CROP YIELD ESTIMATION AND CROP RECOMMENDATION FORECAST USING MACHINE LEARNING

Department of Computer Science & Engineering

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad – 500 043, Telangana

December 2022

By
P. Leena Kamala 19951A0580
N.Naga Sravani 19951A0598
T.Nanditha 19951A0584

INDEX

CERTIFICATE

This is to certify that the project report entitled **Crop Yield Estimation And Crop Recommendation Forcast Using Machine Learning** submitted by **P. Leena Kamala, N. Naga Sravani, T. Nanditha** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in **Computer Science and Engineering** is a bonafide record of work carried out by her under my/our guidance and supervision. The contents of this report, in full or in parts, have not been submitted to any otherInstitute for the award of any Degree.

Supervisor	**Head of the Department**
Ms. S Aswani	**Dr. M.Madhubala**
Assistant professor, CSE AI/ML	**Professor and Head of dept CSE**

APPROVAL SHEET

This project report entitled **Crop Yield Estimation And Crop Recommendation Forcast Using Machine Learning** by **P. Leena Kamala, N. Naga Sravani, T. Nanditha** is approved for the award of the Degree Bachelor of Technology in **Computer Science and Engineering.**

Examiners **Supervisor (s)**

Principal

Date:

Place: Hyderabad

DECLARATION

I certify that

a. the work contained in this report is original and has been done by me under the guidanceof my supervisor(s).
b. the work has not been submitted to any other Institute for any degree or diploma.
c. I have followed the guidelines provided by the Institute in preparing the report.
d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of theInstitute.
e. whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and givingtheir details in the references. Further, I have taken permission from the copyright ownersof the sources, whenever necessary.

Place: Hyderabad

Date:

ACKNOWLEDGEMENT

I am very contempt with the work and research I have done on this project. The pride that comes from the completion of the work would always be unfinished without mentioning the people who guided us to make it possible. I am thankful to our institution management and respected **Sri M. Rajashekar Reddy, Chairman, IARE, Dundigal** for helping me with the necessary inputs that helped us complete this project work.

I would like to thank **Dr. L V N Prasad, Professor and Principal** who was very encouraging and a constant driving force for me. **Dr. M Madhu Bala, Dean of Computational Studies, Professor of CSE** and **Dr. B. J. D. Kalyani, Associate Professor and Head of the Department of CSE**, were a helping hand in developing and improving our work. I am especially thankful to our supervisor**, Ms. S. Aswani, Assistant Professor**, for her extend support, guidance and suggestions that helped me in enhancing the work and get better results.

I would like to use this moment to thank everyone who has supported and assisted me to work hard and make this work to its current form.

LIST OF FIGURES

ABSTRACT

Agriculture is first and foremost factor which is important for survival. Machine learning (ML) could be a crucial perspective for acquiring real-world and operative solution for crop yield issue. Predicting crop yield based on the environmental, soil, water and crop parameters has been a potential research topic. Considering the present system including manual counting, climate smart pest management and satellite imagery, the result obtained aren't really accurate. In this research, we focus on the machine learning algorithms and deep learning algorithms like DEEP QNN, Linear regression, random forest and RNN which can be used for predicting the yield of crop over certain area. The proposed model is enhanced by applying earning techniques and along with the prediction of crop, a clear information is achieved regarding the amounts of soil ingredients needed with their expenses separately. Also we proposed the recommendation where certain factors like nutrients , percentage and rainfall favorable crop to be grown over an area can be recommended. It provides a better accuracy than the existing model. It analyzes the given data and help the farmers in predicting a crop which in return help in gaining profits. The climatic and soil conditions of land are taken into consideration to predict a proper yield.

Keywords: Prediction, Recommendation, DEEP QNN, Linear regression, random forest and RNN

Chapter - 1

Introduction

1.1 Introduction

Since farmers are responsible for the production of a sizable proportion of the world's food supply, agriculture represents one of the major areas of social interest. A growing population and a corresponding decrease in food production have contributed to widespread hunger in many countries. Feeding the world's growing population is an urgent necessity in the fight against hunger. The United Nations has set the year 2030 as a target date for accomplishing two of its most important goals: increasing food security and decreasing hunger. Policymakers in a country need accurate forecasts so they can make informed decisions about food exports and imports. Farmers and growers can use yield predictions to better plan their budgets and operations. An indispensable factor in establishing a region's food security is the level of oversight exercised over agriculture, particularly the monitoring of crop yield. However, due to a number of complexities, predicting crop yields is extremely difficult. Factors such as weather, soil, topography, pests, water, genetics, harvest strategy, and pest control have a significant impact on crop yield. Numerous recent research has shown the superiority of machine learning algorithms over more conventional statistical techniques. To teach a computer something new without explicitly programming it, researchers in the area of artificial intelligence developed a technique called machine learning. When these procedures are used in either a non-linear or linear agricultural setting, very precise forecasts may be obtained. In agricultural frameworks based on machine learning, the methods are learned. Over training for a certain activity is necessary for these procedures. Once the training phase is complete, the model will use its assumptions to validate the data.

1.2 Existing System

The extraction of important crop information for prediction is mostly performed using models trained using Deep Learning. Despite the fact that these techniques have the potential to address the yield prediction issue, they have a few drawbacks. The effectiveness of such models is strongly dependent and there is no way to build a straight Weaknesses of the Current System.

1.2.1 Limitation

Predicting yield manually relies heavily on past design makes it difficult for it to understand the complicated non-linear correlations involved in the process. Due to the development of deep learning, these issues can be dealt with to some degree.

1.3 Proposed System

Our proposed framework is to estimate the crop yield along with predict the favourable crop to grow in the particular area via a web interface. For this, we have taken consideration of different deep learning as well as machine learning algorithms.

Out of the RNN, Deep QNN, Random forest, Linear regression, KNN regressor and XG boost regressor, Random forest has achieved the higher accuracy.

So in our proposed solution we have loaded the Random forest algorithm. The input data set is taken from kaggle and then preprocessed and filtered. For crop yield prediction random forest is used to predict the values.

Also for crop recommendation various algorithms such as KNN , Naïve bayes , Random forest are compared. From the results random forest has higher accuracy hence it is deployed to obtain results.

In order to predict crop productivity, the suggested study builds the data parameters that are used to feed the layers of a Recurrent Neural network in a certain order. Based on the input builds a setting for predicting crop yields. The Q-values are represented as a linear mapping from the In order to forecast crop production, the reinforcement learning agent uses a threshold in conjunction with a set of parametric characteristics. In the end, the agent's efforts to reduce error while increasing prediction precision are summed into a single score.

1.3.1 Advantages

By maintaining the original data distribution, the suggested model efficiently predicts the crop production with an accuracy of 99.3%, exceeding competing models.

1.4 Software and Hardware Requirements

1.4.1 Software Requirements

As a whole, the project's strengths and weaknesses, as well as the best approach to addressing limitations, which include perspective and features, operating system and operating environment, graphics requirements, design limitations, and user documentation.

- Python idle 3.7 version (or)
- Anaconda 3.7 (or)
- Jupiter (or)
- Google collab

1.4.2 Hardware Requirements

The program is a specific thought Python / Canopy / VS Code user is developing will determine the bare minimum in terms of computer hardware needed to complete their work. Apps that work with big arrays of objects in memory will benefit from additional RAM, whereas applications that execute several computations or operations at once will benefit more from a faster CPU.

Operating system	: windows, linux
Processor	: minimum intel i3
Ram	: minimum 4gb
Hard disk	: minimum 250gb

Chapter - 2
Literature Review

1. Probabilistic maize yield prediction over East Africa using dynamic ensemble seasonal climate forecasts:

We tested the efficacy of seasonal climate predictions in forecasting the effects in eastern Africa. We examined whether or not the knowledge shown by these seasonal predictions translated into an ability to predict maize yields in specific locations. Using the World Food Studies (WOFOST) crop model adapted for water-limited maize production and single-season simulation and the European Centre for Medium-Range Weather Forecasts (ECMWF) various initialization dates prior to sowing, we generated a 15-member ensemble of yield predictions. Reference yield simulations are utilized with the WATCH Forcing Data ERA-Interim (WFDEI) to verify maize yield estimates for the most common sowing months (December). These reference yields show good out-of-the-ordinary relationships with FAO and nationally published data; nevertheless. A region's farmers can usually predict yearly yield anomalies two months before planting. The inter annual range of variation for both the reference and forecasted yields is -40%, with the higher inter annual variation in the predicted yield predominating. Above-average and below-average yields may be predicted with high probability using ROCSS at least two months in advance, and there is a generally positive anomalous correlation of +0.3 to +0.6 between reference and forecast yields. We conclude from our examination of the sample sowing dates , when combined with dynamical seasonal climate projections, may predict anomalous water-limited maize yields.

