerable unverified news that lacks conflicting voices in relevant articles, which makes it difficult for the existing methods to identify their credibility. Especially, the differences between true and fake news are not limited to whether there are conflict features in their relevant articles, but also include more extensive hidden differences at the linguistic level, such as the perspectives of emotional expression (like extreme emotion in fake news), writing style (like the shocking title in click bait), etc., the existing methods are difficult to fully capture these differences.

Drawbacks:

- They are not using machine learning.
- Accuracy, Recall F1 score metrics are not calculated and machine learning algorithms are not applied.

Data Science:

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains.

The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science. In 1996, the International Federation of Classification Societies became the first conference to specifically feature data science as a topic. However, the definition was still in flux.

The term ―data science‖ was first coined in 2008 by D.J. Patil, and Jeff Hammer bacher, the pioneer leads of data and analytics efforts at LinkedIn and Facebook. In less than a decade, it has become one of the hottest and most trending professions in the market.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data.

Data science can be defined as a blend of mathematics, business acumen, tools, algorithms and machine learning techniques, all of which help us in finding out the hidden insights or patterns from raw data which can be of major use in the formation of big business decisions.

Data Scientist:

Data scientists examine which questions need answering and where to find the related data. They have business acumen and analytical skills as well as the ability to mine, clean, and present data. Businesses use data scientists to source, manage, and analyze large amounts of unstructured data.

Required Skills for a Data Scientist:

- Programming: Python, SQL, Scala, Java, R, MATLAB.
- Machine Learning: Natural Language Processing, Classification, Clustering.
- Data Visualization: Tableau, SAS, D3.js, Python, Java, R libraries.
- Big data platforms: Mongo DB, Oracle, Microsoft Azure, Cloudera.

1.2. ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving.

Artificial intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by humans or animals. Leading AI textbooks define the field as the study of "intelligent agents" any system that perceives its environment and takes actions that maximize its chance of achieving its goals. Some popular accounts use the term "artificial intelligence" to describe machines that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving", however this definition is rejected by major AI researchers.

AI programming focuses on three cognitive skills: learning, reasoning and self-correction.

Learning processes: This aspect of AI programming focuses on acquiring data and creating rules for how to turn the data into actionable information. The rules, which are called algorithms, provide computing devices with step-by-step instructions for how to complete a specific task.

Reasoning processes: This aspect of AI programming focuses on choosing the right algorithm to reach a desired outcome.

Self-correction processes: This aspect of AI programming is designed to continually fine-tune algorithms and ensure they provide the most accurate results possible.

Natural Language Processing (NLP):

Natural language processing (NLP) allows machines to read and understand human language. A sufficiently powerful natural language processing system would enable natural-language user interfaces and the acquisition of knowledge directly from human-written sources, such as newswire texts. Some straightforward applications of natural language processing include information retrieval, text mining, question answering and machine translation. Many current approaches use word co-occurrence frequencies to construct syntactic representations of text. "Keyword spotting" strategies for search are popular and scalable but dumb; a search query for "dog" might only match documents with the literal word "dog" and miss a document with the word "poodle". "Lexical affinity" strategies use the occurrence of words such as "accident" to assess the sentiment of a document.

1.3. DEEP LEARNING

Deep learning is a subset of machine learning, which itself is a branch of artificial intelligence (AI) focused on creating systems that can learn from data. Deep learning models are inspired by the structure and function of the human brain and are designed to automatically learn to represent data through layers of artificial neurons. These models, also known as **artificial neural networks** (ANNs), consist of multiple layers of interconnected nodes or "neurons" that work together to process information.

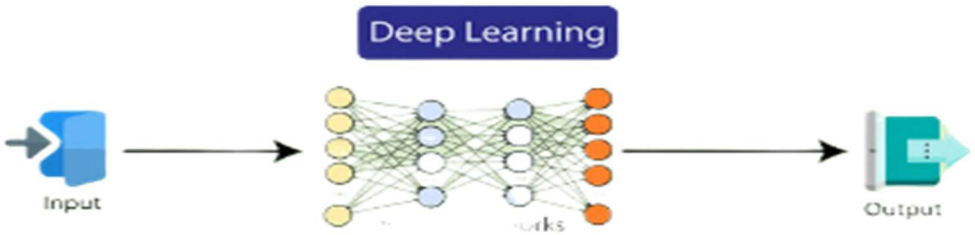
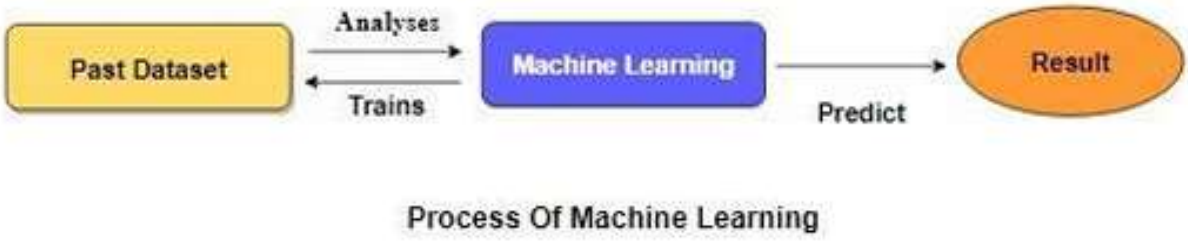


Fig.1.2

Supervised Machine Learning is the majority of practical machine learning uses supervised learning. Supervised learning is where have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output is y = f(X). The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include logistic regression, multi-class classification, Decision Trees and support vector machines etc.

1.4. MACHINE LEARNING

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.



Process Of Machine Learning

Fig.1.3

1.5. PREPARING THE DATASET:

This dataset contains 1400 records of features, which were then classified into 2 classes:

- REAL
- FAKE

1.4.1. Proposed System:

The proposed method is built a deep learning model to classify the real or fake job posting to overcome this method to implement deep learning approach. The dataset is first preprocessed and the columns are analyzed to see the dependent and independent variable and then different deep learning algorithms would be applied to extract patterns and to obtain results with maximum accuracy.

1.4.2. Exploratory Data Analysis

Multiple datasets from different sources would be combined to form a generalized dataset, and then different deep learning algorithms would be applied to extract patterns and to obtain results with maximum accuracy.

1.4.3. Data Wrangling

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

1.4.4. Data collection

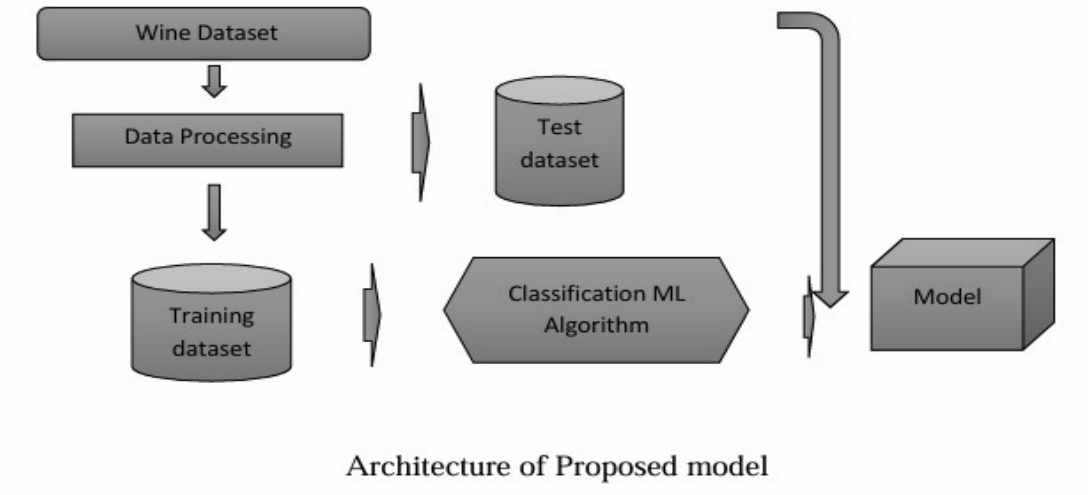
The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using deep learning algorithms are applied on the Training set and based on the test result accuracy, Test set prediction is done.

1.4.5. Building the classification model

It provides better results in classification problem.

- It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.

- It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.



Architecture of Proposed model

Fig.1.4.5

Advantages:

- **High Accuracy** – Deep learning models often outperform traditional methods in classification tasks.
- **Automatic Feature Extraction** – They eliminate the need for manual feature engineering by learning relevant patterns directly from the data.
- **Scalability** – Deep learning can handle large volumes of job postings efficiently.
- **Adaptability** – These models adapt well to different types of job data and formats.
- **Real-Time Processing** – Capable of processing and classifying job posts in real-time.
- **Noise Tolerance** – Effective in dealing with noisy, unstructured text data often found in job postings.
- **Multilingual Support** – Can be trained to classify posts in multiple languages with appropriate datasets.
- **Improved Fraud Detection** – Better at identifying fraudulent or scam job postings through learned patterns.
- **Context Understanding** – Deep learning captures the contextual meaning of words using embeddings like Word2Vec or BERT.
- **Continuous Learning** – Supports model updating as new job data becomes available.
- **Automation** – Reduces the need for manual job classification and human intervention.
- **Customizability** – Models can be fine-tuned for specific industries or job markets.
- **Integration with Other Systems** – Can be embedded into larger recruitment platforms or fraud detection systems.
- **Benchmarking for Research** – Provides a foundation for comparing and improving future algorithms.

CHAPTER 2
LITERATURE SURVEY

2.1. Online Recruitment Fraud Detection: A Study on Contextual Features in Australian Job Industries.

AUTHOURS: Syed Mahbub, Eric Pardede.

PUBLISHEDBY: IEEE.

YEAR: 2022.

Description

The purpose of this study is to investigate the effects of contextual features on automatic detection accuracy of online recruitment frauds in Australian job market. In addition, the study aims to unearth the significance of localization of such approaches. The study first generates a dataset based on a local and semi-structured advertising platform in Australia. The labelled dataset is then used to train a learning model on several content-based and contextual features.

Methodology Used:

- Contextual feature extraction from job listings
- Deep learning models (e.g., BERT, Ro BERT)

Advantages:

- Deep learning models capture semantic nuances
- Handles class imbalance effectively

Disadvantages:

- Computationally expensive models
- Requires large labeled datasets

2.2. A Credit Card Fraud Detection Algorithm Based on SDT and Federated Learning.

AUTHOURS: Yuxuan Tang, Zhanjun Liu.

PUBLISHEDBY: IEEE.

YEAR: 2024.

Description

The rise of digital payment methods and the growth in financial transactions, the issue of credit card fraud has become increasingly severe. Traditional fraud detection methods are currently facing challenges such as poor model performance, difficulty in obtaining accurate results, and limitations in distributed deployment. These challenges stem from constantly evolving fraud strategies, higher volumes of transactions, and the complexity of the financial environment. This study proposes a credit card fraud detection algorithm based on Structured Data Transformer (SDT) and federated learning, which leverages the advanced capabilities of the Transformer model in deep learning. First, we organize credit card data into sequences and introduce a special, learnable token at the beginning of each sequence for classification purposes.

Methodology Used:

- Self-Adaptive Decision Tree (SDT) for dynamic fraud detection.
- Federated Learning (FL) for decentralized data processing.

Advantages:

- Preserves user privacy by avoiding central data storage.
- Adaptable to evolving fraud patterns.

Disadvantages:

- Requires high communication bandwidth for model updates.
- Potential security risks in federated model aggregation.

2.3. A Novel Framework for Credit Card Fraud Detection.

AUTHOURS: Ayoub Mniai, Mouna Tarik, and Khalid Jebari.

PUBLISHEDBY: IEEE.

YEAR: 2023.

Description

Credit card transactions have grown considerably in the last few years. However, this increase has led to significant financial losses around the world. More than that, processing the enormous amount of generated data becomes very challenging, making the datasets highly dimensional and unbalanced. This means the collected data is suffering from two major problems. It is characterized by a severe difference in observation frequency between fraud and non-fraud transactions, and it contains irrelevant, inappropriate, and correlated data that negatively affects their prediction performance.

Methodology Used:

- Machine Learning (e.g., Random Forest, XGBoost, SVM)
- Deep Learning (e.g., LSTM, CNN, Autoencoders)
- Anomaly Detection (Isolation Forest, One-Class SVM)

Advantages:

- High accuracy in detecting fraudulent transactions
- Real-time fraud detection capability
- Reduces false positives using advanced models

Disadvantages:

- Computationally expensive for real-time processing
- Data privacy concerns with sensitive information
- Imbalanced dataset challenges affect model performance

2.4. A Systematic Literature Review of Fraud Detection Metrics in Business Processes.

AUTHOURS: Badr Omair and Ahmad Alturki.

PUBLISHEDBY: IEEE.

YEAR: 2020.

Description

Fraud is a primary source of organization losses, amounting to up to 5% of yearly revenues. Process-based fraud (PBF) is fraud involving a deviation from the standard operating procedure (SOP) of business processes. PBF hinders the achievement of business objectives because business processes operationalize organizational strategies. A systematic content analysis of the literature was conducted on fraud detection metrics in business processes. The current state of fraud detection was surveyed by focusing on PBF metrics while including all relevant conceptual perspectives of PBF detection. The findings indicate that a large body of research has examined detection metrics for possible fraud, but less attention has been paid to PBF.

Methodology Used:

- Systematic Literature Review (SLR)
- PRISMA framework for study selection

Advantages:

- Comprehensive overview of fraud detection metrics
- Identifies trends and gaps in existing research

Disadvantages:

- Time-consuming and resource-intensive
- Potential bias in study selection

2.5. Advanced Credit Card Fraud Detection: An Ensemble Learning Using Random Under Sampling and Two-Stage.

AUTHOURS: Ibrahim Almubark.

