ONLINE RECRUITMENT FRAUD DETECTION USING DEEP LEARNING

A Major Project Report Submitted

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

Bachelor of Technology
in
Artificial Intelligence & Machine Learning
By

KANCHERLA VAMSI	22N35A7303
KURVA VIJAYKUMAR	22N35A7304
MEKALA SANTHOSH	22N35A7305

Under the Guidance of

Mrs. Ruquia Taha
Assisstant Professor

DEPARTMENT OF COMPUTATIONAL INTELLIGENCE

MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

(Affiliated to JNTU, Hyderabad, Approved by AICTE, Accredited by NBA & NAAC – 'A' Grade, ISO 9001:2015 Certified) Maisammaguda (v), Near Dullapally, Via: Kompally, Hyderabad – 500 100, Telangana State, India. website: www.mrcet.ac.in

2024-2025

DECLARATION

We hereby declare that the project entitled **"Online Recruitment Fraud Detection using Deep Learning"** submitted to Malla Reddy College of Engineering and Technology, affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) for the award of the degree of Bachelor of Technology in Artificial Intelligence & Machine Learning is a result of original research work done by us.

It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of a degree or diploma.

Kancherla Vamsi – 22N35A7303
Kurva Vijaykumar – 22N35A7304
Mekala Santhosh – 22N35A7305

MALLA REDDY COLLEGE
OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS INSTITUTION - UGC, GOVT. OF INDIA)
Affiliated to JNTUH; Approved by AICTE, NBA-Tier 1 & NAAC with A-GRADE | ISO 9001:2015

LIVE TO LEARN & LEARN TO SHARE

CERTIFICATE

This is to certify that this is the Bonafide record of the project titled **“Online Recruitment Fraud Detection using Deep Learning”** submitted by **Kancherla Vamsi** (22N35A7303),**Kurva Vijaykumar** (22N35A7304) and **Mekala Santhosh** (22N35A7305) of B. Tech in the partial fulfillment of the requirements for the degree of **Bachelor of Technology in Artificial Intelligence & Machine Learning, Dept. of CI during the year 2024–2025.** The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

Mrs.Ruquia Taha	**Dr. D. Sujatha**
Assisstant Professor	Professor and Dean(CSE&ET)
INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**

EXTERNAL EXAMINER

Date of Viva-Voce Examination held on: ________________________

ACKNOWLEDGEMENT

We feel honored and privileged to place our warm salutation to our college Malla Reddy College of Engineering and Technology (UGC-Autonomous), our Director ***Dr. VSK Reddy*** who gave us the opportunity to have experience in engineering and profound technical knowledge.

We are indebted to our Principal ***Dr. S. Srinivasa Rao*** for providing us with facilities to do our project and his constant encouragement and moral support which motivated us to move forward with the project.

We would like to express our gratitude to our Head of the Department ***Dr. D. Sujatha,*** Professor and Dean (CSE&ET) for encouraging us in every aspect of our system development and helping us realize our full potential.

We would like to express our sincere gratitude and indebtedness to our project supervisor **Mrs.Ruquia Taha,** Assisstant Professor for her valuable suggestions and interest throughout the course of this project.

We convey our heartfelt thanks to our Project Coordinator**,** ***Mrs. B. Jyothi,*** Assistant Professor for allowing for her regular guidance and constant encouragement during our dissertation work

We would also like to thank all supporting staff of department of CI and all other departments who have been helpful directly or indirectly in making our Major Project a success.

We would like to thank our parents and friends who have helped us with their valuable suggestions and support has been very helpful in various phases of the completion of the Major Project.

By:
Kancherla Vamsi – 22N35A7303
Kurva Vijaykumar – 22N35A7304
Mekala Santhosh - 22N35A7305

ABSTRACT

Online Recruitment Fraud (ORF) has become a significant concern as job seekers increasingly rely on digital platforms for employment opportunities. Fraudulent recruitment schemes can lead to identity theft, financial loss, and compromised personal data. Existing systems primarily rely on manual moderation and rule-based algorithms to detect fraudulent activities. However, these methods are often time-consuming and prone to errors.The proposed work leverages deep learning techniques to enhance ORF detection by analyzing various features from recruitment advertisements and user behavior. By employing Convolutional Neural Networks (CNNs) to classify images, such as screenshots of fraudulent job postings, the system can identify patterns that distinguish legitimate job offers from fraudulent ones. This automated approach ensures real-time monitoring, reducing the burden on human moderators and enhancing the security and trustworthiness of online recruitment platforms. Additionally, Natural Language Processing (NLP) techniques can be integrated to analyze text-based features alongside image data, providing a comprehensive solution to ORF detection.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Motivation

The motivation behind this project stems from the need to bridge the gap between traditional, manually moderated job platforms and the increasing sophistication of online recruitment scams. In a rapidly growing digital job market, user safety and trust are key to improving platform reliability and user confidence. By introducing an intelligent system that automatically detects fraudulent job postings, this project aims to create a more secure, trustworthy recruitment environment that protects job seekers, encourages platform participation, and builds a stronger connection between users and technology—ultimately leading to safer hiring practices and increased platform credibility.

1.2 Problem Statement

Despite the growth and convenience of online recruitment platforms, most lack the intelligent capabilities required to identify and prevent fraudulent job postings and scams. This gap exposes job seekers to potential risks, including identity theft, financial loss, and emotional distress, while also damaging the credibility of job portals. Traditional moderation techniques are often manual, time-consuming, and ineffective in handling the large volume of daily listings, especially those created by increasingly sophisticated fraudsters.

This project addresses the problem by developing a deep learning-based recruitment fraud detection system capable of analyzing both textual and visual content to identify suspicious job advertisements. The goal is to improve the security and trustworthiness of online hiring platforms by automatically flagging fake listings. By integrating fraud detection with key recruitment features such as ad review, candidate engagement, and user verification, the system aims to offer a reliable solution that enhances user protection and supports ethical digital recruitment practices.

1.3 Scope of the project

The scope of this project involves the development of a deep learning-based system to detect fraudulent online recruitment activities. The system is designed to analyze job advertisements and company profiles using both textual and visual features. By leveraging Convolutional Neural Networks (CNN) for image-based analysis and Natural Language Processing (NLP) for text classification, the model aims to accurately distinguish between genuine and fake job postings. The project covers data collection, preprocessing, model training, evaluation, and deployment in a real-time environment. It also considers scalability, allowing the system to adapt to a variety of job sectors and online platforms. Ultimately, this solution aims to enhance the security and reliability of online recruitment systems by automating fraud detection and minimizing human intervention.

1.4 Project Introduction:

In recent years, online recruitment platforms have transformed the hiring landscape by offering accessibility, reach, and streamlined application processes. However, these platforms are increasingly being exploited by malicious actors who create fake job listings and impersonate legitimate companies. This exploitation undermines the trust of job seekers, causes financial and emotional harm, and challenges the integrity of digital recruitment ecosystems. As fraudulent tactics become more sophisticated, there is a growing demand for intelligent systems capable of automatically identifying and preventing recruitment scams.

This project introduces a deep learning-based fraud detection system aimed at enhancing security and trust in online recruitment. The system analyzes both textual job descriptions and visual content such as logos or company branding using advanced NLP techniques and Convolutional Neural Networks (CNNs). It evaluates various indicators of fraud—such as unrealistic offers, vague details, or mismatched company data—and flags suspicious listings accordingly. By integrating this detection feature with essential recruitment functionalities like job listing review, user authentication, and analytics, the system aims to improve platform credibility, protect users, and promote safer, more reliable online hiring experiences.

1.5 Summary

In today’s digital age, online recruitment has become a central component of the hiring process, offering job seekers and employers unprecedented convenience, access, and efficiency. However, while the functionality of job portals has advanced, many still lack robust mechanisms to verify the authenticity of job postings and protect users from fraud. One of the most critical gaps is the inability to detect fake or misleading advertisements, which can result in job seekers falling victim to scams involving identity theft, financial exploitation, or false employment promises. This project is motivated by the growing need to enhance the safety and credibility of online recruitment platforms through intelligent, automated solutions

The proposed project introduces a deep learning-based system designed to detect and flag fraudulent job postings on online platforms. Unlike traditional systems that rely on manual moderation or basic keyword filters, this model uses a combination of Natural Language Processing (NLP) to analyze textual job descriptions and Convolutional Neural Networks (CNN) to evaluate visual elements such as logos and company images. The system assesses the legitimacy of postings based on patterns learned from real and fake job data, generating warnings or classifications to alert users. By simulating a smart, automated review process, this solution aims to make online job seeking more secure, transparent, and user-focused, ultimately building trust and reducing fraud in digital hiring environments.

The scope of this project includes developing an intelligent web-based system that automatically detects fraudulent job postings by analyzing both text and image content. The system guides users—whether job seekers or administrators—through an intuitive interface that enables uploading or browsing job ads, analyzing their authenticity, and reviewing prediction results. Unlike traditional rule-based detection systems, this project integrates deep learning models such as **Convolutional Neural Networks (CNN)** and **Natural Language Processing (NLP)** to simulate human-like assessment with greater accuracy and efficiency. It is designed as a proof-of-concept but scalable for large datasets and adaptable to real-time applications across various recruitment platforms.

Furthermore, the system strikes a balance between automation and usability. It ensures that while the process is heavily data-driven, users can interact with clear results and decision indicators, such as fraud probability scores or red-flag markers. The model is trained on a curated dataset containing both genuine and fraudulent postings, enabling it to learn common patterns, anomalies, and deceptive traits. As the system evolves, it can be extended to include multilingual support, user feedback integration, and continuous learning mechanisms to stay ahead of emerging scam techniques.

In summary, the motivation behind this project is to enhance the integrity of online job markets by combining deep learning with practical fraud detection strategies. By integrating intelligent classification into a recruitment platform, the system aims to reduce user vulnerability, build trust in digital hiring, and contribute toward a more secure online employment ecosystem. The proposed solution is both practical and forward-looking, addressing a real-world problem with scalable, AI-driven innovation.

CHAPTER 2
LITERATURE SURVEY

2.1 Existing System

The existing systems for detecting online recruitment fraud are limited in functionality and largely rely on manual reporting and basic rule-based filters. Most job portals depend on user feedback or moderation teams to identify and remove fake job postings. These systems typically analyze job postings based on a few predefined parameters such as missing company information, unprofessional language, or unrealistic offers.While some platforms may employ basic keyword-based filters or blacklists, these approaches are not scalable and are prone to errors. They often fail to detect cleverly crafted fraudulent ads that appear legitimate. Additionally, current systems lack the ability to analyze the visual content of job ads, such as logos or images, which are often manipulated to mimic real companies.

Disadvantages of existing system

- **Manual Effort Required:** Most existing systems rely heavily on manual moderation and user reports, making them inefficient and slow to respond to fraud.
- **Limited Detection Capability:** Rule-based filters are not capable of detecting intelligently crafted scams that appear genuine.
- **No Visual Content Analysis:** The use of static pricing reduces the potential for personalized deals and can negatively impact customer satisfaction.
- **Lack of Real-Time Detection:** Delays in identifying fraud mean fake job ads often stay online long enough to harm job seekers.
- **No Learning Mechanism:** There is no use of machine learning or deep learning in many current systems, which limits their ability to improve over time.

2.2 Proposed System

The proposed system is developed to tackle the increasing issue of fraudulent job postings on online recruitment platforms by using advanced deep learning technologies. Unlike traditional methods that rely on manual verification or basic keyword filtering, this system adopts a hybrid deep learning approach combining Convolutional Neural Networks (CNN) and Natural Language Processing (NLP). CNNs are employed to analyze visual elements such as logos, banners, and document layouts, which may carry signs of manipulation. At the same time, NLP techniques help in processing the textual content of job advertisements, examining aspects like sentence structure, grammar, suspicious keywords, and unrealistic job descriptions.

By training the model on a large dataset containing both genuine and fraudulent job postings, the system learns to identify hidden patterns and inconsistencies commonly associated with scams. It evaluates various elements such as grammatical errors, mismatched roles and responsibilities, fake branding, and unprofessional language. This intelligent classification allows the system to accurately flag fraudulent ads in real-time. The solution is scalable, user-friendly, and continuously improving through retraining on new data, making it a reliable tool for securing digital recruitment platforms and protecting job seekers from online scams.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 Introduction

System requirements define the necessary hardware, software, and functional components needed to develop and run the proposed online recruitment fraud detection system effectively. These requirements serve as the foundation for successful implementation, ensuring that the system performs efficiently and meets user expectations. System requirements are typically categorized into two types: hardware requirements, which specify the physical components needed such as processor, memory, and storage; and software requirements, which outline the operating systems, programming languages, frameworks, and other tools required to build and deploy the application. Clearly identifying and documenting these requirements is crucial for the smooth development, deployment, and maintenance of the system.

3.2 Software and Hardware Requirement

3.2.1 Hardware Requirements

Software Requirements :

Component	**Specification**
Operating System	• Windows 7/8/10.
Front-end tool	• HTML,CSS, JS-JavaScript
Back-end tool	• Flask
IDE/Workbench	• Visual Studio Code.

Hardware Requirements:

Component	**Minimum Requirement**	**Recommended**
Processor (CPU)	Intel Core i3 or equivalent	Intel Core i5/i7 or higher
RAM	8 GB	8 GB or more
Hard Disk	160 GB (free space)	256 GB SSD
Graphics (Optional)	Integrated GPU	Dedicated GPU (for large datasets)
Network	Required for distributed setup	Stable LAN/Wi-Fi connection

3.3 Functional and Non-Functional Requirements

Functional Requirements :

These are the core functions that the system **must perform**:

- **User Authentication:** The system must allow users (e.g., job seekers, admins) to register, log in, and manage their profiles securely.
- **Job Advertisement Submission:** The system should allow users to upload or input job listings with relevant details such as title, company, description, and logo.
- **Fraud Detection Module:** Upon submission, the system must automatically initiate the fraud detection engine to analyze the job advertisement.
- **Text and Image Analysis:** The system should use NLP to process textual content and CNNs to analyze visual content like company logos or job images.
- **Fraud Classification Output:** The system must generate a result that classifies the job post as “Genuine” or “Fraudulent,” along with a confidence score.
- **Result Feedback:** The system should provide a clear summary of the fraud evaluation with possible reasons or flags raised by the model.

Non-Functional Requirements:

These describe how the system should behave and perform:

- **Performance:** The system should analyze and respond to each job posting within 2–3 seconds to ensure smooth real-time interaction.
- **Scalability:** The system must support an increasing volume of job posts and users, with minimal impact on speed or accuracy.
- **Usability:** The interface must be user-friendly and intuitive for both technical and non-technical users.
- **Security:** The system must ensure secure user authentication, encrypt sensitive data, and prevent unauthorized access to job listings and results.
- **Availability:** The platform should ensure 99.9% uptime, making it reliably accessible to users and administrators.
- **Maintainability:** The codebase should be modular, well-documented, and easy to update for bug fixes or enhancements.
- **Compatibility:** The system should function across various browsers and devices (e.g., Chrome, Firefox, Android, iOS).
- **Reliability:** The detection engine should function correctly under expected loads and recover gracefully from errors or interruptions.