2. Predicting future crop yields using remotely sensed water stress and solar radiation data:

Having enough soil moisture (SM) available for evapotranspiration is crucial for food security due to the large inter annual production variability of rainfed crops in expansive agricultural regions. The quantity of sunlight striking a surface also influences its ability to produce food via a process called photosynthesis (Rs). The purpose of this research is to see whether regional yield estimates can be made using data from remotely sensed crop water stress and Rs. The TVI was used to assess the crop water stress and soil moisture availability (TVDI). The TVDI was calculated using MODIS products (MODIS/AQUA) during the critical growth stage of a crop. Validation results demonstrated the adequacy of the model, showing a Relative Mean Squared Error (RMSE) of 330 kg ha-1 to 1300 kg ha-1 and a Relative Error of 13% to 34%. However, things changed radically after the one most important factor in yield was taken into consideration. This is especially true for winter crops grown in damp regions where sunshine is sparse. Semiarid regions, soils with low water retention capacities, and the height of summer are ideal for cultivating plants under crop water stress. The recommended strategy was shown to be sufficient for predicting agricultural production at the regional scale several weeks prior to harvest.

3. High-throughput plant stress phenotyping using machine learning:

Because of advancements in automated and high-throughput imaging technologies, there has been an explosion in the availability of high-resolution pictures and sensor data of plants. However, ML techniques are required to assimilate this massive data set and extract useful information for stress phenotyping. Throughout the four stages of the decision cycle in plant stress phenotyping and plant breeding, different ML approaches may be used. Here, we provide a comprehensive review and user-friendly taxonomy of ML strategies to help the plant community use the most appropriate methods for addressing biotic and abiotic stress aspects.

4. Artificial intelligence techniques for predicting rain in the Indian state of Kerala:

We compare the performance of three different AI methods in making seasonal predictions of monsoon (June–September) and post- . While all of these other methods have done a good job, the ELM methodology has fared the best, with 3.075 and 3.149 during the summer monsoon and post-monsoon periods, respectively. There is a strong correlation between the ELM design and superior accuracy (8-15-1). Findings from this research show that the suggested AI methods can accurately anticipate the onset and duration of the summer monsoon in the Indian state of Kerala.

Chapter - 3
Methodology

3.1 Block diagram

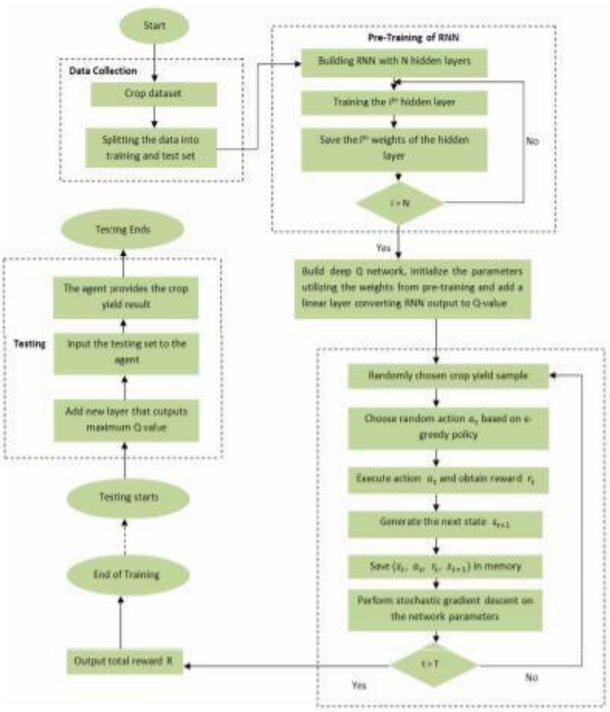


Figure: system Architecture

3.2 Input Data:

In order to build the model, input data has been taken. For testing of crop yield prediction and crop recommendation datasets have been taken from the open sources such as kaggle.com.

The datasets consists of various attributes and records like name of states, district names, season , type of crop, area of the region and production obtained. These are the attributes present in the yield prediction datasets.

Crop recommendation is also implemented. For that in the dataset there are attributes such as nutrients information like nitrogen, potassium and phosphorus quantities in the soil, temperature of the particular area, humidity, ph of the region , rainfall etc.

Dataset have been loaded and preprocessed.

In preprocessing, any null values and invalid records have been removed.

Any duplicates present in the dataset is also been removed.
After that taking the first 10000 values of the dataset we have trained the model.

The dataset the divided into training data and test data.

Model has been developed by the following steps:

Load Dataset:

Data set have been loaded by importing libraries such as pandas, numpy, seaborn , matplotlib from the read_csv file.

Split Data Set:

Split the data set to two types. One is train data test and another one is test data set.

Train data set:

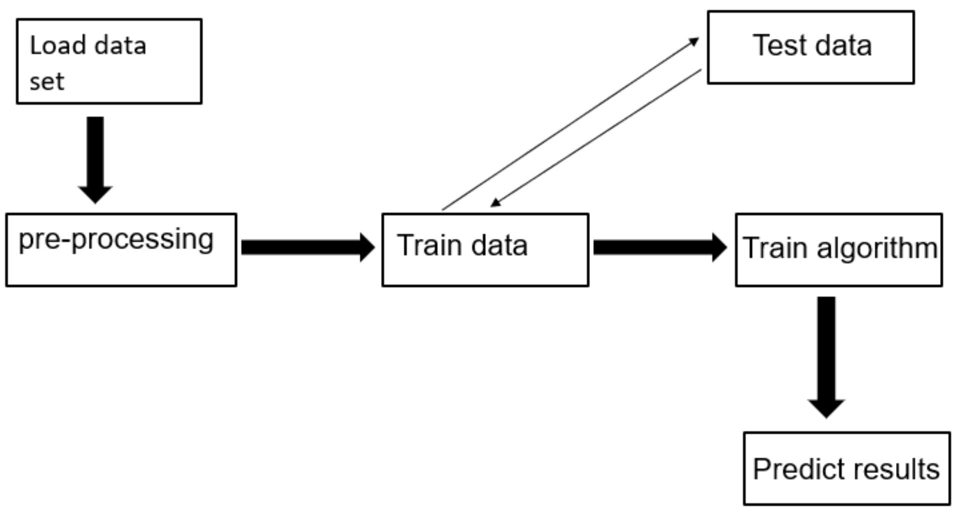
Train data set will train our data set using different algorithms. Many algorithms have taken and at last algorithm with higher accuracy is used to train the data.

Test data set:

Test data set will test the data set using algorithm.

Predict data set:

Predict () method will predict the results.



For crop recommendation and prediction to enter the data for result a front end has been developed. A webpage has been designed by using frameworks such as Flask.

A separate portal for crop prediction and crop recommendation has been created. By clicking on the portal the webpages will be open separately.

For crop yield prediction, we can enter the data to obtain result. We will get the yield percentage.

For crop recommendation, we enter the data and suitable crop to grow at that particular area is displayed along with its image.

3.3 Implementation

Machine Learning:

Machine learning is a subdivision of computer science that enables machines to learn without being separately programmed. One of the most fascinating technologies that one has ever met is machine learning. As the name suggests, it offers the computer the skill to learn, which makes it more human-like. Machine learning is presently in use, possibly in far more locations than one might visualize.

Deep Learning:

Deep learning is a subset of machine learning that is essentially a three-layer neural network. These neural networks aim to imitate the activity of the human brain by allowing it to "learn" from enormous amounts of data, albeit they are far from perfect. While a single layer neural network may produce approximate predictions, additional hidden layers can help to improve and tune for accuracy.

Importing packages:

We have imported pandas,Matplot lib,seaborn,numpy.

Data Exploration:

In order to build the model, input data has been taken. For testing of crop yield prediction and datasets have been taken.

Data Processing:

Data Preprocessing is a method that is used to convert the raw data into a clean data set. The data are gathered from different sources, it is collected in raw format which is not feasible for the analysis. By applying different techniques like replacing missing values and null values, we can transform data into an understandable format. The final step on data preprocessing is the splitting of training and testing data. After that taking the first 10000 values of the dataset we have trained the model. The dataset the divided into training data and test data.