PUBLISHEDBY: IEEE.

YEAR: 2024.

Description

The increase of Credit Card (CC) fraud in recent years requires the development of fraud detection systems that are both efficient and robust. This paper explored the utilization of machine learning models, with a particular emphasis on ensemble methods, to advance the detection of CC fraud. We present an ensemble model that incorporates different classifiers to address the dataset imbalance issue that is present in most CC datasets. We employed synthetic over-sampling and under-sampling techniques in certain machine learning algorithms to tackle the same issue.

Methodology Used:

- Ensemble learning approach
- Random Under Sampling (RUS) for handling class imbalance
- Two-stage detection method

Advantages:

- Improves fraud detection accuracy
- Reduces false positives and negatives
- Handles class imbalance effectively

Disadvantages:

- Risk of losing valuable data due to under-sampling
- Computationally expensive with multiple models
- Requires fine-tuning for optimal performance

2.6. An Adversary Model of Fraudsters' Behavior to Improve Oversampling in Credit Card Fraud Detection.

AUTHOURS: Daniele Lunghi, Gian Marco Paldino, Olivier Caelen

PUBLISHEDBY: IEEE.

YEAR: 2023.

Description

Imbalanced learning jeopardizes the accuracy of traditional classification models, particularly for what concerns the minority class, which is often the class of interest. This paper addresses the issue of imbalanced learning in credit card fraud detection by introducing a novel approach that models fraudulent behavior as a time-dependent process. The main contribution is the design and assessment of an oversampling strategy, called ''Adversary-based Oversampling'' (ADVO), which relies on modeling the temporal relationship among frauds. The strategy is implemented by two learning approaches: first, an innovative regression-based oversampling model that predicts subsequent fraudulent activities based on previous fraud features.

Methodology Used:

- Adversarial modeling of fraudster behavior
- Improved oversampling techniques for class imbalance

Advantages:

- Better fraud detection accuracy
- More realistic synthetic fraud samples

Disadvantages:

- Complexity in modeling adversarial strategies
- Potential risk of generating biased synthetic data

2.7. Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms.

AUTHOURS: Fawaz Khaled Alarfaj, Iqra Malik, Hikmat Ullah Khan

PUBLISHEDBY: IEEE.

YEAR: 2022.

Description

People can use credit cards for online transactions as it provides an efficient and easy-to-use facility. With the increase in usage of credit cards, the capacity of credit card misuse has also enhanced. Credit card frauds cause significant financial losses for both credit card holders and financial companies. In this research study, the main aim is to detect such frauds, including the accessibility of public data, high-class imbalance data, the changes in fraud nature, and high rates of false alarm. The relevant literature presents many machines learning based approaches for credit card detection, such as Extreme Learning Method, Decision Tree, Random Forest, Support Vector Machine, Logistic Regression and XG Boost. However, due to low accuracy, there is still a need to apply state of the art deep learning algorithms to reduce fraud losses.

Methodology Used:

- Supervised learning (SVM, Decision Trees, Random Forest, XGBoost)
- Unsupervised learning (Autoencoders, Isolation Forest, Clustering)

Advantages:

- High accuracy with deep learning models
- Automated feature extraction (CNNs, LSTMs)

Disadvantages:

- Computationally expensive
- High false positives in some models

2.8. Enhancing Fraud Detection in Banking With Deep Learning: Graph Neural Networks and Auto encoders for Real-Time Credit Card Fraud Prevention.

AUTHOURS: Fawaz Khaled Alarfaj, Shabnam Shahzadi.

PUBLISHEDBY: IEEE.

YEAR: 2024.

Description

Under the umbrella of artificial intelligence (AI), deep learning enables systems to cluster data and provide incredibly accurate results. This study explores deep learning for fraud detection, utilizing Graph Neural Networks (GNNs) and Auto encoders to enhance business practices and reduce fraudulent activities in large organizations. For real-time fraud detection, we propose Graph neural network with lambda architecture while for credit card fraud detection, we use an auto encoder, validated through case studies from two banks.

Methodology Used:

- Graph Neural Networks (GNNs) for transaction relationship analysis
- Auto encoders for anomaly detection

Advantages:

- Detects complex fraud patterns
- Real-time transaction monitoring

Disadvantages:

- High computational cost
- Requires large labeled datasets

2.9. Enhancing Medicare Fraud Detection Through Machine Learning: Addressing Class Imbalance With SMOTE-ENN.

AUTHOURS: Rayene Bounab, Karim Zarour, Bouchra Guelib.

PUBLISHEDBY: IEEE.

YEAR: 2024.

Description

The healthcare fraud detection field is constantly evolving and faces significant challenges, particularly when addressing imbalanced data issues. Previous studies mainly focused on traditional machine learning (ML) techniques, often struggling with imbalanced data. This problem arises in various aspects. It includes the risk of over fitting with Random Oversampling (ROS), noise introduction by the Synthetic Minority Oversampling Technique (SMOTE), and potential crucial information loss with Random Under sampling (RUS). Moreover, improving model performance, exploring hybrid re sampling techniques, and enhancing evaluation metrics are crucial for achieving higher accuracy with imbalanced datasets.

Methodology Used:

- Machine learning models for fraud detection
- SMOTE-ENN (Synthetic Minority Over-sampling Technique and Edited Nearest Neighbors) for class imbalance

Advantages:

- Improves fraud detection accuracy
- Reduces class imbalance impact
- Enhances model generalization

Disadvantages:

- Computationally expensive
- Risk of over fitting
- May remove legitimate rare cases

2.10. Ensemble Synthesized Minority Oversampling-Based Generative Adversarial Networks and Random Forest Algorithm for Credit Card Fraud Detection.

AUTHOURS: Fuad A. Ghaleb, Faisal Saeed.

PUBLISHEDBY: IEEE.

YEAR: 2023.

Description

The recent increase in credit card fraud is rapidly has caused huge monetary losses for individuals and financial institutions. Most credit card frauds are conducted online by illegally obtaining payment credentials through data breaches, phishing, or scamming. Many solutions have been suggested to address the credit card fraud problem for online transactions. However, the high-class imbalance is the major challenge that faces the existing solutions to construct an effective detection model. Most of the existing techniques used for class imbalance overestimate the distribution of the minority class, resulting in highly overlapped or noisy and unrepresentative features, which cause either over fitting or imprecise learning.

Methodology Used:

- Uses GANs to generate synthetic fraudulent transactions.
- Balances class distribution by oversampling minority class.
- Combines multiple GAN models for better fraud pattern learning.

Advantages:

- Handles class imbalance effectively.
- Generates realistic fraudulent samples for training.
- Improves detection performance with ensemble learning.

Disadvantages:

- Computationally expensive.
- Requires careful tuning to avoid mode collapse.

2.11. Evaluating the Computational Advantages of the Variational Quantum Circuit Model in Financial Fraud Detection.

AUTHOURS: Antonio Tudisco, Deborah Volpe.

PUBLISHEDBY: IEEE.

YEAR: 2024.

Description

Home banking and digital payments diffusion has greatly increased in recent years. As a result, fraud has also dramatically grown, resulting in the loss of billions of dollars worldwide every year. Therefore, banks and financial institutions are required to offer clients increasingly effective and sophisticated services for illegal transaction detection. Machine learning strategies are commonly employed for this crucial application. However, classical models are not satisfactory enough in highly unbalanced classification tasks like fraud detection.

Methodology Used:

- Variational Quantum Circuits (VQC) for fraud classification
- Hybrid quantum-classical models
- Quantum feature mapping and encoding

Advantages:

- Potential exponential speedup in complex fraud detection
- Better pattern recognition in high-dimensional data
- Enhanced security in data processing

Disadvantages:

- Current quantum hardware limitations (noise, qubit errors)
- Limited scalability for large datasets
- Requires hybrid models due to lack of full quantum advantage yet

2.12. Fed Fusion: Adaptive Model Fusion for Addressing Feature Discrepancies in Federated Credit Card Fraud Detection.

AUTHOURS: Nahid Ferdous Aurna, Md Delwar Hossain.

PUBLISHEDBY: IEEE.

YEAR: 2024.

Description

The digitization of financial transactions has led to a rise in credit card fraud, necessitating robust measures to secure digital financial systems from fraudsters. Nevertheless, traditional centralized approaches for detecting such frauds, despite their effectiveness, often do not maintain the confidentiality of financial data. Consequently, Federated Learning (FL) has emerged as a promising solution, enabling the secure and private training of models across organizations. However, the practical implementation of FL is challenged by data heterogeneity among institutions, complicating model convergence.

Methodology Used:

- Federated Learning (FL) with adaptive model fusion
- Feature discrepancy handling across clients
- Aggregation of model updates via Fed Fusion strategy

Advantages:

- Preserves data privacy across institutions
- Adapts to heterogeneous client data distributions
- Improves fraud detection accuracy over standard FL

Disadvantages:

- Increased computational overhead for adaptive fusion
- Communication costs in federated settings
- Requires careful tuning of fusion weights

2.13. Financial Fraud Detection Using Value-at-Risk with Machine Learning in Skewed Data.

AUTHOURS: Abdullahi Ubale Usman, Sunusi Bala Abdullahi.

PUBLISHEDBY: IEEE.

YEAR: 2024.

Description

The significant losses that banks and other financial organizations suffered due to new bank account (NBA) fraud are alarming as the number of online banking service users increases. The inherent skewness and rarity of NBA fraud instances have been a major challenge to the machine learning (ML) models and happen when non-fraud instances outweigh the fraud instances, which leads the ML models to overlook and erroneously consider fraud as non-fraud instances. Such errors can erode the confidence and trust of customers.

Methodology Used:

- Value-at-Risk (VaR) for risk estimation
- Machine learning models (e.g., SVM, Random Forest, Neural Networks)
- Data preprocessing (handling skewed data with SMOTE, under sampling, or cost-sensitive learning)

Advantages:

- Effective risk estimation with VaR
- Machine learning improves fraud detection accuracy
- Handles large-scale financial data

Disadvantages:

- VaR has limitations in extreme market conditions
- Class imbalance can lead to biased predictions
- High computational cost for complex models

2.14. Generative Adversarial Networks-Based Novel Approach for Fraud Detection for the European Cardholders 2013 Dataset.

AUTHOURS: Fahdah A. Almarshad, Ghada Abdalaziz.

PUBLISHEDBY: IEEE.

YEAR: 2023.

Description

Credit card use poses a significant security issue on a global scale, with rule-based algorithms and traditional anomaly detection being two of the most often used methods. However, they are resource intensive, time-consuming, and erroneous. Given fewer instances than legal payments, the dataset imbalance has become a serious issue. On the other hand, the generative technique is considered an effective way to rebalance the imbalanced class issue, as this technique balances both minority and majority classes before the training. In a more recent period, GAN is considered one of the most popular data generative techniques, as it is used in significant data settings.

Methodology Used:

- GANs generate synthetic fraud samples to improve detection.
- Discriminator distinguishes real vs. fake transactions.
- Model trained with adversarial learning.

Advantages:

- Handles class imbalance effectively.
- Improves fraud detection accuracy.
- Generates realistic fraudulent samples.

Disadvantages:

- Computationally expensive.
- Requires careful hyper parameter tuning.
- Risk of mode collapse in GANs.

2.15. Intelligent Fraud Detection in Financial Statements Using Machine Learning and Data Mining: A Systematic Literature Review

AUTHOURS: Matin N. Ashtiani and Bijan Raahemi.

PUBLISHEDBY: IEEE.

YEAR: 2021.

Description

Fraudulent financial statements (FFS) are the results of manipulating financial elements by overvaluing incomes, assets, sales, and profits while underrating expenses, debts, or losses. To identify such fraudulent statements, traditional methods, including manual auditing and inspections, are costly, imprecise, and time-consuming. Intelligent methods can significantly help auditors in analyzing a large number of financial statements. In this study, we systematically review and synthesize the existing literature on intelligent fraud detection in corporate financial statements.

Methodology Used:

- Machine learning techniques (e.g., SVM, Decision Trees, Random Forest, Neural Networks)
- Data mining approaches (e.g., anomaly detection, clustering, rule-based classification)
- Feature selection methods to identify financial fraud indicators

Advantages:

- High accuracy in fraud detection
- Automates the fraud detection process, reducing manual effort

Disadvantages:

- Requires high-quality labeled data for supervised learning
- Potential bias in model training due to imbalanced datasets

2.16. Machine Learning Methods for Credit Card Fraud Detection: A Survey.

AUTHOURS: Kanishka Ghosh Dastidar, Olivier Caelen.

PUBLISHEDBY: IEEE.

YEAR: 2024.

Description

The widespread adoption of online payments has been accompanied by a significant increase in fraudulent activities, resulting in billions of dollars in financial losses. As payment providers aim to tackle this with various preventive mechanisms, fraudsters also continuously evolve their methods to remain indistinguishable from genuine actors. As the volume of transactions taking place per day is in the millions, relying solely on human investigation is expensive and ultimately unfeasible, leading to an emergence of research into data driven or statistical methods for fraud detection.

Methodology Used:

- **Supervised Learning** – Uses labeled fraudulent and non-fraudulent transactions.
- **Unsupervised Learning** – Detects anomalies without labeled data.
- **Hybrid Models** – Combines supervised and unsupervised approaches for improved accuracy.

Advantages:

- **High Accuracy:** Especially deep learning and ensemble models.
- **Automated Feature Extraction:** Reduces manual effort (Deep Learning).

