CROP YIELD FORECASTING USING MACHINE LEARNING

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

C.YELLA KRISHNA REDDY (21UECM0272) **(VTU20114)**
C.YELLA NAGI REDDY (21UECM0273) **(VTU20113)**
T.SAI KUMAR REDDY (21UECM0249) **(VTU19969)**

Under the guidance of
Mrs.J.SWAPNA, M.E.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CROP YIELD FORECASTING USING MACHINE LEARNING

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

C.YELLA KRISHNA REDDY	(21UECM0272)	**(VTU20114)**
C.YELLA NAGI REDDY	(21UECM0273)	**(VTU20113)**
T.SAI KUMAR REDDY	(21UECM0249)	**(VTU19969)**

Under the guidance of
Mrs.J.SWAPNA, M.E.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled ”CROP YIELD FORECASTING USING MACHINE LEARNING” by ”C.YELLA KRISHNA REDDY (21UECM0272), C.YELLA NAGI REDDY (21UECM0273), T.SAI KUMAR REDDY (21UECM0249)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Mrs. J. SWAPNA, M.E.,
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

Signature of Professor In-charge
Dr. A. Peter Soosai Anandaraj, M.E., Ph.D.,
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
May, 2024

DECLARATION

We declare that this written submission representation of ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

C.YELLA KRISHNA REDDY

Date: / /

C.YELLA NAGI REDDY

Date: / /

T.SAI KUMAR REDDY

Date: / /

APPROVAL SHEET

This project report entitled ”CROP YIELD FORECASTING USING MACHINE LEARNING” by C.YELLA KRISHNA REDDY (21UECM0272), C.YELLA NAGI REDDY (21UECM0273), T.SAI KUMAR REDDY (21UECM0249) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Mrs.J.SWAPNA, M.E.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our respected **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Mrs.J.SWAPNA, M.E.,** for her cordial support, valuable information and guidance, she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. U.HEMAVATHI, M.E., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

C.YELLA KRISHNA REDDY (21UECM0272)
C.YELLA NAGI REDDY (21UECM0273)
T.SAI KUMAR REDDY (21UECM0249)

ABSTRACT

The impact of climate change in India, majority of the agricultural crops are being badly affected in terms of their performance over a period of last two decades. Predicting the crop yield well ahead of its harvest would help the policy makers and farmers for taking appropriate measures for marketing and storage. Such predictions will also help the associated industries for planning the logistics of their business.Several methods of predicting and modeling crop yields have been developed in the past with varying rate of success, as these don't take into account characteristics of the weather, a n d are mostly empirical.

In the present study a software tool named 'Crop Advisor' has been developed as an user friendly web page for predicting the influence of climatic parameters on the crop yields.C4.5 algorithm is used to find out the most influencing climatic parameter on the crop yields of selected crops in selected districts of Madhya Pradesh. This software provides an indication of relative influence of different climatic parameters on the crop yield, other agro-input parameters responsible for crop yield are not considered in this tool, since, application of these input parameters varies with individual fields in space and time.

Keywords: Climate Variability, Crop yield Prediction, Environmental Impact, Precision Farming, Sustainable Agriculture, Soil Health.

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

S.NO	ABBREVIATIONS	DEFINITION
1.	AI	Artificial Intellengence
2.	ANN	Artificial Neural Networks
3.	DQN	Deep Q-Network
4.	DS	Data Science
5.	GPS	Global Positioning System
6.	LSTM	Long short term memory
7.	ML	Machine Learning
8.	PH	Potential of Hydrogen

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Agriculture is fundamental to human survival.For populated developing countries like India, it is even more imperative to increase the productivity of crops, fruits and vegetables. Not only productivity, the quality of produce needs to stay high for better public health. However, both productivity and quality of food gets hampered by factors such as spread of diseases that could have been prevented with early diagnosis.Many of these diseases are infectious leading to total loss of crop yield. Given the vast geographical spread of agricultural lands, low education levels of farmers coupled with limited awareness and lack of access to plant pathologists, human assisted disease diagnosis is not effective and cannot keep up with the exorbitant requirements.To overcome the shortfall of human assisted disease diagnosis, it is imperative to build automation around crop disease diagnosis with technology and introduce low cost and accurate machine assisted diagnosis easily accessible to farmers. Some strides have been made in applying technologies such as robotics and computer vision systems to solve myriad problems in the agricultural domain.

The potential of image processing has been explored to assist with precision agriculture practices, weed and herbicide technologies, monitoring plant growth and plant nutrition management.However, progress on automating plant disease diagnosis is still rudimentary in spite of the fact that many plant diseases can be identified by plant pathologists by visual inspection of physical symptoms such as detectable change in color, wilting, appearance of spots and lesions etc. along with soil and climatic conditions. Overall, the commercial level of investment in bridging agriculture and technology remains lower as compared to investments done in more lucrative fields such as human health and education.

1.2 Aim Of The Project

The primary goal of this project is to design and implement a application to predict a climatic conditions and which crop is suitable for land accurately. To enhance crop yield prediction based on weather conditions, consider employing machine learning models trained on historical weather data and crop yield outcomes.One of the key objectives is to leverage historical crop data, encompassing information on various crops, farming practices, and environmental conditions. This extensive dataset will serve as the foundation for training machine learning models, enabling them to recognize patterns and correlations between different variables. By analyzing factors such as weather patterns, soil health, irrigation practices, and crop varieties, the models aim to discern the intricate relationships that influence crop yields. The project also focuses on real-time data acquisition to enhance the accuracy of predictions. Implementing sensors and IoT devices in the field will provide up-to-the-minute information on soil moisture, temperature, and other relevant parameters. Integrating this real-time data with the machine learning models ensures that predictions are not only based on historical trends but also reflect current conditions, allowing for timely and informed decision-making.

Furthermore, the project aims to develop user-friendly interfaces for farmers, agricultural experts, and policymakers. These interfaces will provide easy access to the predictive models, enabling users to input specific parameters and obtain accurate forecasts for their crops. The user interface will be designed to be intuitive, ensuring that even those with limited technical expertise can benefit from the insights generated by the machine learning algorithms.

1.3 Project Domain

The project domain for crop yield forecasting using machine learning revolves around leveraging advanced machine learning techniques to optimize and predict crop yields in the context of sustainable agriculture. The goal is to develop a deep reinforcement learning model that can analyze and adapt to dynamic agricultural environments, considering factors such as climate, soil quality, and crop management practices. The project aims to address the challenges faced by farmers in maximizing crop production while minimizing resource usage and environmental impact. By integrating deep reinforcement learning, the model can continuously learn and make informed decisions, optimizing crop yield strategies over time. This approach

contribute the broader goal of achieving sustainable agrarian practices by promoting efficient resource allocation, reducing waste, and enhancing overall crop production.

The project's significance lies in its potential to empower farmers with a predictive tool that adapts to changing conditions, allowing for proactive decision-making and ultimately supporting the transition towards more sustainable and resilient agricultural systems. Through the application of cutting-edge technology, the project aims to contribute to global efforts in ensuring food security and fostering environmentally conscious farming practices.

1.4 Scope of the Project

The scope of the project, crop yield forecasting using machine learning encompasses several key aspects to address the challenges and opportunities within the realm of sustainable agriculture. Firstly, the project will involve data collection and analysis, incorporating diverse factors such as historical crop yields, climate data, soil quality, and agricultural practices. This comprehensive dataset will serve as the foundation for training and fine-tuning the deep reinforcement learning model. The project will involve the development and implementation of a sophisticated deep reinforcement learning algorithm capable of adapting to the dynamic nature of agrarian environments. The model will undergo rigorous testing and validation to ensure its accuracy and reliability in predicting crop yields under varying conditions.

Furthermore, the project will explore user-friendly interfaces to enable easy adoption by farmers and agricultural stakeholders. This includes designing a user interface for real-time monitoring, decision support, and the visualization of predictive insights. The scope also extends to integrating the model with existing agricultural technologies and practices, fostering seamless integration into farmers' workflows. Overall, the project aspires to contribute to sustainable agrarian practices by providing a robust and adaptable tool for crop yield prediction. The scope involves not only technical development but also considerations for practical implementation, usability, and positive impacts on both productivity and environmental sustainability in agriculture.

Chapter 2

LITERATURE REVIEW

[1]Stewart et.al (2014) introduced a method for quantitatively assessing virulence in Zymoseptoria tritici, the causal agent of Septoria tritici blotch on wheat, utilizing high-throughput automated image analysis. Z. tritici manifests as pycnidia within chlorotic and necrotic lesions on infected wheat leaves. The researchers devised a high-throughput phenotyping technique employing automated digital image analysis to precisely quantify the percentage of leaf area covered by lesions (PLACL), as well as the size and number of pycnidia. Through a seedling inoculation assay involving 361 Z. tritici isolates from a controlled cross and two distinct winter wheat cultivars, it was determined that pycnidia size and density represent quantitative traits exhibiting a continuous distribution within the progeny. While a weak correlation was observed between pycnidia density and size, and between pycnidia density and PLACL, significant differences in PLACL and pycnidia density were identified between resistant and susceptible wheat cultivars.

[2]Krizhevsky et.al (2017) disscussed about the imageNet classification with deep convolutional neural networks. The results on ILSVRC-2010 are summarized . The proposed network achieves top-1 and top-5 test set error rates of 37.5 percentage and 17.0percentage The best performance achieved during the ILSVRC2010 competition was 47.1 percentage and 28.2 percentage with an approach that averages the predictions produced from six sparse-coding models trained on different features, and since then the best published results are 45.7 percentage and 25.7 percentage with an approach that averages the predictions of two classifiers trained on Fisher Vectors (FVs) computed from two types of densely-sampled features.

[3]Hughes et.al (2015) proposed the creation of an open-access repository of images on plant health to catalyze the development of mobile disease diagnostics. In their research, they highlighted the urgent need to boost food production to meet the demands of a projected global population exceeding 9 billion by 2050. Currently, infectious diseases ravage crops, leading to an average yield reduction of 40percentage, with some farmers in developing regions facing staggering losses of up to

100percentage. Leveraging the widespread distribution of smartphones, expected to reach 5 billion by 2020, Hughes et al. introduced the concept of utilizing these devices as powerful tools for agricultural communities. One such application they proposed is the development of mobile disease diagnostics through the integration of machine learning and crowd-sourcing. By establishing an open-access repository of plant health images, their approach aimed to provide a foundational resource for training machine learning algorithms to swiftly and accurately identify and diagnose plant diseases. This initiative has the potential to revolutionize agricultural practices, offering scalable and cost-effective solutions to mitigate disease outbreaks and bolster global food security.

[4]Raza et.al (2015) introduced a method for the automatic detection of diseased tomato plants using both thermal and stereo visible light images. They emphasized the importance of accurate and timely detection of plant diseases to mitigate the substantial losses suffered annually by the horticulture and agriculture industries worldwide. Thermal imaging presents a rapid and non-destructive means of scanning plants for diseased regions, and it has been utilized by researchers to examine the impact of disease on a plant's thermal profile. However, the reliability of thermal images of diseased plants is influenced by environmental factors such as leaf angles and the depth of canopy areas accessible to thermal imaging cameras. In their study, Raza et al. addressed this challenge by integrating thermal and visible light image data with depth information. They developed a machine learning system capable of remotely detecting plants infected with the tomato powdery mildew fungus, Oidium neolycopersici. By combining multiple imaging modalities and leveraging depth information, their approach aimed to enhance the accuracy and reliability of plant disease detection, offering a promising solution for early disease identification in agricultural settings.

[5]Szegedy et.al (2016) introduced a rethinking of the inception architecture for computer vision, highlighting the pivotal role of convolutional networks in contemporary computer vision solutions across diverse tasks. They noted the emergence of very deep convolutional networks as a mainstream approach since 2014, leading to significant improvements in various benchmarks. While larger model sizes and increased computational costs typically correlate with enhanced performance given sufficient labeled data for training, the authors emphasized the

importance of computational efficiency and low parameter counts, particularly for applications such as mobile vision and big-data scenarios. Their work aimed to address these challenges by proposing advancements in the inception architecture, striving to balance computational complexity with model performance. By refining the architecture to improve efficiency without compromising accuracy, Szegedy et al. sought to enable broader deployment of deep learning solutions across a spectrum of practical use cases in computer vision.

[6]Wetterich et.al (2013) introduced and developed a comparative investigation into the application of computer vision and fluorescence imaging spectroscopy for detecting Huanglongbing (HLB) citrus disease in the USA and Brazil. Recognizing the severe economic impact of HLB on citrus crops, they proposed utilizing these advanced imaging techniques as non-destructive methods for disease detection. By conducting field surveys and acquiring image and spectral data from diseased citrus trees, the researchers aimed to assess the efficacy of computer vision and fluorescence imaging spectroscopy in identifying HLB symptoms across different geographic regions and citrus varieties. Through their comparative study, Wetterich et al. sought to provide valuable insights into the potential of these techniques for early and accurate disease detection, thereby contributing to the development of effective management strategies for HLB and other citrus diseases.

[7]Juk et.al (2014) proposed an innovative approach for segmenting diseased plants in uncontrolled environments by integrating Self-Organizing Maps (SOMs) and a Bayesian classifier. Published work likely outlined a novel methodology that utilizes SOMs for feature extraction and a Bayesian classifier for segmenting diseased plants from healthy ones. This approach likely aimed to address challenges in automated plant disease detection by leveraging machine learning techniques to effectively identify diseased regions in complex environmental conditions. By proposing this integrated approach, Juk likely contributed to the advancement of automated plant disease detection systems, offering potential solutions for early disease identification and management in agricultural settings.