Data selection:

The training data is taken and processed along with test data. The first 10000 values are taken into consideration and model is trained with the target columns production and crop.

Visualization :

By taking the target columns crop year on x axis and production on y axis we visualize line plot between them.
Bar plot is visualized to show relation between state name(x-axis) and production(y-axis)

Data cleaning:

The columns which are not needed such as district name and crop year are dropped and then again the model is trained.

Feature selection:

Total number of features are extracted are 80 and target coloumn is 1 i.e is production other are independent features used for prediction.

Model implementation:

Out of the compared algorithms the random forest has higher accuracy and less error rate. So we have imported random forest into the model and dumped using the pickle module. This predicts the yield percentage for a given crop in a particular area also crop recommendation is done.

RNF- Random Forest, or RF, is a robust and flexible supervised machine learning technique that builds a "forest" of decision trees from individual ones. In R and Python, you may use it for both classification and regression.

MODULES:

- We will use this module to load data into the system for exploratory purposes and to read data for processing.
- Through the use of this module, data will be separated into train and test sets.
- User registration and login are facilitated by this module.
- Using this module will provide data that may be used for forecasting.
- Forecast: ultimate forecast shown

Tensorflow

TensorFlow is a freely available and open-source software framework for dataflow and differentiable programming in a variety of contexts. It is a library for symbolic mathematics and is also put to use in machine learning programmes, especially those using neural networks.

It has dual functions at Google, serving both research and production needs. TensorFlow was first created by the Google Brain team for usage inside the company. As of the 9th of November, 2015, it was made available to the public under the Apache 2.0 open-source license.

Numpy

The Numpy package performs array processing for a wide variety of uses. It offers a powerful object for manipulating multidimensional arrays and techniques for maximizing their performance.

Numpy has many applications in the scientific community, but it also serves as a powerful multidimensional container for non-specialist data. Since Numpy lets users to design their own data types, it may quickly and easily be integrated

Pandas

Pandas is a free and open-source Python library that offers advanced features for working with and analyzing data. Primarily, When it came to analyzing data, it added virtually little. In this case, pandas were the hero.

Regardless of the data's origin, Pandas makes it possible to do the five standard phases involved in processing and analyzing the data: loading, preparing, manipulating, modeling, and analyzing. Finance, economics, Statistics, analytics, etc. are just a few of the many academic and industrial sectors that make use of Python with Pandas.

Matplotlib

Matplotlib is a 2D plotting toolkit written in Python that can generate publishable figures in a variety are all supported by Matplotlib. The purpose of Matplotlib is to simplify complicated activities while increasing the difficulty of routine ones. Scatter plots, bar charts, error bars, histograms, and other visualizations may be made with little coding. Examine the illustration plot samples and the picture gallery for more clarification.

When used in conjunction with IPython, the pyplot module creates an interface similar to that of MATLAB, making it ideal for creating basic plots. If you're an advanced user, you can use the object-oriented interface or a set of functions similar to those in MATLAB to take complete control of the appearance of your plots in terms of line styles, font properties, axis properties, and so on.

Scikit-Learn

Scikit-learn is a set of Python modules that, via a unified interface, make accessible a variety of learning techniques, both supervised and unsupervised. It is provided with several versions of Linux and is free for both personal and commercial usage thanks to its liberal simplified BSD license.

Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas.

Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

Python:

Python's interpreter does the heavy lifting at runtime, so it's a very interpretive language. Your code will run just fine without being compiled first. As with PERL and PHP, this is very similar to those languages. Download and install Python SciPy and get the most useful package for machine learning in Python.

Load a dataset and understand it’s structure using statistical summaries and data visualization. Create machine learning models, pick the best and build confidence that the accuracy is reliable. Python is a high-level scripting language. It has a growing ecosystem of libraries, frameworks, and tools. These tools and libraries are equipped with pre-written codes, that help users to perform a myriad of functions while saving an adequate amount of time spent in code generation

Python offers readable and concise codes. Since machine learning and artificial intelligence involve complex algorithms, the simplicity of Python adds value and enables the creation of reliable systems. This helps the developer to remain focused on the machine learning problem without worrying about the technical details of the language.

A big reason that adds to the success of Python is its simplicity to learn. It is easier to understand and this helps to easily create machine learning models. Another feature of Python is that it is intuitive and is perfect for a collaborative deployment. It allows faster prototyping and product testing as it is a general-purpose language.

Flask:

Flask is a web application framework written in Python. It is developed by **Armin Ronacher**, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Web Server Gateway Interface (WSGI) has been adopted as a standard for Python web application development. WSGI is a specification for a universal interface between the web server and the web applications.

It is a WSGI toolkit, which implements requests, response objects, and other utility functions. This enables building a web framework on top of it. The Flask framework uses Werkzeug as one of its bases.

Jinja2 is a popular templating engine for Python. A web templating system combines a template with a certain data source to render dynamic web pages.

Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have form a validation support. Instead, Flask supports the extensions to add such functionality to the application. Some of the popular Flask extensions are discussed later in the tutorial.

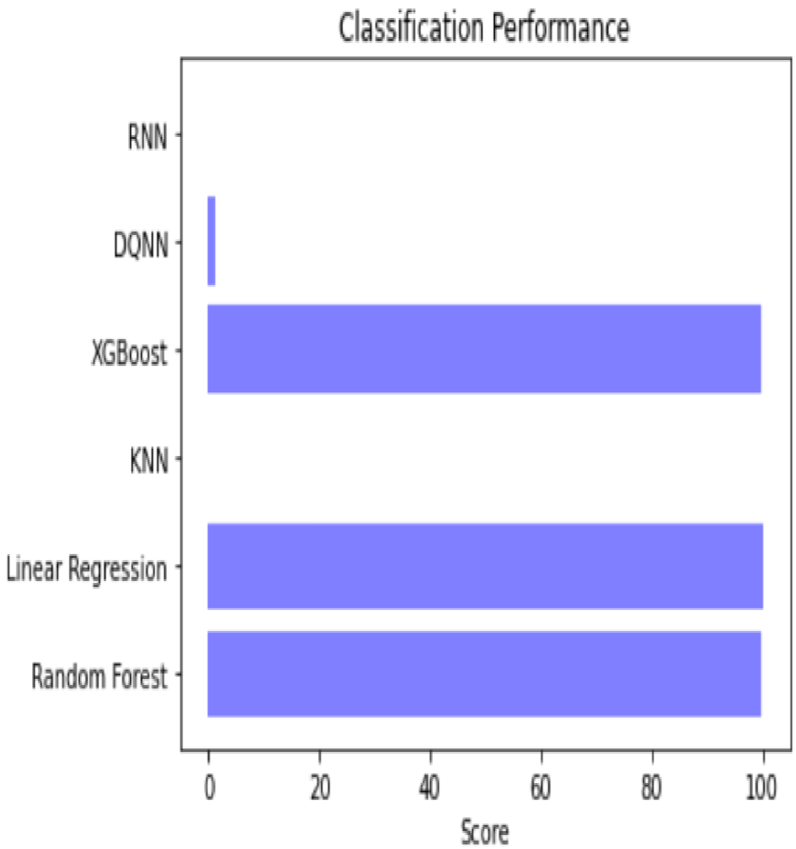
Algorithms

- RNN- Recurrent neural networks, or RNNs, are a special kind of artificial neural network that has found widespread use in fields including voice recognition and NLP. By analyzing input in a sequential fashion, recurrent neural networks may extrapolate probable future outcomes based on learned patterns.
- LSTM- The letters LSTM stand for a specific kind of Deep Learning network termed a long short-term memory network. For sequence prediction problems in particular, RNNs' ability to learn long-term dependencies is invaluable.
- RNF- Random Forest, or RF, is a robust and flexible supervised machine learning technique that builds a "forest" of decision trees from individual ones. In R and Python, you may use it for both classification and regression.
- XGBOOST- Classifiers with the help of XGBoost, which is a fast, versatile, and transportable library for distributed gradient boosting with a focus on

optimization. It utilizes the Gradient Boosting framework to put into action ML algorithms.