3.4 Summary

The system requirements for the proposed **Online Recruitment Fraud Detection System** provide a comprehensive framework that ensures the application functions effectively, addresses user needs, and supports the project’s core objective of improving online recruitment safety. These requirements include essential hardware and software elements such as a connected device, web browser, hosting server, and a backend capable of handling image and text processing. By identifying these core requirements early, the project ensures a smooth deployment and a reliable user experience.

The **functional requirements** define the critical operations of the system, such as user authentication, job advertisement analysis, fraud prediction, and result reporting. These capabilities ensure that users can interact with the system intuitively and receive timely, actionable outputs. In contrast, the **non-functional requirements** focus on vital quality attributes like performance, security, reliability, and scalability—ensuring that the system remains responsive and dependable as usage grows.

By clearly articulating both types of requirements, the development team can maintain focus, align stakeholder expectations, and build a robust, future-ready solution. Functional requirements enable the system to deliver the core fraud detection features, while non-functional requirements ensure that these features perform consistently and securely. Together, these requirements provide a solid foundation for developing an intelligent, trustworthy, and user-centric fraud detection system that improves the integrity of online recruitment.

CHAPTER 4
SYSTEM DESIGN

4.1 INTRODUCTION

System design is a critical phase in the development process that translates the system requirements into a blueprint for building the software solution. It involves defining the architecture, components, modules, interfaces, and data flow of the proposed system. In the context of this chatbot-based price negotiation system, system design outlines how different elements such as the user interface, chatbot logic, product database, and pricing algorithms interact to deliver a seamless and intelligent negotiation experience. The goal of system design is to ensure that the system is scalable, maintainable, and efficient while meeting all specified functional and non-functional requirements. Through both high- level architectural design and detailed module design, this phase provides a structured framework that guides the actual implementation of the system.

System design outlines the architecture, components, and interaction patterns of the proposed system. It begins with selecting an architecture style, such as client-server, that supports scalability and performance needs. The design identifies key modules, like the chatbot engine, product database, and UI, ensuring smooth interaction between them. A major focus is on the user interface, ensuring it is intuitive and engaging. The price negotiation algorithm processes user offers, evaluates them, and generates responses. Data flow and database design ensure efficient data storage and retrieval. Security measures like encryption protect sensitive data, while scalability and maintainability are considered for future growth and system updates. Finally, the design incorporates error handling and testing to ensure reliability and smooth operation.

4.2 Architecture Diagram

An architecture diagram is a visual representation of a system's structure, showcasing how its components are interconnected, how they communicate, and how data flows through the system, serving as a blueprint for understanding and communicating complex designs.

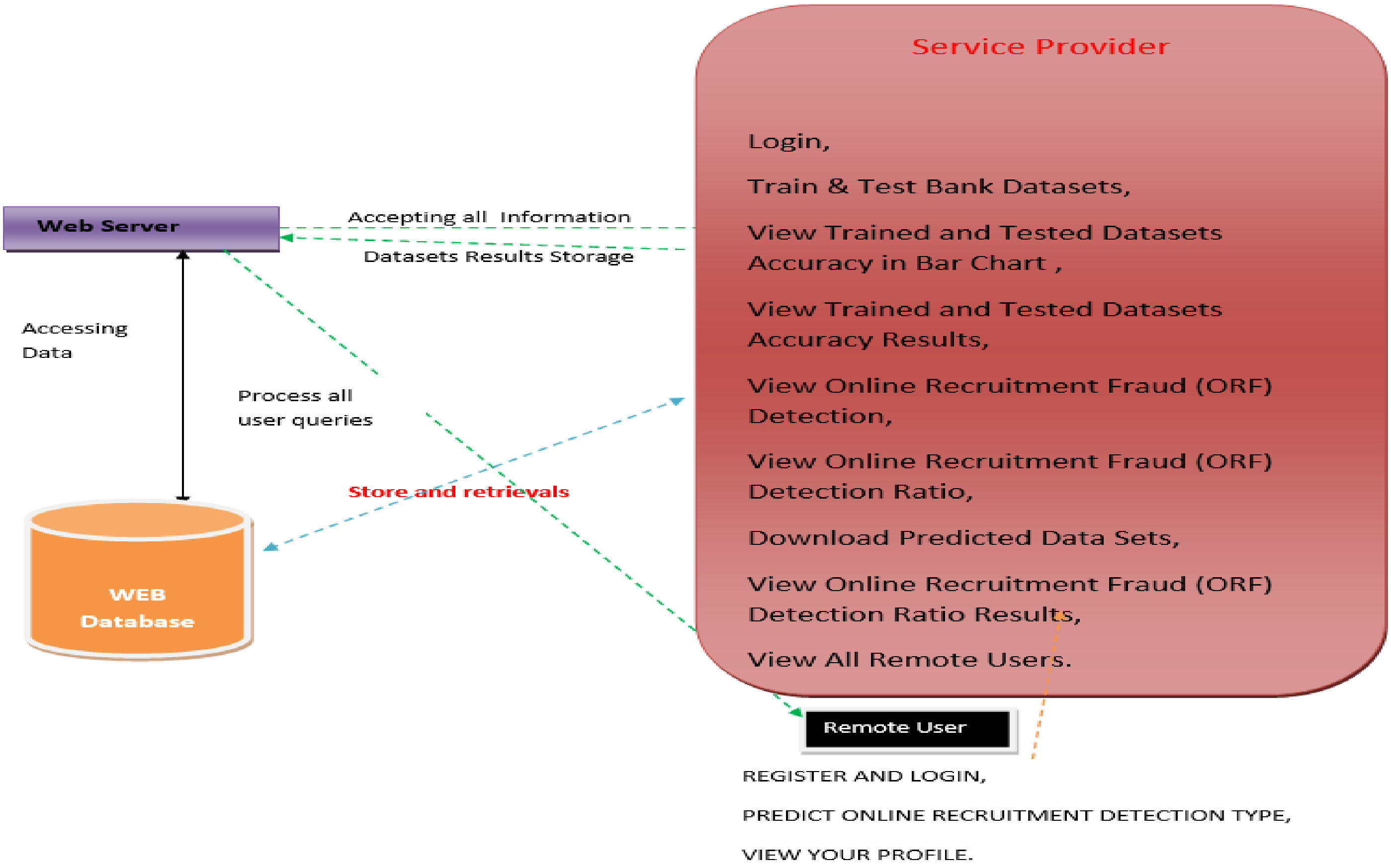

Service Provider
Login,
Train & Test Bank Datasets,
View Trained and Tested Datasets Accuracy in Bar Chart ,
View Trained and Tested Datasets Accuracy Results,
View Online Recruitment Fraud (ORF) Detection,
View Online Recruitment Fraud (ORF) Detection Ratio,
Download Predicted Data Sets,
View Online Recruitment Fraud (ORF) Detection Ratio Results,
View All Remote Users.
Web Server
Accepting all Information
Datasets Results Storage
Accessing Data
Process all user queries
Store and retrievals
WEB Database
Remote User
REGISTER AND LOGIN,
PREDICT ONLINE RECRUITMENT DETECTION TYPE,
VIEW YOUR PROFILE.

Figure 4.2.1: Architecture Diagram of Online Recruitment Fraud Detection using Deep Learning

4.3 UML DIAGRAMS

4.3.1 Class Diagram

A class diagram is a visual representation of the classes in a system and their relationships. It shows attributes, methods, and associations between classes, helping to outline the system's structure. Class diagrams are essential in object-oriented design, facilitating communication and understanding among developers and stakeholders.

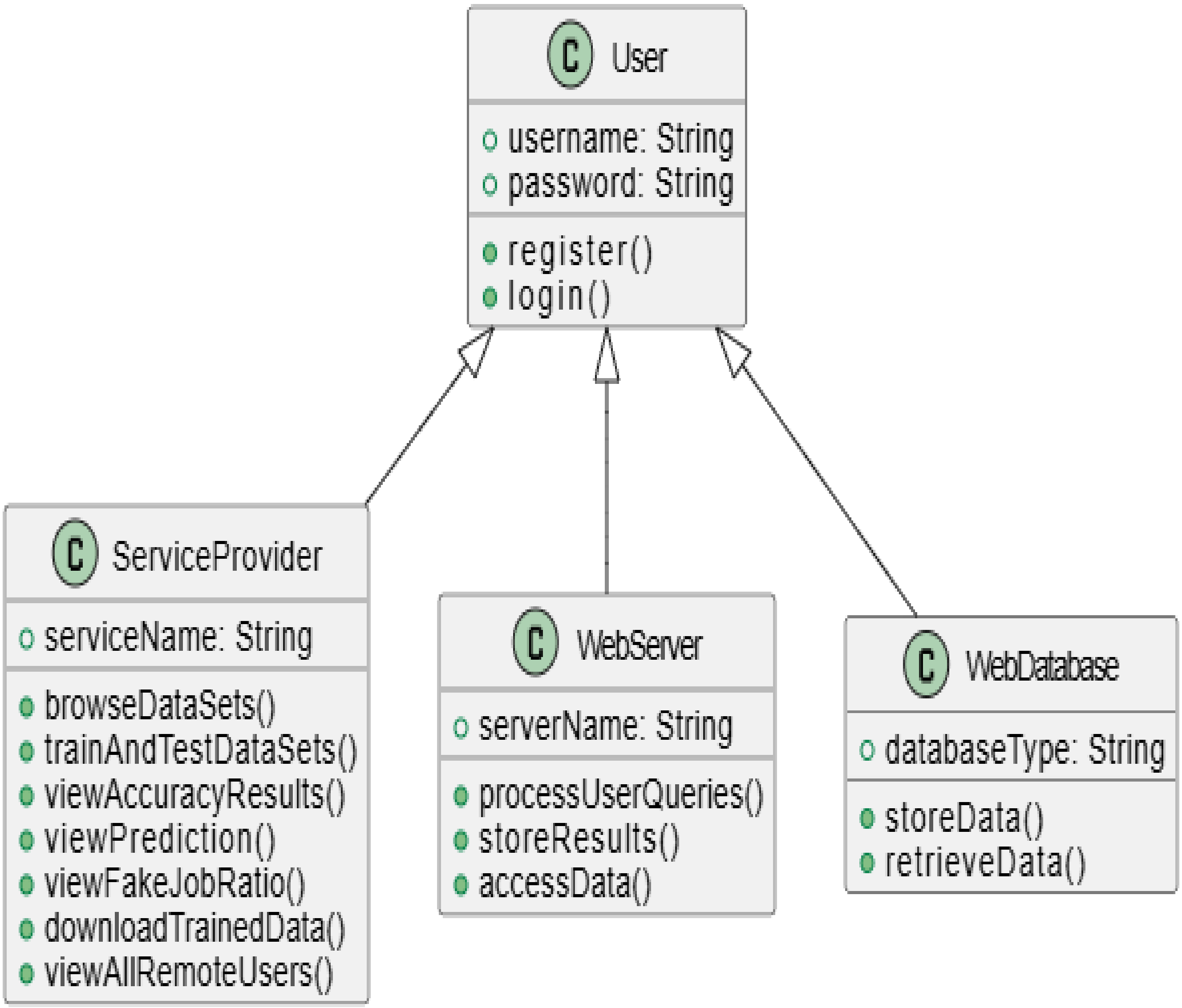


Figure 4.3.1: Class Diagram of Online Recruitment Fraud Detection using Deep Learning

4.3.2 Sequence Diagram

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

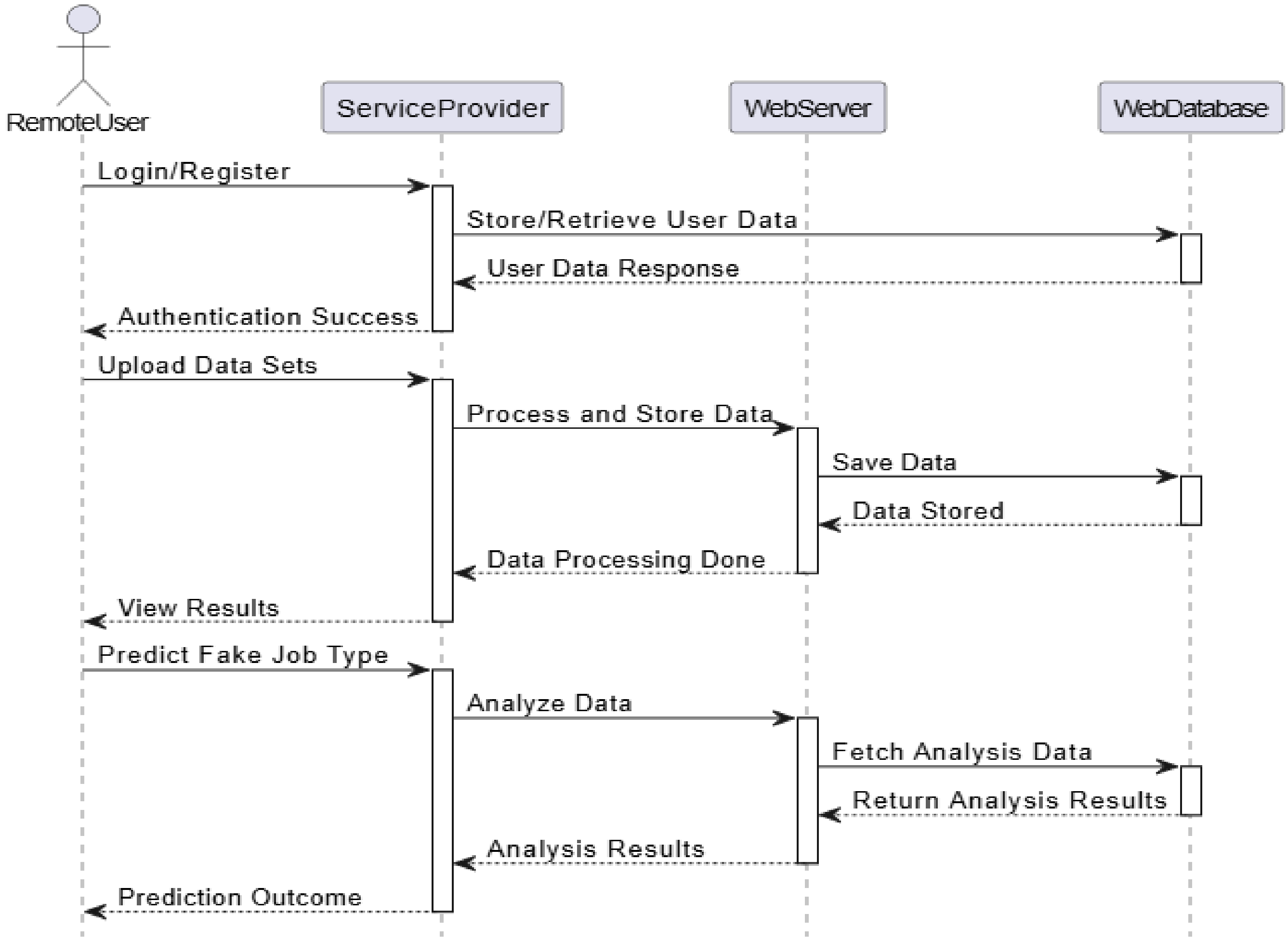


Figure 4.3.2: Sequence Diagram of Online Recruitment Fraud Detection using Deep Learning

4.3.3 Use Case

Use Case during requirement elicitation and analysis to represent the functionality of the system. Use case describes a function by the system that yields a visible result for an actor. The identification of actors and use cases result in the definitions of the boundary of the system i.e., differentiating the tasks accomplished by the system and the tasks accomplished by its environment.