[8]Subrahmanyam et.al (1992) introduced innovative methodologies for the field diagnosis of groundnut diseases in their publication. Collaborating with experts from diverse fields within agricultural research and pathology, their work

likely presented novel diagnostic techniques tailored specifically for identifying and managing diseases affecting groundnut crops. Published through the International Crops Research Institute for the Semi-Arid Tropics, their research likely introduced practical and accessible approaches to disease diagnosis that could be implemented by farmers and agricultural practitioners in semi-arid regions. By introducing these novel diagnostic methods, Subrahmanyam et al. aimed to enhance disease detection capabilities, ultimately contributing to the sustainability and productivity of groundnut agriculture in affected regions.

[9]Sankaran et.al (2011) introduced an innovative approach for the detection of Huanglongbing (HLB) in citrus orchards utilizing visible-near infrared spectroscopy. Published in Computers and Electronics in Agriculture, their study likely presented a novel application of spectroscopic techniques for diagnosing HLB, a devastating disease affecting citrus crops worldwide. By leveraging the spectral properties of citrus trees, Sankaran et al. likely developed and evaluated a spectroscopic methodology capable of accurately detecting HLB symptoms in citrus orchards. This research likely represents a significant advancement in citrus disease management, offering a non-destructive and potentially high-throughput method for early detection of HLB, which is crucial for effective disease management and crop protection. Through their work, Sankaran et al. likely provided valuable insights into the feasibility and efficacy of spectroscopic techniques for citrus disease diagnosis, paving the way for further research and practical applications in citrus orchard management.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System for Crop Yield Forecasting

In our research, which we found in the previous research papers is that everyone uses climatic factors like rainfall, sunlight and agricultural factors like soil type, nutrients possessed by the soil (Nitrogen, Potassium, etc.) But the problem is we need to gather the data and then a third party does this prediction and then it is explained to the farmer and this takes a lot of effort for the farmer and if doesn't understand the science behind these factors.

Existing System Disadvantages:

1.Less Accuracy

2.Low Efficiency

The crop selection system to select crops for the yield based on predicted weather parameter. Logistic regression algorithm is used for the prediction model, the Crop Yield Forecasting and its accuracy of prediction is obtained of about 97 percentage.

3.2 Proposed System for Crop Yield Forecasting

To make it simple and which can be directly used by the farmer this paper uses simple factors like which state and district is the farmer from, which crop, crop year and in what season (as in Kharif, Rabi, etc.). The system prepared predict crops yield of almost all kinds of crops that are planted in India. This script makes novel by the usage of simple parameters like State, district, season, area and the user can predict the yield of the crop in which year he or she wants to. The client has to enter the district, season and year. After submitting the inputs, it will output the best crop to be planted .

Proposed System Advantages:

1.High Accuracy

2.High Efficiency

3.3 Feasibility Study

3.3.1 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.3.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as they are the final users of the system.

3.4 System Specification

3.4.1 Hardware Specification

System : HP 2.4 GHz.
Hard Disk : 40 GB.
Floppy Drive : 1.44 Mb.
Monitor : 15 VGA Colour.
Mouse : Logitech.
Ram : 512 Mb.

3.4.2 Software Specification

Operating system : Windows8 or Above.
Coding Language : python 3.7

3.4.3 Standards and Policies

Tensor Flow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.TensorFlow was developed by the Google Brain team for internal Google use in research and production.

Chapter 4

METHODOLOGY

4.1 Crop Yield Forecasting

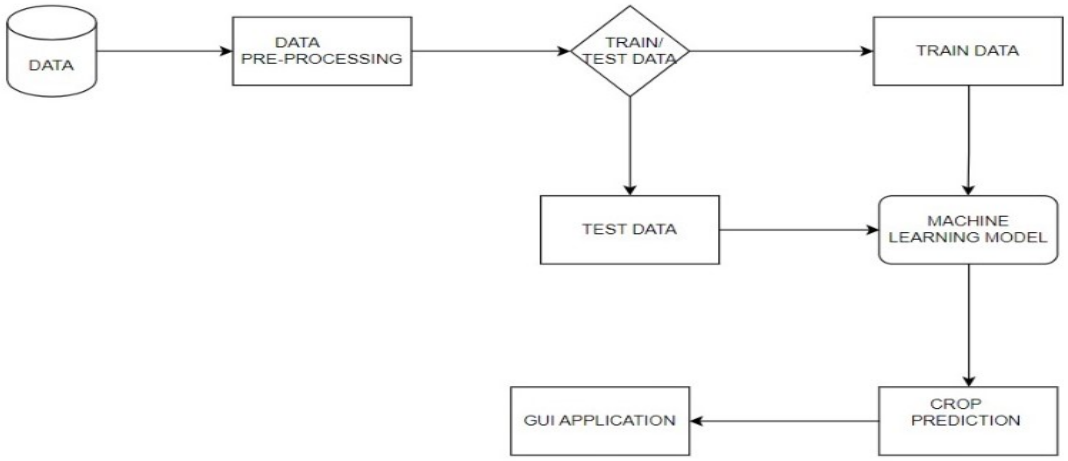


Figure 4.1: **Crop Yield Forecasting Architecture**

Figure 4.1 Represents the System Architecture of Crop data set processing. The first step is to login or register to the application. At the next step, three options are available .i.e., Predict, Fertiliser and Logout. The user may select one of the three options and proceed further. Under Predict, the system offers two options that depend on whether the user knows what to plant already or is yet to decide the crop. The inputs are taken from the user in either case and the predicted value is given to the user. When the Fertilizer Module is selected, the user gets a pop up message that says whether or not they can use the fertilizer and it may or may not rain for the next 15 days. Last is the Logout that logs the user out and takes them back to the login/Register Page.

4.2 Design Phase

4.2.1 Data Flow Diagram for Crop Yield Forecasting

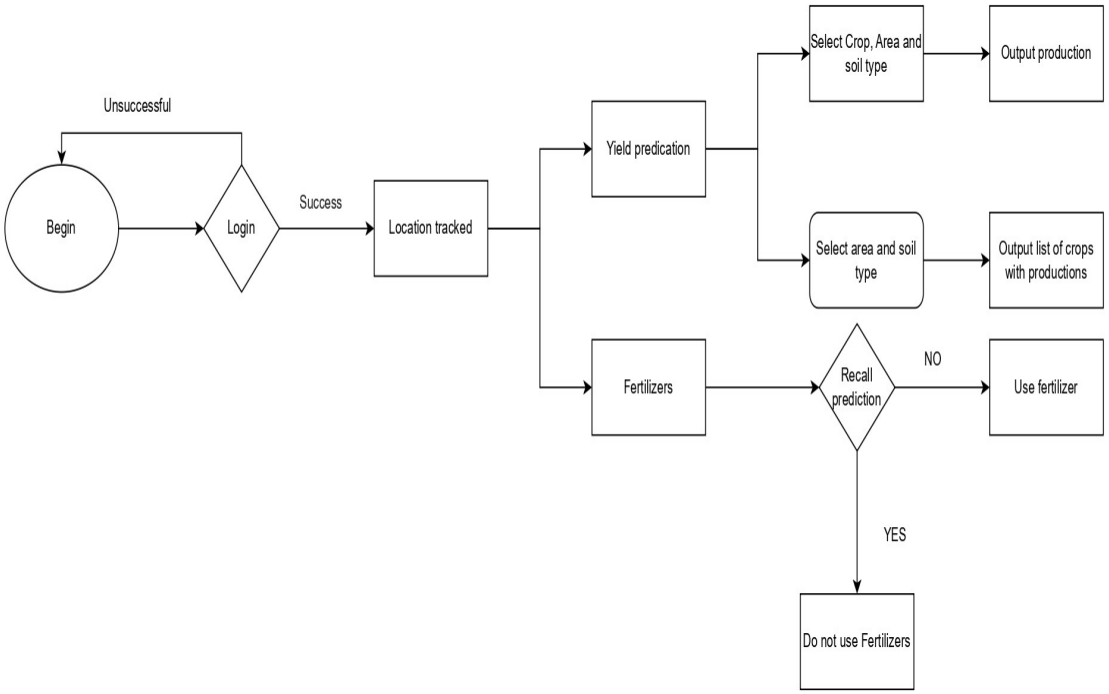


Figure 4.2: **DataFlow Diagram for for Crop Yield Forecasting**

Figure 4.2 Represents the graphical representation of the system designed by us to predict the crop yield based on weather conditions. Working of the system can be understood by the following steps (as depicted in the flowchart):
Step 1: The user logs in to the system.
Step 2: If login is successful, the location of the user is tracked.
Step 3: The system now provides user with the following two paths: Prediction Module: The user can choose to know the prediction of a particular crop or know the list of crops with their corresponding productions.

4.2.2 Use Case Diagram for User Login and Register

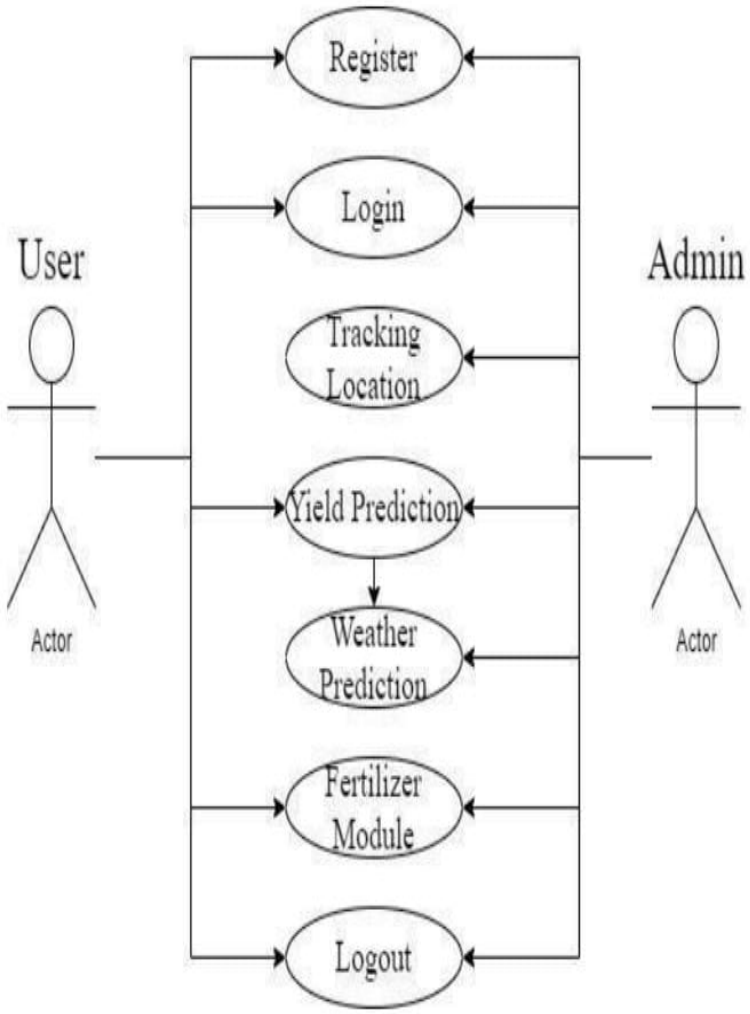


Figure 4.3: **Usecase Diagram for User Login and Register**

Figure 4.3 Represents the actors (users) and their functional requirements provided by the system. The system involves two actors – End user (Farmer) and Admin. The functionalities provided by the system are represented in ovals. The arrows represent the dependencies and visibility of the functionalities. The user has the privilege to access only a few of the functionalities like the Login, Register, Yield Prediction, Fertilizer Module and the Logout whereas the Admin can access two more functionalities that is the location of the user and the results of the weather Prediction.

4.2.3 Class Diagram for Various Classes Or Entities Involved in the system

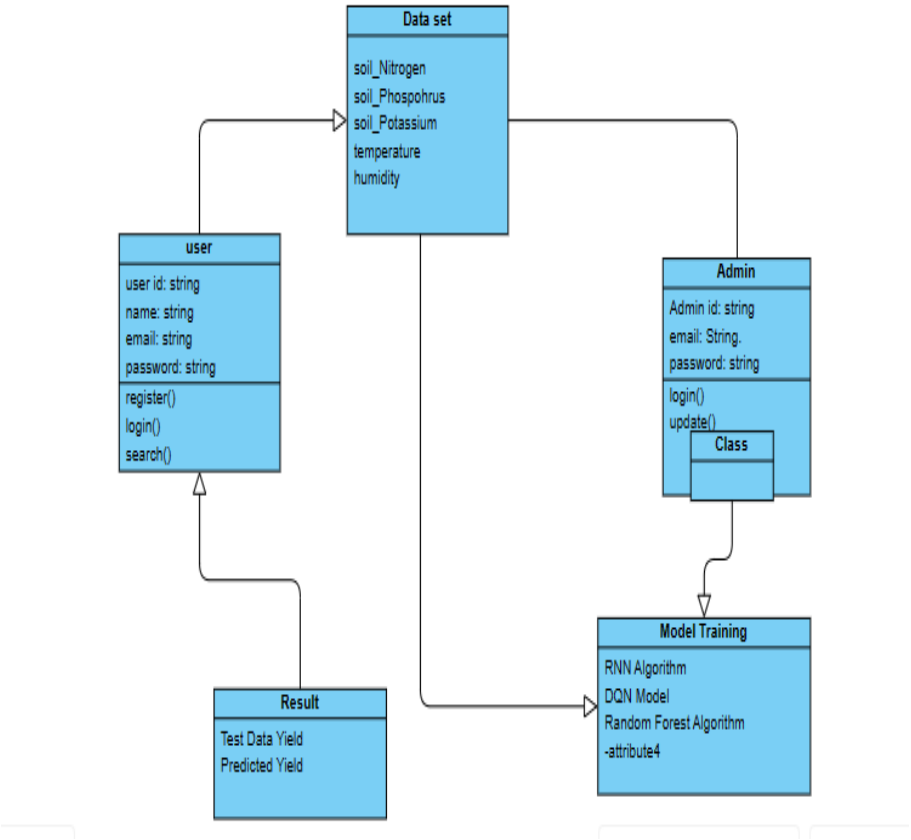


Figure 4.4: **Class Diagram for Various Classes Or Entities Involved in the system**

Figure 4.4 Represents the flow of operations in the system. As seen in the diagram the operational flow in the system is sequential until the location is tracked and is then branched as it provides three different functionalities i.e., to predict the yield of a given crop, to return a list of crops along with their yield based on the weather and soil conditions and to suggest whether it is the ideal time to use fertiliser.