- KNN: K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm



3.4 Code Implementation:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

data = df.dropna()
print(data.shape)
test = df[~df["Production"].notna()].drop("Production",axis=1)
print(test.shape)
from sklearn.ensemble import RandomForestRegressor
RF = RandomForestRegressor()
RF.fit(x_train, y_train)
predictions = RF.predict(x_test)
from sklearn.metrics import r2_score
r_rf = r2_score(y_test,predictions)
print("R2score when we predict using Randomn forest is ",r_rf)
import pickle

# Saving model to disk
pickle.dump(RF, open('model/model.pkl','wb'))

import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
import pandas as pd
from model import cd
import sqlite3

app = Flask(__name__)

model = pickle.load(open('model/model.pkl', 'rb'))

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

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

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

Chapter 4
Results and Discussions

4.1 Model Screens

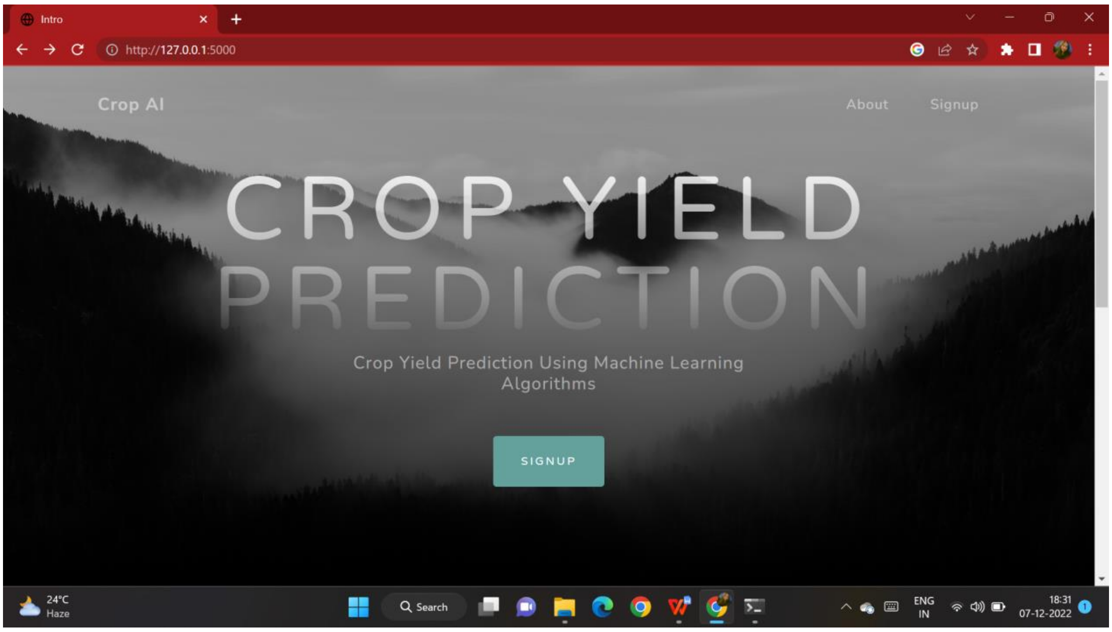


Figure 1: Home page

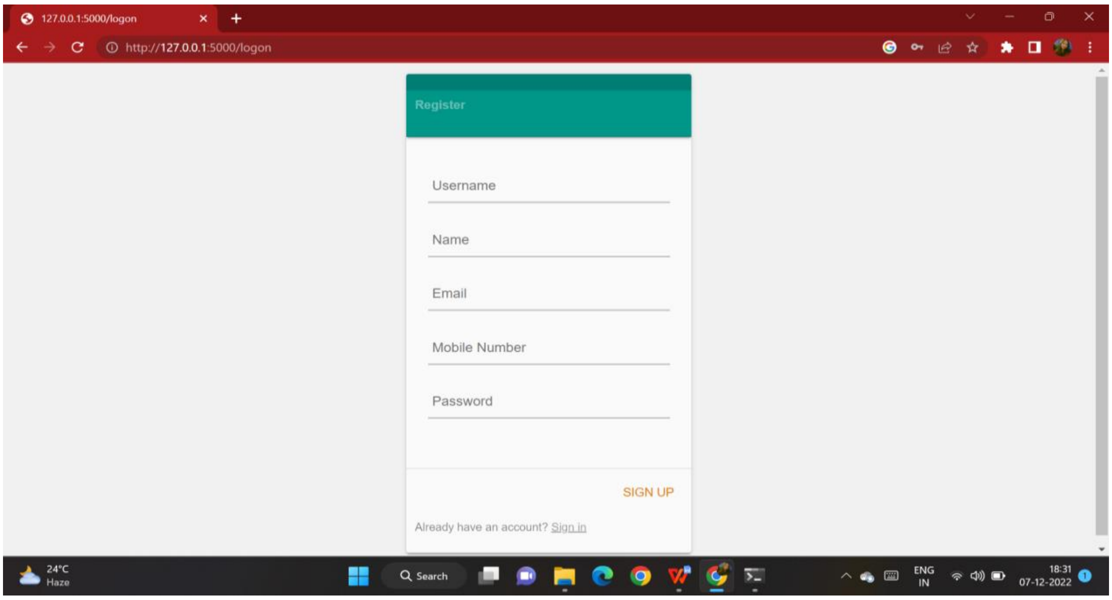


Figure 2: Login page

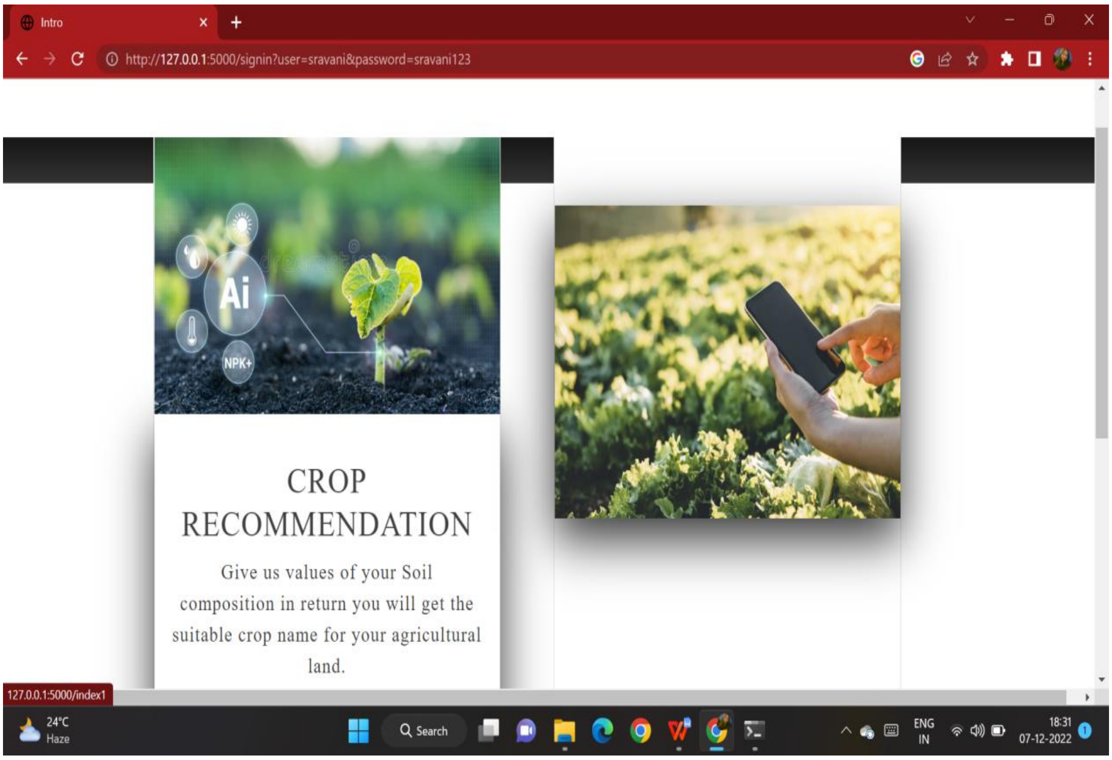