Disadvantages:

- **Data Imbalance:** Fraud cases are rare, leading to biased models.
- **High Computational Cost:** Deep learning requires significant resources.

2.17. Medicare Fraud Detection Using Graph Analysis: A Comparative Study of Machine Learning and Graph Neural Networks.

AUTHOURS: Yeeun Yoo, Jinho Shin, Sunghyon Kyeong.

PUBLISHEDBY: IEEE.

YEAR: 2023.

Description

Insurance companies have focused on Medicare fraud detection to reduce financial losses and reputational harm because Medicare fraud causes tens of billions of dollars in damage annually. This study demonstrates that Medicare fraud detection can be significantly enhanced by introducing graph analysis with considering the relationships among medical providers, beneficiaries, and physicians. We use open-source tabular datasets containing beneficiary information, inpatient claims, outpatient claims, and indications about potential fraudulent providers.

Methodology Used:

- Constructing a graph representation of Medicare claims data.
- Feature extraction from network structures (e.g., provider-patient relationships).

Advantages:

- Captures complex relationships in Medicare fraud cases.
- GNNs improve fraud detection by leveraging graph structures.

Disadvantages:

- High computational cost for large-scale graph processing.
- Requires quality graph construction for accurate results.
- Interpretability of GNNs is lower compared to traditional models.

2.18. Online Payment Fraud Detection Model Using Machine Learning Techniques.

AUTHOURS: Abdulwahab Ali Almazroi, Nasir Ayub.

PUBLISHEDBY: IEEE.

YEAR: 2023.

Description

In a world where wireless communications are critical for transferring massive quantities of data while protecting against interference, the growing possibility of financial fraud has become a significant concern. The ResNeXt-embedded Gated Recurrent Unit (GRU) model (RXT) is a unique artificial intelligence approach precisely created for real-time financial transaction data processing. Motivated by the need to address the rising threat of financial fraud, which poses major risks to financial institutions and customers, our artificial intelligence technique takes a systematic approach.

Methodology Used:

- Data preprocessing (handling missing values, feature selection)
- Feature engineering (transaction patterns, user behavior analysis)

Advantages:

- High accuracy in fraud detection
- Real-time transaction monitoring
- Reduces false positives with advanced ML techniques

Disadvantages:

- Requires high-quality labeled data
- Computationally expensive for real-time processing
- Possible model bias leading to false positives/negatives

2.19. Opt Dev Net: A Optimized Deep Event-Based Network Framework for Credit Card Fraud Detection.

AUTHOURS: Muhammad Adil, Zhang Yinjun.

PUBLISHEDBY: IEEE.

YEAR: 2024.

Description

In recent times, credit card fraud has emerged as a substantial financial challenge for both cardholders and the issuing authorities. To address this demanding issue, researchers have employed machine learning techniques to identify fraudulent activities within labeled transaction records. However, these techniques have primarily been evaluated on limited or specific datasets, which may not adequately represent the broader real-world scenario. These limitations motivated us to comprehensively assess the existing machine learning classifiers and propose an Optimized Deep Event-based Network (OptDevNet) framework capable of addressing these challenges.

Methodology Used:

- Deep learning-based event-driven network
- Feature extraction from transaction sequences
- Optimization techniques for performance improvement

Advantages:

- High accuracy in fraud detection
- Real-time processing capability
- Handles sequential transaction patterns effectively

Disadvantages:

- Computationally expensive
- Requires large labeled datasets
- May struggle with adaptive fraud patterns

2.20. Quantum Auto encoder for Enhanced Fraud Detection in Imbalanced Credit Card Dataset.

AUTHOURS: Chansreynich Huot, Sovanmonynuth Heng.

PUBLISHEDBY: IEEE.

YEAR: 2024.

Description

Credit card fraud detection is crucial for financial security which entails identifying unauthorized transactions that can result in significant financial losses. Detection is inherently challenging due to the rarity and indistinguishability of fraudulent transactions from genuine ones, which makes it an anomaly detection problem. Traditional detection systems struggle with the highly imbalanced nature of transaction datasets, where genuine transactions vastly outnumber fraudulent cases. In response to these challenges, we propose a novel detection model utilizing Quantum Auto Encoders-based Fraud Detection (QAE-FD).

Methodology Used :

- Quantum autoencoder for feature extraction and dimensionality reduction.
- Hybrid quantum-classical model combining quantum encoding with classical classification.

Advantages:

- Efficient feature compression improves fraud detection.
- Quantum computing enhances model scalability.
- Handles high-dimensional data effectively.

Disadvantages:

- Requires quantum hardware, limiting real-world deployment.
- Noisy quantum circuits can affect accuracy.

CHAPTER 3
METHODOLOGY

Objectives

The goal is to develop a deep learning model for real or fake job Prediction, to potentially replace the updatable supervised machine learning classification models by predicting results in the form of best accuracy by comparing supervised algorithm.

Project Goals

- Exploration data analysis of variable identification
    - Loading the given dataset
    - Import required libraries packages
    - Analyze the general properties
    - Find duplicate and missing values
    - Checking unique and count values
- Uni-variate data analysis
    - Rename, add data and drop the data
    - To specify data type
- Exploration data analysis of bi-variate and multi-variate
    - Plot diagram of pairplot, heatmap, bar chart and Histogram
- Method of Outlier detection with feature engineering
    - Pre-processing the given dataset
    - Splitting the test and training dataset
    - Comparing the Decision tree and Logistic regression model and random forest etc.

Scope of the Project

The main Scope is to detect the fake job, which is a classic text classification problem with a help of NLP and machine learning algorithm. It is needed to build a model that can differentiate between ―Real‖ job and ―Fake‖ job.

3.1. EXISTING SYSTEM

the growing reliance on digital platforms for hiring, many companies now post job vacancies online to simplify the recruitment process. However, this surge has also led to a rise in online recruitment fraud (ORF), where scammers post fake job advertisements for monetary gain. Traditional machine learning and deep learning methods have been applied to identify such fraudulent listings. These existing approaches largely rely on outdated benchmark datasets, which contain limited and specific job postings. Consequently, models trained on these datasets struggle to generalize and detect fake job posts effectively across diverse and evolving job markets.

SMoreover, class imbalance—where fake job postings form the minority—further challenges model performance, leading to biased predictions and lower accuracy for detecting fraudulent listings.

3.2. PROPOSED SYSTEM

The proposed system introduces a more robust and accurate approach to detecting fake job postings using transformer-based deep learning models, specifically BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa (Robustly Optimized BERT-Pretraining Approach).

A novel dataset has been developed by combining job postings from three distinct sources, overcoming the limitations of outdated benchmark datasets. This enriched dataset enhances model training by incorporating recent and diverse job listings.

To address the issue of class imbalance, the system implements ten high-performing variants of the Synthetic Minority Oversampling Technique (SMOTE). Each SMOTE variant is applied to balance the dataset, and its impact on model performance is evaluated.

Experimental results show that all models perform competitively; however, the BERT model combined with SMOBD SMOTE achieves the highest balanced accuracy and recall, approximately 90%, making it the most effective configuration for detecting fake job postings.

3.3. FEASIBILITY STUDY:

3.3.1. Data Wrangling

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

3.3.2. Data collection

The data set collected for predicting given data is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, logistic, Decision tree algorithms and Support vector classifier (SVC) are applied on the Training set and based on the test result accuracy, Test set prediction is done.

3.3.3. Preprocessing

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed so as to improve the efficiency of the algorithm. The outliers have to be removed and also variable conversion need to be done.

3.3.4. Building the classification model

The prediction of wine quality, A high accuracy prediction model is effective because of the following reasons: It provides better results in classification problem.

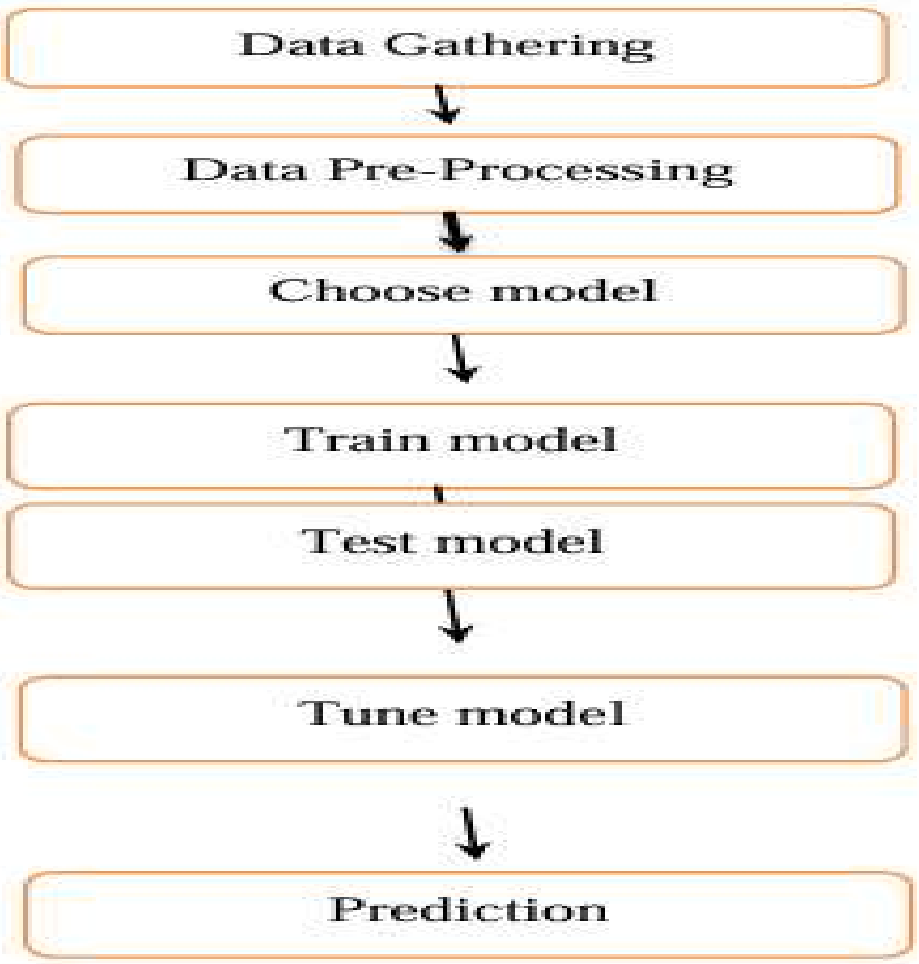
- It is strong in preprocessing outliers, irrelevant variables, and a mix of continuous, categorical and discrete variables.

- It produces out of bag estimate error which has proven to be unbiased in many tests and it is relatively easy to tune with.

3.3.5. Construction of a Predictive Model

Deep learning needs data gathering have lot of past data‘s. Data gathering have sufficient historical data and raw data. Before data pre-processing, raw data can‘t be used directly. It‘s used to preprocess then, what kind of algorithm with model. Training and testing this model working and predicting correctly with minimum errors. Tuned model involved by tuned time to time with improving the accuracy.

To construct an effective predictive model, the preprocessed data is split into training and testing sets to evaluate the model's performance. Various deep learning algorithms such as Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), or Recurrent Neural Networks (RNN) may be applied depending on the nature of the problem. During training, the model learns the patterns and relationships within the data, while testing evaluates its generalization capability. Performance metrics like accuracy, precision, recall, and F1-score help determine the model's effectiveness. Continuous tuning of hyperparameters, along with techniques like cross-validation and regularization, ensures that the model adapts over time and maintains high accuracy on new, unseen data.



Process of dataflow diagram

Fig.3.3.5

3.4. LIST OF MODULES:

- Data Pre-processing
- Data Analysis of Visualization
- Comparing Algorithm with prediction in the form of best accuracy result

3.5. PROJECT REQUIREMENTS:

3.5.1. General:

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements
2. Non-Functional requirements
3. Environment requirements
    A. Hardware requirements
    B. software requirements

3.5.2. Functional requirements:

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

3.5.2. Non-Functional Requirements:

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

3.6. ENVIRONMENTAL REQUIREMENTS:

1. Software Requirements:

    Operating System : Windows

    Tool : Anaconda with Jupyter Notebook

2. Hardware requirements:

    Processor : Pentium IV/III

Hard disk : minimum 80 GB

RAM : minimum 2 GB

3.7. SOFTWARE DESCRIPTION

Anaconda is a free the Python and R programming and languages open-source distribution for scientific of computing (data science, machine learning, Deep learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with Anaconda.

3.8. ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage Anaconda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda.org or in a local Anaconda Repository.

Anaconda Now, if you are primarily doing data science work, Anaconda is also a great option. Anaconda is created by Continuum Analytics, and it is a Python distribution that comes preinstalled with lots of useful python libraries for data science.