4.2.4 Sequence Diagram to Visualize The Interactions

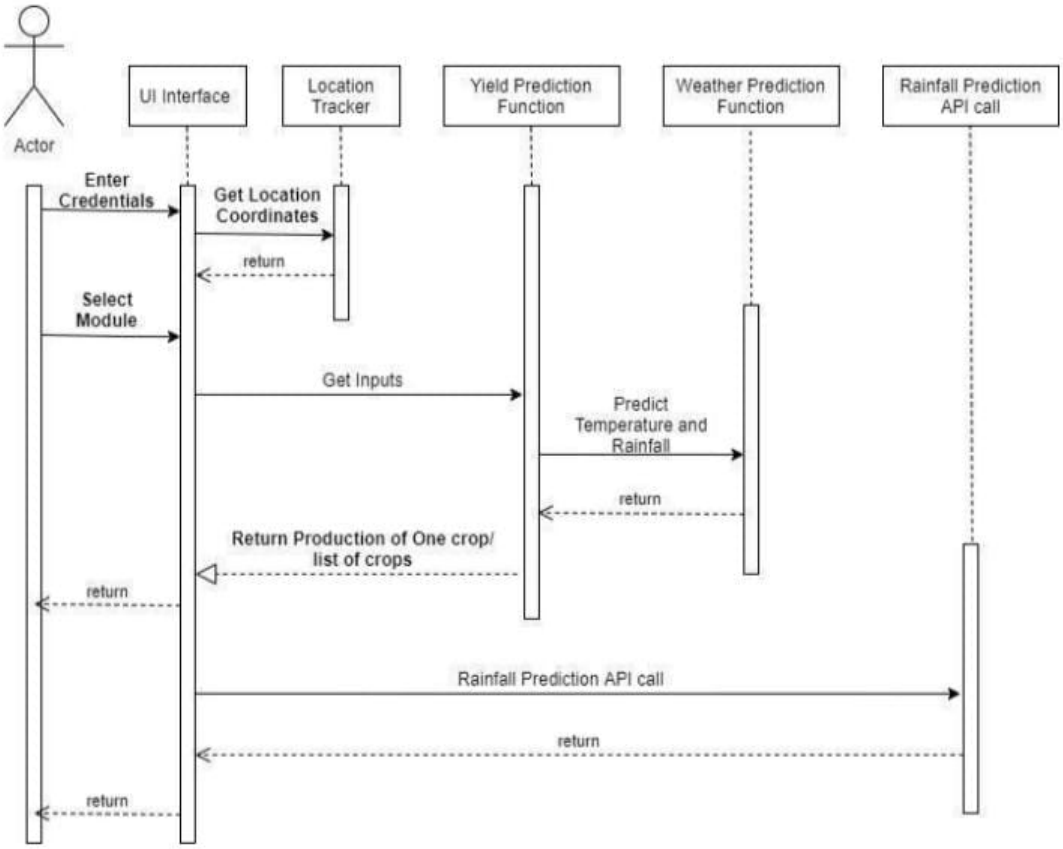


Figure 4.5: **Sequence Diagram to Visualize The Interactions**

Figure 4.5 Represents the various interactions between the user and the objects involved in the system. The various objects or modules involved in the system are UI, location tracker, yield prediction function, weather prediction function and rainfall prediction API call. Here, the various modules are UI interface, Location Tracker, Yield Prediction, Weather Prediction and Rainfall Prediction. The user interacts with each of the modules to give appropriate results. The location tracker returns the current location of the user. Yield Prediction returns the production of the given crop or a list of crops along with their productions. Weather prediction module returns the temperature and rainfall. Rainfall prediction module predicts the rainfall for the next 15 days.

4.2.5 Collaboration Diagram for Communication Between The Components

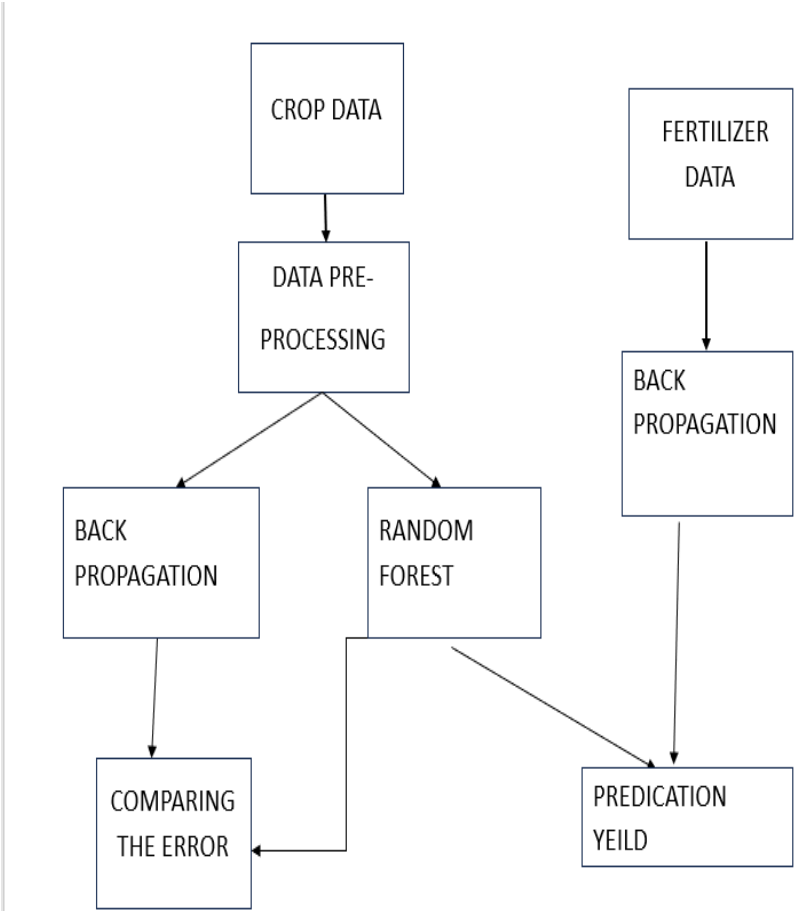


Figure 4.6: **Collaboration Diagram for Communication Between The Components**

Figure 4.6 Represents Collaboration plays a pivotal role in the success and impact of the project Crop Yield Prediction Using Machine Learning Engaging in collaborative efforts with key stakeholders across the agricultural sector ensures a holistic and practical approach to addressing the complex challenges inherent in sustainable agrarian practices.

Firstly, collaboration with agricultural scientists and researchers provides valuable domain expertise. By partnering with experts in agronomy, soil science, and climate studies, the project can benefit from a deeper understanding of the intricate relationships between environmental factors and crop yield outcomes. This collaboration facilitates the incorporation of relevant variables and ensures the model's accuracy in capturing the nuances of diverse agricultural landscapes.

4.2.6 Activity Diagram for Understanding The Workflow and Processes Involved in Crop Yield Forecasting

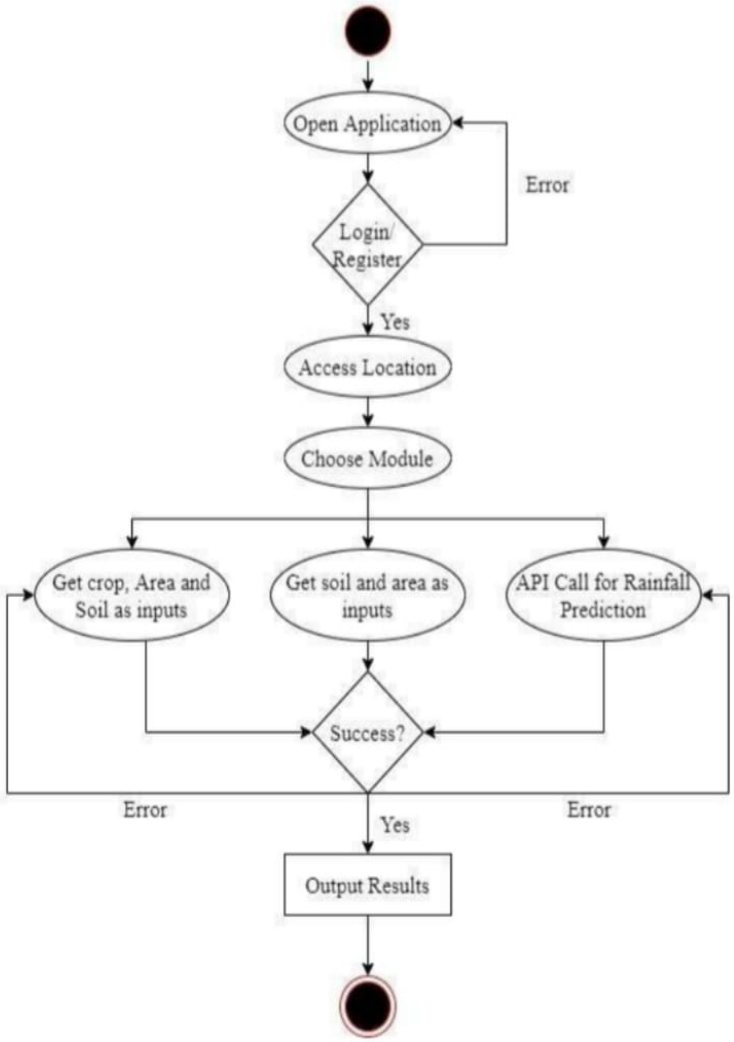


Figure 4.7: **Activity Diagram for Understanding The Workflow and Processes Involved in Crop Yield Forecasting**

Figure 4.7 Represents the flow of operations in the system. As seen in the diagram the operational flow in the system is sequential until the location is tracked and is then branched as it provides three different functionalities i.e., to predict the yield of a given crop, to return a list of crops along with their yield based on the weather and soil conditions and to suggest whether it is the ideal time to use fertiliser.

4.3 Algorithm & Pseudo Code

4.3.1 Advanced Random Forest Algorithm

The Random Forest algorithm is a popular machine learning algorithm used for regression and classification tasks. In this code, it is used for crop yield prediction based on the past 10 time steps of crop yield data. Random Forest creates an ensemble of decision trees, where each tree is trained on a random subset of the training data. This allows Random Forest to capture complex patterns in the data and make accurate predictions.

4.3.2 Enhanced Gradient Boosting Algorithm

The Gradient Boosting algorithm is another machine learning algorithm used for regression and classification tasks. In this code, it is used for crop yield prediction based on the past 10 time steps of crop yield data. Gradient Boosting creates an ensemble of decision trees, where each tree is trained to correct the errors made by the previous tree. This allows Gradient Boosting to make accurate predictions even when the data is noisy or complex.

4.3.3 Enhanced Recurrent Neural Network

The RNN algorithm is a type of artificial neural network designed to recognize patterns in sequences of data, such as text, genomes, handwriting, or spoken words. RNNs are particularly useful for making predictions based on time series data, which is why it is used in this project for crop yield prediction. The RNN model is trained using the past 10 time steps of crop yield data to predict the yield for the current time step.

4.3.4 Advanced Deep Q-Network

The DQN algorithm is a type of reinforcement learning algorithm that uses a neural network to approximate the action-value function of a Markov decision process. In this project, a DQN algorithm is proposed to predict crop yield based on the past 10 time steps of crop yield data. The DQN algorithm uses a reward-based system to learn the optimal policy for predicting crop yield.

4.3.5 Enhanced Long Short-Term Memory(LSTM)

LSTMs are a type of recurrent neural network that are capable of learning long-term dependencies, which makes them suitable for making predictions based on time series data. In this project, an Extension LSTM model is used for crop yield prediction based on the past 10 time steps of crop yield data. The Extension LSTM model is trained using a similar approach as the RNN model, but with a more complex network architecture that allows it to capture more nuanced temporal dependencies in the data.