Figure 3: Portal for recommendation

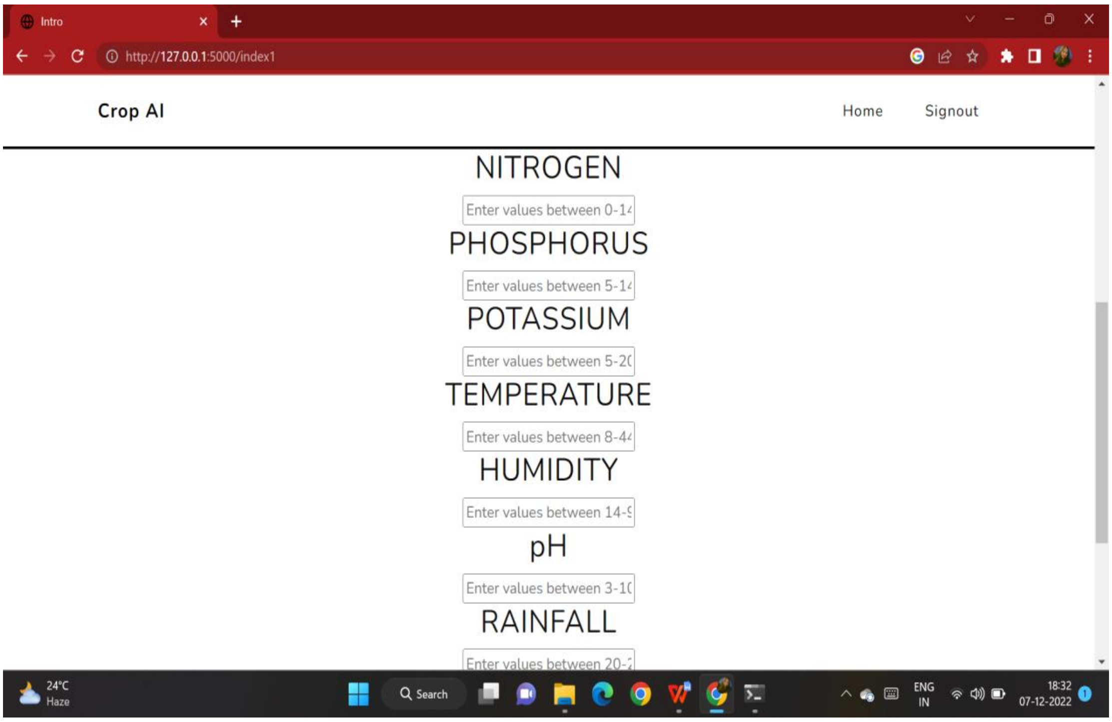


Figure 4: data attributes for recommendation

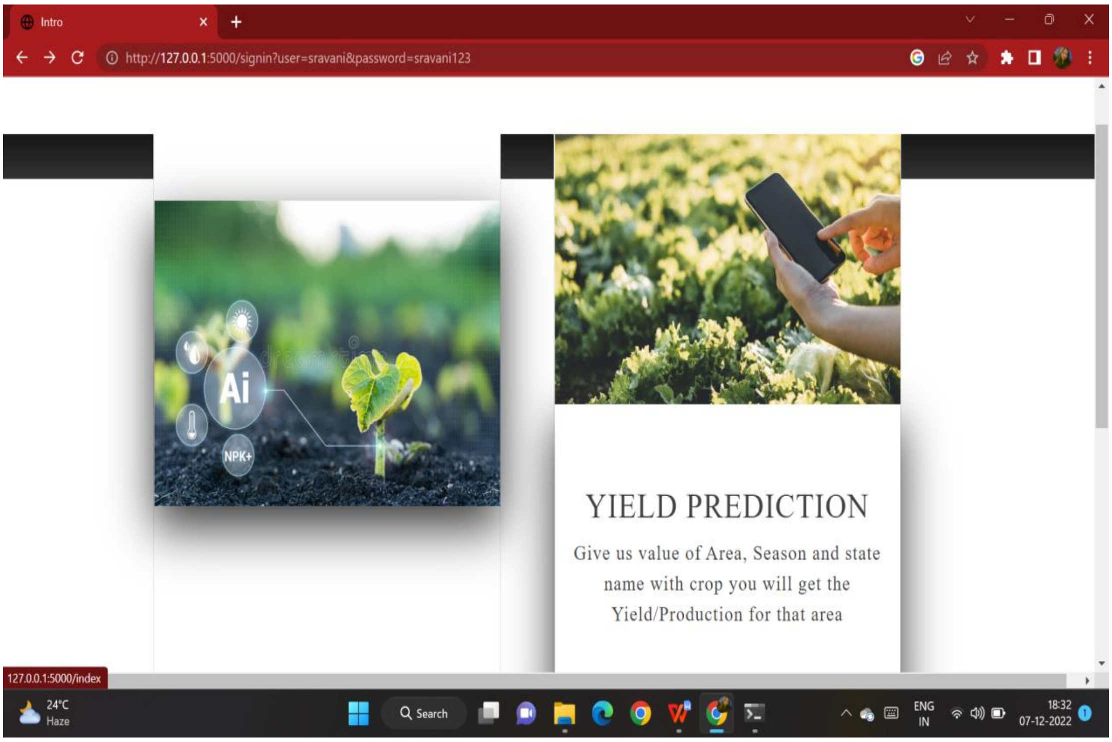


Figure 5: portal for yield prediction

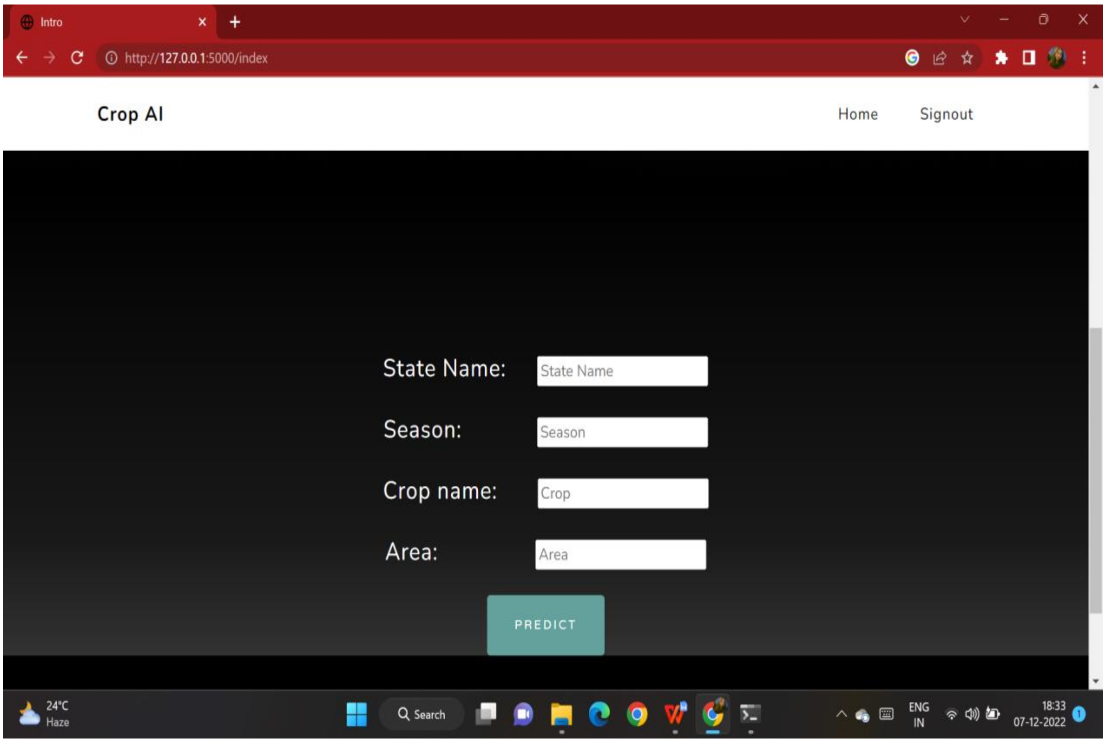


Figure 6: Data attributes for yield prediction

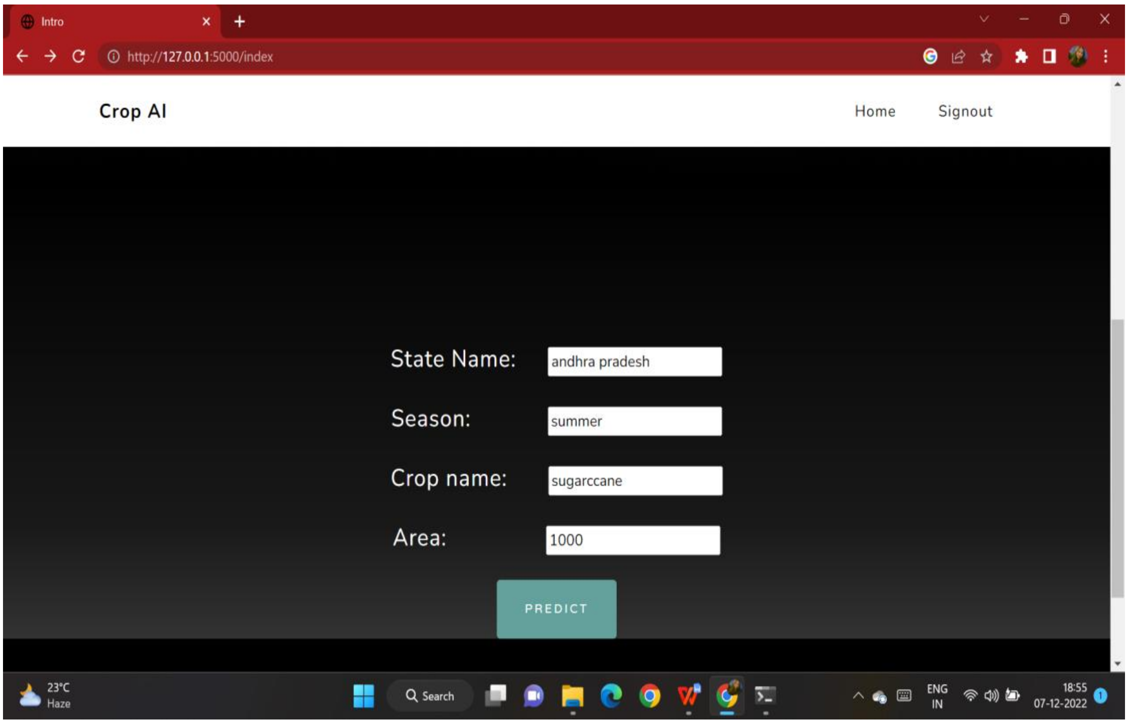


Figure 7: Data entered

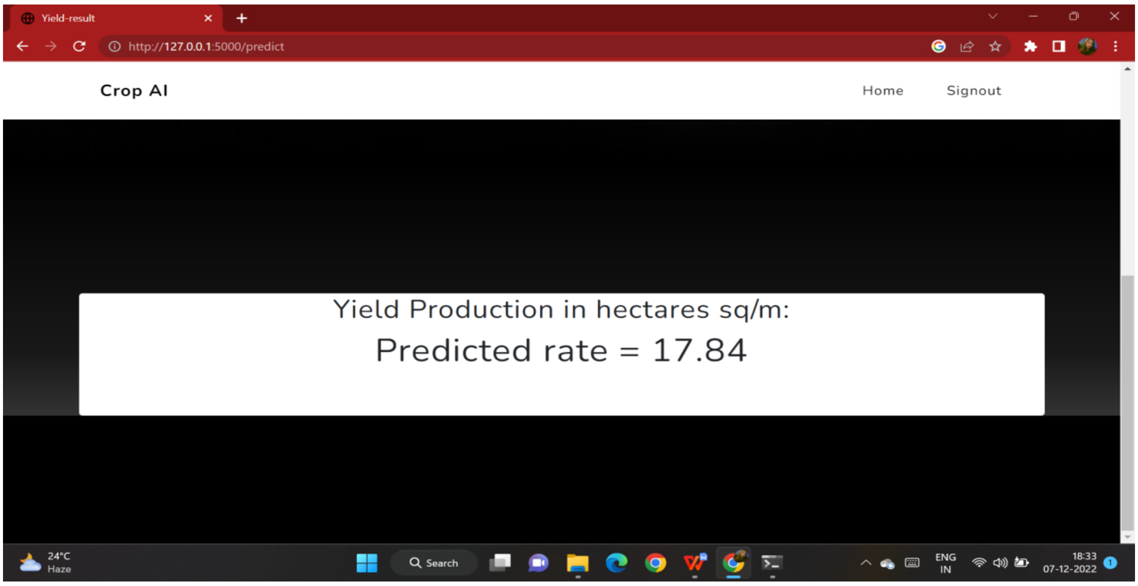


Figure 8: Outcome of yield

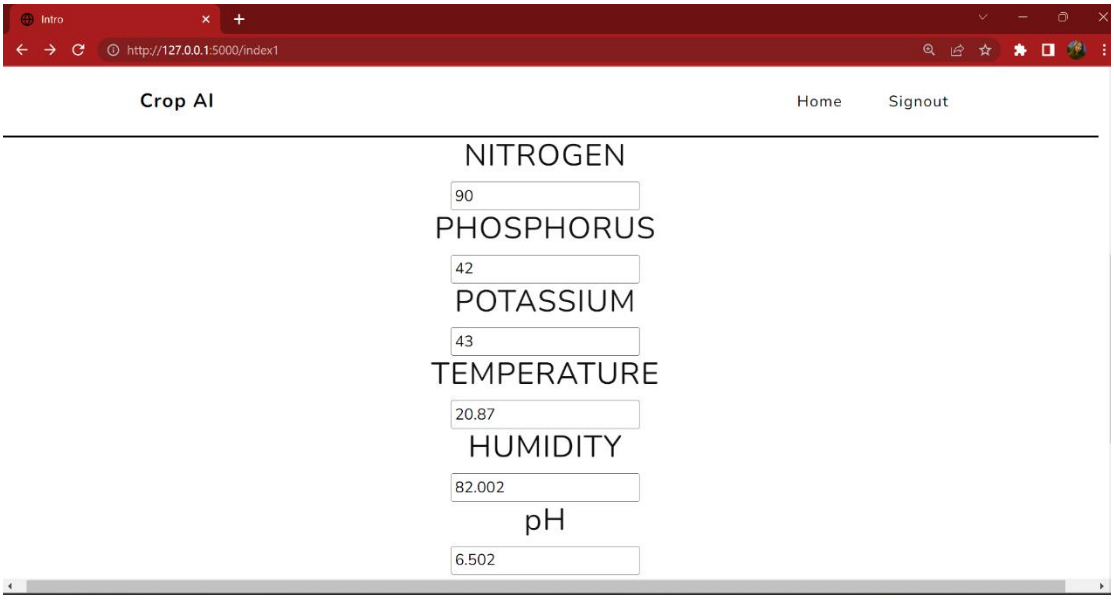


Figure 9: Values entered for recommendation

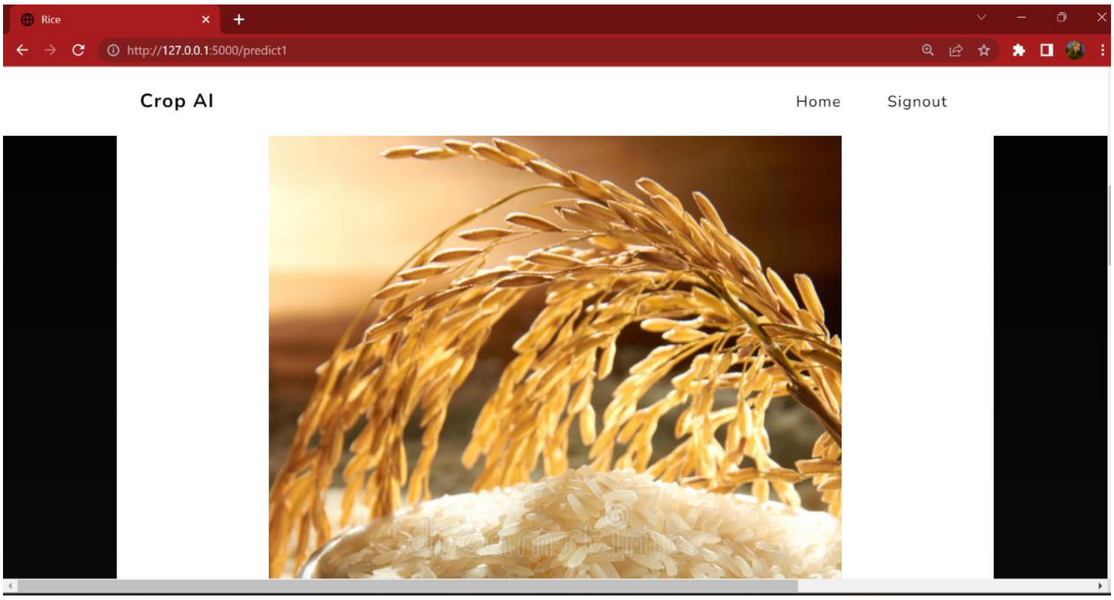


Figure 10: Obtained result(rice)

