Crop Yield Prediction

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

RK VALLEY,KADAPA

submitted by

A.Vanaja - R170550

M.Rekha - R170505

Under the Esteemed Guidance of

N.SATYANANDARAM SIR

Dept of CSE

RGUKT,RK VALLEY

CERTIFICATE

This is to certify the report entitled **"CROP YIELD PREDICTION"** submitted by **A.Vanaja (R170550)** and **M.Rekha (R170505)** in partial fulfillment of the requirement for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out here under my supervision and guidance.

The report hasn't been submitted previously in part or in full to this or any other university or institution for the award of any degree.

Project Guide
N.Satyanandaram sir
Project Internal Guide
Computer Science and Engineering,
Rgukt rk valley

HEAD OF THE DEPT
N.Satyanandaram
HOD CSE

DECLARATION

We hereby declare that the report of the B.Tech Major Project Work entitled **"CROP YIELD PREDICTION"** which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for award of any degree.

Acknowledgment

We would like to express our sincere gratitude to **N.Satyanandaram,** our project internal guide for valuable suggestions and interest in the progress.

We are grateful to **N.Satyanandaram** sir, HOD CSE, for providing excellent computing facilities and a congenial atmosphere for progressing with our project.

At the outset, we would like to thank **Rajiv Gandhi University of Knowledge Technologies** for providing all the necessary resources for the successful completion of our course work.

ABSTRACT

Agriculture is the one of the most focused area of interest in the society because a very huge portion of food is produced by the agriculture itself. Agriculture is the most important sector that influences the frugality of India. Utmost agricultural crops are always been highly affected by the effect of change in the global world in India. Predicting the crop yield grounded on the place and the season has been a highly crucial and difficult content. Agriculture for times but the results are no way satisfying due to colorful factors that affect the crop yield. This design will allow growers to find the yield of their crops before they cultivate in the field and therefore helps them to make the necessary opinions at earlier stages itself before the cultivation. It utilizes a Random Forest Algorithm. It will allow the makers and growers to take an effective marketing and provides a way to predict crop yields before in their crop. The input is taken from the dataset. Eventually, the experimental results show the accuracy score and also forecasts the production of the crop yield.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 GENERAL INTRODUCTION

Agriculture is one of the areas of interest as most of the food is produced by them. At present, many countries are experiencing famine because of a shortage or lack of food. Increasing food production is a compelling process to end shortages. Improving food security and reducing hunger by 2030 are the stated and most important goals of the United Nations. Therefore crop protection, land surveying, and crop yield forecasting are critical to global food production. Agriculture has been a major trading partner in recent times since the development of neuroscience, information science, and mechanical learning (ML) techniques. Agriculture aims to increase and increase crop yields and thus crop quality to

support human health. Machine learning is like an umbrella that holds important color strategies and techniques. When we look at outstanding agricultural models, we can see the use of machine learning using the Random Forest algorithm. Random Forest is a popular machine learning algorithm.

1.2 PROBLEM STATEMENT

Crop yield prediction is the problem which is mostly faced in the field of the agriculture. The agricultural yield mainly depends on season and the place where the land is present. Accurate information about the crop yield history place an important role for making decisions related to risk management and future predictions in the agriculture or cultivation. It is actually difficult to predict the crop, to overcome this difficulty we are using the machine learning with the Random Forest algorithm.

1.3 OBJECTIVES

The main objective of the project is,

- To predict the crop yield by using the Random Forest algorithm.
- To enhance the overall performance analysis.

1.4 SCOPE

The main objective of this project is to predict the crop yield to overcome the shortage of the food. It is done by using the Machine learning with the Random Forest algorithm. It also enhances the overall performance analysis and accuracy of the model.

CHAPTER 2

ANALYSIS

2.1 EXISTING SYSTEM

The existing model constructs a Deep Recurrent Q-Network model which is a Recurrent Neural Network deep learning algorithm over the Q-Learning reinforcement learning algorithm to forecast the crop yield. The stacked layers are placed in a sequential manner. Recurrent Neural network is generally initiated by using the data parameters from the data set.

2.1.1 DISADVANTAGES

- Low accuracy
- Doesn’t Efficient for handling the large volumes of data
- Theoretical Limits
- Incorrect Classification Results
- Less Prediction Accuracy

2.2.PROPOSED SYSTEM

In this system, the crop yield dataset was taken as input from the dataset repository. Then, we need to implement the data pre-processing step. In this step, we have to handle the missing values or irrelevant values that make sure that we are predicting in a correct manner. Then, we have to split the data into test and train. In this step, test data is used for predicting the model and train data is used to evaluate the model. Later, we have to implement the machine learning algorithms such as the Random Forest algorithm. Finally, the experimental results shows the performance results such as accuracy score and the predicted crop yield.