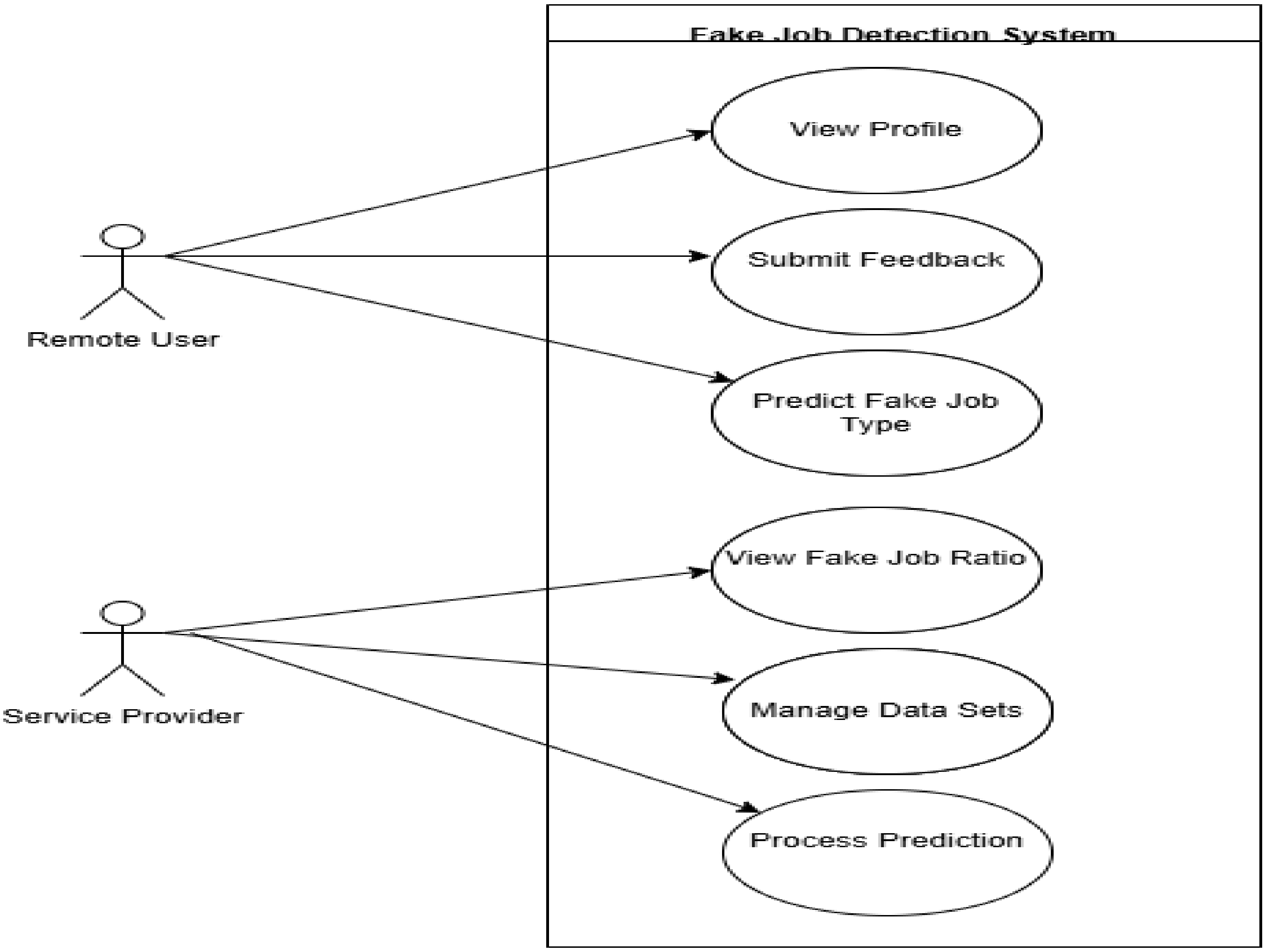


Figure 4.3.3: Use Case diagram of Online Recruitment Fraud Detection using Deep Learning

4.3.4 Activity Diagram

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

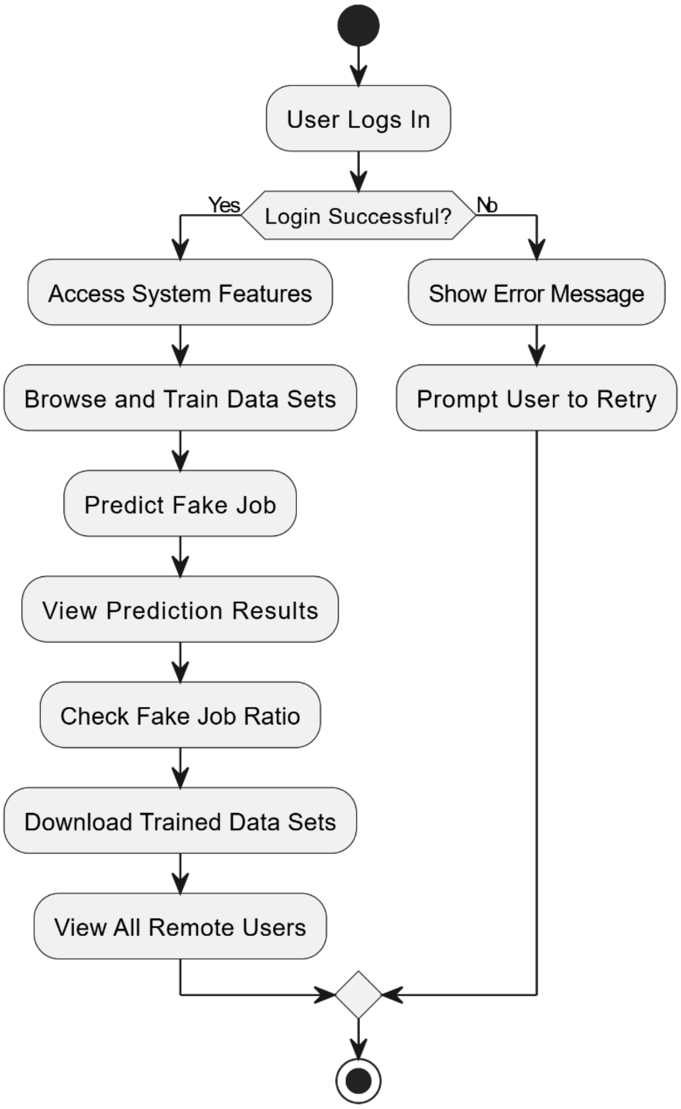


Figure 4.3.4: Activity Diagram of Online Recruitment Fraud Detection using Deep Learning

4.4 Data Flow Diagram

Data Flow Diagrams (DFDs) are graphical representations used to model the flow of data within a system. They provide a visual overview of how data moves through different processes, stores, and external entities within the system.

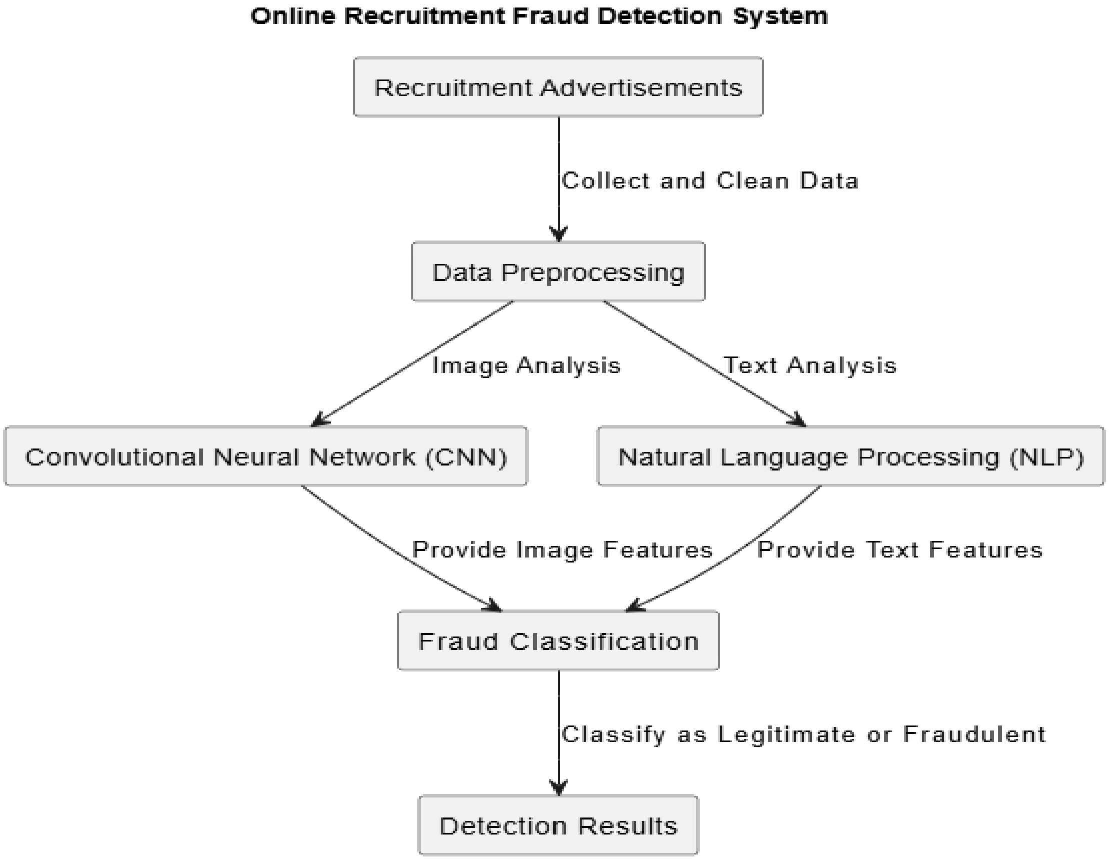

Online Recruitment Fraud Detection System
Recruitment Advertisements
Collect and Clean Data
Data Preprocessing
Image Analysis
Text Analysis
Convolutional Neural Network (CNN)
Natural Language Processing (NLP)
Provide Image Features
Provide Text Features
Fraud Classification
Classify as Legitimate or Fraudulent
Detection Results

Fig 4.4 Data Flow Diagram of Online Recruitment Fraud Detection using Deep Learning

4.5 Summary

System design serves as a blueprint for building a robust, scalable, and intelligent recruitment fraud detection system. It outlines the system's architecture, components, and interactions necessary to meet both user and business objectives. The system is structured to follow a modular, client-server architecture that enables secure, fast, and accurate analysis of job advertisements. Each component is designed to handle specific tasks such as user interaction, job post submission, image and text analysis, and fraud classification. Emphasis is placed on designing an intuitive interface for users and administrators while ensuring that security, error handling, and scalability are built into the core architecture. Provisions for data integrity, model updates, and ongoing system maintenance are also incorporated to ensure long-term functionality and reliability.

Objective:

The objective of the proposed system is to improve the safety and credibility of online job platforms by detecting and flagging fraudulent job postings using deep learning techniques. By analyzing both textual and visual content from job ads, the system aims to automatically identify suspicious listings and alert users. This will enhance user trust, reduce the risk of scams, and contribute to a more secure online recruitment environment.

Approach:

The development of the recruitment fraud detection system involves several stages. First, a dataset of real and fake job advertisements will be collected and labeled. The system will use **Natural Language Processing (NLP)** to process and analyze job descriptions and **Convolutional Neural Networks (CNN)** for image/logo analysis. Users can upload or input job ads, which are then evaluated by the model to predict whether the job is genuine or fraudulent. Based on the prediction, the system will provide feedback in a clear, user-friendly format. The architecture will be modular, allowing for future enhancements such as multilingual support and real-time scanning. Security protocols will protect user data, and performance testing will ensure that results are delivered efficiently and accurately.

System Modules:

- **Job Post Management Module:** This module allows users to submit job advertisements by entering details such as job title, company name, description, and uploading logos or related images. It ensures all data is collected for processing and can be accessed later for analysis.
- **Fraud Detection Engine:**This is the core module responsible for classifying job posts as genuine or fraudulent. It combines NLP for text analysis and CNN for image detection. It uses trained deep learning models to evaluate and score job postings based on learned patterns and indicators of fraud.
- **User Account Management Module:**Handles user registration, login, and profile management. This module stores user activity history, such as previously analyzed job posts, flagged results, and administrative actions.
- **Admin Dashboard Module:** Allows administrators to manage the dataset, monitor system performance, retrain the model with new data, and view flagged job posts. It plays a crucial role in maintaining the accuracy and adaptability of the detection system.
- **Report and Feedback Module:** After analysis, the system generates a detailed report for each job posting, providing the user with classification results, fraud probability, and possible indicators. It also allows users to give feedback on the prediction accuracy, supporting continuous model improvement.

CHAPTER 5
IMPLEMENTATION

5.1 Algorithms:

1. Fraud Detection Algorithm:

Input: Job posting content, including job title, description, company name, and uploaded images such as company logos.

Process: The system uses a deep learning model trained on labeled datasets of genuine and fraudulent job postings.

- The **textual data** is processed using Natural Language Processing (NLP) techniques to extract linguistic and semantic patterns.
- The **visual data** (images) are analyzed using Convolutional Neural Networks (CNN) to detect anomalies or fake branding.
- Features are combined and passed through a classification model.

Output: A prediction label (e.g., "Genuine" or "Fraudulent") along with a confidence score.

2. Input Validation Algorithm:

Input: User-submitted job posting data (text fields, image uploads).

Process: Validates inputs to ensure they are complete and correctly formatted. Text fields like job title and description must not be empty or suspiciously short. Image files must be in accepted formats (e.g., PNG, JPG) and below a size threshold. Email addresses and URLs are validated for structure and domain checks.

Output: Cleaned and validated data, or error messages prompting correction.
Architectural Components

5.2 Architectural Components

The proposed chatbot-based price negotiation system is composed of several key architectural components that ensure efficient communication, data processing, and user interaction.

- **User Interface (UI):**

  A web-based front end that enables users to upload job posts, view results, and navigate the system. Technologies like HTML, CSS, and JavaScript are used for ease of use.
- **Prediction Engine::**

  The core component where deep learning models (CNN for images and NLP models for text) are executed to classify job listings.
- **Preprocessing Layer:**

  Handles tasks like cleaning textual input, resizing/normalizing images, and converting data into formats suitable for the model.
- **Database:**

  Stores all the necessary data including product details, pricing information, user credentials, and negotiation logs. It ensures quick retrieval and storage for seamless interactions.
- **Admin Dashboard:**

  Provides system administrators with tools to manage datasets, retrain models, view usage statistics, and oversee flagged listings.