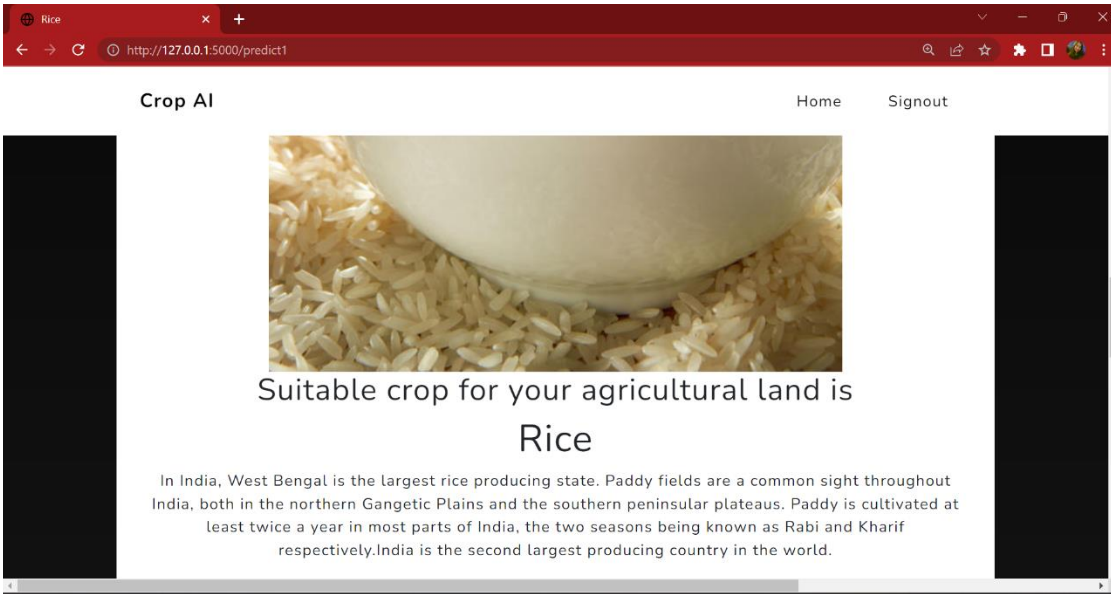


Figure 11: Data about the crop

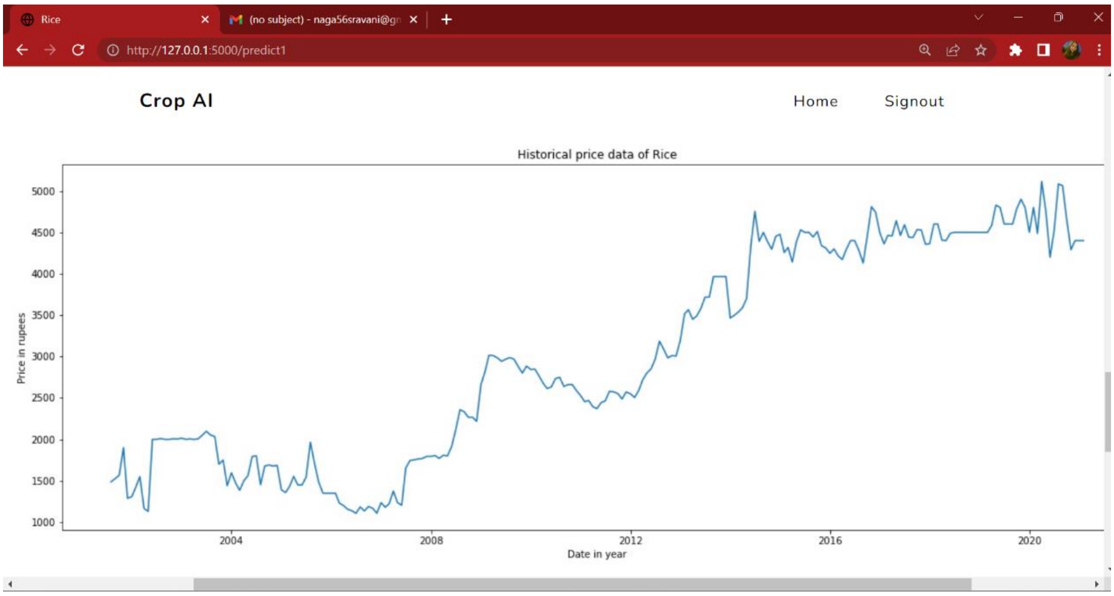


Figure 12: Graph depicting historical price of crop

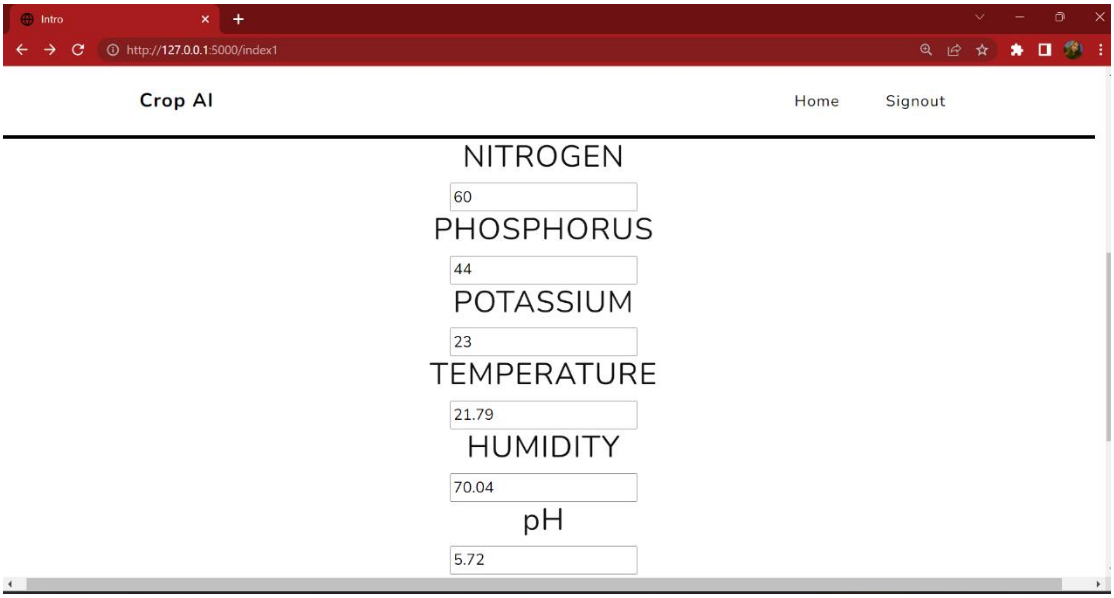


Figure 13: Values entered for crop recommendation

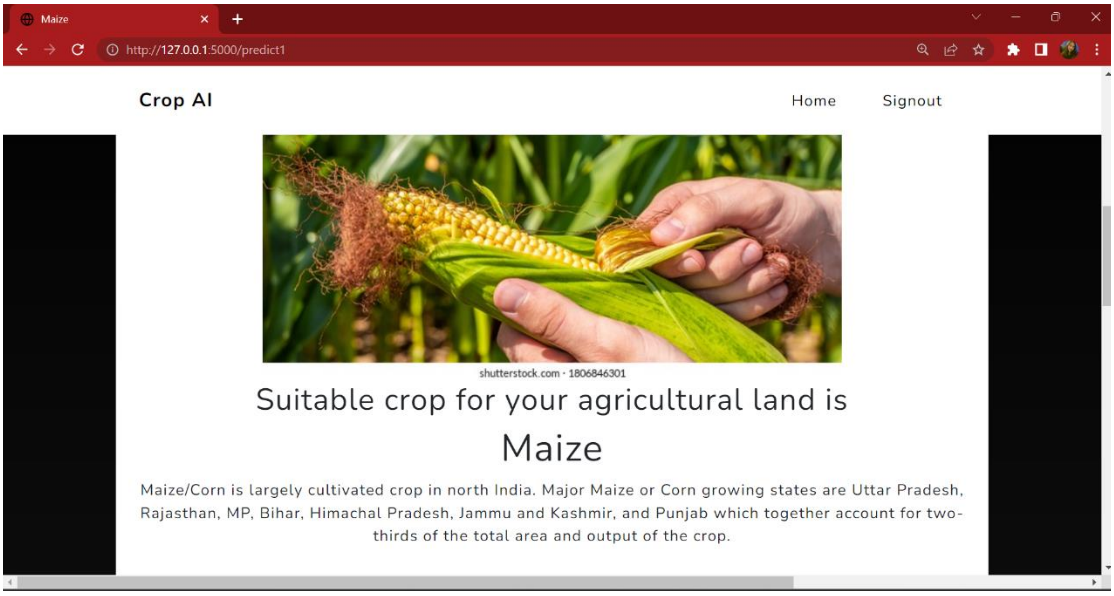


Figure 14: Result obtained (Maize crop)