The following applications are available by default in Navigator:

- Jupyter Lab
- Jupyter Notebook
- Spyder
- Py Charm
- VS Code
- Glueviz
- Orange 3 App
- R Studio
- Anaconda Prompt (Windows only)
- Anaconda Power Shell (Windows only)

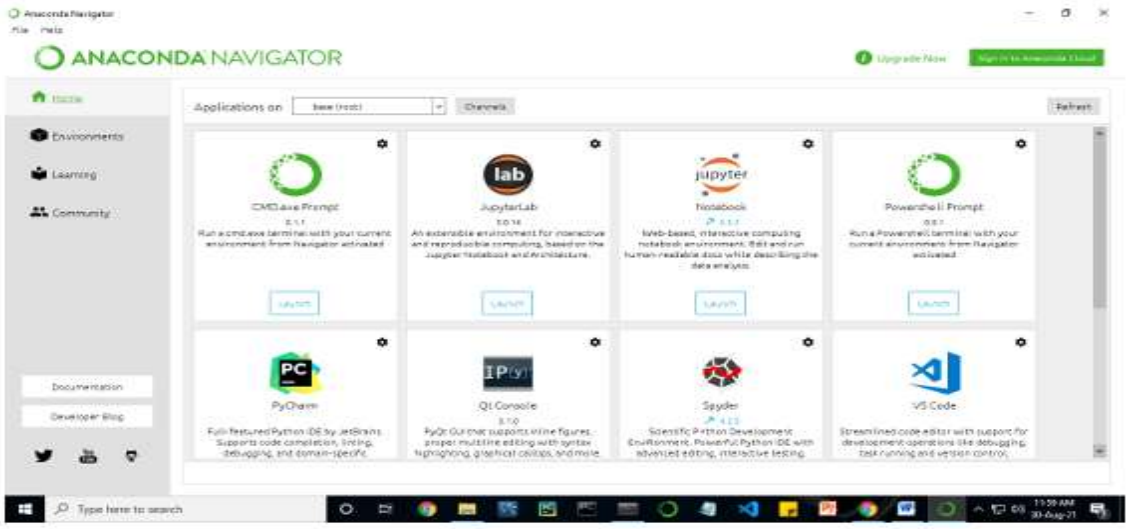


Fig 3.8.1

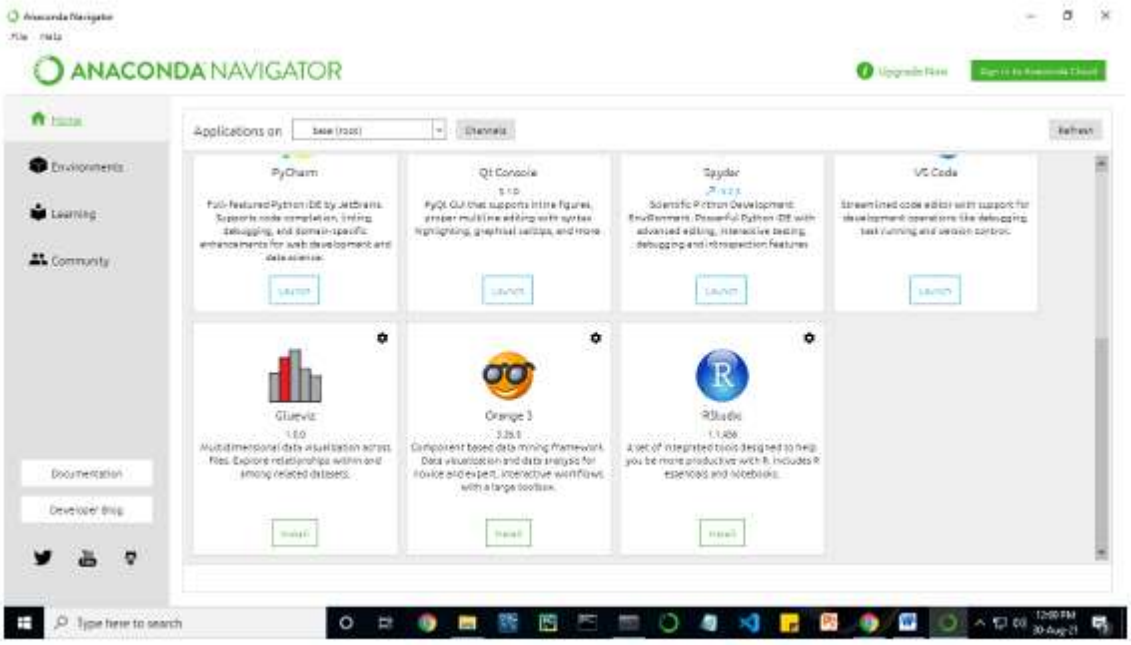


Fig 3.8.2

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution.

3.9. JUPYTER NOTEBOOK

This website acts as ―meta documentation for the Jupyter ecosystem. It has a collection of resources to navigate the tools and communities in this ecosystem, and to help you get started.

Project Jupyter is a project and community whose goal is to "develop open source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Perez.

3.10. PYTHON

Introduction:

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Designed to promote code clarity, Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. It is dynamically typed and includes automatic garbage collection. Python is widely used for both small scripts and large-scale applications. Its extensive standard library—often referred to as “batteries included”—provides built-in modules and tools for various tasks, making development faster and more efficient. Python’s versatility and ease of use have made it a popular choice in web development, data science, automation, AI, and many other domains.

Python is a high-level, interpreted, general-purpose programming language known for its emphasis on code readability, primarily through the use of significant indentation. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it suitable for a wide range of applications. Python is dynamically typed and includes automatic garbage collection, which simplifies memory management.

3.11. SYSTEM ARCHITECTURE:

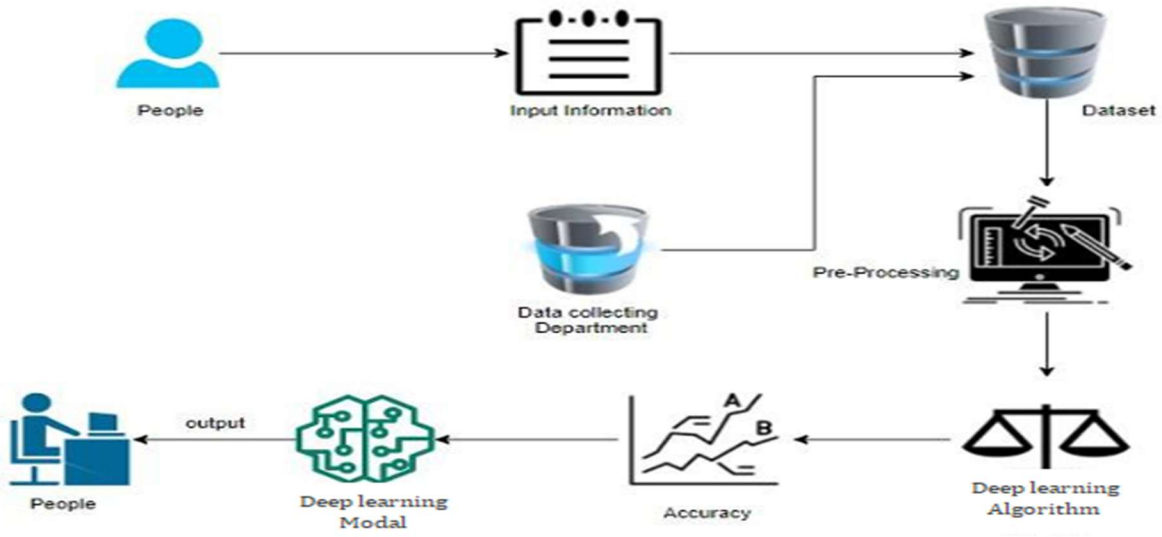


Fig.3.11.1

3.12. WORK FLOW DIAGRAM:

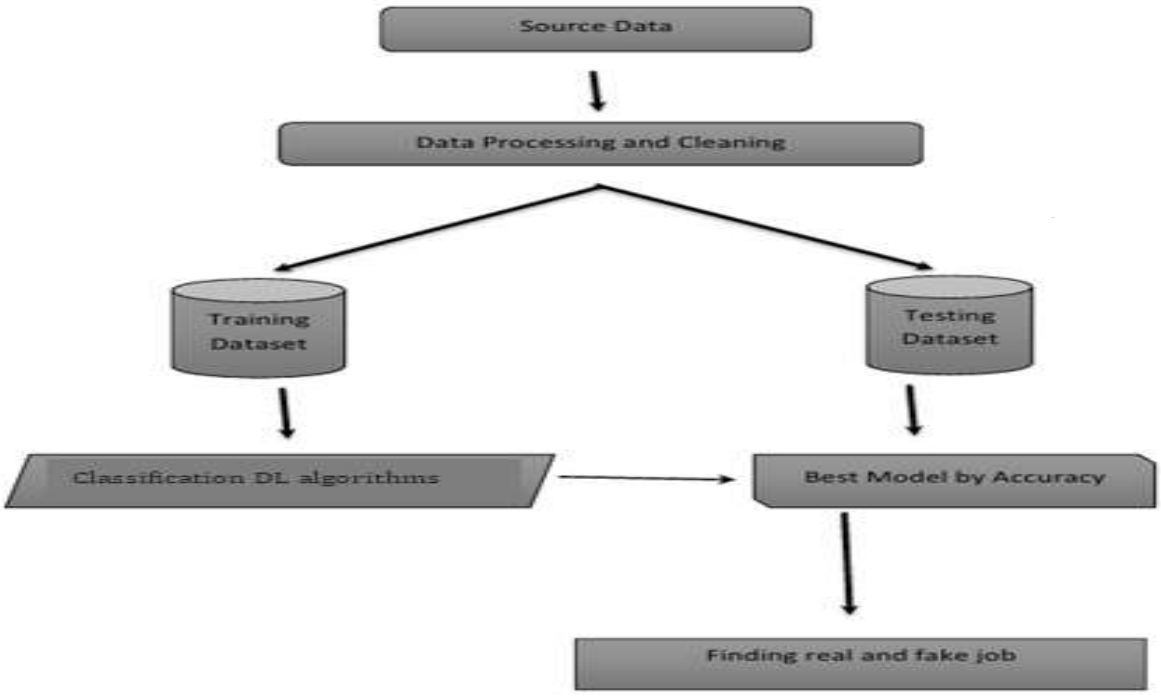


Fig.3.12.1

3.13. USE CASE DIAGRAM:

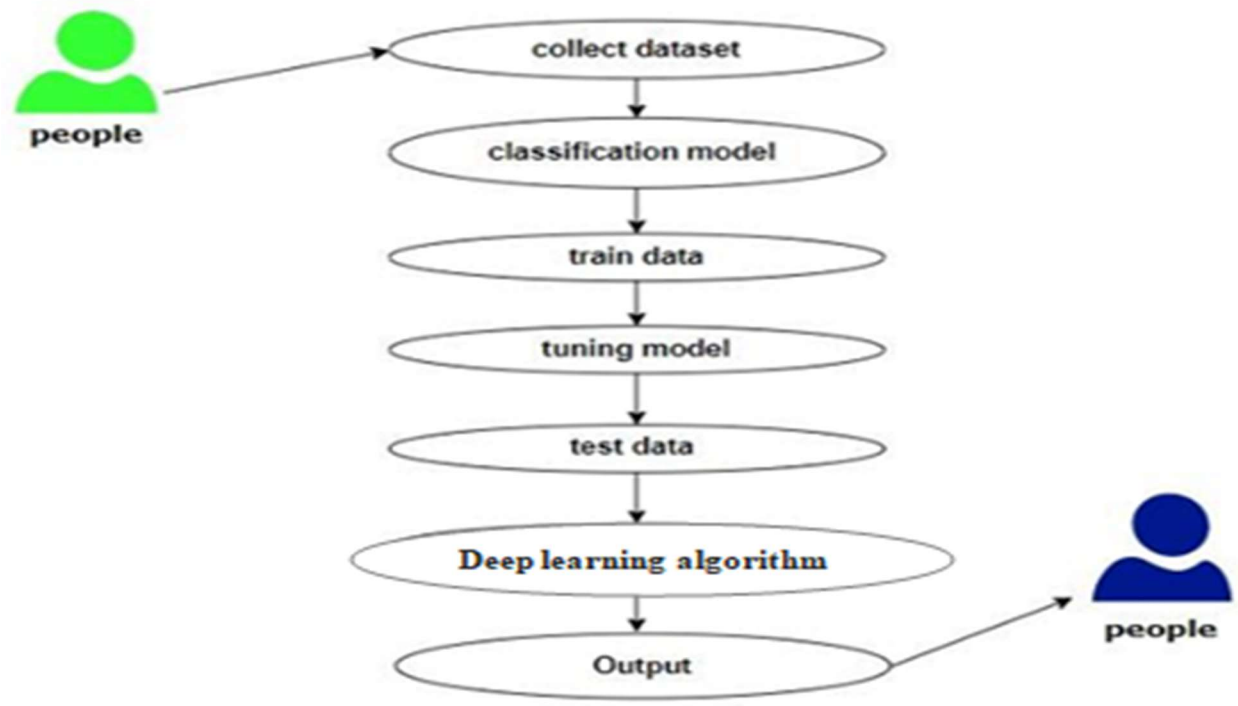


Fig.3.13.1

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner. Use case diagrams are essential tools for high-level requirement analysis in a system. They visually represent the interactions between users (actors) and the system to capture all possible functionalities in an organized format. By analyzing the requirements, use cases help identify what the system should do, making it easier to understand and communicate system behavior. They serve as a blueprint for both development and validation, ensuring that all user needs are addressed. Use case diagrams are especially valuable in early stages of system design, as they provide clarity, improve documentation, and support collaboration among stakeholders, developers, and analysts.