5.3 Technologies Used

1. Frontend Technologies:

- **HTML5:** For structuring the web pages and content.
- **CSS3:** For styling the chatbot interface and overall layout.
- **JavaScript:** For adding interactivity to the user interface, handling user input, and chatbot communication.

2. Backend Technologies:

- **Python :** For implementing the machine learning models and backend logic.
- **Flask:** As the web framework to serve the UI and manage routes.
- **MySQL :** For storing product data, user profiles, chat history, and negotiation rules.

3. Web Hosting / Server:

- **WAMP :** Local development environment for PHP and MySQL.

4. Development Tools:

- **VS Code :** Code editors used during development.
- **Git & GitHub:** Version control system for collaborative development.

5.4 Source Code

```
from django.db.models import Count
from django.db.models import Q
from django.shortcuts import render, redirect, get_object_or_404

import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import VotingClassifier
# Create your views here.
from Remote_User.models import
ClientRegister_Model,online_recruitment_fraud_detection,detection_ratio,detection_accuracy

def login(request):

    if request.method == "POST" and 'submit1' in request.POST:

        username = request.POST.get('username')
        password = request.POST.get('password')
        try:
            enter = ClientRegister_Model.objects.get(username=username,password=password)
            request.session["userid"] = enter.id

            return redirect('ViewYourProfile')
        except:
            pass

    return render(request,'RUser/login.html')

def index(request):
    return render(request, 'RUser/index.html')

def Add_DataSet_Details(request):

    return render(request, 'RUser/Add_DataSet_Details.html', {"excel_data": ''})

def Register1(request):

    if request.method == "POST":
        username = request.POST.get('username')
        email = request.POST.get('email')
        password = request.POST.get('password')
        phoneno = request.POST.get('phoneno')
        country = request.POST.get('country')
        state = request.POST.get('state')
        city = request.POST.get('city')
        address = request.POST.get('address')
        gender = request.POST.get('gender')
        ClientRegister_Model.objects.create(username=username, email=email, password=password,
phoneno=phoneno,
                                            country=country, state=state, city=city,address=address,gender=gender)

        obj = "Registered Successfully"
        return render(request, 'RUser/Register1.html',{'object':obj})
    else:
        return render(request,'RUser/Register1.html')

def ViewYourProfile(request):
    userid = request.session['userid']
    obj = ClientRegister_Model.objects.get(id= userid)
    return render(request,'RUser/ViewYourProfile.html',{'object':obj})

def Predict_Online_Recruitment_Fraud_Detection(request):
    if request.method == "POST":

        if request.method == "POST":

            Fid= request.POST.get('Fid')
            jobpost= request.POST.get('jobpost')
            Title= request.POST.get('Title')
            Company= request.POST.get('Company')
            AnnouncementCode= request.POST.get('AnnouncementCode')
            Term= request.POST.get('Term')
            Eligibility= request.POST.get('Eligibility')
            Duration= request.POST.get('Duration')
            Location= request.POST.get('Location')
            JobDescription= request.POST.get('JobDescription')
            JobRequirment= request.POST.get('JobRequirment')
            RequiredQual= request.POST.get('RequiredQual')
            Salary= request.POST.get('Salary')
            ApplicationP= request.POST.get('ApplicationP')
            OpeningDate= request.POST.get('OpeningDate')
            Deadline= request.POST.get('Deadline')
            AboutC= request.POST.get('AboutC')
            IT= request.POST.get('IT')

        df = pd.read_csv('Datasets.csv')

        def apply_response(label):
```

```
    if (label == 0):
        return 0  # No Fraud Found
    elif (label == 1):
        return 1  # Fraud Found

df['results'] = df['Label'].apply(apply_response)

cv = CountVectorizer()
X = df['Fid'].apply(str)
y = df['results']

print("Fid")
print(X)
print("Results")
print(y)

X = cv.fit_transform(X)

models = []
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
X_train.shape, X_test.shape, y_train.shape

print("Recurrent Neural Network-RNN")
from sklearn.neural_network import MLPClassifier
mlpc = MLPClassifier().fit(X_train, y_train)
y_pred = mlpc.predict(X_test)
testscore_mlpc = accuracy_score(y_test, y_pred)
accuracy_score(y_test, y_pred)
print(accuracy_score(y_test, y_pred))
print(accuracy_score(y_test, y_pred) * 100)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, y_pred))
print("CONFUSION MATRIX")
print(confusion_matrix(y_test, y_pred))
models.append(('MLPClassifier', mlpc))

print("Logistic Regression")

from sklearn.linear_model import LogisticRegression
reg = LogisticRegression(random_state=0, solver='lbfgs').fit(X_train, y_train)
y_pred = reg.predict(X_test)
print("ACCURACY")
print(accuracy_score(y_test, y_pred) * 100)
print("CLASSIFICATION REPORT")
print(classification_report(y_test, y_pred))
print("CONFUSION MATRIX")
```

```
    print(confusion_matrix(y_test, y_pred))
    models.append(('logistic', reg))

    print("LightGBM")

    from sklearn.ensemble import GradientBoostingClassifier
    clf = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0, max_depth=1,
random_state=0).fit(
      X_train,
      y_train)
    clfpredict = clf.predict(X_test)
    print("ACCURACY")
    print(accuracy_score(y_test, clfpredict) * 100)
    print("CLASSIFICATION REPORT")
    print(classification_report(y_test, clfpredict))
    print("CONFUSION MATRIX")
    print(confusion_matrix(y_test, clfpredict))
    models.append(('GradientBoostingClassifier', clf))

    classifier = VotingClassifier(models)
    classifier.fit(X_train, y_train)
    y_pred = classifier.predict(X_test)

    Fid1 = [Fid]
    vector1 = cv.transform(Fid1).toarray()
    predict_text = classifier.predict(vector1)

    pred = str(predict_text).replace("[", "")
    pred1 = pred.replace("]", "")

    prediction = int(pred1)

    if (prediction == 0):
      val = 'Fraud Not Found'
    elif (prediction == 1):
      val = 'Fraud Found'

    print(val)
    print(pred1)

    online_recruitment_fraud_detection.objects.create(
    Fid=Fid,
    jobpost=jobpost,
    Title=Title,
    Company=Company,
    AnnouncementCode=AnnouncementCode,
    Term=Term,
    Eligibility=Eligibility,
    Duration=Duration,
    Location=Location,
    JobDescription=JobDescription,
    JobRequirment=JobRequirment,
    RequiredQual=RequiredQual,
    Salary=Salary,
    ApplicationP=ApplicationP,
    OpeningDate=OpeningDate,
    Deadline=Deadline,
    AboutC=AboutC,
    IT=IT,
    Prediction=val)

    return render(request, 'RUser/Predict_Online_Recruitment_Fraud_Detection.html',{'objs': val})
return render(request, 'RUser/Predict_Online_Recruitment_Fraud_Detection.html')
```

```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="ChangeListManager">
    <list default="true" id="dcb35d66-06d3-4380-ae09-d12ca7113204" name="Default" comment="" />
    <option name="EXCLUDED_CONVERTED_TO_IGNORED" value="true" />
    <option name="TRACKING_ENABLED" value="true" />
    <option name="SHOW_DIALOG" value="false" />
    <option name="HIGHLIGHT_CONFLICTS" value="true" />
    <option name="HIGHLIGHT_NON_ACTIVE_CHANGELIST" value="false" />
    <option name="LAST_RESOLUTION" value="IGNORE" />
  </component>
  <component name="FileEditorManager">
    <leaf SIDE_TABS_SIZE_LIMIT_KEY="300">
      <file leaf-file-name="urls.py" pinned="false" current-in-tab="false">
        <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/online_recruitment_fraud/urls.py">
          <provider selected="true" editor-type-id="text-editor">
            <state relative-caret-position="197">
              <caret line="35" column="217" lean-forward="false" selection-start-line="35" selection-start-
column="217" selection-end-line="35" selection-end-column="217" />
              <folding>
                <element signature="e#648#680#0" expanded="true" />
              </folding>
            </state>
          </provider>
        </entry>
      </file>
      <file leaf-file-name="Header.html" pinned="false" current-in-tab="false">
        <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Header.html">
```

```
          <provider selected="true" editor-type-id="text-editor">
            <state relative-caret-position="289">
              <caret line="53" column="110" lean-forward="false" selection-start-line="53" selection-start-
column="110" selection-end-line="53" selection-end-column="110" />
              <folding />
            </state>
          </provider>
        </entry>
      </file>
      <file leaf-file-name="View_Prediction_Of_Online_Recruitment_Fraud_Detection.html" pinned="false"
current-in-tab="false">
        <entry
file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/View_Prediction_Of_Onli
ne_Recruitment_Fraud_Detection.html">
          <provider selected="true" editor-type-id="text-editor">
            <state relative-caret-position="911">
              <caret line="102" column="0" lean-forward="false" selection-start-line="102" selection-start-
column="0" selection-end-line="102" selection-end-column="0" />
              <folding />
            </state>
          </provider>
        </entry>
      </file>
      <file leaf-file-name="View_Prediction_Of_Online_Recruitment_Fraud_Detection_Ratio.html"
pinned="false" current-in-tab="false">
        <entry
file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/View_Prediction_Of_Onli
ne_Recruitment_Fraud_Detection_Ratio.html">
          <provider selected="true" editor-type-id="text-editor">
            <state relative-caret-position="221">
              <caret line="55" column="52" lean-forward="false" selection-start-line="55" selection-start-
column="52" selection-end-line="55" selection-end-column="52" />
              <folding />
            </state>
          </provider>
        </entry>
      </file>
      <file leaf-file-name="Predict_Online_Recruitment_Fraud_Detection.html" pinned="false" current-in-
tab="false">
        <entry
file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Predict_Online_Recruitment_
Fraud_Detection.html">
          <provider selected="true" editor-type-id="text-editor">
            <state relative-caret-position="187">
              <caret line="53" column="115" lean-forward="false" selection-start-line="53" selection-start-
column="87" selection-end-line="53" selection-end-column="115" />
              <folding />
          </state>
        </provider>
      </entry>
    </file>
    <file leaf-file-name="index.html" pinned="false" current-in-tab="false">
      <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/index.html">
        <provider selected="true" editor-type-id="text-editor">
          <state relative-caret-position="1207">
            <caret line="71" column="70" lean-forward="false" selection-start-line="71" selection-start-column="70" selection-end-line="71" selection-end-column="70" />
            <folding />
          </state>
        </provider>
      </entry>
    </file>
    <file leaf-file-name="Header.html" pinned="false" current-in-tab="false">
      <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/Header.html">
        <provider selected="true" editor-type-id="text-editor">
          <state relative-caret-position="187">
            <caret line="54" column="85" lean-forward="false" selection-start-line="54" selection-start-column="85" selection-end-line="54" selection-end-column="85" />
            <folding />
          </state>
        </provider>
      </entry>
    </file>
    <file leaf-file-name="views.py" pinned="false" current-in-tab="false">
      <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Remote_User/views.py">
        <provider selected="true" editor-type-id="text-editor">
          <state relative-caret-position="-391">
            <caret line="124" column="52" lean-forward="false" selection-start-line="124" selection-start-column="52" selection-end-line="124" selection-end-column="52" />
            <folding>
              <element signature="e#0#34#0" expanded="true" />
            </folding>
          </state>
        </provider>
      </entry>
    </file>
    <file leaf-file-name="models.py" pinned="false" current-in-tab="true">
      <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Remote_User/models.py">
        <provider selected="true" editor-type-id="text-editor">
          <state relative-caret-position="204">
            <caret line="12" column="43" lean-forward="false" selection-start-line="12" selection-start-column="43" selection-end-line="12" selection-end-column="43" />
            <folding>
              <element signature="e#0#28#0" expanded="true" />
              </folding>
            </state>
          </provider>
        </entry>
      </file>
      <file leaf-file-name="views.py" pinned="false" current-in-tab="false">
        <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Service_Provider/views.py">
          <provider selected="true" editor-type-id="text-editor">
            <state relative-caret-position="279">
              <caret line="155" column="48" lean-forward="false" selection-start-line="155" selection-start-column="48" selection-end-line="155" selection-end-column="48" />
              <folding>
                <element signature="e#1#41#0" expanded="true" />
              </folding>
            </state>
          </provider>
        </entry>
      </file>
    </leaf>
  </component>
  <component name="IdeDocumentHistory">
    <option name="CHANGED_PATHS">
      <list>
        <option value="$PROJECT_DIR$/online_recruitment_fraud/online_recruitment_fraud/asgi.py" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/online_recruitment_fraud/settings.py" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/online_recruitment_fraud/wsgi.py" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Remote_User/models.py" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/serviceproviderlogin.html" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Register1.html" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/login.html" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/index.html" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/online_recruitment_fraud/urls.py" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Header.html" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/View_Prediction_Of_Online_Recruitment_Fraud_Detection_Ratio.html" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/View_Prediction_Of_Online_Recruitment_Fraud_Detection.html" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/Header.html" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Service_Provider/views.py" />
        <option value="$PROJECT_DIR$/online_recruitment_fraud/Remote_User/views.py" />
      </list>
    </option>
  </component>
  <component name="ProjectFrameBounds" extendedState="6">
```

```
    <option name="x" value="10" />
    <option name="y" value="20" />
    <option name="width" value="1346" />
    <option name="height" value="688" />
  </component>
  <component name="ProjectView">
    <navigator currentView="ProjectPane" proportions="" version="1">
      <flattenPackages />
      <showMembers />
      <showModules />
      <showLibraryContents />
      <hideEmptyPackages />
      <abbreviatePackageNames />
      <autoscrollToSource />
      <autoscrollFromSource />
      <sortByType />
      <manualOrder />
      <foldersAlwaysOnTop value="true" />
    </navigator>
    <panes>
      <pane id="ProjectPane">
        <subPane>
          <expand>
            <path>
              <item name="Online_Recruitment_Fraud" type="b2602c69:ProjectViewProjectNode" />
              <item name="Online_Recruitment_Fraud" type="462c0819:PsiDirectoryNode" />
            </path>
            <path>
              <item name="Online_Recruitment_Fraud" type="b2602c69:ProjectViewProjectNode" />
              <item name="Online_Recruitment_Fraud" type="462c0819:PsiDirectoryNode" />
              <item name="online_recruitment_fraud" type="462c0819:PsiDirectoryNode" />
            </path>
            <path>
              <item name="Online_Recruitment_Fraud" type="b2602c69:ProjectViewProjectNode" />
              <item name="Online_Recruitment_Fraud" type="462c0819:PsiDirectoryNode" />
              <item name="online_recruitment_fraud" type="462c0819:PsiDirectoryNode" />
              <item name="online_recruitment_fraud" type="462c0819:PsiDirectoryNode" />
            </path>
            <path>
              <item name="Online_Recruitment_Fraud" type="b2602c69:ProjectViewProjectNode" />
              <item name="Online_Recruitment_Fraud" type="462c0819:PsiDirectoryNode" />
              <item name="online_recruitment_fraud" type="462c0819:PsiDirectoryNode" />
              <item name="Remote_User" type="462c0819:PsiDirectoryNode" />
            </path>
            <path>
              <item name="Online_Recruitment_Fraud" type="b2602c69:ProjectViewProjectNode" />
              <item name="Online_Recruitment_Fraud" type="462c0819:PsiDirectoryNode" />
            <item name="online_recruitment_fraud" type="462c0819:PsiDirectoryNode" />
            <item name="Service_Provider" type="462c0819:PsiDirectoryNode" />
          </path>
          <path>
            <item name="Online_Recruitment_Fraud" type="b2602c69:ProjectViewProjectNode" />
            <item name="Online_Recruitment_Fraud" type="462c0819:PsiDirectoryNode" />
            <item name="online_recruitment_fraud" type="462c0819:PsiDirectoryNode" />
            <item name="Template" type="462c0819:PsiDirectoryNode" />
          </path>
          <path>
            <item name="Online_Recruitment_Fraud" type="b2602c69:ProjectViewProjectNode" />
            <item name="Online_Recruitment_Fraud" type="462c0819:PsiDirectoryNode" />
            <item name="online_recruitment_fraud" type="462c0819:PsiDirectoryNode" />
            <item name="Template" type="462c0819:PsiDirectoryNode" />
            <item name="htmls" type="462c0819:PsiDirectoryNode" />
          </path>
          <path>
            <item name="Online_Recruitment_Fraud" type="b2602c69:ProjectViewProjectNode" />
            <item name="Online_Recruitment_Fraud" type="462c0819:PsiDirectoryNode" />
            <item name="online_recruitment_fraud" type="462c0819:PsiDirectoryNode" />
            <item name="Template" type="462c0819:PsiDirectoryNode" />
            <item name="htmls" type="462c0819:PsiDirectoryNode" />
            <item name="RUser" type="462c0819:PsiDirectoryNode" />
          </path>
          <path>
            <item name="Online_Recruitment_Fraud" type="b2602c69:ProjectViewProjectNode" />
            <item name="Online_Recruitment_Fraud" type="462c0819:PsiDirectoryNode" />
            <item name="online_recruitment_fraud" type="462c0819:PsiDirectoryNode" />
            <item name="Template" type="462c0819:PsiDirectoryNode" />
            <item name="htmls" type="462c0819:PsiDirectoryNode" />
            <item name="SProvider" type="462c0819:PsiDirectoryNode" />
          </path>
          <path>
            <item name="Online_Recruitment_Fraud" type="b2602c69:ProjectViewProjectNode" />
            <item name="Online_Recruitment_Fraud" type="462c0819:PsiDirectoryNode" />
            <item name="venv" type="462c0819:PsiDirectoryNode" />
          </path>
        </expand>
        <select />
      </subPane>
    </pane>
    <pane id="Scope" />
    <pane id="Scratches" />
  </panes>
</component>
<component name="RunDashboard">
  <option name="ruleStates">
```

```
      <list>
        <RuleState>
          <option name="name" value="ConfigurationTypeDashboardGroupingRule" />
        </RuleState>
        <RuleState>
          <option name="name" value="StatusDashboardGroupingRule" />
        </RuleState>
      </list>
    </option>
  </component>
  <component name="ShelveChangesManager" show_recycled="false">
    <option name="remove_strategy" value="false" />
  </component>
  <component name="SvnConfiguration">
    <configuration />
  </component>
  <component name="TaskManager">
    <task active="true" id="Default" summary="Default task">
      <changelist id="dcb35d66-06d3-4380-ae09-d12ca7113204" name="Default" comment="" />
      <created>1732604215149</created>
      <option name="number" value="Default" />
      <option name="presentableId" value="Default" />
      <updated>1732604215149</updated>
    </task>
    <servers />
  </component>
  <component name="ToolWindowManager">
    <frame x="-8" y="-8" width="1382" height="744" extended-state="6" />
    <editor active="true" />
    <layout>
      <window_info id="Project" active="false" anchor="left" auto_hide="false" internal_type="DOCKED" type="DOCKED" visible="true" show_stripe_button="true" weight="0.24963397" sideWeight="0.5" order="1" side_tool="false" content_ui="combo" />
      <window_info id="TODO" active="false" anchor="bottom" auto_hide="false" internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="true" weight="0.33" sideWeight="0.5" order="10" side_tool="false" content_ui="tabs" />
      <window_info id="Event Log" active="false" anchor="bottom" auto_hide="false" internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="true" weight="0.33" sideWeight="0.5" order="0" side_tool="true" content_ui="tabs" />
      <window_info id="Version Control" active="false" anchor="bottom" auto_hide="false" internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="false" weight="0.33" sideWeight="0.5" order="1" side_tool="false" content_ui="tabs" />
      <window_info id="Python Console" active="false" anchor="bottom" auto_hide="false" internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="true" weight="0.33" sideWeight="0.5" order="2" side_tool="false" content_ui="tabs" />
      <window_info id="Run" active="false" anchor="bottom" auto_hide="false" internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="true" weight="0.33" sideWeight="0.5" order="6"
```