2.2.1 ADVANTAGES

- Implements the machine learning algorithm
- Capable of handling large datasets with high dimensionality.
- Enhances the accuracy of the model and prevents the overfitting issue.

2.3 HARDWARE REQUIREMENTS

- Hard Disk : 200 GB
- Mouse : Logitech
- Keyboard : 110 keys enhanced
- Ram : 8 GB

2.4 SOFTWARE REQUIREMENTS

- OS : Windows 10

- Language : Python

CHAPTER 3

DESIGN

Project design is an early phase in the implementation of the project. This mainly helps to understand the features of the model, structure of the model and also help to understand the criteria for the success of the project. The main goal of this phase is to develop a correct an accurate plan that enhances the overall performance and accuracy of any model or project.

3.1 SYSTEM ARCHITECTURE

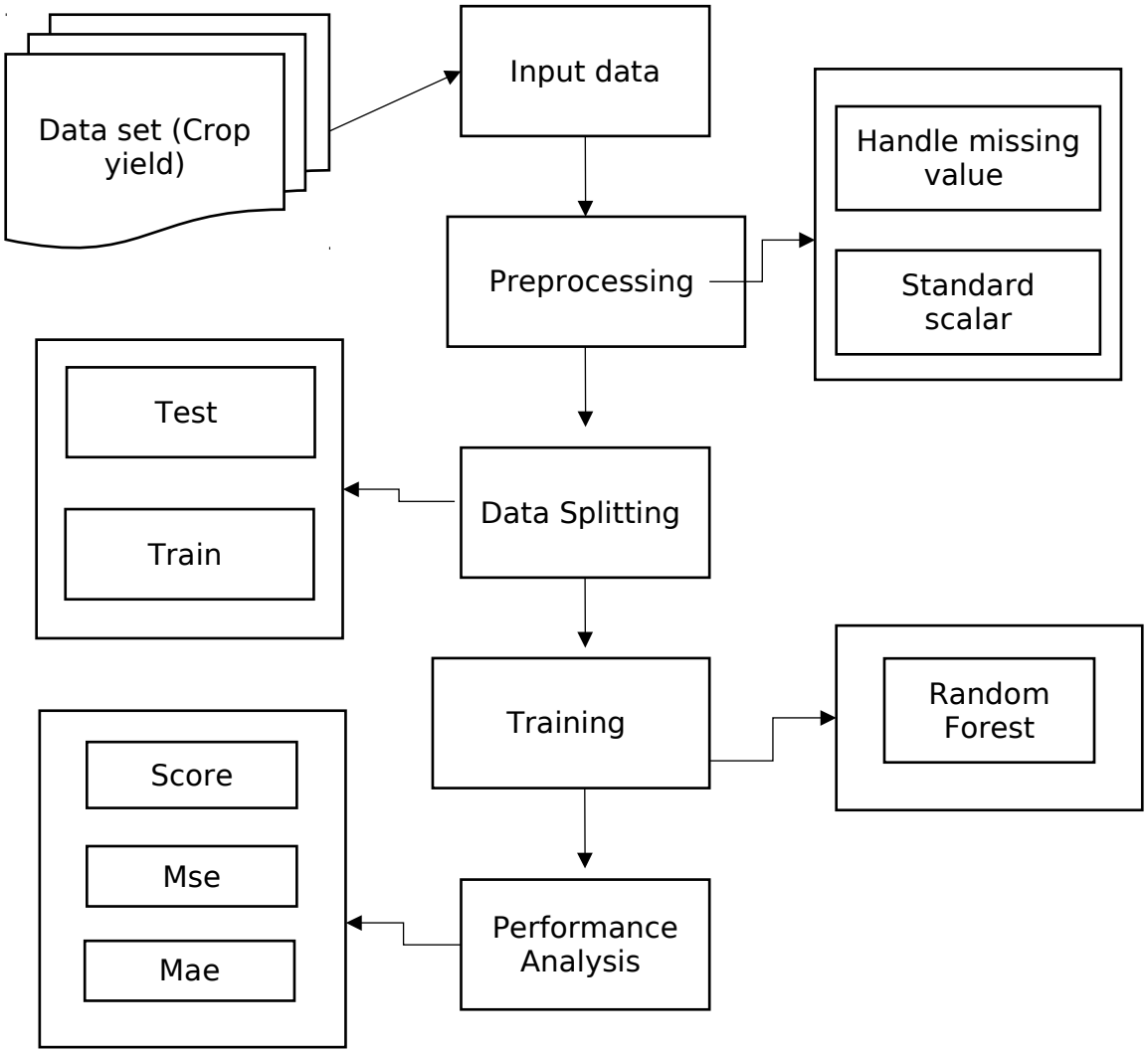


FIGURE 3.1: SYSTEM ARCHITECTURE

3.2 FLOW DIAGRAM

Flow diagram is generally the flow of the process of an algorithm. These helps to understand even the complex process in a easy and clear manner. Because of this understanding, the development of the project model can be done easily. It involves shapes that may include rectangle, oval, diamond etc.