3.14. CLASS DIAGRAM:

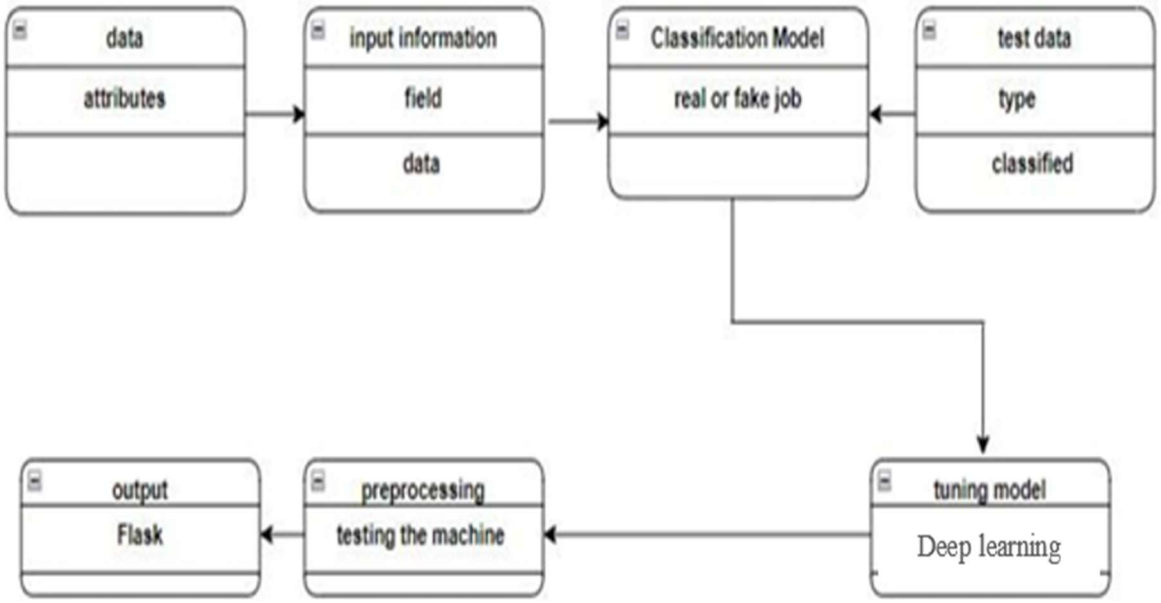


Fig 3.14.1

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because, unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

3.15. ACTIVITY DIAGRAM:

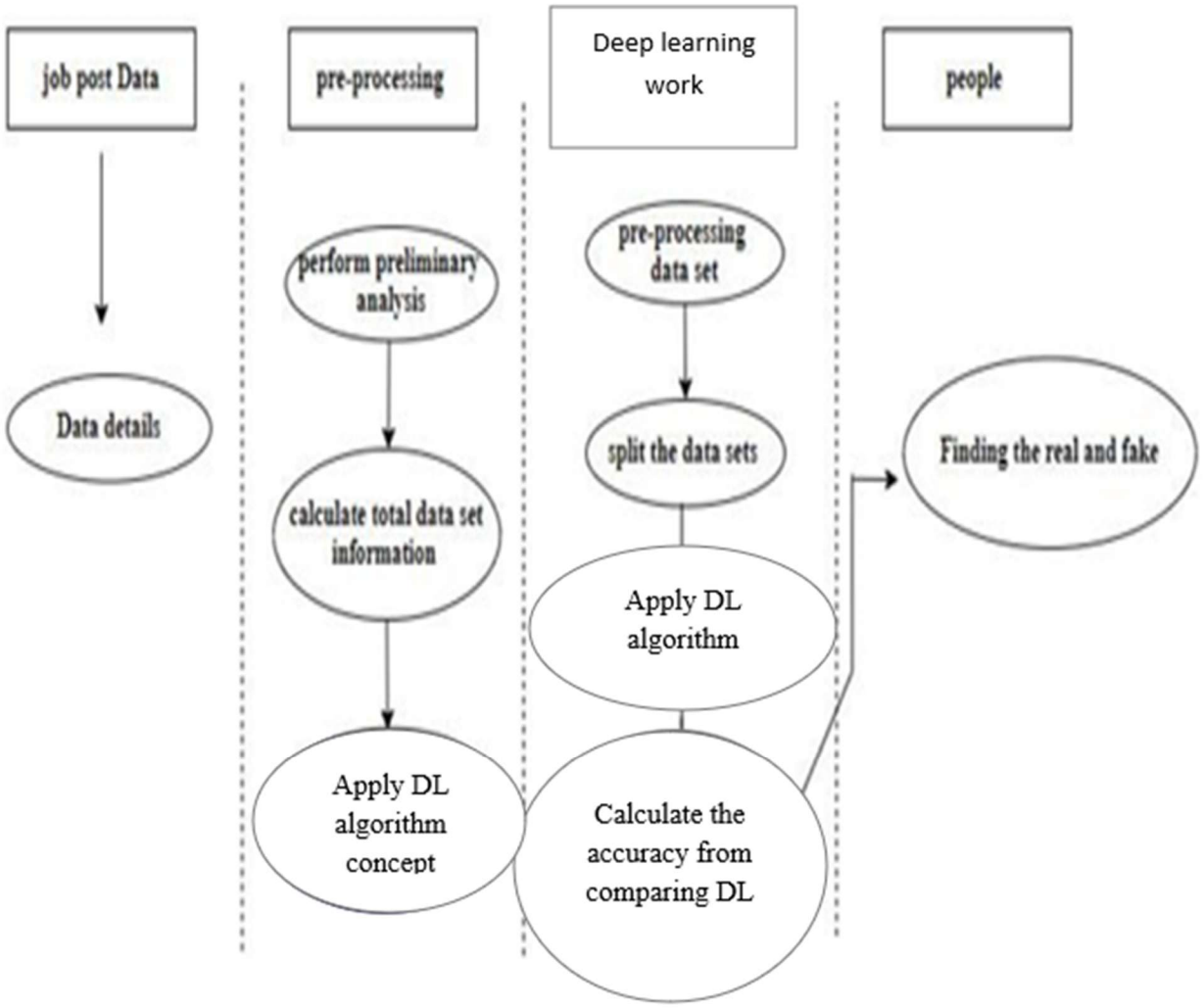


Fig 3.15.1

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

3.16. SEQUENCE DIAGRAM:

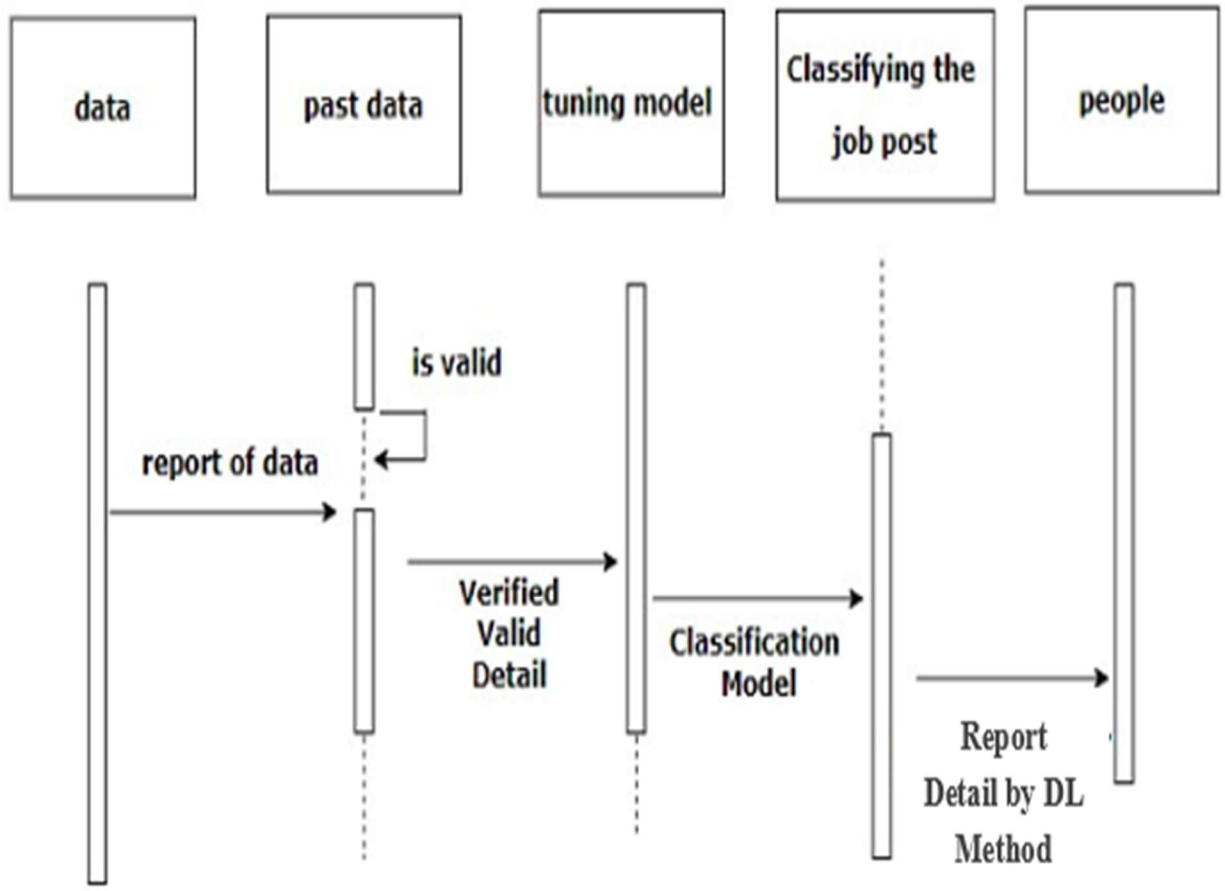


Fig 3.16.1

Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system. Other dynamic modeling techniques include activity diagramming, communication diagramming, timing diagramming, and interaction overview diagramming. Sequence diagrams, along with class diagrams and physical data models are in my opinion the most important design level models for modern business application development.

3.17. ENTITY RELATIONSHIP DIAGRAM (ERD):

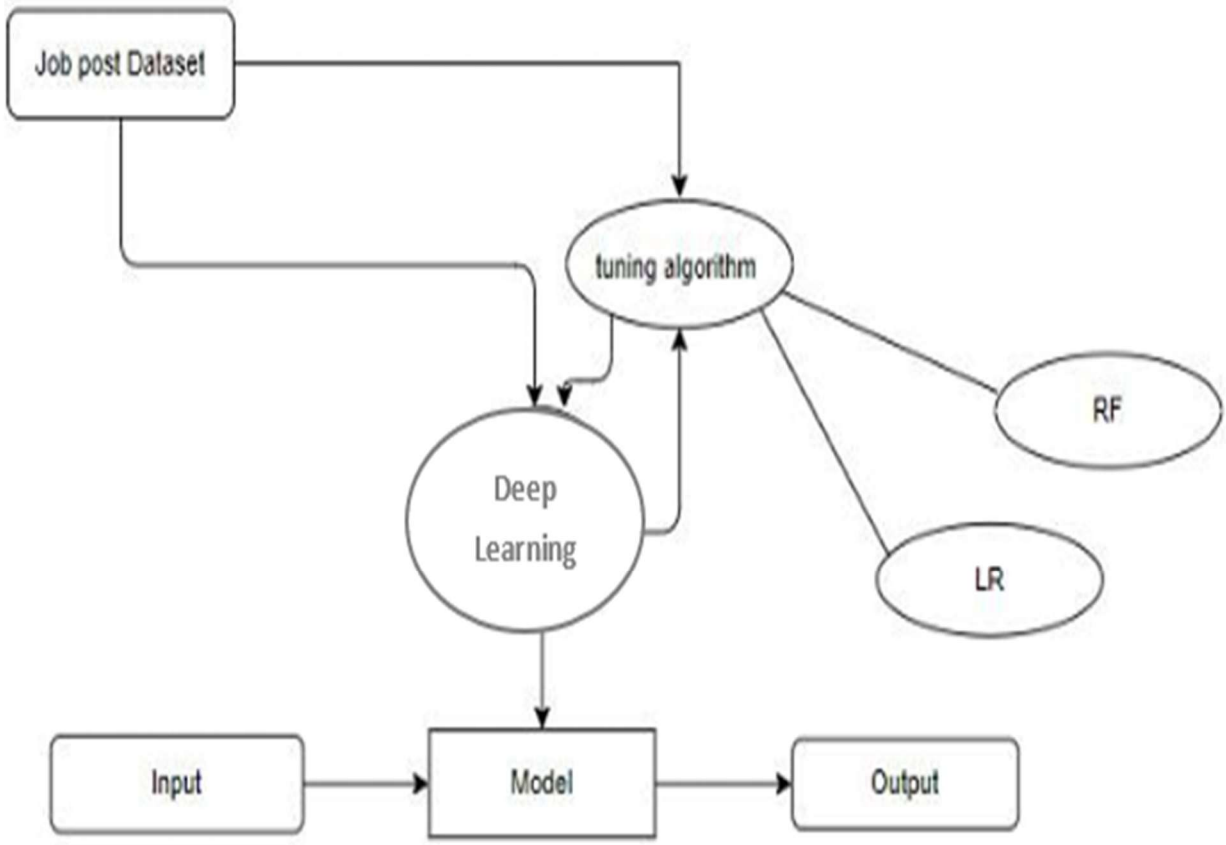


Fig 3.17.1

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database. Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

CHAPTER 4
RESULT AND DISCUSSION

The deep learning approach is developed in this model to predict the real or fake job posting.To view the dependent and independent variable it is preprocessed and examined, and to get the results various machine learning algorithms are used. Dataset are combined to form a common dataset and these would be used to get the pattern and result with higher accuracy .The data will be loaded,cleaned and trimmed for result.

4.1. DISCUSSION

4.1.1. Data Pre-processing

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters.

Some of these sources are just simple random mistakes. Other times, there can be a deeper reason why data is missing. It‘s important to understand these different types of missing data from a statistics point of view. The type of missing data will influence how to deal with filling in the missing values and to detect missing values, and do some basic imputation and detailed statistical approach for dealing with missing data. Before, joint into code, it‘s important to understand the sources of missing data.