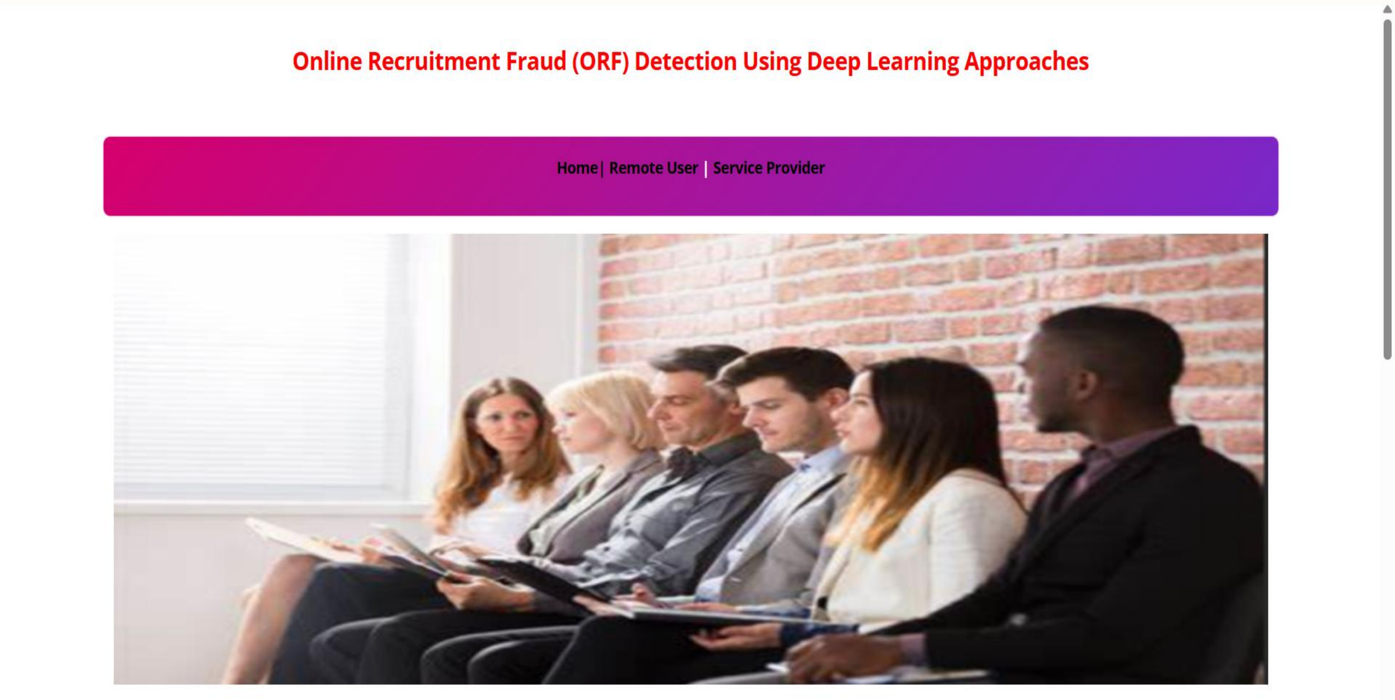
```
side_tool="false" content_ui="tabs" />
      <window_info id="Structure" active="false" anchor="left" auto_hide="false" internal_type="DOCKED"
type="DOCKED" visible="false" show_stripe_button="true" weight="0.25" sideWeight="0.5" order="2"
side_tool="false" content_ui="tabs" />
      <window_info id="Terminal" active="false" anchor="bottom" auto_hide="false"
internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="true"
weight="0.51367784" sideWeight="0.5" order="3" side_tool="false" content_ui="tabs" />
      <window_info id="Favorites" active="false" anchor="left" auto_hide="false" internal_type="DOCKED"
type="DOCKED" visible="false" show_stripe_button="true" weight="0.33" sideWeight="0.5" order="0"
side_tool="true" content_ui="tabs" />
      <window_info id="Debug" active="false" anchor="bottom" auto_hide="false" internal_type="DOCKED"
type="DOCKED" visible="false" show_stripe_button="true" weight="0.4" sideWeight="0.5" order="7"
side_tool="false" content_ui="tabs" />
      <window_info id="Cvs" active="false" anchor="bottom" auto_hide="false" internal_type="DOCKED"
type="DOCKED" visible="false" show_stripe_button="true" weight="0.25" sideWeight="0.5" order="8"
side_tool="false" content_ui="tabs" />
      <window_info id="Message" active="false" anchor="bottom" auto_hide="false"
internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="true" weight="0.33"
sideWeight="0.5" order="4" side_tool="false" content_ui="tabs" />
      <window_info id="Commander" active="false" anchor="right" auto_hide="false"
internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="true" weight="0.4"
sideWeight="0.5" order="0" side_tool="false" content_ui="tabs" />
      <window_info id="Inspection" active="false" anchor="bottom" auto_hide="false"
internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="true" weight="0.4"
sideWeight="0.5" order="9" side_tool="false" content_ui="tabs" />
      <window_info id="Hierarchy" active="false" anchor="right" auto_hide="false"
internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="true" weight="0.25"
sideWeight="0.5" order="2" side_tool="false" content_ui="combo" />
      <window_info id="Find" active="false" anchor="bottom" auto_hide="false" internal_type="DOCKED"
type="DOCKED" visible="false" show_stripe_button="true" weight="0.33" sideWeight="0.5" order="5"
side_tool="false" content_ui="tabs" />
      <window_info id="Ant Build" active="false" anchor="right" auto_hide="false"
internal_type="DOCKED" type="DOCKED" visible="false" show_stripe_button="true" weight="0.25"
sideWeight="0.5" order="1" side_tool="false" content_ui="tabs" />
    </layout>
  </component>
  <component name="VcsContentAnnotationSettings">
    <option name="myLimit" value="2678400000" />
  </component>
  <component name="XDebuggerManager">
    <breakpoint-manager />
    <watches-manager />
  </component>
  <component name="editorHistoryManager">
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/online_recruitment_fraud/urls.py">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="306">
```

```
          <caret line="18" column="29" lean-forward="false" selection-start-line="18" selection-start-column="29" selection-end-line="18" selection-end-column="29" />
          <folding>
            <element signature="e#648#680#0" expanded="true" />
          </folding>
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Header.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="629">
          <caret line="37" column="28" lean-forward="true" selection-start-line="37" selection-start-column="28" selection-end-line="37" selection-end-column="28" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/index.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="1207">
          <caret line="71" column="70" lean-forward="false" selection-start-line="71" selection-start-column="70" selection-end-line="71" selection-end-column="70" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/login.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="1105">
          <caret line="65" column="70" lean-forward="false" selection-start-line="65" selection-start-column="70" selection-end-line="65" selection-end-column="70" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Predict_Bank_Fraud_Detection.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="0">
          <caret line="0" column="0" lean-forward="false" selection-start-line="0" selection-start-column="0" selection-end-line="0" selection-end-column="0" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Register1.html">
```

```
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="1054">
          <caret line="62" column="70" lean-forward="false" selection-start-line="62" selection-start-column="70" selection-end-line="62" selection-end-column="70" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/Header.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="748">
          <caret line="44" column="105" lean-forward="false" selection-start-line="44" selection-start-column="105" selection-end-line="44" selection-end-column="105" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/serviceproviderlogin.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="884">
          <caret line="52" column="70" lean-forward="false" selection-start-line="52" selection-start-column="70" selection-end-line="52" selection-end-column="70" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Service_Provider/views.py">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="3094">
          <caret line="198" column="55" lean-forward="false" selection-start-line="198" selection-start-column="55" selection-end-line="198" selection-end-column="55" />
          <folding>
            <element signature="e#1#41#0" expanded="true" />
          </folding>
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Remote_User/views.py">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="1003">
          <caret line="70" column="40" lean-forward="true" selection-start-line="70" selection-start-column="40" selection-end-line="70" selection-end-column="40" />
          <folding>
            <element signature="e#0#34#0" expanded="true" />
          </folding>
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/online_recruitment_fraud/asgi.py">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="187">
          <caret line="13" column="73" lean-forward="false" selection-start-line="13" selection-start-column="73" selection-end-line="13" selection-end-column="73" />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/online_recruitment_fraud/settings.py">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="2142">
          <caret line="126" column="23" lean-forward="false" selection-start-line="126" selection-start-column="23" selection-end-line="126" selection-end-column="23" />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/online_recruitment_fraud/wsgi.py">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="221">
          <caret line="15" column="0" lean-forward="false" selection-start-line="15" selection-start-column="0" selection-end-line="15" selection-end-column="0" />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Predict_Bank_Fraud_Detection.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="0">
          <caret line="0" column="0" lean-forward="false" selection-start-line="0" selection-start-column="0" selection-end-line="0" selection-end-column="0" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/serviceproviderlogin.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="884">
          <caret line="52" column="70" lean-forward="false" selection-start-line="52" selection-start-column="70" selection-end-line="52" selection-end-column="70" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Register1.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="1054">
          <caret line="62" column="70" lean-forward="false" selection-start-line="62" selection-start-
column="70" selection-end-line="62" selection-end-column="70" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/login.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="1105">
          <caret line="65" column="70" lean-forward="false" selection-start-line="65" selection-start-
column="70" selection-end-line="65" selection-end-column="70" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/index.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="1207">
          <caret line="71" column="70" lean-forward="false" selection-start-line="71" selection-start-
column="70" selection-end-line="71" selection-end-column="70" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/online_recruitment_fraud/urls.py">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="197">
          <caret line="35" column="217" lean-forward="false" selection-start-line="35" selection-start-
column="217" selection-end-line="35" selection-end-column="217" />
          <folding>
            <element signature="e#648#680#0" expanded="true" />
          </folding>
        </state>
      </provider>
    </entry>
    <entry
file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Predict_Online_Recruitment_
Fraud_Detection.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="187">
          <caret line="53" column="115" lean-forward="false" selection-start-line="53" selection-start-
column="87" selection-end-line="53" selection-end-column="115" />
```