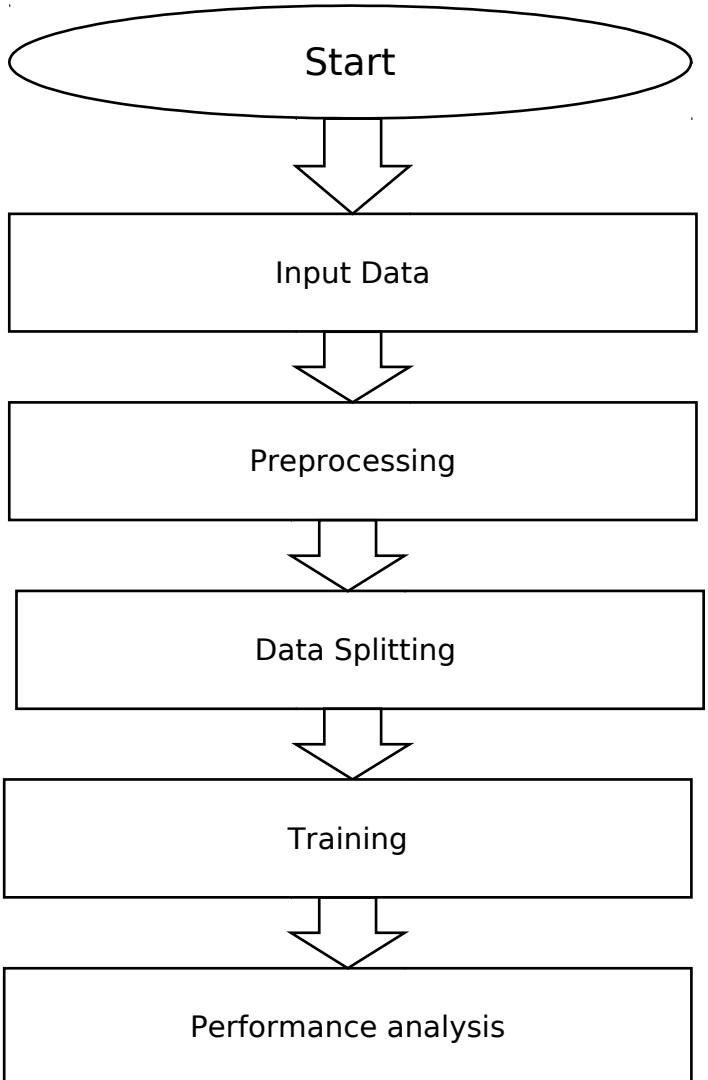


FIGURE 3.2: FLOW DIAGRAM

3.3 DESIGN USING UML DIAGRAMS

UML (Unified Modeling Language) is a standard language that is used for specifying, visualizing, constructing, and documenting the important information about software systems. The key for developing an Unified Modeling Language diagram is creating the shapes and connecting them that represent an object or class with other shapes that demonstrate the relationships and the flow of information and the data for the implementation of the system. UML diagrams are used in many sectors, this makes the implementation process easier.

Building Blocks of UML:

Things:

Anything that is a real-world object or entity is called as Things. These are the abstractions for the first-class citizens in a model. These things are the basic building blocks of the object oriented analysis and design. There are four types of Things. They are,

- Annotational Things
- Behavioural Things

- Grouping Things
- Structural Things

Relationships:

Relationships provides the meaningful connections between things.The functionality of an application mainly depends on these relationships. The Relationships tie the things together. These relationships are of four types namely,

- Association
- Dependency
- Generalization
- Realization

Diagrams:

The diagrams are the graphical representation of the models that contains symbols and text.There are three types of Unified Modeling Language diagrams. They are,

- Structural Diagrams
- Behavioural Diagrams
- Interaction Diagrams

These diagrams group interesting collections of things. This graphical representation helps in understanding and analyzing the data or information in an easy way.

3.3.1 USE CASE DIAGRAM

- Use case diagrams generally involves the users and the actions that are performed by the user for the interaction.