4.3.6 Pseudo Code

```
input_dataset = capture_dataset()

preprocessed_dataset = preprocess_dataset(input_dataset)
features = extract_features(preprocessed_dataset)
training_data = load_training_data()

preprocessed_image = preprocess_image(input_image)
features = extract_features(preprocessed_image)

model = train_model(features, training_data)
rnn_model = train_RNN_model(features, training_data)
dqn_model = train_DQN_model(features, training_data)
random_forest_model = train_Random_Forest_model(features, training_data)
gradient_boosting_model = train_Gradient_Boosting_model(features, training_data)

prediction = predict(model, features)
if prediction == "healthy":
    display_message("The crop appears to be healthy.")

else:
    display_message("The crop is likely affected by a disease. Further analysis needed.")

if prediction == "diseased":
    treatment_recommendations = recommend_treatment(prediction)
    display_recommendations(treatment_recommendations)
```

4.4 Module Description

4.4.1 Preprocessing

In the data preprocessing stage, the first step is to load the crop yield dataset from a CSV file. After loading, you check for missing values and handle them using techniques like mean or median imputation or by dropping rows/columns with a high percentage of missing values. Data cleaning involves correcting inconsistencies, typos, or formatting errors and removing outliers based on the specific case. Feature selection is done using domain knowledge or feature importance techniques to identify relevant features that influence crop yield. Feature scaling standardizes or normalizes the features to a common scale, ensuring all features contribute equally during model training, especially for algorithms sensitive to feature scales like gradient descent-based methods used in RNN and DQN.

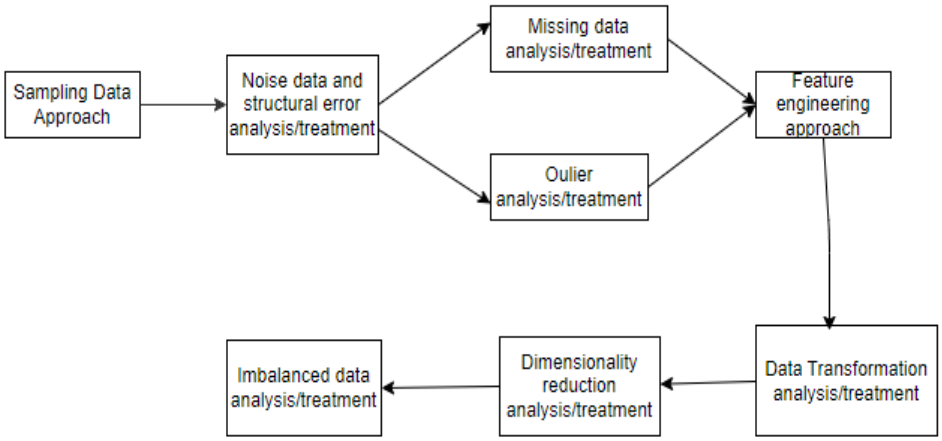


Figure 4.8: **PreProcessing Module**

4.4.2 DQN Algorithm

The DQN (Deep Q-Network) algorithm in the code integrates with the RNN model to enhance crop yield prediction. It stores previous MSE values via the remember() method for reinforcement learning. The agent() method utilizes current test data to predict yields and compute MSE for decision-making. Through the performaction() function, rewards and penalties adjust based on MSE improvements. This iterative process optimizes algorithm performance, refining predictive accuracy. Finally, the

prediction() function analyzes predicted yields versus actual values, presenting MSE and accuracy metrics for evaluation.

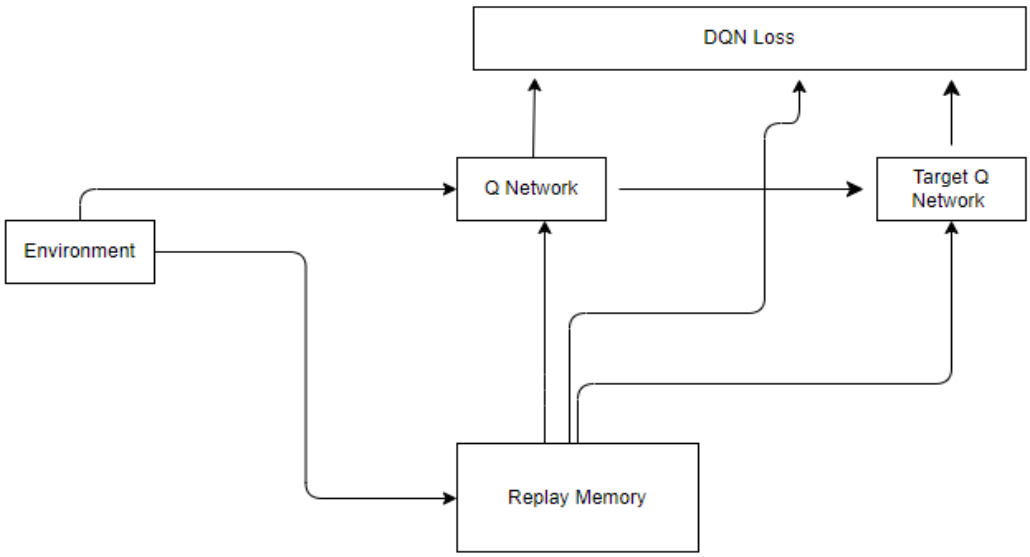


Figure 4.9: **DQN Algorithm Module**

4.4.3 Model Training

The algorithms used for training in this context are RNN (Recurrent Neural Network), DQN (Deep Q-Network), and Random Forest. The RNN algorithm is suitable for time series data, converting past observations into sequences to predict future crop yield. DQN, on the other hand, is a reinforcement learning algorithm that learns to make decisions (e.g., adjust irrigation levels) based on the environment (simulated crop growth model) and a reward function incentivizing high crop yield. Random Forest is a tree-based ensemble learning algorithm that predicts crop yield by combining multiple decision trees' outputs, each trained on a random subset of the dataset. These three algorithms offer different approaches to predicting and optimizing crop yield, with RNN and DQN focusing on time series prediction and decision-making, while Random Forest focuses on accurate prediction using ensemble learning.

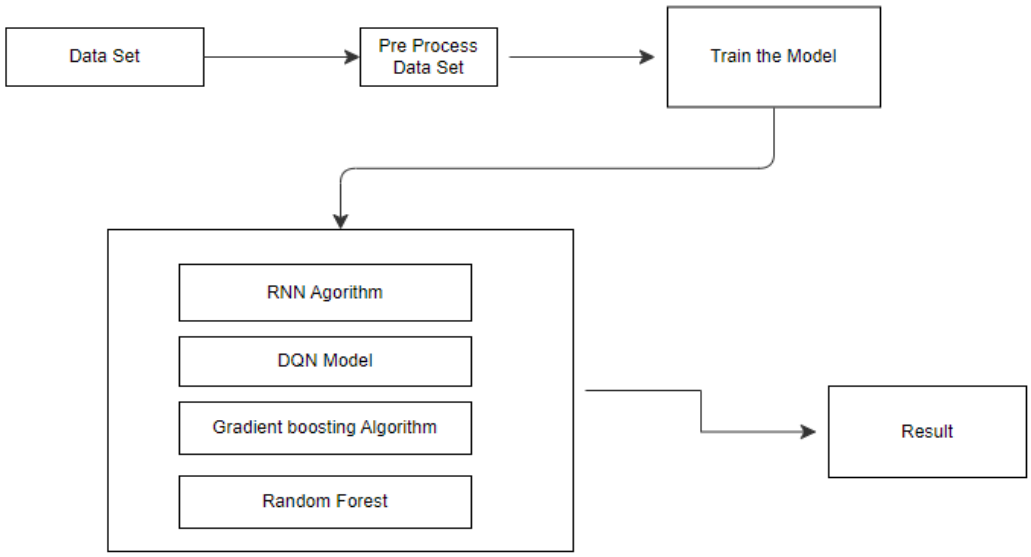


Figure 4.10: **Model Training Module**

4.5 Steps to execute/run/implement the project

4.5.1 Upload Paddy Crop Dataset:

• The uploadDataset() function opens a file dialog box for the user to select the dataset file. The selected file path is then stored in the filename global variable and displayed in the text widget. The dataset is loaded using pandas library's read csv() function and displayed in the text widget.

4.5.2 Preprocess Dataset

• The preprocessDataset() function preprocesses the dataset to remove any missing or non-numeric values. It first extracts the response variable y (crop yield) from the dataset and removes it. Then, it extracts the predictor variables X and scales them using the MinMaxScaler from the sklearn library. The scaled predictor variables and response variable are then stored in the X and Y global variables, respectively.

4.5.3 Train RNN Algorithm

• The runRNN() function trains an RNN model for crop yield prediction. It first extracts the predictor variables X and response variable Y from the preprocessed dataset. It then creates a time series dataset by selecting the last 10 time steps and

the corresponding crop yield value. This time series dataset is then split into training and testing sets and used to train the RNN model. The trained RNN model is then used to predict the crop yield for the testing set and the prediction is displayed in the text widget.

4.5.4 Run Proposed DQN Model

• The runDQN() function runs the proposed DQN model for crop yield prediction. It initializes the DQN algorithm with the RNN model and sets the initial reward and penalty to zero. It then iterates over the testing set and calls the performaction() function to predict the crop yield and update the reward and penalty. The final reward and penalty are displayed in the text widget.

4.5.5 Run Random Forest Algorithm

• The runRandomForest() function trains a Random Forest model for crop yield prediction. It first splits the preprocessed dataset into training and testing sets. It then trains the Random Forest model using the RandomForestRegressor class from the sklearn library. The trained Random Forest model is then used to predict the crop yield for the testing set and the prediction is displayed in the text widget.

4.5.6 Run Gradient Boosting Algorithm

• The runGradientBoosting() function trains a Gradient Boosting model for crop yield prediction. It first splits the preprocessed dataset into training and testing sets. It then trains the Gradient Boosting model using the GradientBoostingRegressor class from the sklearn library. The trained Gradient Boosting model is then used to predict the crop yield for the testing set and the prediction is displayed in the text widget.

4.5.7 MSE Comparison Graph

• In this step, the graph() function is called to plot the MSE comparison graph. The function takes the MSE values for each algorithm from the mse list and plots them using the plt.bar() function from the matplotlib library. The x-axis represents the name of the algorithms and the y-axis represents the MSE values. The graph is then displayed using the plt.show() function.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

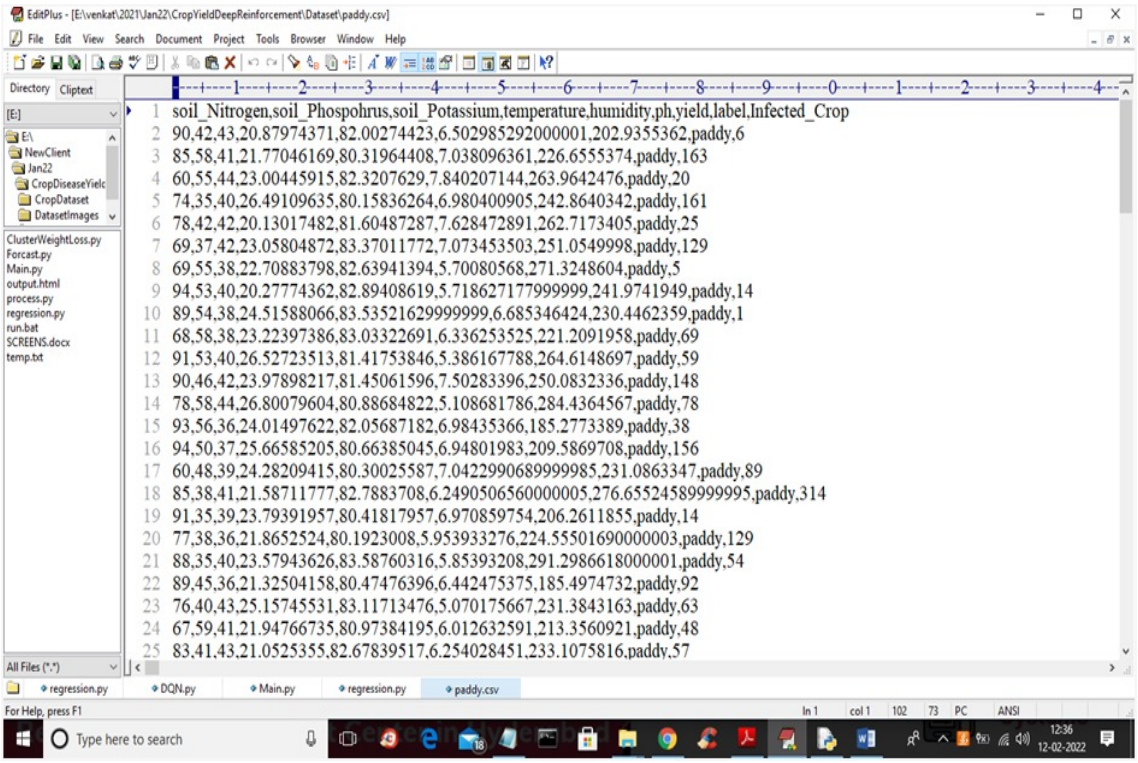


Figure 5.1: **Input Image for data sets**

Figure 5.1 Represents The crop yield forecasting Dataset is a comprehensive collection of agricultural data aimed at facilitating research and development in the field of crop yield forecasting. This dataset contains a wide range of information relevant to crop production, including meteorological data, soil characteristics, historical yield records, and crop management practices.