Here are some typical reasons why data is missing:

- User forgot to fill in a field.
- Data was lost while transferring manually from a legacy database.
- There was a programming error.
- Users chose not to fill out a field tied to their beliefs about how the results would be used or interpreted.

Variable identification with Uni-variate, Bi-variate and multi-variate analysis:

- import libraries for access and functional purpose and read the given dataset

- General Properties of Analyzing the given dataset
- Display the given dataset in the form of data frame
- Show columns
- Shape of the data frame
- To describe the data frame
- Checking data type and information about dataset
- Checking for duplicate data
- Checking Missing values of data frame
- Checking unique values of data frame
- Checking count values of data frame
- Rename and drop the given data frame
- To specify the type of values
- To create extra columns

4.1.2. Data Validation/ Cleaning/Preparing Process

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

The process begins by importing the necessary library packages and loading the provided dataset. Initial analysis involves identifying variables through examining the dataset's shape, data types, and assessing the presence of missing or duplicate values. A validation dataset is separated from the training data to estimate model performance during the tuning process, ensuring the model's ability to generalize well to unseen data.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1404 entries, 0 to 1403
Data columns (total 19 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Unnamed: 0           1404 non-null   int64
 1   job_id               1404 non-null   int64
 2   title                1404 non-null   object
 3   location             1404 non-null   object
 4   department           1404 non-null   object
 5   salary_range         1404 non-null   object
 6   company_profile      1404 non-null   object
 7   description          1404 non-null   object
 8   requirements         1404 non-null   object
 9   benefits             1404 non-null   object
 10  telecommuting        1404 non-null   int64
 11  has_company_logo     1404 non-null   int64
 12  has_questions        1404 non-null   int64
 13  employment_type      1404 non-null   object
 14  required_experience  1404 non-null   object
 15  required_education   1404 non-null   object
 16  industry             1404 non-null   object
 17  function             1404 non-null   object
 18  fraudulent           1404 non-null   object
dtypes: int64(5), object(14)
memory usage: 219.4+ KB
```

Fig.(a)

	Unnamed: 0	job_id	telecommuting	has_company_logo	has_questions
count	1404.000000	1404.000000	1404.000000	1404.000000	1404.000000
mean	701.500000	7967.114672	0.128205	0.965100	0.687322
std	405.444201	4862.817023	0.334437	0.183593	0.463750
min	0.000000	7.000000	0.000000	0.000000	0.000000
25%	350.750000	4403.000000	0.000000	1.000000	0.000000
50%	701.500000	7091.000000	0.000000	1.000000	1.000000
75%	1052.250000	11740.000000	0.000000	1.000000	1.000000
max	1403.000000	17866.000000	1.000000	1.000000	1.000000

Fig.(b)

MODULE DIAGRAM

Fig.(c)

GIVEN INPUT EXPECTED OUTPUT

- ✓ Input: data
- ✓ Output: removing noisy data

4.1.3. Exploration data analysis of visualization

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

- ➢ How to chart time series data with line plots and categorical quantities with bar charts.
- ➢ How to summarize data distributions with histograms and box plots.

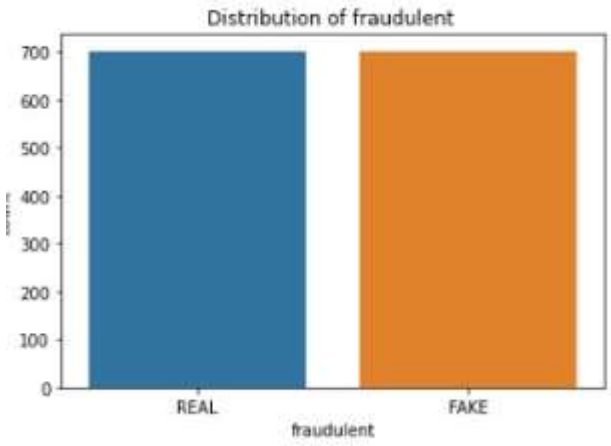


Fig.(a)

MODULE DIAGRAM

Fig.(b)

GIVEN INPUT EXPECTED OUTPUT

- ✓ Input: data
- ✓ Output: visualized data

4.1.4. Comparing Algorithm with prediction in the form of best accuracy result

It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics.

In the example below 4 different algorithms are compared:

- ➢ Logistic Regression
- ➢ Random Forest

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set. And another aspect is that data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms are executed in given dataset.

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. It needs the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

True Positive Rate (TPR) = TP / (TP + FN)

False Positive Rate (FPR) = FP / (FP + TN)

Accuracy:

The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non defaulters.

Accuracy calculation:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct.

Precision = TP / (TP + FP)

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = TP / (TP + FN)

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it‘s better to look at both Precision and Recall.

General Formula:

F- Measure = 2TP / (2TP + FP + FN)

F1-Score Formula:

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

4.2. ALGORITHM AND TECHNIQUES

4.2.1. Algorithm Explanation

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc. In Supervised Learning, algorithms learn from labeled data. After understanding the data, the algorithm determines which label should be given to new data based on pattern and associating the patterns to the unlabeled new data.

4.2.2. Used Python Packages:

sklearn:

- In python, sklearn is a machine learning package which include a lot of ML algorithms.
- Here, we are using some of its modules like train_test_split, DecisionTreeClassifier or Logistic Regression and accuracy_score.

NumPy:

- It is a numeric python module which provides fast maths functions for calculations.
- It is used to read data in numpy arrays and for manipulation purpose.

Pandas:

- Used to read and write different files.
- Data manipulation can be done easily with data frames.

Matplotlib:

- Data visualization is a useful way to help with identify the patterns from given dataset.
- Data manipulation can be done easily with data frames.

4.2.3. Natural Language Processing (NLP):

Natural language processing (NLP) allows machines to read and understand human language. A sufficiently powerful natural language processing system would enable natural-language user interfaces and the acquisition of knowledge directly from human-written sources, such as newswire texts. Some straightforward applications of natural language processing include information retrieval, text mining, question answering and machine translation. Many current approaches use word co-occurrence frequencies to construct syntactic representations of text. ―Keyword spotting‖ strategies for search are popular and scalable but dumb; a search query for ―dog‖ might only match documents with the literal word ―dog‖ and miss a document with the word ―poodle‖. ―Lexical affinity‖ strategies use the occurrence of words such as ―accident‖ to assess the sentiment of a document. Modern statistical NLP approaches can combine all these strategies as well as others, and often achieve acceptable accuracy at the page or paragraph level. Beyond semantic NLP, the ultimate goal of ―narrative‖ NLP is to embody a full understanding of commonsense reasoning.

1. Remove stopwords: There are a few words which are very commonly used when humans interact, but these words don‘t make any sense or add any extra value. Additionally, there might be few words which are not required for the business case given in hand. So, these words need to be deleted from the data.

2. Tokenization: This is one of the common practices while working on text data. This helps to split a phrase, sentence, or paragraph into small units like words or terms. Each unit is called a token. There are different types of tokenization. We have already used this in above examples for stemming, POS tagging, and NER. Below are different ways to tokenize the text.

4.2.4. Vectorization/Word Embedding

Once cleaning and tokenization is done, extracting features from the clean data is very important as the machine doesn‘t understand the words but numbers. Vectorization helps to map the words to a vector of real numbers, which further helps into predictions. This helps to extract the important features.

4.2.5. Logistic Regression

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts P(Y=1) as a function of X. Logistic regression Assumptions:

- Binary logistic regression requires the dependent variable to be binary.
- For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
- Only the meaningful variables should be included.
- The independent variables should be independent of each other. That is, the model should have little.
- The independent variables are linearly related to the log odds.

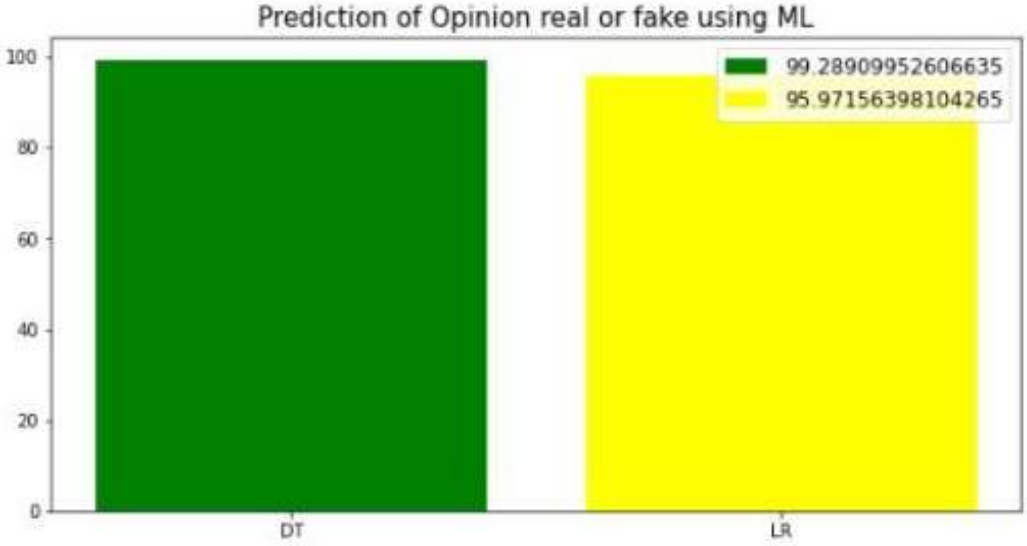


Fig.(a)

```
Accuracy result of Random Forest Classifierr  is: 99.28909952606635


Classification report of Random Forest Classifier : Results:

              precision    recall  f1-score   support

           0       1.00      0.99      0.99       211
           1       0.99      1.00      0.99       211

    accuracy                           0.99       422
   macro avg       0.99      0.99      0.99       422
weighted avg       0.99      0.99      0.99       422

Confusion Matrix result of Random Forest Classifier : is:
 [[208   3]
 [  0 211]]

Sensitivity :  0.985781990521327

Specificity :  1.0
```

Fig.(b)

MODULE DIAGRAM

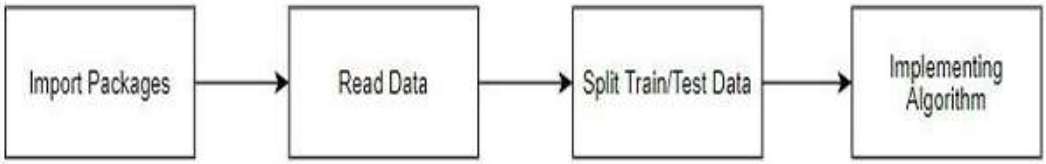


Fig.(c)

GIVEN INPUT EXPECTED OUTPUT

- ✓ Input: data
- ✓ Output: getting accuracy

4.2.6. Random Forest Classifier

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees 'habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithms of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

- ➢ Pick N random records from the dataset.
- ➢ Build a decision tree based on these N records.
- ➢ Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

```
In [16]: from sklearn.feature_extraction.text import CountVectorizer
         from sklearn.model_selection import train_test_split

In [17]: cv = CountVectorizer()
         X = cv.fit_transform(x) # Fit the Data
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)

In [18]: from sklearn.linear_model import LogisticRegression
         clf = LogisticRegression()
         clf.fit(X_train,y_train)
```

Fig.(a)

```
In [21]: input_word=input("ENTER THE SENTENCE:")

ENTER THE SENTENCE:Qualified candidates are encouraged to apply directly to this job posting. Â Direct email and phone calls ar
e not being considered. Thank you for your cooperation. Â Please no recruiters. Â UST Testing Technician IIÂ Bakersfield, CAÂ L
ocal PetroleumÂ Company operates primarily in retail and wholesale of motor fuels and other related petroleum products and is s
eeking talented, experienced, maintenance technicians to perform troubleshooting and maintenance on their retail gasoline equip
ment.Â RESPONSIBILITIESThe ideal candidate will have experience working with retail gasoline dispensing and peripheral equipmen
t, UST systems, weights and measures compliance, and computer applications. Certifications with Gilbarco, Ruby, Sapphire, and V
eederRoot Tank Gauging preferred.Schedule testing with local agencies.Â

In [22]: data = cv.transform([input_word]).toarray()
         print(clf.predict(data))

['FAKE']
```

Fig.(b)

MODULE DIAGRAM

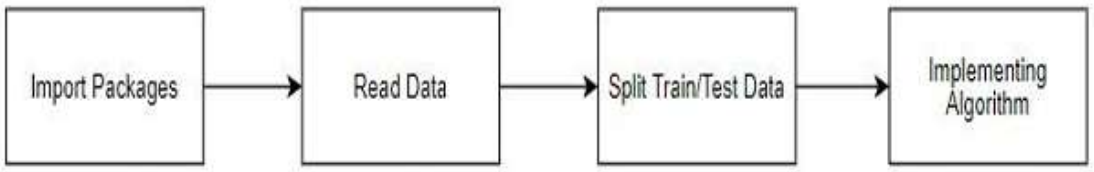


Fig.(c)

GIVEN INPUT EXPECTED OUTPUT

- ✓ input: data
- ✓ output: getting accuracy

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1. CONCLUSION

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. The best accuracy on public test set is higher accuracy score will be found out. This application can help to find the Prediction of Real and fake job.

5.2. FUTURE WORK

In the future, the proposed system will be expanded to integrate with cloud-based infrastructure for real and fake job prediction. This will allow for more scalable processing, enabling the handling of vast amounts of recruitment data from multiple sources in real time. Cloud integration will also support centralized data access, easier updates, and faster deployment of detection models, significantly improving system performance and efficiency.

Further enhancements will involve optimizing the system within an advanced Artificial Intelligence (AI) environment. By incorporating adaptive learning models, the system will be capable of continuously learning from new patterns and behaviors, improving accuracy over time. Real-time detection capabilities will be developed to ensure immediate identification and flagging of fraudulent job postings. These dynamic learning techniques will reduce manual intervention and ensure the system remains up to date with emerging threats in the online recruitment landscape.

Additionally, graph-based analysis will be introduced to map relationships between fraudulent users, job listings, and suspicious activity, revealing hidden networks. A cross-platform detection system will also be developed to ensure consistency and reliability across various devices and platforms. Ethical and legal frameworks will be strictly followed to ensure compliance with data privacy laws and responsible AI practices. Lastly, user behavior analytics will be implemented to monitor interaction patterns and enhance fraud detection based on suspicious user activity, making the system more proactive and secure.

APPENDICES

6.1 SOURCE CODE

Module-1

Main.py

```
from app import app
from routes import *
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000, debug=True)
```

Module-2

Preprocessing.py

```
import re
import string
class TextPreprocessor:
    """Class for preprocessing text data for the BERT model."""
    def __init__(self):
        # Simplified preprocessor without NLTK dependencies
        self.stop_words = set([
            'i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',
            "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself',
            'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her',
            'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them',
            'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom',
            'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was',
```

```
        'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do',
        'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or',
        'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with',
        'about', 'against', 'between', 'into', 'through', 'during', 'before',
        'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out',
        'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once',
        'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both',
        'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor',
        'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't',
        'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now',
        'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn',
        "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't",
        'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn',
        "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't",
        'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won',
        "won't", 'wouldn', "wouldn't"
    ])


def preprocess(self, text):
    """
    Preprocess text for BERT model.