- A proper use case diagram shows a high-level interaction of the model by considering the actors, system and their interaction.

- The users are represented by using the stick diagram.

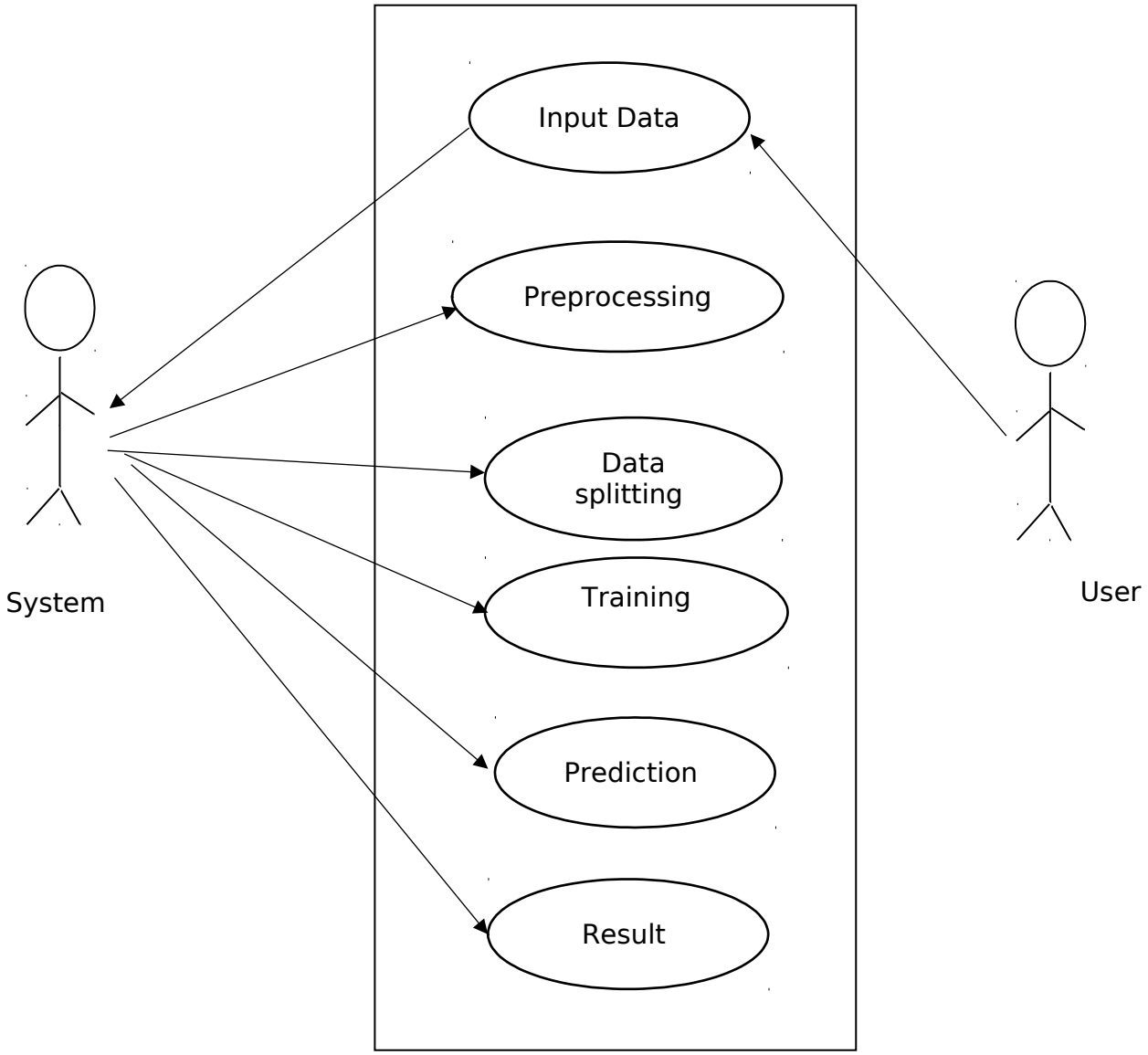
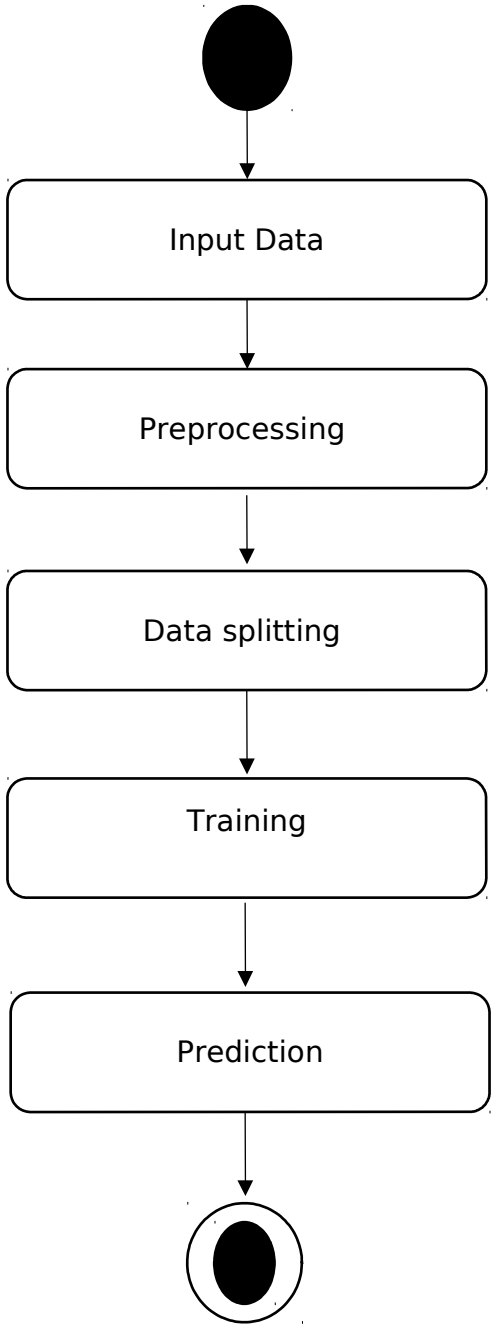