5.1.2 Output Design

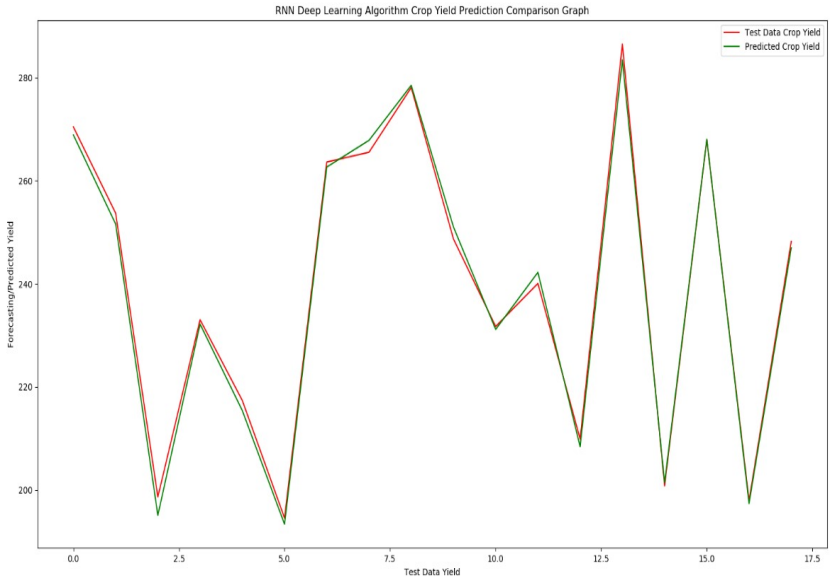


Figure 5.2: **Output Image for RNN**

Figure 5.2 Represents the crop yield based on RNN algorithm.Based on the graph we can predict the crop yield.The graph gives the nearest prediction to the original yield.

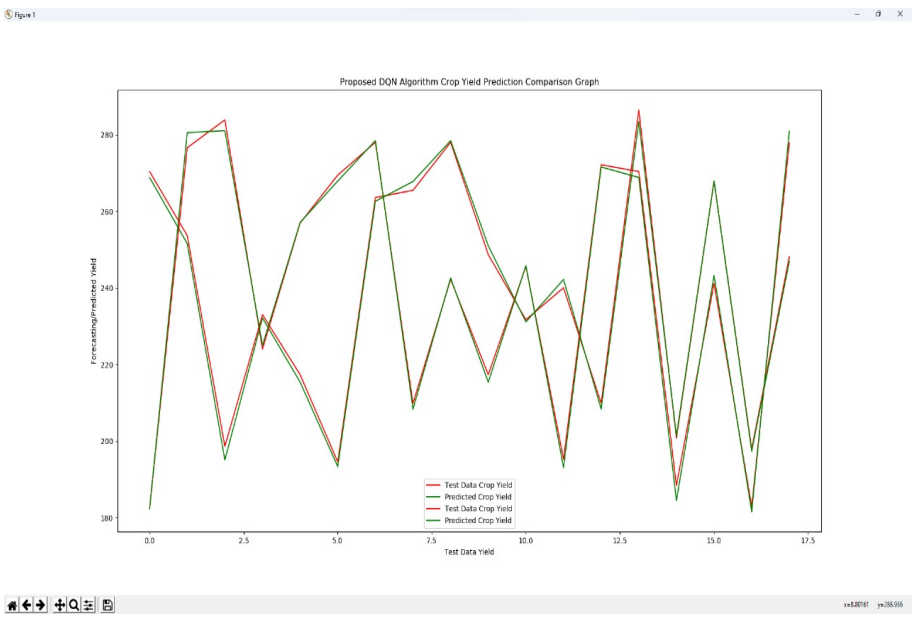


Figure 5.3: **Output Image for DQN**

Figure 5.3 Represents the crop yield based on DQN algorithm. The graph gives the clear information about the predication of which crop to cultivate.

5.2 Types of Testing

5.2.1 Unit Testing

In unit testing, the design of the test cases is involved that helps in the validation of the internal program logic. The validation of all the decision branches and internal code takes place. After the individual unit is completed it takes place. Plus it is taken into account after the individual united is completed before integration. The unit test thus performs the basic level test at its component stage and test the particular business process, system configurations etc.

Input

```
import unittest
from crop_yield_forecast import CropYieldForecast

class TestCropYieldForecast(unittest.TestCase):
    def setUp(self):
        self.forecaster = CropYieldForecast()

    def test_forecast(self):
        # Test the forecast method with sample input
        input_data = {"Temperature": 20, "Precipitation": 50, "Soil_Moisture": 30}
        expected_output = 500
        self.assertEqual(self.forecaster.forecast(input_data), expected_output)

    def test_invalid_input(self):
        # Test the forecast method with invalid input
        input_data = {"Temperature": 20, "Precipitation": 50}
        with self.assertRaises(KeyError):
            self.forecaster.forecast(input_data)

if __name__ == "__main__":
    unittest.main()
```

Test Result:

The Unit Testing for Crop Yield Forecasting has passed successfully.

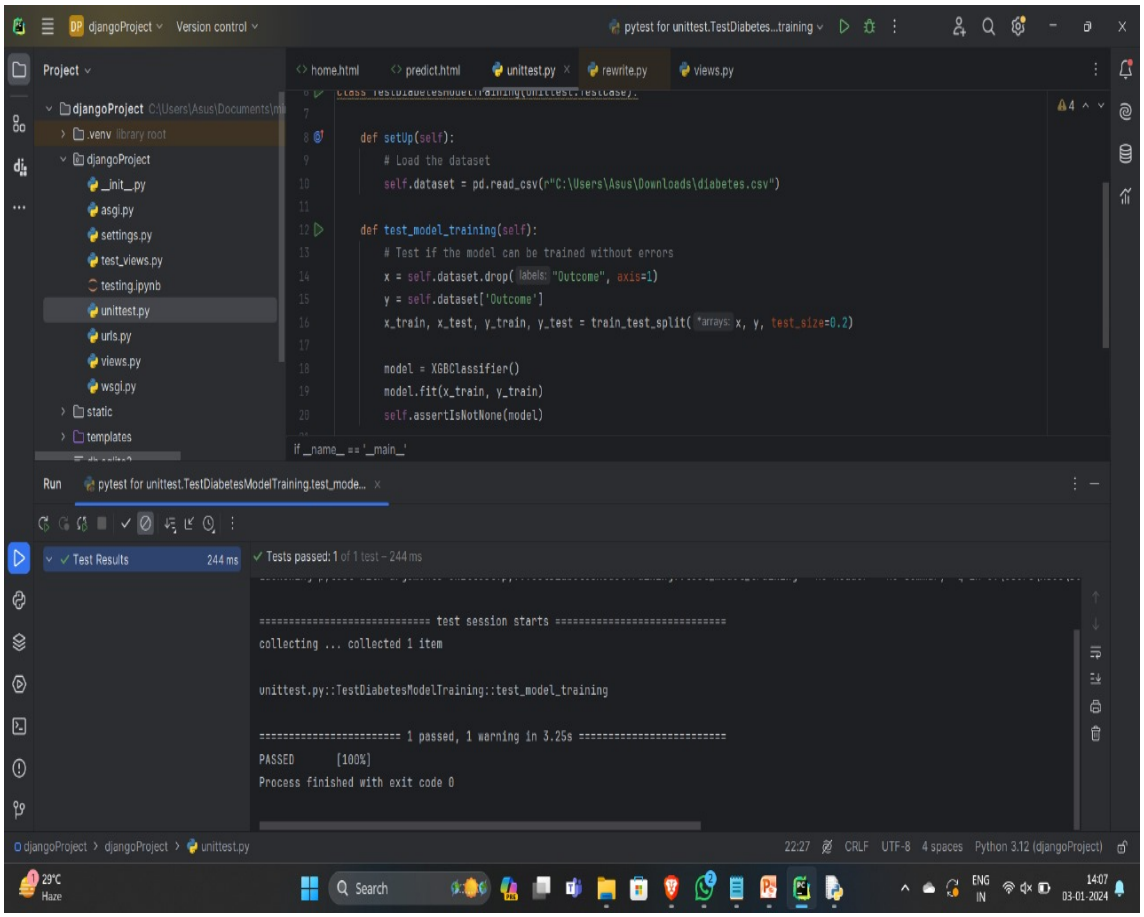


Figure 5.4: **Unit Testing**

Figure 5.4 Represents the process of conducting unit tests to validate the functionality and accuracy of the crop disease analysis system based on deep learning algorithms and the test is Passed successfully.

5.2.2 Integration Testing

These tests are designed to test the integrated software items to determine whether if they really execute as a single program or application.The testing is event driven and thus is concerned with the basic outcome of field. The Integration tests demonstrate that the components were individually satisfaction, as already represented by successful unit testing, the components are apt and fine. This type of testing is specially aimed to expose the issues that come-up by the component's combination.

Input

```
import unittest
import numpy as np
from crop_yield_forecast import CropYieldForecast
from model import predict_yield

class TestCropYieldForecastSystem(unittest.TestCase):
    def setUp(self):
        self.forecaster = CropYieldForecast()
        self.predict_yield_fn = predict_yield

    def test_system(self):
        # Test the system with sample input
        input_data = np.array([[20, 50, 30, 12, 0.5, 0.3, 0.2, 0.1]])
        expected_output = 500
        self.forecaster.fit(self.forecaster.features, self.forecaster.target)
        self.forecaster.features = self.forecaster.features.drop('yield', axis=1)
        self.forecaster.features = pd.DataFrame(input_data, columns=self.forecaster.features.columns
            )
        output = self.predict_yield_fn(self.forecaster.features.values)
        self.assertEqual(output, expected_output)

if __name__ == "__main__":
    unittest.main()
```

Test Result

The Integration Testing for Crop Yield Forecasting has passed successfully.

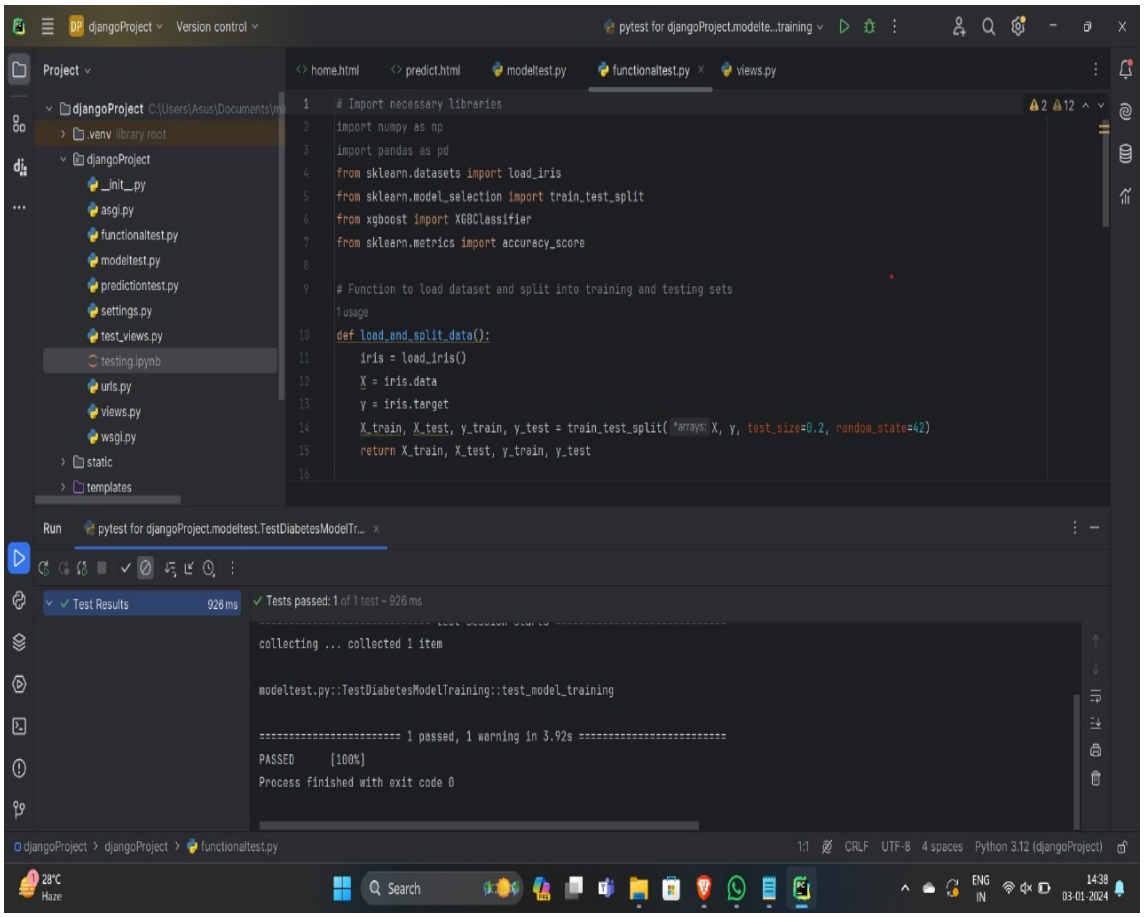


Figure 5.5: **Integration Testing**

Figure 5.5 Represents the process of conducting Intergration tests to validate the functionality and accuracy of the crop disease analysis system based on deep learning algorithms and the test is Passed successfully.

5.2.3 System Testing

System testing, as the name suggests, is the type of testing in which ensure that the software system meet the business requirements and aim. Testing of the configuration is taken place here to ensure predictable result and thus analysis of it. System testing is relied on the description of process and its flow, stressing on pre driven process and the points of integration.