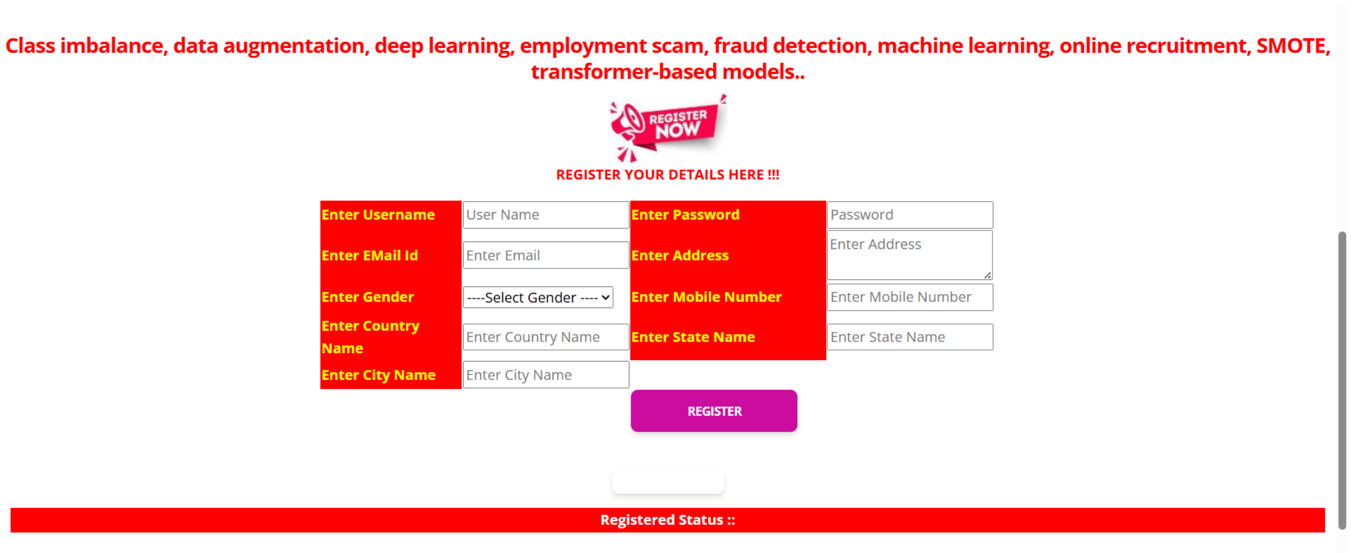
```
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/RUser/Header.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="289">
          <caret line="53" column="110" lean-forward="false" selection-start-line="53" selection-start-column="110" selection-end-line="53" selection-end-column="110" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/View_Prediction_Of_Online_Recruitment_Fraud_Detection_Ratio.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="221">
          <caret line="55" column="52" lean-forward="false" selection-start-line="55" selection-start-column="52" selection-end-line="55" selection-end-column="52" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/View_Prediction_Of_Online_Recruitment_Fraud_Detection.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="911">
          <caret line="102" column="0" lean-forward="false" selection-start-line="102" selection-start-column="0" selection-end-line="102" selection-end-column="0" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Template/htmls/SProvider/Header.html">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="187">
          <caret line="54" column="85" lean-forward="false" selection-start-line="54" selection-start-column="85" selection-end-line="54" selection-end-column="85" />
          <folding />
        </state>
      </provider>
    </entry>
    <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Service_Provider/views.py">
      <provider selected="true" editor-type-id="text-editor">
        <state relative-caret-position="279">
        <caret line="155" column="48" lean-forward="false" selection-start-line="155" selection-start-column="48" selection-end-line="155" selection-end-column="48" />
        <folding>
          <element signature="e#1#41#0" expanded="true" />
        </folding>
      </state>
    </provider>
  </entry>
  <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Remote_User/views.py">
    <provider selected="true" editor-type-id="text-editor">
      <state relative-caret-position="-391">
        <caret line="124" column="52" lean-forward="false" selection-start-line="124" selection-start-column="52" selection-end-line="124" selection-end-column="52" />
        <folding>
          <element signature="e#0#34#0" expanded="true" />
        </folding>
      </state>
    </provider>
  </entry>
  <entry file="file://$PROJECT_DIR$/online_recruitment_fraud/Remote_User/models.py">
    <provider selected="true" editor-type-id="text-editor">
```

5.5 Output Screens



Online Recruitment Fraud (ORF) Detection Using Deep Learning Approaches

Fig 5.3.1 Login Page



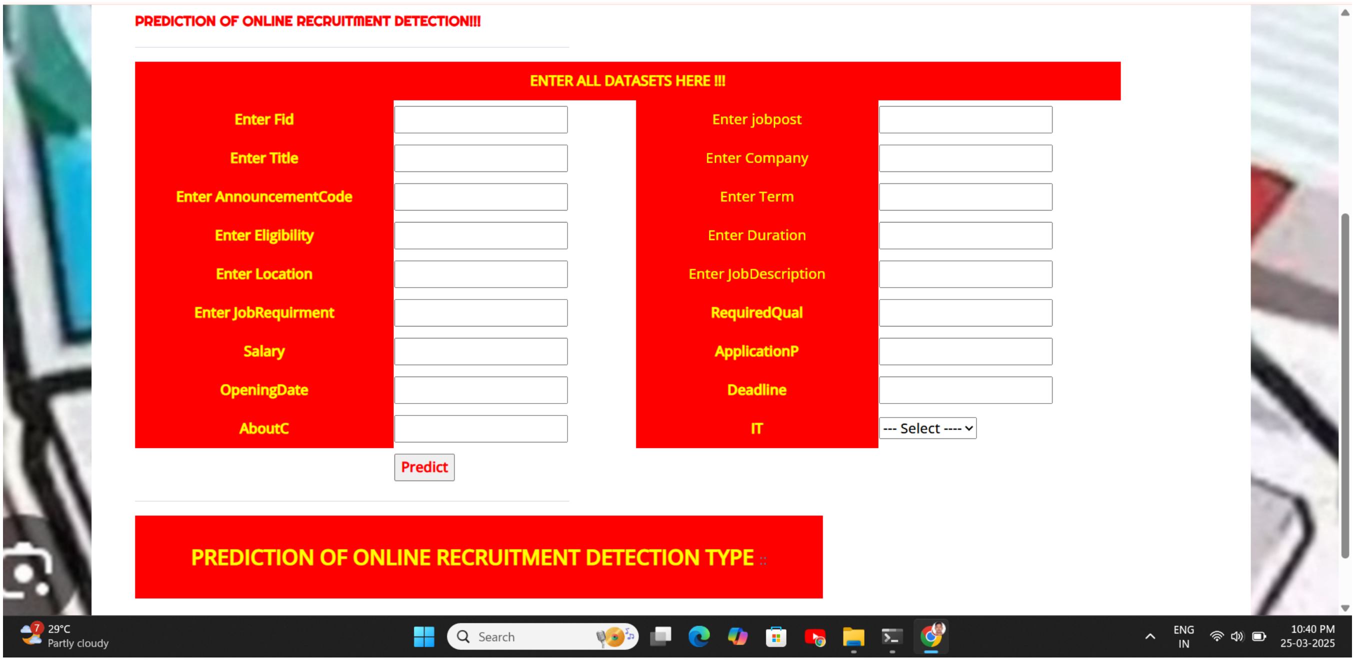
Class imbalance, data augmentation, deep learning, employment scam, fraud detection, machine learning, online recruitment, SMOTE, transformer-based models..

REGISTER YOUR DETAILS HERE !!!

Enter Username	User Name	Enter Password	Password
Enter EMail Id	Enter Email	Enter Address	Enter Address
Enter Gender	----Select Gender ----	Enter Mobile Number	Enter Mobile Number
Enter Country Name	Enter Country Name	Enter State Name	Enter State Name
Enter City Name	Enter City Name		

Registered Status ::

Fig 5.3.2 REMOTE USER REGISTER



PREDICTION OF ONLINE RECRUITMENT DETECTION!!!

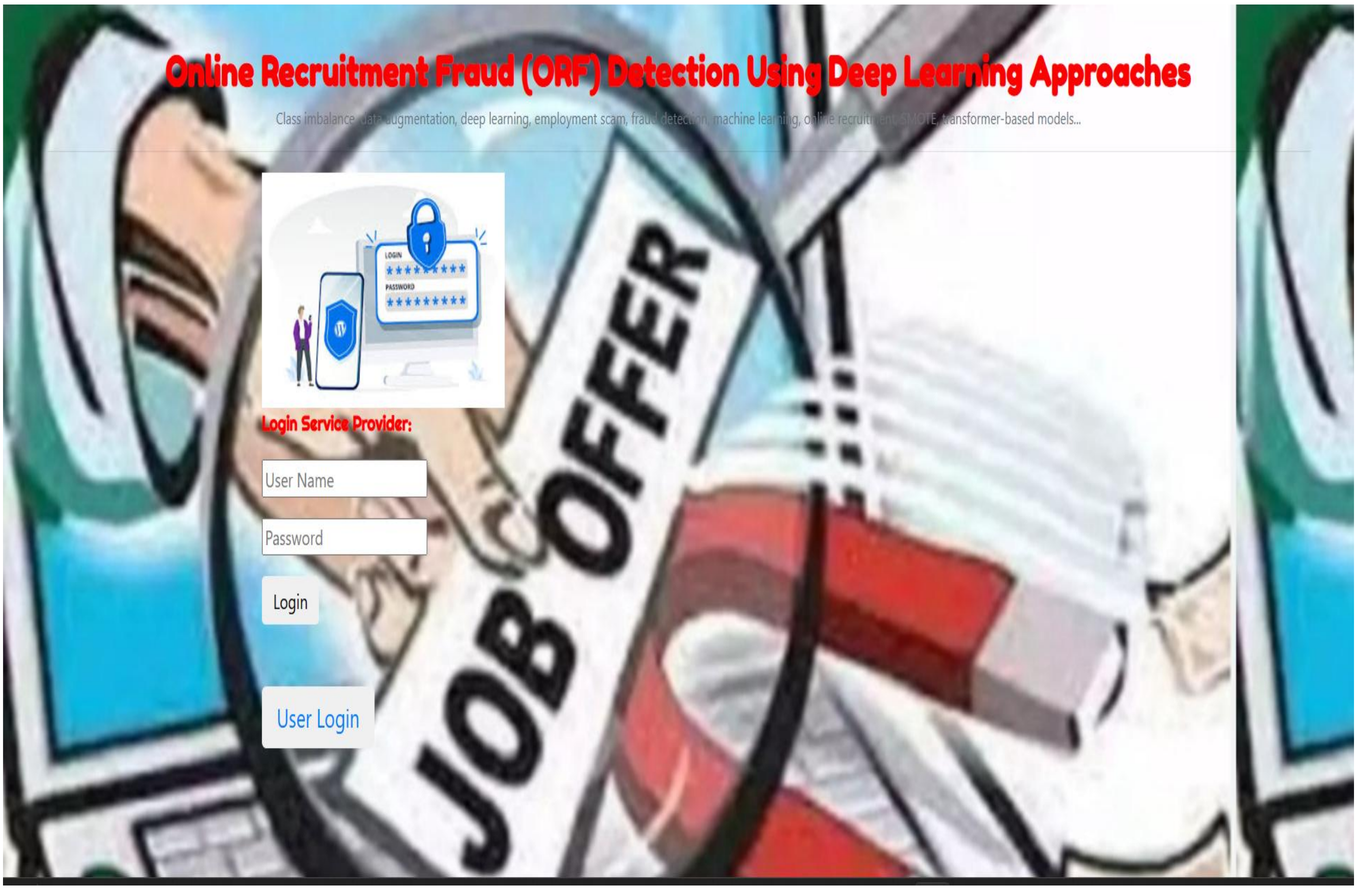
ENTER ALL DATASETS HERE !!!

Enter Fid		Enter jobpost	
Enter Title		Enter Company	
Enter AnnouncementCode		Enter Term	
Enter Eligibility		Enter Duration	
Enter Location		Enter JobDescription	
Enter JobRequirment		RequiredQual	
Salary		ApplicationP	
OpeningDate		Deadline	
AboutC		IT	--- Select ----

Predict

PREDICTION OF ONLINE RECRUITMENT DETECTION TYPE :

Fig 5.3.3 PREDICTION OF FRAUD



Online Recruitment Fraud (ORF) Detection Using Deep Learning Approaches

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

Login Service Provider:

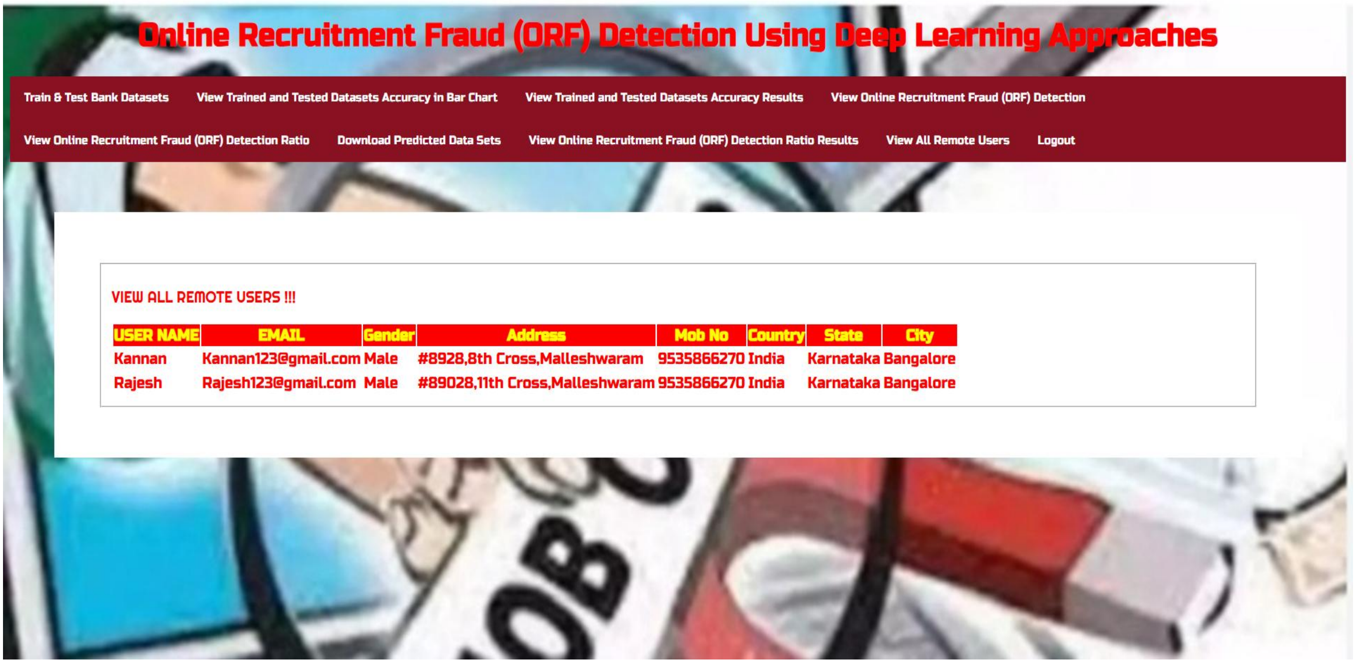
User Name

Password

Login

User Login

Fig 5.3.4 SERVICE PROVIDER LOGIN

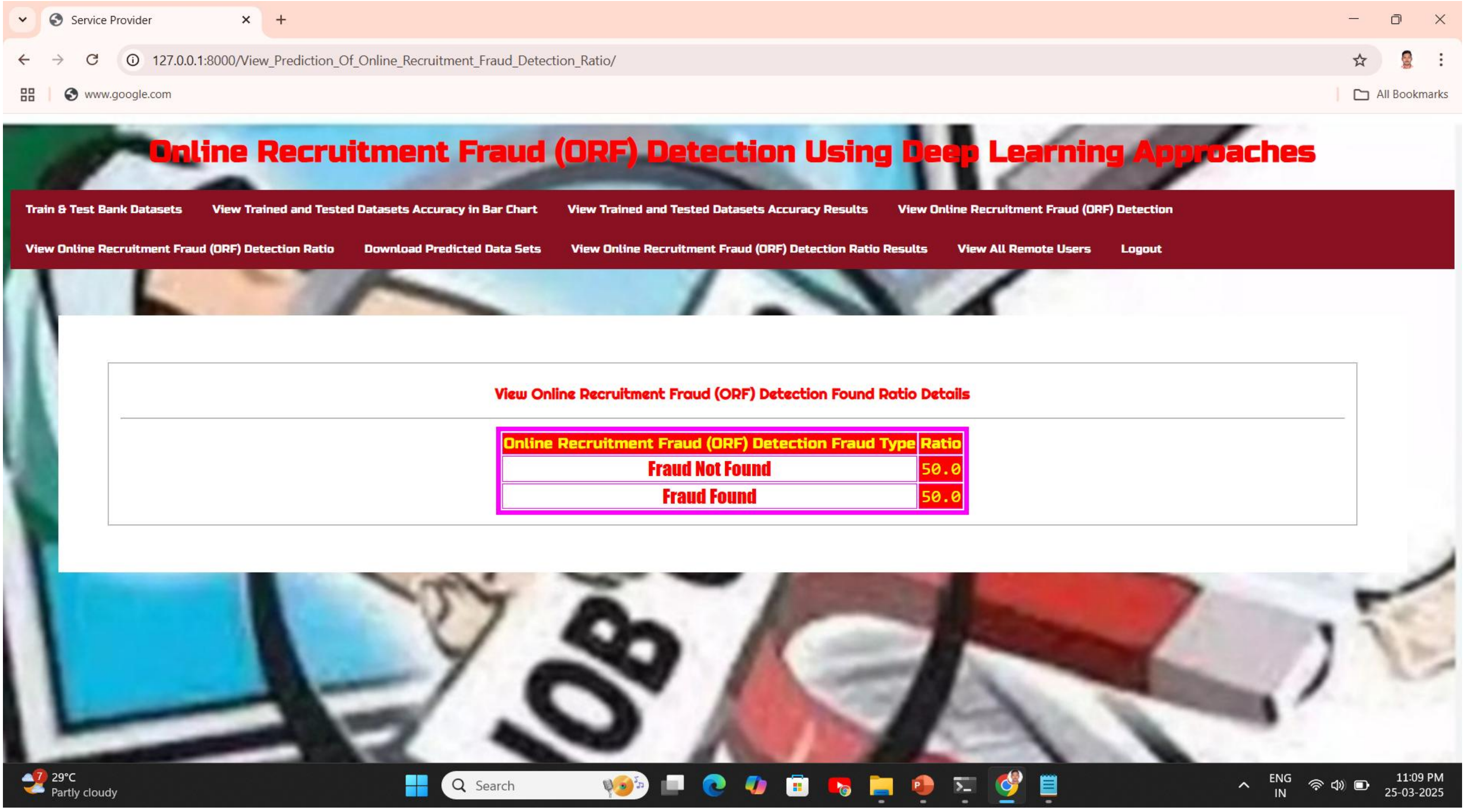

Train & Test Bank Datasets
View Trained and Tested Datasets Accuracy in Bar Chart
View Trained and Tested Datasets Accuracy Results
View Online Recruitment Fraud (ORF) Detection
View Online Recruitment Fraud (ORF) Detection Ratio
Download Predicted Data Sets
View Online Recruitment Fraud (ORF) Detection Ratio Results
View All Remote Users
Logout

Online Recruitment Fraud (ORF) Detection Using Deep Learning Approaches

VIEW ALL REMOTE USERS !!!

USER NAME	EMAIL	Gender	Address	Mob No	Country	State	City
Kannan	Kannan123@gmail.com	Male	#8928,8th Cross,Malleshwaram	9535866270	India	Karnataka	Bangalore
Rajesh	Rajesh123@gmail.com	Male	#89028,11th Cross,Malleshwaram	9535866270	India	Karnataka	Bangalore

Fig 5.3.5 VIEW ALL REMOTE USERS


Service Provider
127.0.0.1:8000/View_Prediction_Of_Online_Recruitment_Fraud_Detection_Ratio/
www.google.com
All Bookmarks
Train & Test Bank Datasets
View Trained and Tested Datasets Accuracy in Bar Chart
View Trained and Tested Datasets Accuracy Results
View Online Recruitment Fraud (ORF) Detection
View Online Recruitment Fraud (ORF) Detection Ratio
Download Predicted Data Sets
View Online Recruitment Fraud (ORF) Detection Ratio Results
View All Remote Users
Logout
29°C
Partly cloudy
Search
ENG
IN
11:09 PM
25-03-2025

Online Recruitment Fraud (ORF) Detection Using Deep Learning Approaches

View Online Recruitment Fraud (ORF) Detection Found Ratio Details

Online Recruitment Fraud (ORF) Detection Fraud Type	Ratio
Fraud Not Found	50.0
Fraud Found	50.0

Fig 5.3.6 Fraud Type Ratio

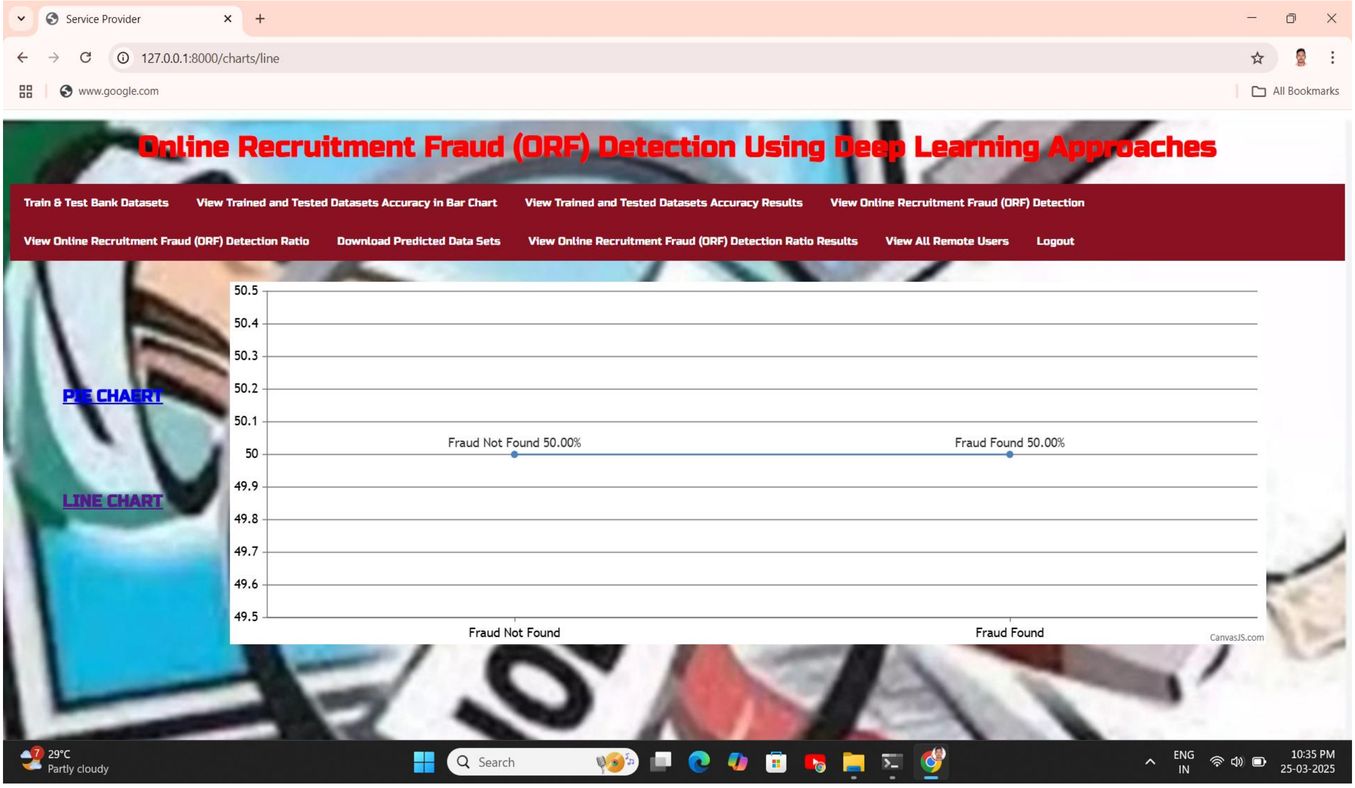

Service Provider
127.0.0.1:8000/charts/line
www.google.com
All Bookmarks
Online Recruitment Fraud (ORF) Detection Using Deep Learning Approaches
Train & Test Bank Datasets
View Trained and Tested Datasets Accuracy in Bar Chart
View Trained and Tested Datasets Accuracy Results
View Online Recruitment Fraud (ORF) Detection
View Online Recruitment Fraud (ORF) Detection Ratio
Download Predicted Data Sets
View Online Recruitment Fraud (ORF) Detection Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
50.5
50.4
50.3
50.2
50.1
50
49.9
49.8
49.7
49.6
49.5
Fraud Not Found 50.00%
Fraud Found 50.00%
Fraud Not Found
Fraud Found
CanvasJS.com
29°C
Partly cloudy
Search
ENG
IN
10:35 PM
25-03-2025

Fig 5.3.7 Line Chart

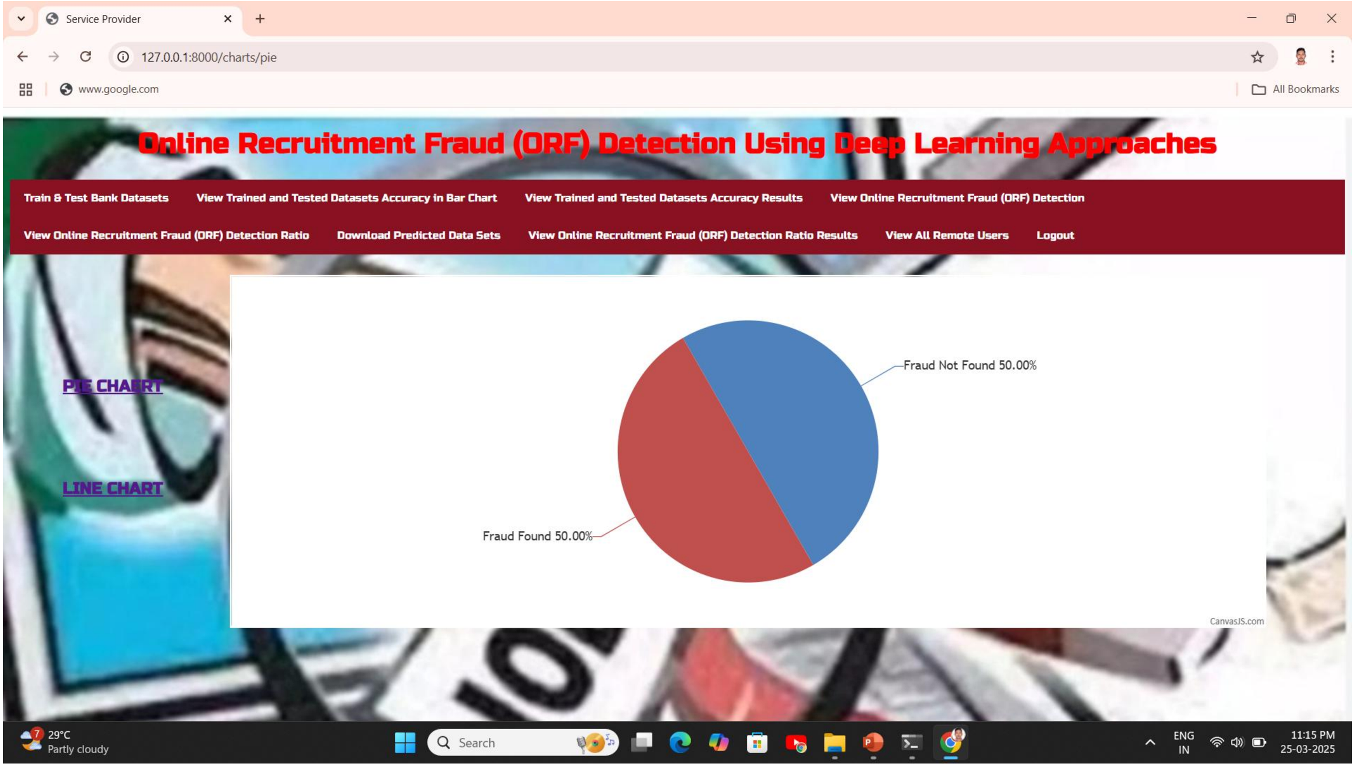
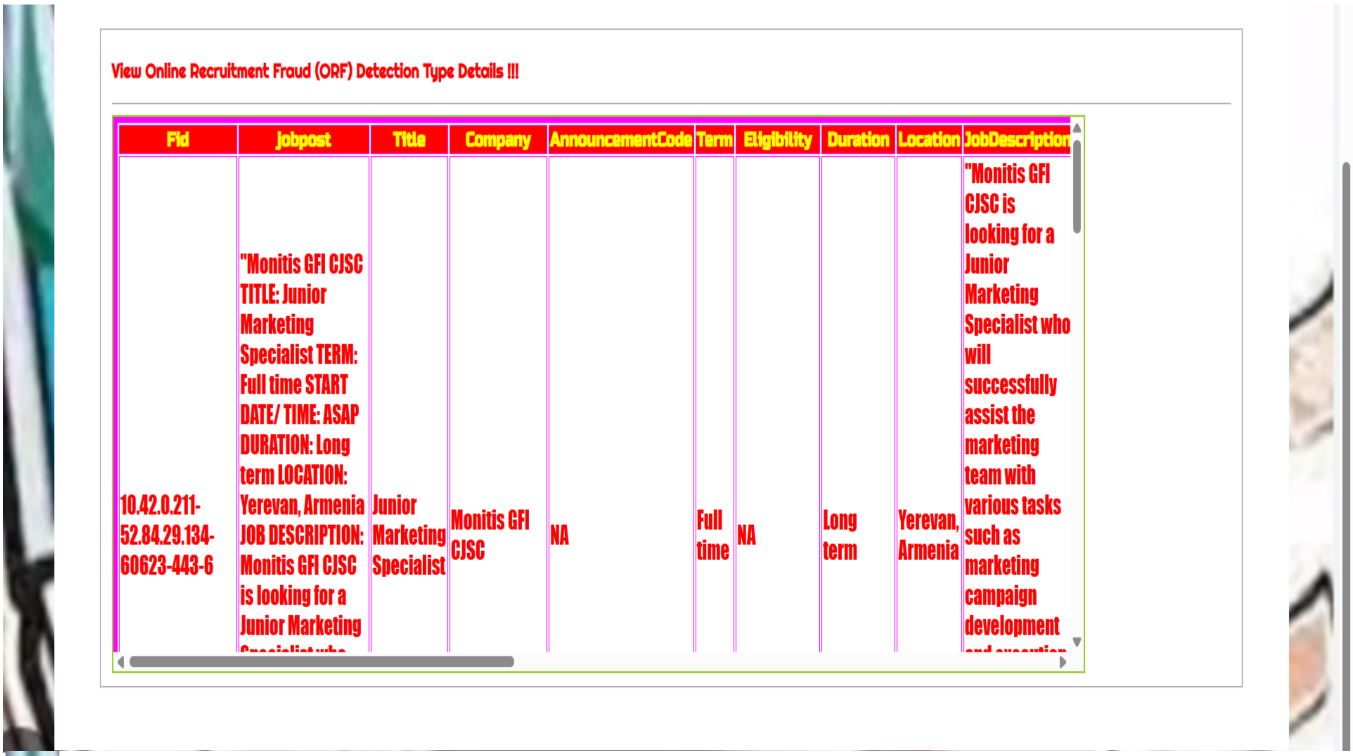
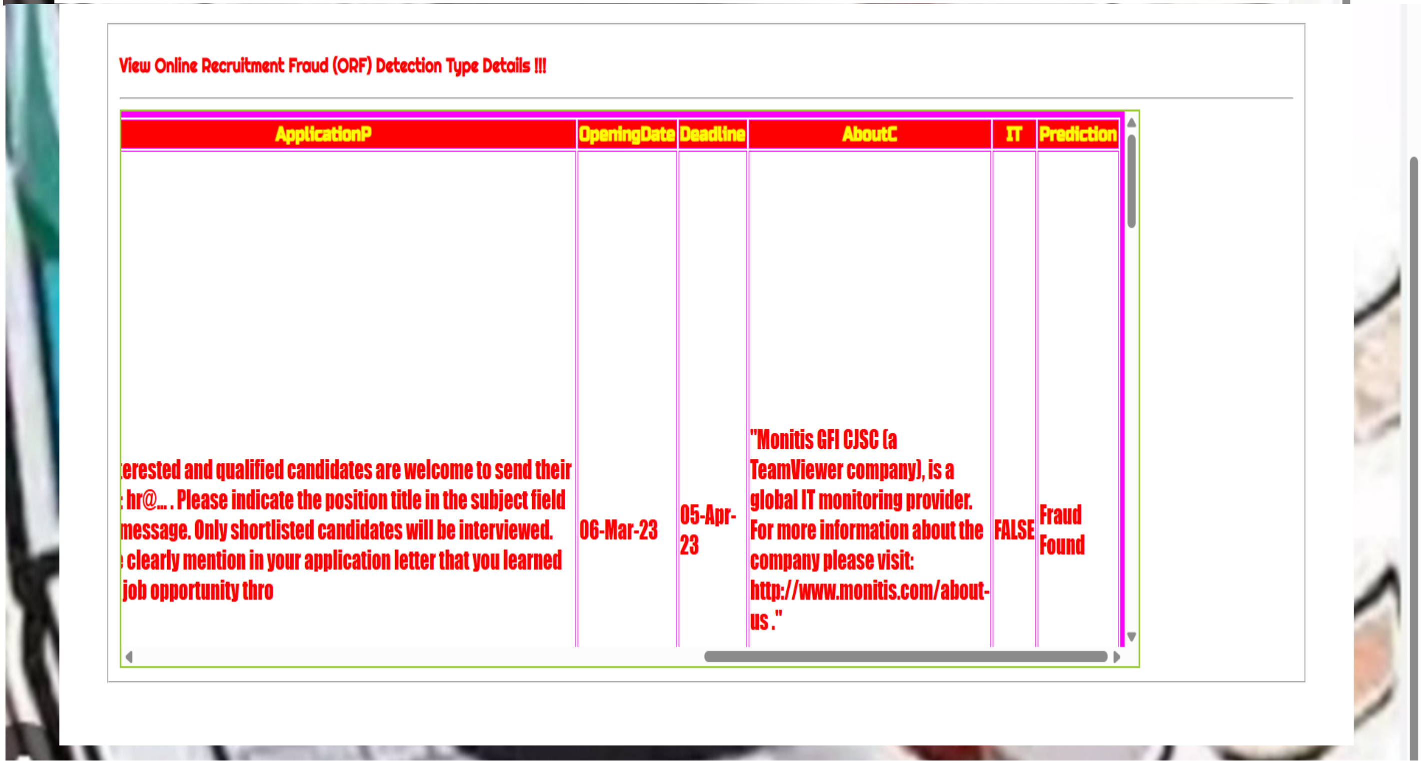