FIGURE 3.3.1: USE CASE DIAGRAM

3.3.2 ACTIVITY DIAGRAM

- Activity diagrams are nothing but the flow diagrams that generally shows or displays the flow of the process.
- We can also consider joining and branching in activity diagrams.



3.3.3 SEQUENCE DIAGRAM

- Sequence diagram is considered as an interactive diagram because of the interactive behaviour of the model.
- It consists of stages with life lines and the messages that need to transferred.

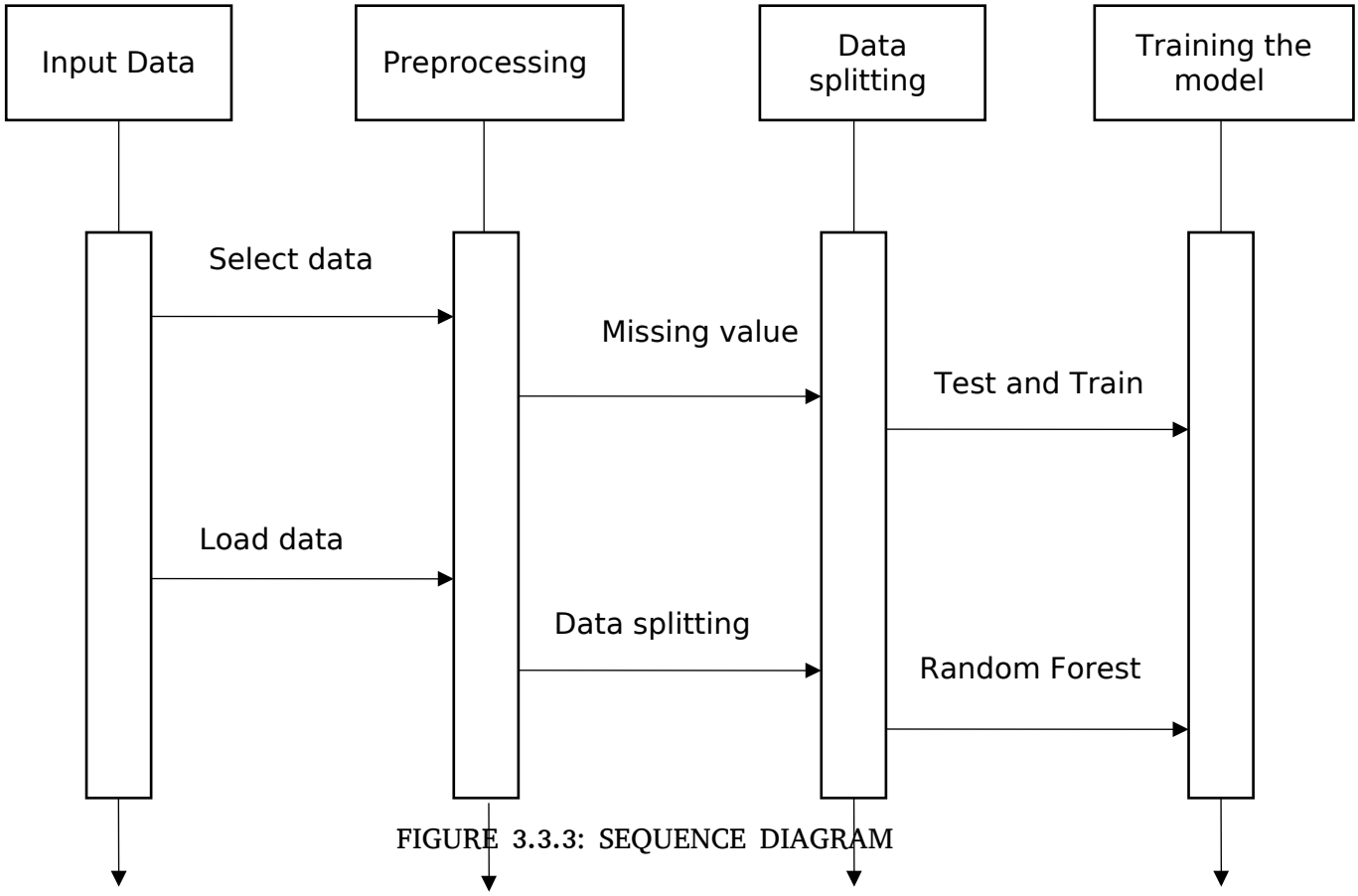


FIGURE 3.3.3: SEQUENCE DIAGRAM

3.3.4 CLASS DIAGRAM

- Class diagram is generally a static representation of the model.
- We can use access modifiers for these attributes and methods. The most used access modifiers are,

- Public (+)
- Private (-)
- Protected (#)