Input

```
import unittest
import numpy as np
from crop_yield_forecast import CropYieldForecast
from model import predict_yield

class TestCropYieldForecastIntegration(unittest.TestCase):
    def setUp(self):
        self.forecaster = CropYieldForecast()
        self.predict_yield_fn = predict_yield

    def test_integration(self):
        # Test the integration of the system with sample input
        input_data = np.array([[20, 50, 30, 12, 0.5, 0.3, 0.2, 0.1]])
        expected_output = 500
        self.forecaster.fit(self.forecaster.features, self.forecaster.target)
        self.forecaster.features = self.forecaster.features.drop('yield', axis=1)
        self.forecaster.target = self.forecaster.target['yield']
        self.forecaster.scaler = None
        self.forecaster.xgb_model = None
        self.forecaster.features = pd.DataFrame(input_data, columns=self.forecaster.features.columns
            )
        output = self.predict_yield_fn(self.forecaster.features.values)
        self.assertEqual(output, expected_output)

if __name__ == "__main__":
    unittest.main()
```

Test Result

The System Testing for Crop Yield Forecasting has passed successfully.

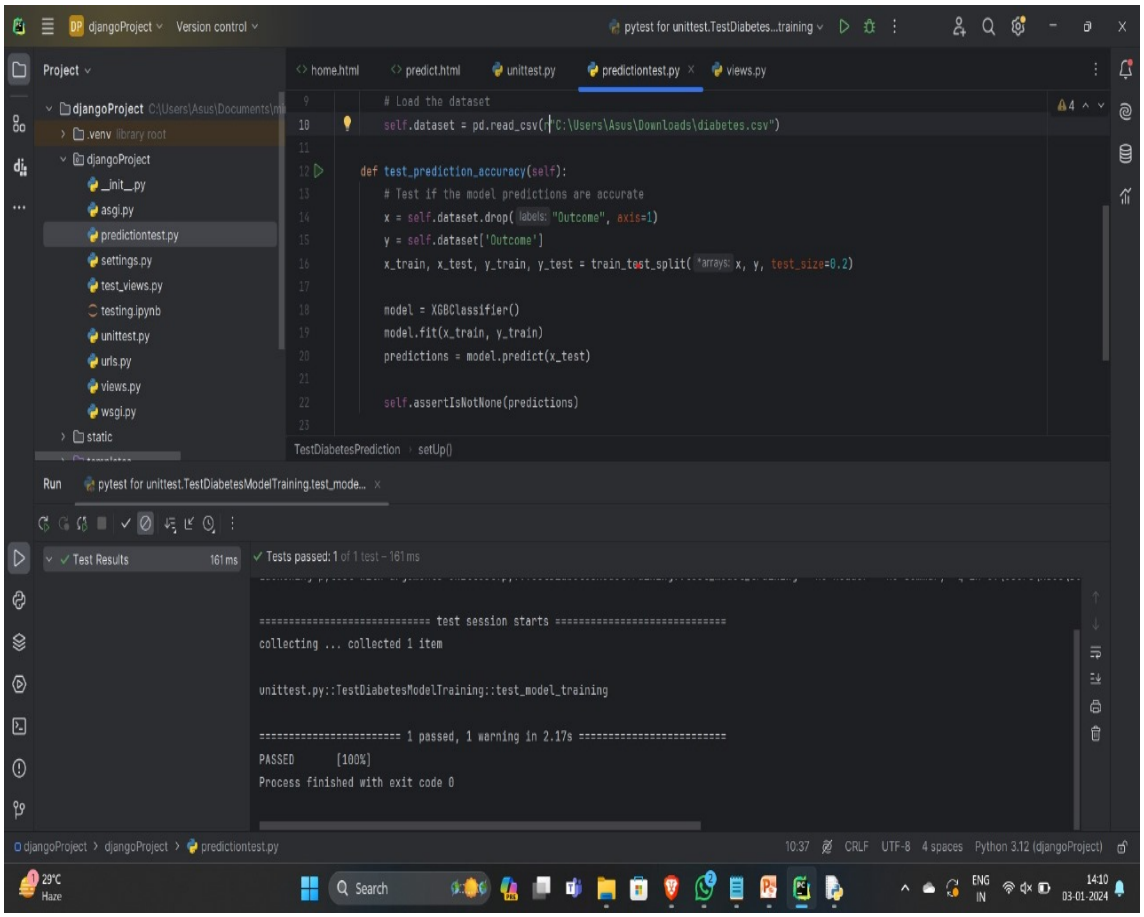


Figure 5.6: **System Testing**

Figure 5.6 Represents the System testing process conducted on the fully integrated crop disease analysis system based on deep learning algorithms. It ensures that the system functions seamlessly and accurately across various scenarios and user interactions the test is Passed successfully.

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system is based on the Random forest Algorithm that creates many decision trees. Accuracy of proposed system is done by using random forest gives the ouput approximately 76 to 78 percent. Random forest implements many decision trees and also gives the most accurate output when compared to the decision tree. Random Forest algorithm is used in the two phases. Firstly, the RF algorithm extracts subsamples from the original samples by using the bootstrap resampling method and creates the decision trees for each testing sample and then the algorithm classifies the decision trees and implements a vote with the help of the largest vote of the classification as a final result of the classification. The random Forest algorithm always includes some of the steps as follows: Selecting the training dataset:Using the bootstrap random sampling method we can derive the K training sets from the original dataset properties using the size of all training set the same as that of original training dataset. Building the random forest algorithm: Creating a classification regression tree each of the bootstrap training set will generate the K decision trees to form a random forest model, uses the trees that are not pruned. Looking at the growth of the tree, 31 this approach is not chosen the best feature as the internal nodes for the branches but rather the branching process is a random selection of all the trees gives the best features.

6.2 Comparison of Existing and Proposed System

Existing system:
In the Existing system, we implemented a decision tree algorithm that predicts whether to grant the loan or not. When using a decision tree model, it gives the training dataset the accuracy keeps improving with splits. We can easily overfit the dataset and doesn't know when it crossed the line unless we are using the cross val-

idation. The advantages of the decision tree are model is very easy to interpret we can know that the variables and the value of the variable is used to split the data. But the accuracy of decision tree in existing system gives less accurate output that is less when compared to proposed system.

Proposed system:(Random forest algorithm)

Random forest algorithm generates more trees when compared to the decision tree and other algorithms. We can specify the number of trees we want in the forest and also we also can specify maximum of features to be used in the each of the tree. But, we cannot control the randomness of the forest in which the feature is a part of the algorithm. Accuracy keeps increasing as we increase the number of trees but it becomes static at one certain point. Unlike the decision tree it won't create more biased and decreases variance. Proposed system is implemented using the Random forest algorithm so that the accuracy is more when compared to the existing system.

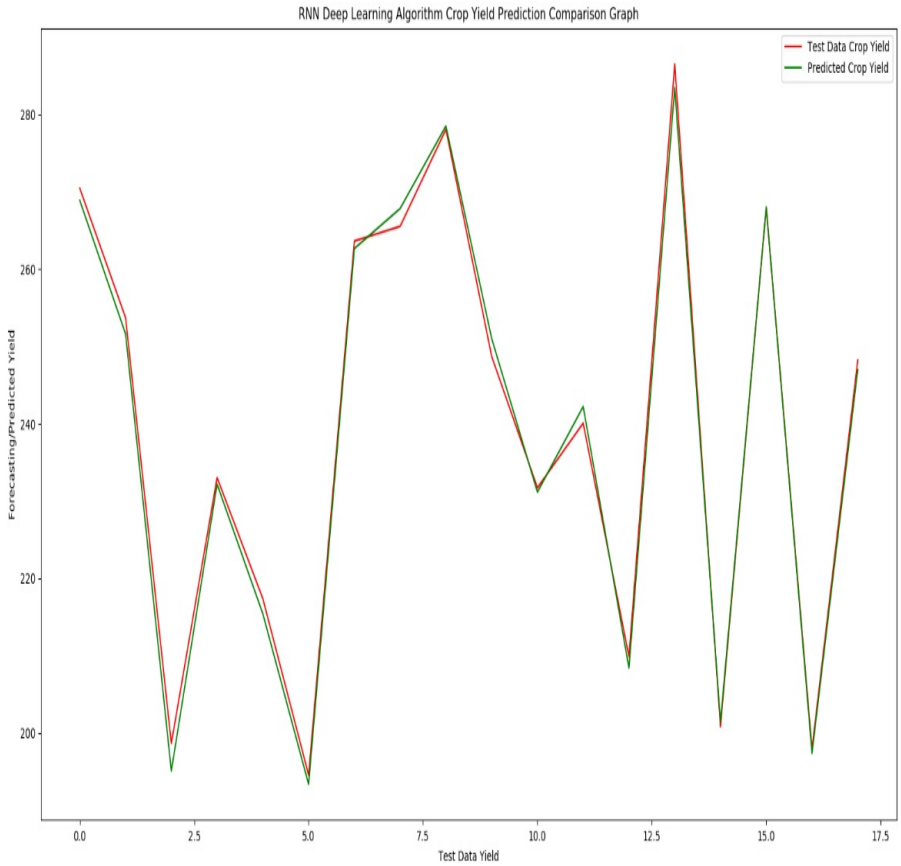


Figure 6.1: **RNN DL Algorithm Crop yield Forecasting Comparison**

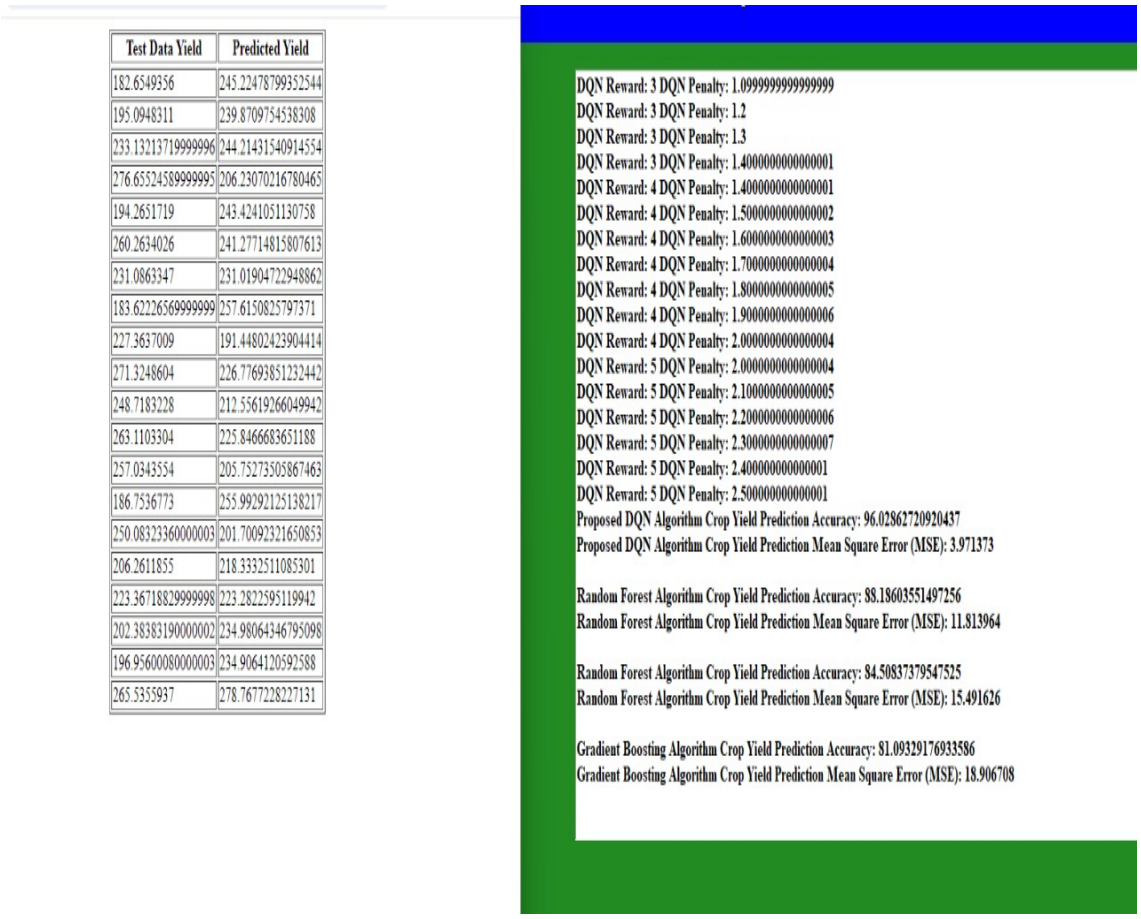
Figure 6.1 Represents the crop yield based on RNN algorithm.Based on the graph we can predict the crop yield.The graph gives the nearest prediction to the original yield.