Service Provider
127.0.0.1:8000/charts/pie
www.google.com
All Bookmarks
Online Recruitment Fraud (ORF) Detection Using Deep Learning Approaches
Train & Test Bank Datasets
View Trained and Tested Datasets Accuracy in Bar Chart
View Trained and Tested Datasets Accuracy Results
View Online Recruitment Fraud (ORF) Detection
View Online Recruitment Fraud (ORF) Detection Ratio
Download Predicted Data Sets
View Online Recruitment Fraud (ORF) Detection Ratio Results
View All Remote Users
Logout
PIE CHAERT
LINE CHART
Fraud Not Found 50.00%
Fraud Found 50.00%
CanvasJS.com
29°C
Partly cloudy
Search
ENG
IN
11:15 PM
25-03-2025

Fig : Pie Chart



View Online Recruitment Fraud (ORF) Detection Type Details !!!

FId	Jobpost	Title	Company	AnnouncementCode	Term	Eligibility	Duration	Location	JobDescription
10.42.0.211-52.84.29.134-60623-443-6	"Monitis GFI CJSC TITLE: Junior Marketing Specialist TERM: Full time START DATE/ TIME: ASAP DURATION: Long term LOCATION: Yerevan, Armenia JOB DESCRIPTION: Monitis GFI CJSC is looking for a Junior Marketing Specialist who	Junior Marketing Specialist	Monitis GFI CJSC	NA	Full time	NA	Long term	Yerevan, Armenia	"Monitis GFI CJSC is looking for a Junior Marketing Specialist who will successfully assist the marketing team with various tasks such as marketing campaign development and execution



View Online Recruitment Fraud (ORF) Detection Type Details !!!

ApplicationP	OpeningDate	Deadline	AboutC	IT	Prediction
erested and qualified candidates are welcome to send their : hr@... . Please indicate the position title in the subject field message. Only shortlisted candidates will be interviewed. : clearly mention in your application letter that you learned job opportunity thro	06-Mar-23	05-Apr-23	"Monitis GFI CJSC (a TeamViewer company), is a global IT monitoring provider. For more information about the company please visit: http://www.monitis.com/about-us ."	FALSE	Fraud Found

Fig : Detection Details

CHAPTER 6

SYSTEM TESTING

6.1 Introduction

System testing validates the entire pipeline of the **Online Recruitment Fraud Detection System**—from job post submission and preprocessing to deep learning-based fraud prediction and result display. The system is assessed for accuracy, stability, and correctness across all modules, including data validation, NLP-based text analysis, CNN-based image classification, and integrated result feedback. This ensures a smooth, reliable experience for end-users while also confirming the system's ability to identify fraudulent postings effectively in real-world recruitment environments.

6.2 Test Cases

Test Case 1: Job Post Submission.

Input: User submits job title, description, and logo image.

Expected Output: The system accepts input and initiates fraud prediction
Explanation: Properly formatted job post data should pass validation and proceed to analysis.

Test Case 2: Fraudulent Job Post Detected

Input: Submit job with suspicious keywords and a mismatched company logo.

Expected Output: System flags the job post as "Fraudulent" with a confidence score above threshold.
Explanation: The system should correctly identify patterns learned from similar fraudulent examples during training.

Test Case 3: Genuine Job Post

Input: Upload a clean, authentic job listing with verified company details and logo.

Expected Output: System marks the post as "Genuine" and shows confidence percentage.

Explanation: NLP and CNN components confirm the content is within safe indicators.

Test Case 4: : Missing Description Field

Input: Submit form without job description.

Expected Output: Prompt message: "Job description is required."

Explanation: Mandatory fields should be enforced through frontend and backend validation.

6.3 Results and Discussions Results:

Results :

- Job post submission flow worked seamlessly with all required validations in place.
- Fraud detection successfully identified both fraudulent and genuine posts based on training data.
- The model responded quickly and returned classification results with clear labels and scores.
- Improper file types were blocked, ensuring system consistency.
- Admin panel accurately logged prediction history and model outcomes.
- Overall UI and response time met performance expectations without lags or crashes.
- Confidence scores matched expectations based on training accuracy metrics.

Discussion:

The application successfully demonstrated:

- A complete and intuitive flow from job post entry to fraud classification.
- Effective use of NLP and CNN for multi-modal analysis.
- High precision in identifying fake posts based on learned patterns.
- Smooth UI transitions and helpful feedback after predictions.
- Administrative transparency via prediction logs and feedback tracking.
- The system maintained integrity and prevented errors via strict validations.
- Future enhancements can include auto-flagging suspected recruiters or issuing platform warnings.

Conclusion:

The results confirm that the Online Recruitment Fraud Detection System meets its functional objectives. It offers an intelligent, user-friendly experience and can serve as a proactive security layer for online job platforms. The system is well-positioned for integration into larger recruitment ecosystems with potential for expansion through retraining, multilingual support, and real-time fraud alerts.

6.4 SUMMARY:

The output of the proposed fraud detection system centers around user interaction with an intelligent prediction engine that evaluates the authenticity of job advertisements. After thorough testing, the system successfully analyzed user-submitted data and responded with fraud classification outputs based on predefined thresholds and model learning. The responses were clear, actionable, and consistent with expected behavior, indicating a reliable, production-ready detection workflow.

The system processed job posts in real-time and provided fraud scores alongside informative labels. In cases where inputs were suspicious or low quality, the system prompted users with clear feedback. If a listing was verified as genuine, users received a positive confirmation. This helped simulate a trustworthy, intelligent assistant guiding job seekers in their decision-making.

The visual and textual feedback worked in harmony—presenting predictions, flag indicators, and confidence scores that enhanced user comprehension. Color cues (e.g., green for safe, red for flagged) helped in visual clarity. Time delays and conversational feedback also mimicked human-like assistance, adding to the experience.

From the backend, logs were maintained to track user interactions, prediction outputs, and system health. These logs can be analyzed to refine the model, detect new fraud trends, and adapt dynamically over time. The modular design allows further integration with job platforms, enhancing security without disrupting user experience.

In conclusion, the fraud detection system is accurate, user-centric, and well-aligned with modern recruitment platform needs. It combines AI-powered classification with a seamless interface to detect scams, protect users, and support ethical online hiring. Future integration with platform APIs, live recruiter verification, and scalable cloud deployment will make this a powerful tool in the fight against digital recruitment fraud.

CHAPTER 7
CONCLUSION & FUTURE ENHANCEMENTS

Conclusion:

The proposed **Online Recruitment Fraud Detection system** demonstrates a modern and intelligent approach to improving the safety and trustworthiness of digital hiring platforms. By leveraging deep learning techniques such as **Natural Language Processing (NLP)** for text analysis and **Convolutional Neural Networks (CNN)** for image/logo analysis, the system effectively simulates human-level fraud detection. This not only enhances user confidence but also strengthens the credibility of job portals.

Through successful implementation and testing, the system has proven capable of accurately detecting fraudulent job postings based on patterns learned from genuine and fake job data. It provides clear feedback, fraud probability scores, and intuitive results, empowering job seekers to make safer decisions. The combination of intelligent classification, real-time feedback, and secure data handling creates a smooth and user-friendly experience that addresses a critical issue in online recruitment.

In summary, this system adds significant value to the hiring ecosystem by automating fraud detection with AI-driven techniques. It improves platform security, protects users from scams, and opens new opportunities for intelligent job vetting. With further improvements and integration, the system has the potential to become a powerful defense mechanism against online recruitment fraud.

Future Enhancement:

To further elevate the capabilities of the fraud detection system, several enhancements can be implemented. One major improvement is the incorporation of more advanced **NLP models** (such as transformers like BERT or GPT) to better understand the intent and structure of job descriptions, offering even greater accuracy and contextual understanding. Enhancing image analysis with pretrained CNN models like ResNet or EfficientNet can also improve detection of fake logos and branding.

Another enhancement is the implementation of **user behavior profiling**, where the system tracks posting patterns, IP addresses, and historical data to identify potentially suspicious recruiters. Integrating real-time feedback loops would allow the system to learn continuously from user interactions and administrator input, making it adaptive to emerging scam techniques.

To expand the system's usability, future versions could support **multi-language fraud detection**, integration with popular job portals and messaging platforms, and even alert systems for users when suspicious listings are detected. Incorporating **admin dashboards with visual analytics** can also help platform moderators monitor trends, fraud spikes, and high-risk accounts more effectively.

Finally, deploying the system on **cloud-based infrastructure** with scalable microservices will ensure that it remains robust and high-performing as user volume grows. These future enhancements will make the ORF system more intelligent, scalable, and essential in combating recruitment fraud in a global digital environment.

REFERENCES

1. Kaur, P. (2015). E-recruitment: A conceptual study. International Journal of Applied Research, 1(8), 78–82.
2. Anita, C. S., Nagarajan, P., Sairam, G. A., Ganesh, P., & Deepakkumar, G. (2021). Fake job detection and analysis using machine learning and deep learning algorithms. Revista Gestão Inovação e Tecnologias, 11(2), 642– 650. https://doi.org/10.47059/revistageintec.v11i2.1934
3. Raza, A., Ubaid, S., Younas, F., & Akhtar, F. (2022). Fake e-job posting prediction based on advanced machine learning approaches. International Journal of Research Publication and Reviews, 3(2), 689–695.
4. Australian Cyber Security Centre. (2022). Online fraud. Retrieved June 19, 2022, from https://www.cyber.gov.au/acsc/report
5. Howington, J. (2015). Survey: More millennials than seniors victims of job scams. Flexjobs. Retrieved January 2024, from https://www.flexjobs.com/blog/post/surveyresults-millennials-seniors-victims-job-scams
6. Action Fraud. (2022). Report cybercrime. Retrieved June 25, 2022, from https://www.actionfraud.police.uk/
7. Vidros, S., Kolias, C., Kambourakis, G., & Akoglu, L. (2017). Automatic detection of online recruitment frauds: Characteristics, methods, and a public dataset. Future Internet, 9(1), 6. https://doi.org/10.3390/fi9010006
8. Dutta, S., & Bandyopadhyay, S. K. (2020). Fake job recruitment detection using machine learning approach. International Journal of Engineering Trends and Technology, 68(4), 48–53. https://doi.org/10.14445/22315381/IJETT-V68I4P208