    Args:
        text (str): Raw text to preprocess
```

```
    Returns:
        str: Preprocessed text
    """
    # Convert to lowercase
    text = text.lower()

    # Remove URLs
    text = re.sub(r'http\S+|www\S+|https\S+', '', text)

    # Remove HTML tags
    text = re.sub(r'<.*?>', '', text)

    # Remove special characters and numbers
    text = re.sub(r'[^\w\s]', '', text)
    text = re.sub(r'\d+', '', text)

    # Remove extra whitespace
    text = re.sub(r'\s+', ' ', text).strip()
    return text
def combine_features(self, title, description, location, url):
    """
    Combine job posting features into a single text.
    Args:
        title (str): Job title
        description (str): Job description
```

```
        location (str): Job location
        url (str): Job posting URL
    Returns:
        str: Combined preprocessed text
    """
    # Preprocess each field
    title = self.preprocess(title if title else "")
    description = self.preprocess(description if description else "")
    location = self.preprocess(location if location else "")

    # Extract domain from URL if present
    domain = ""
    if url:
        match = re.search(r'https?://(?:www\.)?([^/]+)', url)
        if match:
            domain = match.group(1)

    # Combine all features with special tokens for the model to distinguish them
      combined_text = f"title: {title} location: {location} description: {description} domain: {domain}"

    return combined_text
```

Module-3

Model.py

```
from datetime import datetime
from app import db
class JobPosting(db.Model):
    """Model for job postings to be analyzed."""
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(255), nullable=False)
    description = db.Column(db.Text, nullable=False)
    location = db.Column(db.String(255))
    url = db.Column(db.String(2048))
    is_fraudulent = db.Column(db.Boolean)
    confidence_score = db.Column(db.Float)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)
    def __repr__(self):
        return f"<JobPosting {self.title}>"
    def to_dict(self):
        """Convert job posting to a dictionary."""
        return {
            'id': self.id,
            'title': self.title,
            'description': self.description[:100] + '...' if len(self.description) > 100 else self.description,
            'location': self.location,
            'url': self.url,
            'is_fraudulent': self.is_fraudulent,
```

```
            'confidence_score': self.confidence_score,
            'created_at': self.created_at.strftime('%Y-%m-%d %H:%M:%S')
        }
```

Module-4

Froud_Detector.py

```
import logging
import random
class FraudDetector:
    """Class for detecting fraudulent job postings (simplified version without ML dependencies)."""
    def __init__(self):
        self.model_name = "dummy-model"  # Placeholder for real model
        self.max_length = 512
        self.tokenizer = None
        self.model = None
        self.logger = logging.getLogger(__name__)
    def load_model(self):
        """Simplified model loading (no actual model for demo)."""
        try:
            self.logger.info(f"Loading simplified model {self.model_name}...")
            self.model = "dummy-model"
            self.tokenizer = "dummy-tokenizer"
            self.logger.info("Model loaded successfully.")
            return True
```

```
    except Exception as e:
        self.logger.error(f"Error loading model: {str(e)}")
        return False
def predict(self, text):
    """
    Predict if a job posting is fraudulent based on text.
    This is a simplified version that uses rule-based detection instead of ML.
    Args:
        text (str): Preprocessed job posting text
    Returns:
        tuple: (is_fraudulent, confidence_score)
    """
    # Count suspicious keywords and phrases
    suspicious_keywords = [
        "immediate start", "work from home", "no experience",
        "flexible hours", "high income", "easy money", "apply now",
        "urgent hiring", "money back guarantee", "payment required",
        "no interview", "start today", "unlimited earning"
    ]
    text_lower = text.lower()
    count = sum(1 for keyword in suspicious_keywords if keyword in text_lower)
    # Simple heuristic - more keywords = higher probability
    keyword_ratio = count / len(suspicious_keywords)
    # Add some randomness for demo purposes
    base_score = keyword_ratio * 0.7 + random.random() * 0.3
```

```
    confidence = min(max(base_score, 0.1), 0.95)  # Keep between 0.1 and 0.95
    # Classify based on confidence threshold
    is_fraudulent = confidence > 0.5
    return is_fraudulent, confidence
def get_risk_factors(self, text):
    """
    Extract potential risk factors from job posting.
    Args:
        text (str): Job posting text
    Returns:
        list: Risk factors found in text
    """
    risk_factors = []
    # Common red flags in fraudulent job postings
    red_flags = {
        "payment upfront": "Requests for payment upfront",
        "no interview": "No formal interview process mentioned",
        "high salary": "Unusually high salary with minimal qualifications",
        "work from home": "Work from home with high salary promises",
        "personal information": "Requests for excessive personal information",
        "poor grammar": "Poor grammar or spelling errors",
        "no company details": "Vague or missing company information",
        "too good to be true": "Offers that seem too good to be true",
        "immediate start": "Immediate start without proper vetting",
        "vague job details": "Vague job descriptions or requirements"
```

```
    }
    # Check for red flags in the text
    text_lower = text.lower()
    for key, description in red_flags.items():
        if key in text_lower:
            risk_factors.append(description)
    return risk_factors
```

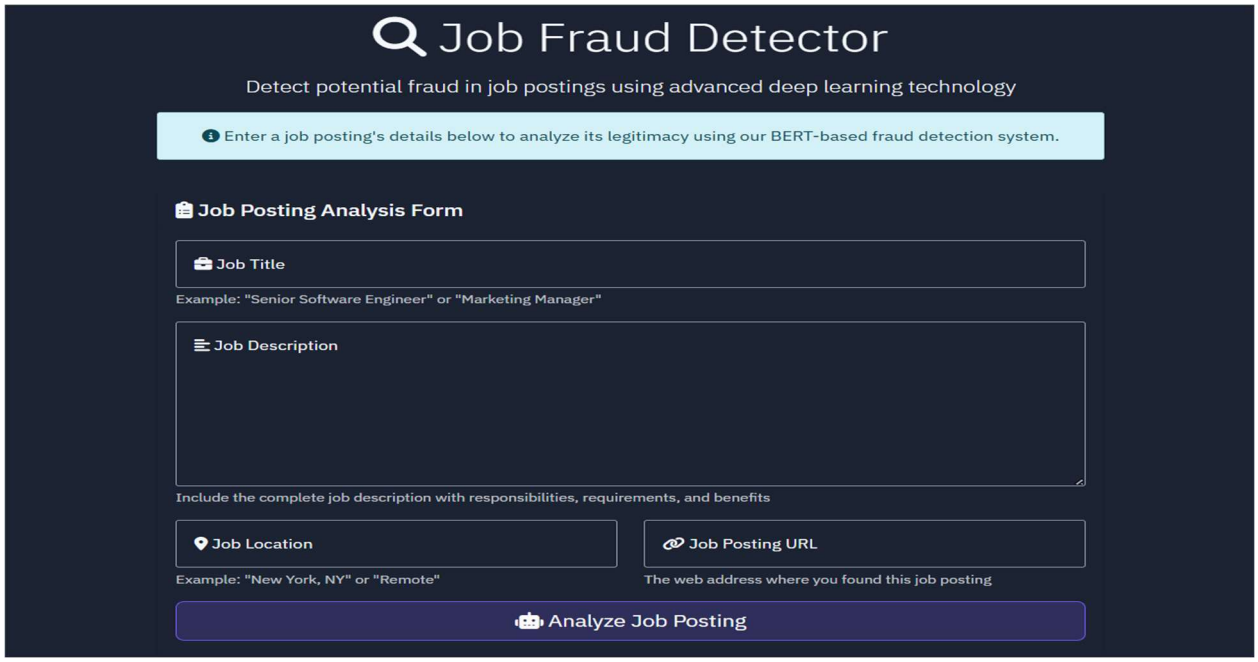
Model-5

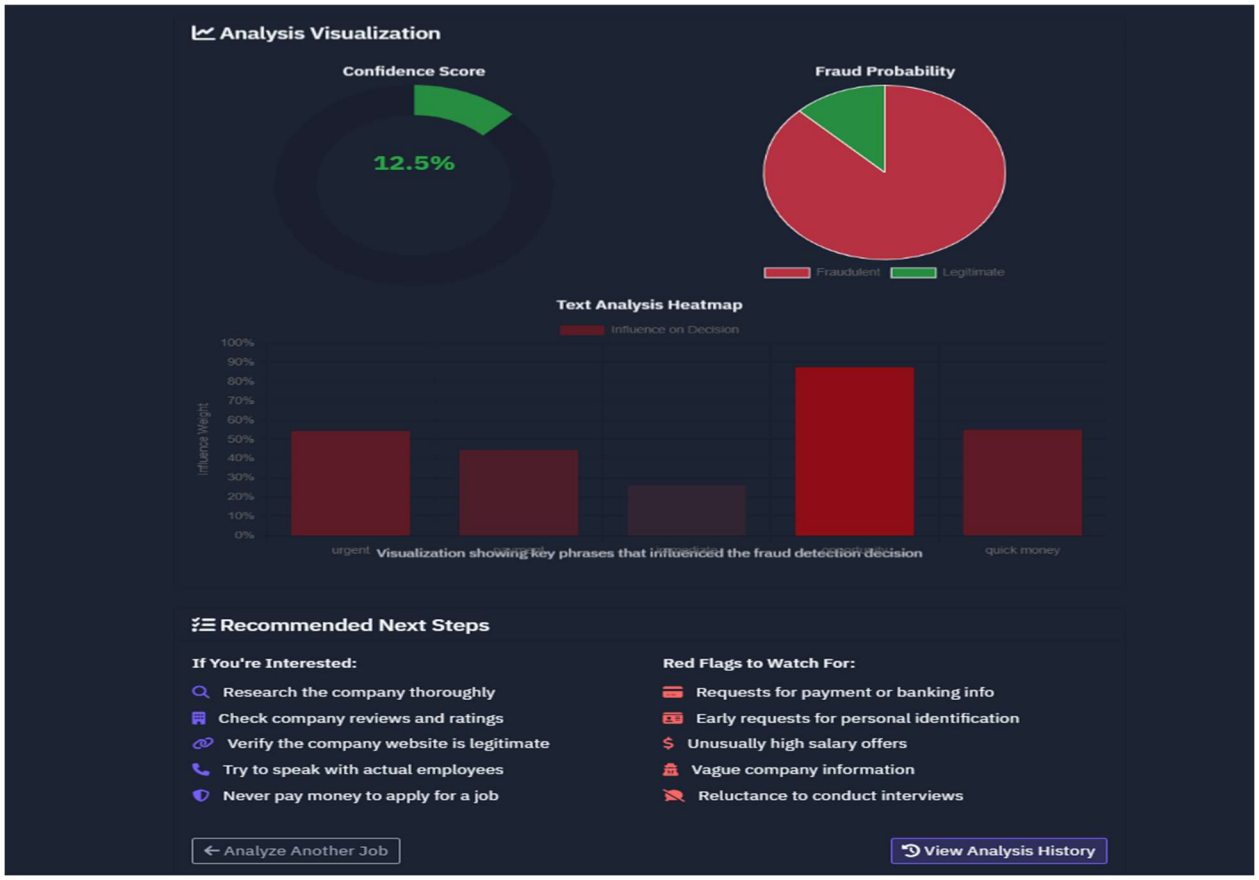
App.py

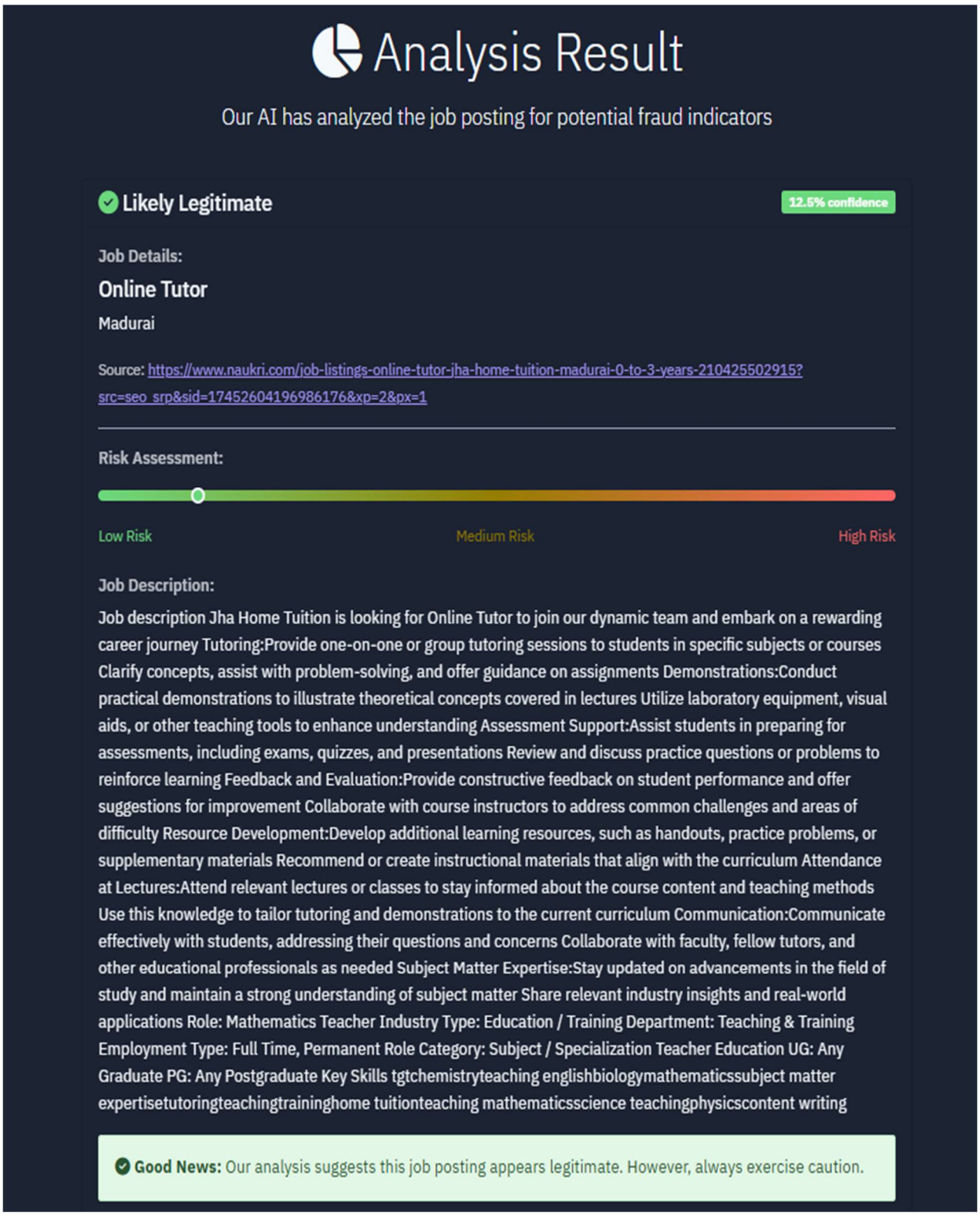
```
import os
import logging
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy.orm import DeclarativeBase
# Set up logging
logging.basicConfig(level=logging.DEBUG)
# Create SQLAlchemy base class
class Base(DeclarativeBase):
    pass
# Initialize SQLAlchemy
db = SQLAlchemy(model_class=Base)
# Create the Flask app
app = Flask(__name__)
app.secret_key = os.environ.get("SESSION_SECRET", "default_secret_key")
```