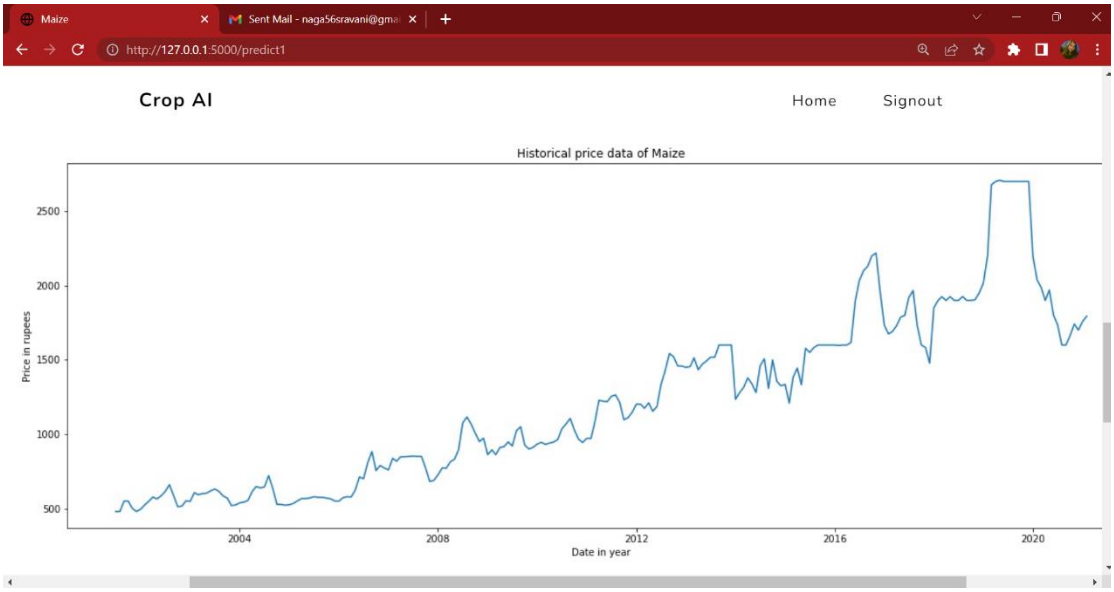


Figure 15: graph depicting historical price of maize

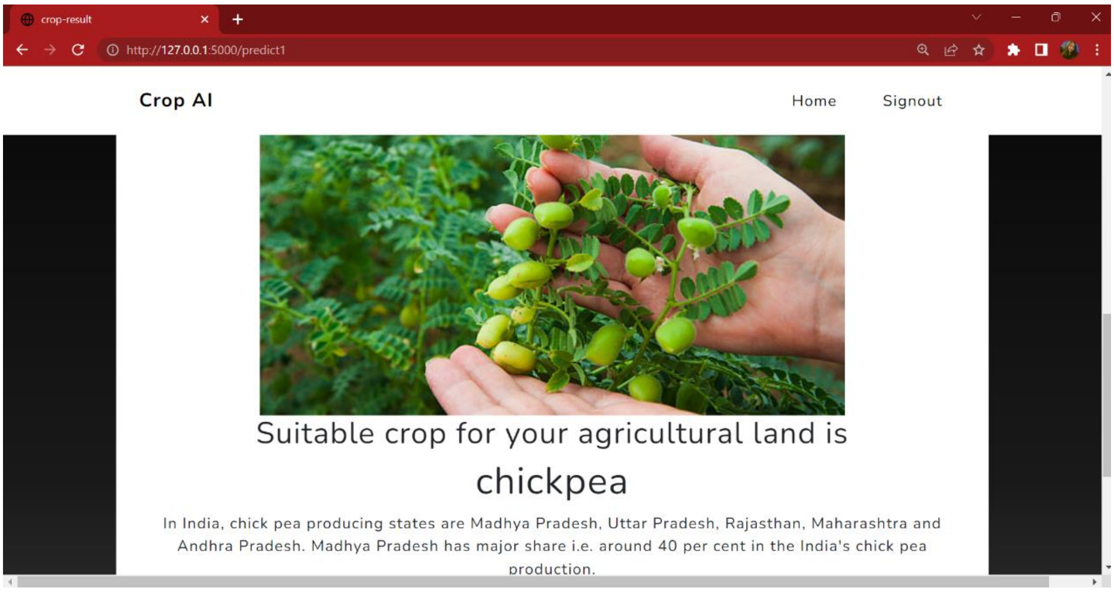


Figure 16: Result obtained (Chickpea)

4.3 Test cases

USER REQUIREMENTS:

1. Home:

Use case ID	Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications
Use case Name	Home button
Description	Display home page of application
Primary actor	User
Precondition	User must open application
Post condition	Display the Home Page of an application
Frequency of Use case	Many times
Alternative use case	N/A
Use case Diagrams	
Attachments	N/A

Figure 17: User specifications

Chapter 5
Conclusions and Future Scope of Study

5.1 Conclusion

Improved autonomy and intelligence in AI algorithms made possible by easier and useful estimation of crop yield and crop recommendation. The effectiveness and flexibility of the proposed Random forest for yield prediction are shown by the results of the accuracy and efficiency tests. The suggested technique builds a yield prediction environment in which it selects random samples from the given data and constructs decision tree for each training data and then by averaging the voting takes place and prediction can be obtained. To conclude, the suggested technique can properly predict. The random forest technique provides a more thorough solution than the unsupervised learning-based on since it runs on the large dataset the accuracy would is higher.

Also the estimation of missing values becomes easier. It also overfits the training data as multiple decision trees are working. Because of this perk, less specialized input and background information may be required for modeling agricultural yields. Because of this, it seems as if the suggested technique is using a more general model for yield prediction than is really the case. However, the RNN-based DRL may cause the gradients to expand or vanish if the time series is too lengthy. Data prediction experiments utilizing different ML predictive algorithms provide visible results, but it is essential to assess the statistical uncertainty connected to these predictions before making any decisions. Therefore, it is crucial to create a theory that can predict not just the objective but also the uncertainty in that prediction. may be used in future extensions of the current model to account for uncertainty in statistical predictions. Alternatively, one may use a DRL that is built on a network of short-term and long-term memories.

5.2 Future Enhancement:

The existing framework may be extended to incorporate the investigation of additional agricultural production prediction characteristics with respect to pest and infestations and crop damage, hence allowing the construction of a more robust working model. It would be interesting to see whether we might increase the computational efficiency of the training procedure.

References

[1] S. Li, S. Peng, W. Chen, and X. Lu, ‘‘INCOME: Practical land monitoring in precision agriculture with sensor networks,’’ Comput. Commun., vol. 36, no. 4, pp. 459–467, Feb. 2013.

[2] A. D. Jones, F. M. Ngure, G. Pelto, and S. L. Young, ‘‘What are we assessing when we measure food security? A compendium and review of current metrics,’’ Adv. Nutrition, vol. 4, no. 5, pp. 481–505, 2013.

[3] G. E. O. Ogutu, W. H. P. Franssen, I. Supit, P. Omondi, and R. W. Hutjes, ‘‘Probabilistic maize yield prediction over East Africa using dynamic ensemble seasonal climate forecasts,’’ Agriculture. Forest Meteorol., vols. 250–251, pp. 243–261, Mar. 2018.

[4] M. E. Holzman, F. Carmona, R. Rivas, and R. Niclòs, ‘‘Early assessment of crop yield from remotely sensed water stress and solar radiation data,’’ ISPRS J. Photogramm. Remote Sens., vol. 145, pp. 297–308, Nov. 2018.

[5] A. Singh, B. Ganapathysubramanian, A. K. Singh, and S. Sarkar, ‘‘Machine learning for high-throughput stress phenotyping in plants,’’ Trends Plant Sci., vol. 21, no. 2, pp. 110–124, 2016.