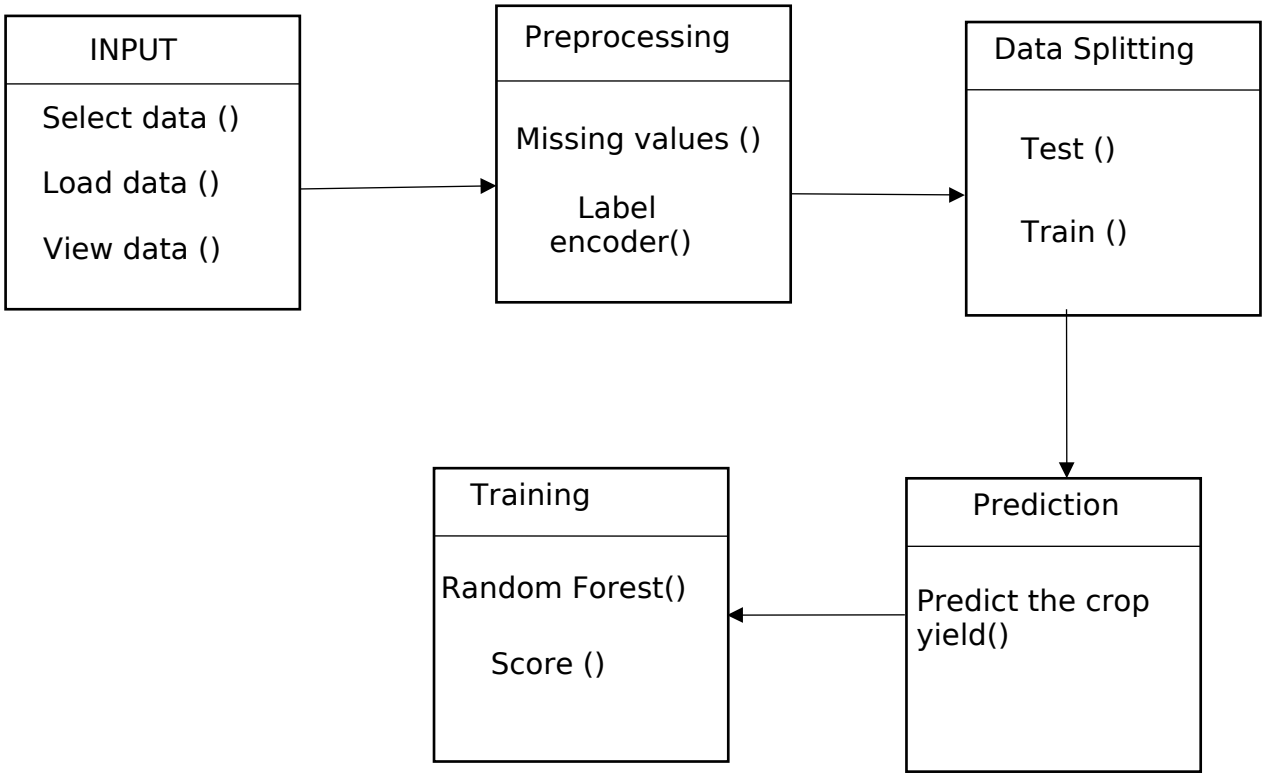


FIGURE 3.3.4: CLASS DIAGRAM

3.4 ER DIAGRAM

- The ER diagram stands for Entity-Relationship diagrams. These are also called as ERD.
- ER diagrams generally contain various symbols for representation. Some of them are,
  - Rectangle- to represent entity
  - Oval- to represent attributes
  - Diamonds- to represent relationships

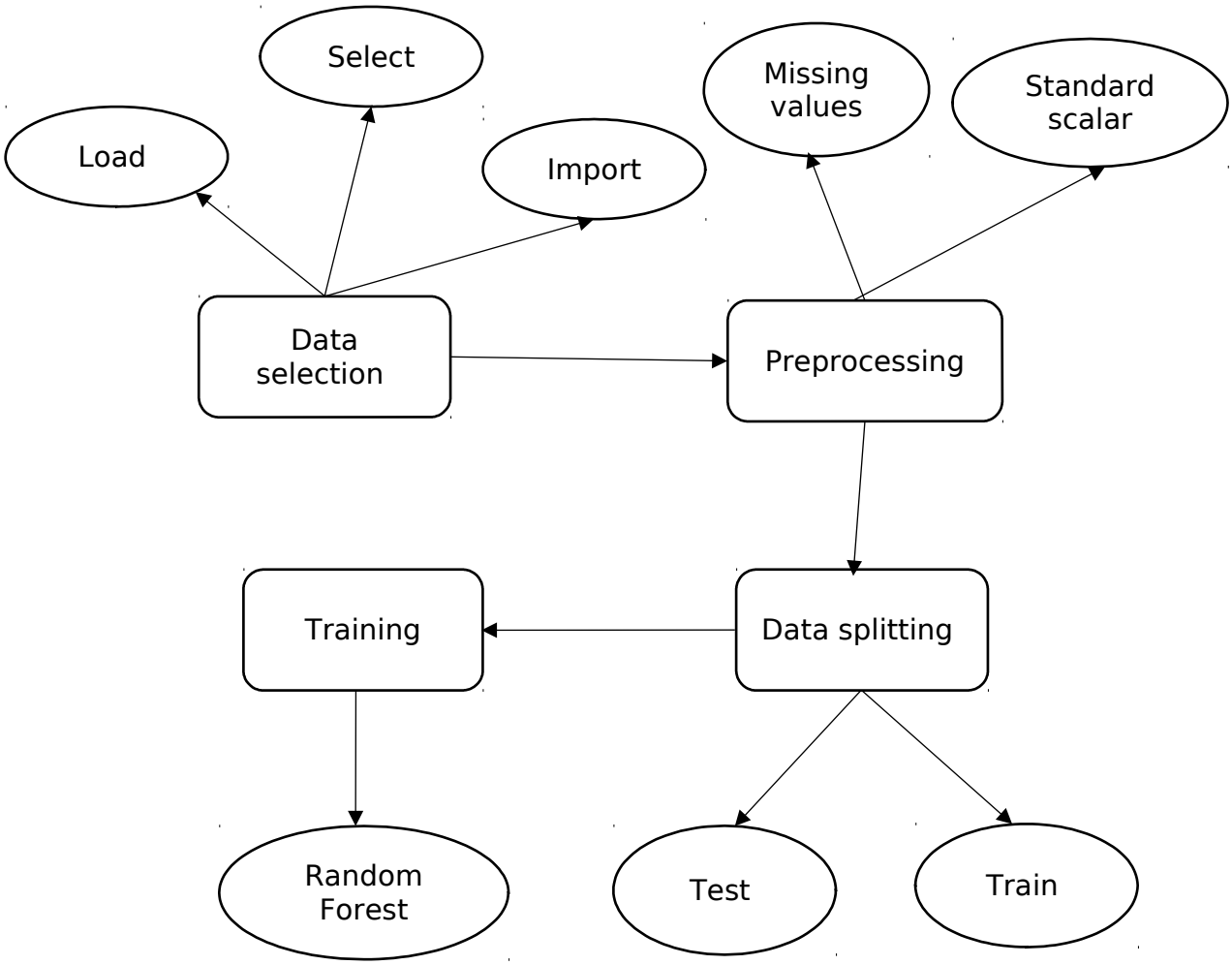


FIGURE 3.4: ER DIAGRAM

CHAPTER 4

IMPLEMENTATION

4.1 MODULES

- Data selection
- Data preprocessing

- Data splitting
- Training the model
- Performance Analysis

4.1.1 DATA SELECTION

- The input data was taken from a data set repository like Kaggle repository.
- The dataset contains information about the place, crop name, season, area and the production.
- We have used the python to implement the project. To load the data we have used the pandas package.