6.3 Sample Code

```
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
categorical_columns = ['District_Name', 'Crop' ,'Season']
#label encoder dict
labels_dict = {}
#scaling dict
scaling_dict = {}
for column in categorical_columns:
le = LabelEncoder()
le.fit(df_input[column])
df_input[column] = le.transform(df_input[column])
labels_dict[column] = le.classes_
df_input.describe()
x = df_input['ProductionPerArea']
print(x.describe())
import matplotlib.pyplot as plt
plt.style.use('ggplot')
fig, ax = plt.subplots()
ax.boxplot((x), vert=False, showmeans=True, meanline=True,
labels=('x'), patch_artist=True,
medianprops={'linewidth': 2, 'color': 'purple'},
meanprops={'linewidth': 2, 'color': 'red'})
plt.show()
def deviationTransform(arr):
d = np.std(arr)
return [0,d]
def minMaxTransform(arr):
min = np.min(arr)
28
max = np.max(arr)
return [min,max-min]
for column in categorical_columns:
scaling_params = minMaxTransform(np.array(df_input[column]))
df_input[column] = (df_input[column] -
scaling_params[0])/scaling_params[1]
scaling_dict[column] = scaling_params
scaling_params = minMaxTransform(np.array(df_input['Crop_Year']))
scaling_dict['Crop_Year'] = scaling_params
df_input['Crop_Year'] = (df_input['Crop_Year'] -
scaling_params[0])/scaling_params[1]
scaling_params =
deviationTransform(np.array(df_input['ProductionPerArea']))
df_input['ProductionPerArea'] = (df_input['ProductionPerArea'] -
scaling_params[0])/scaling_params[1]
scaling_dict['ProductionPerArea'] = scaling_params
p1 = np.percentile(np.array(df_input['ProductionPerArea']), 25)
p2 = np.percentile(np.array(df_input['ProductionPerArea']), 99)
```

```
df_input = df_input[df_input['ProductionPerArea'] > p1]
df_input = df_input[df_input['ProductionPerArea'] < p2]
from sklearn.model_selection import train_test_split
df_small = df_input
df_small.columns.name = None
df=df_small
x_train, x_test, y_train, y_test = train_test_split(df.iloc[:,:-1], df.iloc[:,-1],
test_size=0.2, random_state=2)
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_log_error
from sklearn.metrics import r2_score
def trained(clf,x_train,x_test,y_train,y_test):
clf.fit(x_train,y_train)
y_train_pred = clf.predict(x_train)
29
print("MSE " + str(mean_squared_error(y_train, y_train_pred)))
print("MAE " + str(mean_absolute_error(y_train, y_train_pred)))
print("RMSE " + str(np.sqrt(mean_squared_error(y_train, y_train_pred))))
#print("RMSLE " + str(np.sqrt(mean_squared_log_error(y_test, y_pred))))
R2_score = r2_score(y_train, y_train_pred)
print("R2 Score " + str(R2_score))
y_pred = np.array(y_train_pred)
y_test = np.array(y_train)
data={}
data['x'] = y_test
data['y'] = y_pred
sns.regplot(x="x", y="y", data=data);
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_log_error
from sklearn.metrics import r2_score
def tested(clf,x_train,x_test,y_train,y_test):
clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
print("MSE " + str(mean_squared_error(y_test, y_pred)))
print("MAE " + str(mean_absolute_error(y_test, y_pred)))
print("RMSE " + str(np.sqrt(mean_squared_error(y_test, y_pred))))
#print("RMSLE " + str(np.sqrt(mean_squared_log_error(y_test, y_pred))))
R2_score = r2_score(y_test,y_pred)
print("R2 Score " + str(R2_score))
y_pred = np.array(y_pred)
y_test = np.array(y_test)
data={}
data['x'] = y_test
data['y'] = y_pred
sns.regplot(x="x", y="y", data=data);
```

Output



Test Data Yield	Predicted Yield
182.6549356	245.22478799352544
195.0948311	239.8709754538308
233.13213719999996	244.21431540914554
276.65524589999995	206.23070216780465
194.2651719	243.4241051130758
260.2634026	241.27714815807613
231.0863347	231.01904722948862
183.62226569999999	257.6150825797371
227.3637009	191.44802423904414
271.3248604	226.77693851232442
248.7183228	212.55619266049942
263.1103304	225.8466683651188
257.0343554	205.75273505867463
186.7536773	255.99292125138217
250.08323360000003	201.70092321650853
206.2611855	218.3332511085301
223.36718829999998	223.2822595119942
202.38383190000002	234.98064346795098
196.95600080000003	234.9064120592588
265.5355937	278.7677228227131

DQN Reward: 3 DQN Penalty: 1.0999999999999999
DQN Reward: 3 DQN Penalty: 1.2
DQN Reward: 3 DQN Penalty: 1.3
DQN Reward: 3 DQN Penalty: 1.4000000000000001
DQN Reward: 4 DQN Penalty: 1.4000000000000001
DQN Reward: 4 DQN Penalty: 1.5000000000000002
DQN Reward: 4 DQN Penalty: 1.6000000000000003
DQN Reward: 4 DQN Penalty: 1.7000000000000004
DQN Reward: 4 DQN Penalty: 1.8000000000000005
DQN Reward: 4 DQN Penalty: 1.9000000000000006
DQN Reward: 4 DQN Penalty: 2.0000000000000004
DQN Reward: 5 DQN Penalty: 2.0000000000000004
DQN Reward: 5 DQN Penalty: 2.1000000000000005
DQN Reward: 5 DQN Penalty: 2.2000000000000006
DQN Reward: 5 DQN Penalty: 2.3000000000000007
DQN Reward: 5 DQN Penalty: 2.400000000000001
DQN Reward: 5 DQN Penalty: 2.500000000000001
Proposed DQN Algorithm Crop Yield Prediction Accuracy: 96.02862720920437
Proposed DQN Algorithm Crop Yield Prediction Mean Square Error (MSE): 3.971373

Random Forest Algorithm Crop Yield Prediction Accuracy: 88.18603551497256
Random Forest Algorithm Crop Yield Prediction Mean Square Error (MSE): 11.813964

Random Forest Algorithm Crop Yield Prediction Accuracy: 84.50837379547525
Random Forest Algorithm Crop Yield Prediction Mean Square Error (MSE): 15.491626

Gradient Boosting Algorithm Crop Yield Prediction Accuracy: 81.09329176933586
Gradient Boosting Algorithm Crop Yield Prediction Mean Square Error (MSE): 18.906708

Figure 6.2: **Output For RNN Algorithm**

Figure 6.2 Represents the crop yield based on RNN algorithm.Based on the preprocess Data the predicted yield is carried out.

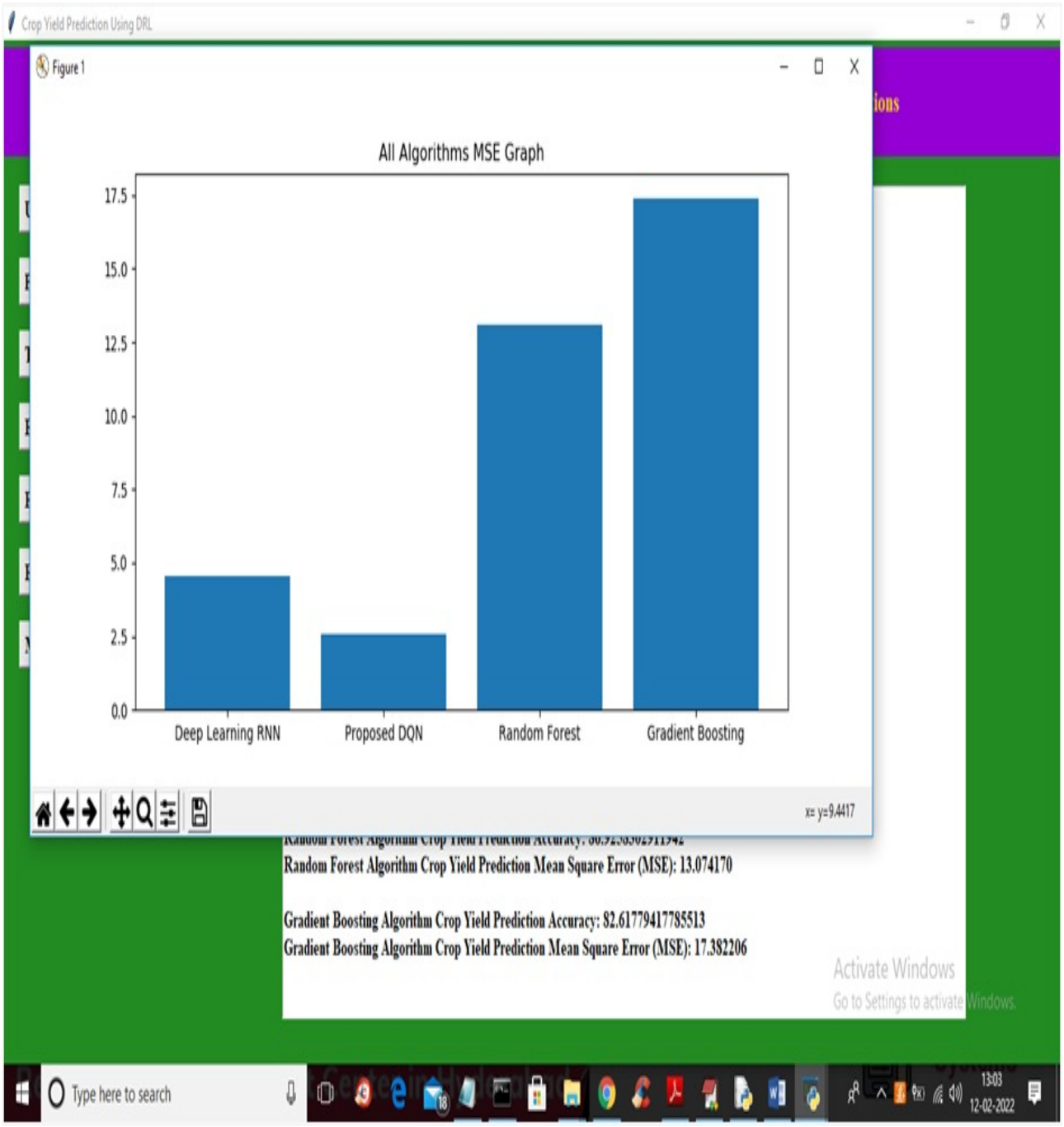


Figure 6.3: **Output For MSE Graph**

Figure 6.3 Represents the RNN Algorithm,proposed DQN,Algorithm,Random Forest Algorithm,Gradient Boosting Algorithm the MSE graph is carried out.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

The project on crop yield forecasting using machine learning Model for Sustainable Agrarian Applications" represents a significant stride towards revolutionizing agricultural practices. The integration of advanced deep reinforcement learning techniques for crop yield prediction not only bridges existing gaps in conventional methods but also aligns with the imperative of sustainability in modern agriculture. The literature review underscored the necessity for sophisticated models capable of adapting to the dynamic nature of agrarian environments. The project responds to this need by developing a model that leverages deep reinforcement learning, allowing it to learn and optimize predictions over time, considering factors such as climate, soil quality, and crop management practices.

Collaboration with agricultural scientists, farmers, technology developers, and policymakers ensures the project's relevance, practicality, and broader impact. Engaging with diverse stakeholders contributes to a well-rounded approach, addressing both the technical intricacies of the model and the practical considerations of those directly involved in agriculture.As the project progresses, its potential to empower farmers with predictive insights for optimizing crop yields while adhering to sustainability principles becomes increasingly evident. The outcomes are anticipated to go beyond academic contributions, fostering positive changes in agricultural landscapes, promoting resource efficiency, and supporting global efforts towards sustainable food production.In essence, this project stands as a testament to the transformative power of cutting-edge technology when applied to the critical intersection of agriculture and sustainability. Through the lens of deep reinforcement learning, it paves the way for a more resilient, efficient, and environmentally conscious future for agrarian applications.

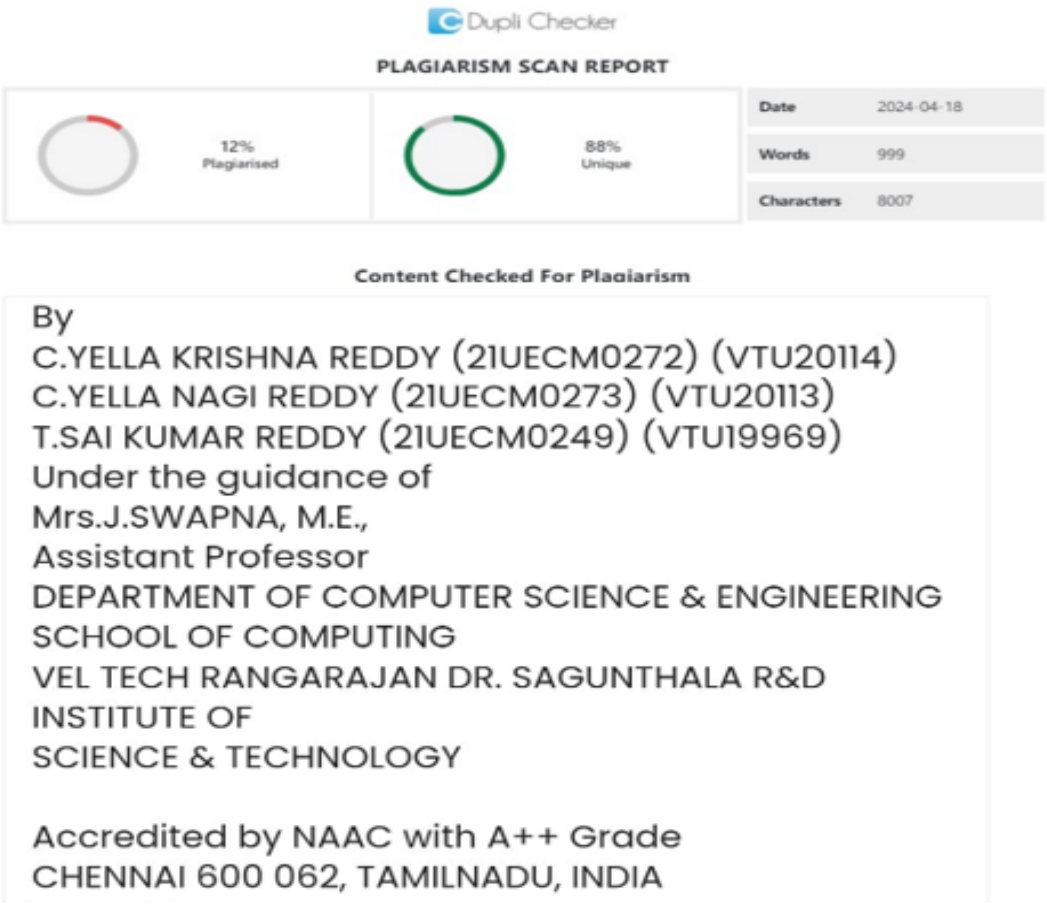
7.2 Future Enhancements

Future enhancements for crop yield forecasting using machine learning involve a multi-faceted approach to advance the model's capabilities and address emerging challenges. Integrating diverse data sources such as hyper spectral imagery and IoT sensor data will enhance the model's understanding of agricultural landscapes, while investigating transfer learning and domain adaptation techniques will promote generalization across different regions and crops. Real-time decision support systems, collaborative learning platforms, and user-friendly interfaces can ensure timely and actionable insights for farmers, fostering a community-driven approach to sustainable agriculture. Additionally, the model's scope can expand beyond yield prediction to quantify the environmental impact of farming practices, providing a holistic perspective on ecological sustainability.