```
# Configure SQLAlchemy
app.config["SQLALCHEMY_DATABASE_URI"] = os.environ.get("DATABASE_URL", "sqlite:///orf_detector.db")
app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
app.config["SQLALCHEMY_ENGINE_OPTIONS"] = {
    "pool_recycle": 300,
    "pool_pre_ping": True,
}
# Initialize the app with SQLAlchemy
db.init_app(app)
# Import routes after app initialization to avoid circular imports
with app.app_context():
    from routes import *
    from models import *
    # Create all database tables
    db.create_all()
```

6.2 SCREENSHORT






Analysis Result
Our AI has analyzed the job posting for potential fraud indicators
Likely Legitimate
12.5% confidence
Job Details:
Online Tutor
Madurai
Source: https://www.naukri.com/job-listings-online-tutor-jha-home-tuition-madurai-0-to-3-years-210425502915?src=seo_srp&sid=17452604196986176&xp=2&px=1
Risk Assessment:
Low Risk
Medium Risk
High Risk
Job Description:
Job description Jha Home Tuition is looking for Online Tutor to join our dynamic team and embark on a rewarding career journey Tutoring:Provide one-on-one or group tutoring sessions to students in specific subjects or courses Clarify concepts, assist with problem-solving, and offer guidance on assignments Demonstrations:Conduct practical demonstrations to illustrate theoretical concepts covered in lectures Utilize laboratory equipment, visual aids, or other teaching tools to enhance understanding Assessment Support:Assist students in preparing for assessments, including exams, quizzes, and presentations Review and discuss practice questions or problems to reinforce learning Feedback and Evaluation:Provide constructive feedback on student performance and offer suggestions for improvement Collaborate with course instructors to address common challenges and areas of difficulty Resource Development:Develop additional learning resources, such as handouts, practice problems, or supplementary materials Recommend or create instructional materials that align with the curriculum Attendance at Lectures:Attend relevant lectures or classes to stay informed about the course content and teaching methods Use this knowledge to tailor tutoring and demonstrations to the current curriculum Communication:Communicate effectively with students, addressing their questions and concerns Collaborate with faculty, fellow tutors, and other educational professionals as needed Subject Matter Expertise:Stay updated on advancements in the field of study and maintain a strong understanding of subject matter Share relevant industry insights and real-world applications Role: Mathematics Teacher Industry Type: Education / Training Department: Teaching & Training Employment Type: Full Time, Permanent Role Category: Subject / Specialization Teacher Education UG: Any Graduate PG: Any Postgraduate Key Skills tgtchemistryteaching englishbiologymathematicssubject matter expertisetutoringteachingtraininghome tuitionteaching mathematicsscience teachingphysicscontent writing
Good News: Our analysis suggests this job posting appears legitimate. However, always exercise caution.

REFERENCE

[1] N. Akram, M. B. Amin, and S. M. A. Shah, "Online Recruitment Fraud (ORF) Detection Using Deep Learning Approaches", IEEE Access, 2022.

[2] P. R. Rani, P. Rajalakshmi, "Detection of fake job postings using ensemble learning and NLP", Materials Today: Proceedings, 2021.

[3] R. Bhat and S. Pathak, "Detecting Fake Job Postings Using Machine Learning", Journal of Information Technology and Software Engineering, 2020.

[4] S. Singh, A. Sharma, "Detecting Fraudulent Job Postings Using Natural Language Processing", International Journal of Computer Applications, 2020.

[5] Y. Kim, "Convolutional Neural Networks for Sentence Classification", EMNLP, 2014.

[6] Z. Yang et al., "Hierarchical Attention Networks for Document Classification", NAACL, 2016.

[7] Vaswani et al., "Attention is All You Need", NeurIPS, 2017.

[8] Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", NAACL, 2019.

[9] X. Zhang, J. Zhao, Y. LeCun, "Character-level Convolutional Networks for Text Classification", NeurIPS, 2015.

[10] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification", ACL, 2018.

[11] S. Sharma et al., "Fake News Detection using NLP and Machine Learning", Procedia Computer Science, 2019.

[12] V. Varshney, N. Vishwakarma, "Phishing URL Detection using Deep Learning Techniques", International Journal of Information Security, 2021.

[13] L. Wang et al., "Detecting Fake Online Reviews with Deep Learning", ACM Transactions on the Web, 2020

[14] T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space", arXiv, 2013. (Word2Vec foundational paper)

[15] D. F. Horsman et al., "Online scams: Exploring the psychological and emotional impact", Cyberpsychology, Behavior, and Social Networking, 2020.

[16] M. J. Moore and T. Florez, "Online Recruitment Fraud: Emerging Trends and Challenges", Journal of Cybersecurity Research, 2021.

[17] F. A. Shaikh et al., "Detection of Fraudulent Job Advertisements using Machine Learning", IEEE International Conference on Big Data, 2021.

[18] D. Kowsari et al., "Text Classification Algorithms: A Survey", Information, 2019.

[19] Kaggle Dataset – Fake Job Postings: https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction

[20] G. L. Ciampaglia et al., "The Spread of Fake Content: A Survey of Detection Techniques", IEEE Intelligent Systems, 2018.

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 04/70400243493

DOI: https://www.doi.org/10.56726/IRJMETS73923

Date: 28/04/2025

Certificate of Publication

This is to certify that author "**Dr. R. Gopi**" with paper ID "**IRJMETS70400243493**" has published a paper entitled "ONLINE RECRUITMENT FRAUD (ORF) DETECTION USING DEEP LEARNING APPROACHES" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 04, April 2025**

Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 04/70400243493

DOI: https://www.doi.org/10.56726/IRJMETS73923

Date: 28/04/2025

Certificate of Publication

This is to certify that author "**Sanshay Kumar Tiwary**" with paper ID "**IRJMETS70400243493**" has published a paper entitled "ONLINE RECRUITMENT FRAUD (ORF) DETECTION USING DEEP LEARNING APPROACHES" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 04, April 2025**

Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 04/70400243493

DOI: https://www.doi.org/10.56726/IRJMETS73923

Date: 28/04/2025

Certificate of Publication

This is to certify that author "**Raushan Kumar**" with paper ID "**IRJMETS70400243493**" has published a paper entitled "ONLINE RECRUITMENT FRAUD (ORF) DETECTION USING DEEP LEARNING APPROACHES" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 04, April 2025**

Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 04/70400243493

DOI: https://www.doi.org/10.56726/IRJMETS73923

Date: 28/04/2025

Certificate of Publication

This is to certify that author "**Sanni K Ranjan Kumar**" with paper ID "**IRJMETS70400243493**" has published a paper entitled "ONLINE RECRUITMENT FRAUD (ORF) DETECTION USING DEEP LEARNING APPROACHES" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 04, April 2025**

Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 04/70400243493

DOI: https://www.doi.org/10.56726/IRJMETS73923

Date: 28/04/2025

Certificate of Publication

This is to certify that author "**Suraj Kumar**" with paper ID "**IRJMETS70400243493**" has published a paper entitled "ONLINE RECRUITMENT FRAUD (ORF) DETECTION USING DEEP LEARNING APPROACHES" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 04, April 2025**

Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)
Re-Accredited with 'A' Grade by NAAC
Re-Accredited by NBA (BME, ECE & EEE), Accredited by NBA (CSE, IT, MECH & AERO)
Perambalur - 621 212, Tamil Nadu, India.

CERTIFICATE

This is to certify that ~~Dr.~~ / Mr. / ~~Ms.~~ SANSHAY KUMAR TIWARY of DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE has presented a Research Paper entitled ONLINE RECRUITMENT FRAUD DETECTION USING DEEP LEARNING APPROACHES in the ***International Conference on Smart Intelligent Computing and Applications (ICSICA) - 2025,*** organized by Department of CSE, IT, AI & DS, CST and MCA on ***25th April, 2025.***

ORGANIZING CHAIR **CHAIRPERSON**

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)
Re-Accredited with 'A' Grade by NAAC
Re-Accredited by NBA (BME, ECE & EEE), Accredited by NBA (CSE, IT, MECH & AERO)
Perambalur - 621 212, Tamil Nadu, India.

CERTIFICATE

This is to certify that ~~Dr.~~ / Mr. / ~~Ms.~~ RAUSHAN KUMAR of DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE has presented a Research Paper entitled ONLINE RECRUITMENT FRAUD DETECTION USING DEEP LEARNING APPROACHES in the *International Conference on Smart Intelligent Computing and Applications (ICSICA) - 2025,* organized by Department of CSE, IT, AI & DS, CST and MCA on *25th April, 2025.*

ORGANIZING CHAIR

CHAIRPERSON

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)
Re-Accredited with 'A' Grade by NAAC
Re-Accredited by NBA (BME, ECE & EEE), Accredited by NBA (CSE, IT, MECH & AERO)
Perambalur - 621 212, Tamil Nadu, India.

CERTIFICATE

This is to certify that Dr. / Mr. / Ms. SANNI KRANJAN KUMAR of DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE has presented a Research Paper entitled ONLINE RECRUITMENT FRAUD DETECTION USING DEEP LEARNING APPROACHES in the *International Conference on Smart Intelligent Computing and Applications (ICSICA) - 2025,* organized by Department of CSE, IT, AI & DS, CST and MCA on *25th April, 2025.*

ORGANIZING CHAIR **CHAIRPERSON**

DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE

(AUTONOMOUS)

(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)
Re-Accredited with 'A' Grade by NAAC
Re-Accredited by NBA (BME, ECE & EEE), Accredited by NBA (CSE, IT, MECH & AERO)
Perambalur - 621 212, Tamil Nadu, India.

CERTIFICATE

This is to certify that ~~Dr.~~ / Mr. / ~~Ms.~~ SURAJ KUMAR of DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE has presented a Research Paper entitled ONLINE RECRUITMENT FRAUD DETECTION USING DEEP LEARNING APPROACHES in the ***International Conference on Smart Intelligent Computing and Applications (ICSICA) - 2025,*** organized by Department of CSE, IT, AI & DS, CST and MCA on ***25th April, 2025.***

ORGANIZING CHAIR **CHAIRPERSON**