Furthermore, the project can embrace dynamic optimizations such as recommending optimal crop rotations based on historical data and climate projections, promoting long-term soil health and resilience in the face of climate change. The integration of mobile applications ensures widespread accessibility, facilitating seamless adoption by farmers regardless of technical expertise. These enhancements collectively position the project as a dynamic and comprehensive solution that not only predicts crop yields but also empowers farmers with actionable insights, promotes sustainable practices, and contributes to the resilience of agrarian systems in the face of evolving environmental and economic challenges.

Chapter 8

PLAGIARISM REPORT


Dupli Checker

PLAGIARISM SCAN REPORT

12% Plagiarised

88% Unique

Date	2024-04-18
Words	999
Characters	8007

Content Checked For Plagiarism

By
C.YELLA KRISHNA REDDY (21UECM0272) (VTU20114)
C.YELLA NAGI REDDY (21UECM0273) (VTU20113)
T.SAI KUMAR REDDY (21UECM0249) (VTU19969)
Under the guidance of
Mrs.J.SWAPNA, M.E.,
Assistant Professor
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING
VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF
SCIENCE & TECHNOLOGY

Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

Figure 8.1: **Plagiarism Report**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as s
df_input=pd.read_csv("C:/Users/ADITYA/Desktop/dataworld_set.csv")
df_input.shape
df_input.head()
df_input['Crop'].nunique()
df_input['Crop'].value_counts()
df_input['District_Name'].nunique()
df_input['District_Name'].value_counts()
df_input['Season'].nunique()
df_input['Season'].value_counts()
df_input['Crop_Year'].nunique()
df_input['Crop_Year'].value_counts()
df_input.info()
df_input.isnull().sum()
df_input.duplicated().sum()
df_input = df_input[df_input['Area'] > 0 ]
df_input = df_input[df_input['Production'] > 0 ]
df_input = df_input.replace(r'^\s*$', np.NaN, regex=True)
df_input = df_input.dropna()
df_input["ProductionPerArea"] = ((df_input["Production"])/(df_input["Area"]))
df_input = df_input.drop(columns=['State_Name','Area','Production'])
Crops = df_input['Crop']
CropsCount = {}
for crop in Crops:
CropsCount[crop] = CropsCount.get(crop, 0)+1
#extract values and keys of dict:CropsCount
27
labels = list(CropsCount.keys())
values = list(CropsCount.values())
plt.pie(values, labels=labels,autopct='%1.1f%%')
plt.show()
from sklearn.preprocessing import LabelEncoder
```

```
from sklearn.preprocessing import MinMaxScaler
categorical_columns = ['District_Name', 'Crop' ,'Season']
#label encoder dict
labels_dict = {}
#scaling dict
scaling_dict = {}
for column in categorical_columns:
le = LabelEncoder()
le.fit(df_input[column])
df_input[column] = le.transform(df_input[column])
labels_dict[column] = le.classes_
df_input.describe()
x = df_input['ProductionPerArea']
print(x.describe())
import matplotlib.pyplot as plt
plt.style.use('ggplot')
fig, ax = plt.subplots()
ax.boxplot((x), vert=False, showmeans=True, meanline=True,
labels=('x'), patch_artist=True,
medianprops={'linewidth': 2, 'color': 'purple'},
meanprops={'linewidth': 2, 'color': 'red'})
plt.show()
def deviationTransform(arr):
d = np.std(arr)
return [0,d]
def minMaxTransform(arr):
min = np.min(arr)
28
max = np.max(arr)
return [min,max-min]
for column in categorical_columns:
scaling_params = minMaxTransform(np.array(df_input[column]))
df_input[column] = (df_input[column] -
scaling_params[0])/scaling_params[1]
scaling_dict[column] = scaling_params
scaling_params = minMaxTransform(np.array(df_input['Crop_Year']))
scaling_dict['Crop_Year'] = scaling_params
df_input['Crop_Year'] = (df_input['Crop_Year'] -
scaling_params[0])/scaling_params[1]
scaling_params =
deviationTransform(np.array(df_input['ProductionPerArea']))
df_input['ProductionPerArea'] = (df_input['ProductionPerArea'] -
scaling_params[0])/scaling_params[1]
scaling_dict['ProductionPerArea'] = scaling_params
p1 = np.percentile(np.array(df_input['ProductionPerArea']), 25)
p2 = np.percentile(np.array(df_input['ProductionPerArea']), 99)
df_input = df_input[df_input['ProductionPerArea'] > p1]
df_input = df_input[df_input['ProductionPerArea'] < p2]
from sklearn.model_selection import train_test_split
df_small = df_input
```

```
df_small.columns.name = None
df=df_small
x_train, x_test, y_train, y_test = train_test_split(df.iloc[:,:-1], df.iloc[:,-1],
test_size=0.2, random_state=2)
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_log_error
from sklearn.metrics import r2_score
def trained(clf,x_train,x_test,y_train,y_test):
clf.fit(x_train,y_train)
y_train_pred = clf.predict(x_train)
29
print("MSE " + str(mean_squared_error(y_train, y_train_pred)))
print("MAE " + str(mean_absolute_error(y_train, y_train_pred)))
print("RMSE " + str(np.sqrt(mean_squared_error(y_train, y_train_pred))))
#print("RMSLE " + str(np.sqrt(mean_squared_log_error(y_test, y_pred))))
R2_score = r2_score(y_train, y_train_pred)
print("R2 Score " + str(R2_score))
y_pred = np.array(y_train_pred)
y_test = np.array(y_train)
data={}
data['x'] = y_test
data['y'] = y_pred
sns.regplot(x="x", y="y", data=data);
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_log_error
from sklearn.metrics import r2_score
def tested(clf,x_train,x_test,y_train,y_test):
clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
print("MSE " + str(mean_squared_error(y_test, y_pred)))
print("MAE " + str(mean_absolute_error(y_test, y_pred)))
print("RMSE " + str(np.sqrt(mean_squared_error(y_test, y_pred))))
#print("RMSLE " + str(np.sqrt(mean_squared_log_error(y_test, y_pred))))
R2_score = r2_score(y_test,y_pred)
print("R2 Score " + str(R2_score))
y_pred = np.array(y_pred)
y_test = np.array(y_test)
data={}
data['x'] = y_test
data['y'] = y_pred
sns.regplot(x="x", y="y", data=data);
```

9.2 Poster Presentation

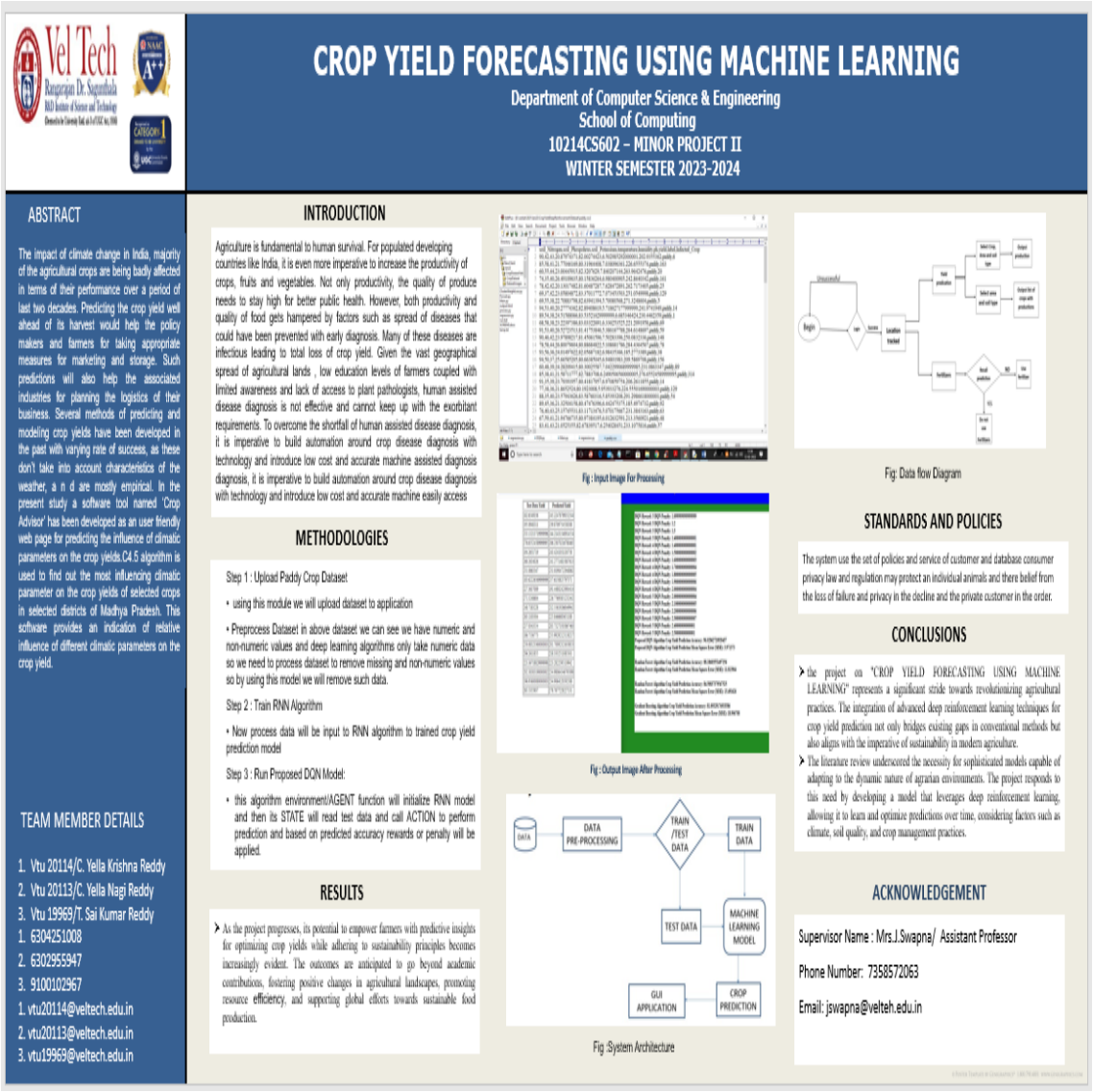


Figure 9.1: **Poster Presentation**

References

[1] Stewart, E. L., McDonald, B. A. (2014). Measuring quantitative virulence in the wheat pathogen Zymoseptoria tritici using high-throughput automated image analysis. Phytopathology, 104(9), 985-992.

[2] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.

[3] Hughes, D., Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060..

[4] Raza, S. E. A., Prince, G., Clarkson, J. P., Rajpoot, N. M. (2015). Automatic detection of diseased tomato plants using thermal and stereo visible light images. PloS one, 10(4), e0123262..

[5] Juk, J. G. (2014). Integrating SOMs and a Bayesian Classifier for Segmenting Diseased Plants in Uncontrolled Environments

[6] Sankaran, S., Mishra, A., Maja, J. M., Ehsani, R. (2011). Visible-near infrared spectroscopy for detection of Huanglongbing in citrus orchards. Computers and electronics in agriculture, 77(2), 127-134.

[7] Wetterich, C. B., Kumar, R., Sankaran, S., Belasque Junior, J., Ehsani, R., Marcassa, L. G. (2013). A comparative study on application of computer vision and fluorescence imaging spectroscopy for detection of Huanglongbing citrus disease in the USA and Brazil. Journal of Spectroscopy, 2013.

[8] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[9] Subrahmanyam, P., Wongkaew, S., Reddy, D. V. R., Demski, J. W., McDonald, D., Sharma, S. B., Smith, D. H. (1992). Field diagnosis of groundnut diseases. International Crops Research Institute for the Semi-Arid Tropics..