4.1.2 DATA PREPROCESSING

- Data preprocessing is nothing but the preprocess of the data before the data is used for the implementation.
- In this step, we also clean the dataset by removing the corrupted data or any data which is not relevant for implementation. Because of this the accuracy of the dataset increases which indirectly increases the efficiency of the model.
- The two major actions performed here are,
    - Missing data removal
    - Standard scalar
- **Missing data removal:** In this step, the null values like Nan values are replaced by 0.

- **Standard scalar:** In this step, the mean and scales of every attribute or variable are changed to unit variance.

4.1.3 DATA SPLITTING

- In the machine learning process, the role of the data is very crucial for the learning process.
- Data splitting is the process of dividing the data from the dataset into two portions. It is usually performed for the purpose of cross-validation.

4.1.4 TRAINING THE MODEL

- We need to use the train data for training the model. Here we are using the Random Forest Regression technique to train and evaluate the model.
- Random Forest Regression technique is a machine learning technique that contains multiple decision trees for the learning of the models.

4.2. LIBRARIES

In this project we have used the Python for the implementation. Python is a language which is easy and simple to implement. Python contains the very efficient high level data structures and also suites good towards the object oriented programming. Python has many features. Some of them are,

  - Simple

- Easy to learn
- Free and open source
- Portable
- Interpreted
- Object Oriented
- Extensive Libraries

Python has many libraries. The following are the used libraries in the implementation of the model.

4.2.1 Pandas:

- Pandas is a Python library used for running with information units.
- It has functions for studying, cleaning, exploring, and manipulating records.

4.2.2 Matplotlib:

- Matplotlib is a plotting library for creating static, lively, and makes the visualizations interactive in the Python. Matplotlib can be used in the Python scripts, the Python and IPython shell, internet software servers, and diverse graphical person interface toolkits like Tkinter, awxPython, and so on.

4.2.3 RandomForestRegressor:

- A random forest is a meta estimator that suits a various classifying decision trees on numerous sub-samples of

the dataset *and makes use of combining to improve the accuracy of the prediction and also manages the over-fitting.

4.2.4 Minmaxscaler:

- The MinMaxscaler is a sort of scaler that scales the minimal and maximum values to be zero and 1 respectively.

4.2.5 Seaborn:

- Seaborn is a Python information visualization library constructed on pinnacle of Matplotlib.

4.2.6 Sklearn:

- Scikit-research is probably the maximum beneficial library for system gaining knowledge of in Python.
  - **Sklearn.model_selection.train_test_split:**

train_test_split is a characteristic in Sklearn model selection for splitting information streams into different sets called as subsets, these are used for training statistics and then for testing information.

  - **Sklearn.ensemble:**

The sklearn. ensemble module includes the algorithms for averaging which is dependent on the choice of bushes in a random way.

  - **sklearn.preprocessing.labelencoder:**

Target labels are encoded with the values between 0 and n_classes-1. This transformation is used to encode the global values which is y and no longer the x.

4.3 SOURCE CODE

```
import pandas
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
#_____Dataselection_____
_____
data = pandas.read_csv("crop_production.csv")
print("-----------------------------------------------------")
print("Data Selection")
print()
print(data.head(10))
print()
print("-----------------------------------------------------")
#_____Preprocessing_____
df = data.copy()
print("Dropping null values")
df.dropna(axis=0, inplace=True)
print("Checking the data")
print(df.isna().sum())
```

```
#df["Crop"].value_counts()
#_____________________________________________________________
print("Taking only the crops which have data more than 1500")
crop_count = df["Crop"].value_counts()
df = df.loc[df["Crop"].isin(crop_count.index[crop_count > 1500])]
print(df.head(10))
print()
print()
#_____________________________________________________________
_____
print("printing the crop names whose count is greater than 1500")
names=list(set(df["Crop"].values))
print(names)
print()
print()
#_____________________Giving the crop name as input_____________________
cro=input("Enter the crop name:")
Name = df[(df["Crop"] == cro)]
print("dispyaling the selected crop data")
print(Name.head(10))
print()
print()
```

```
#_____________________________________________________________
dt = Name.copy()
le = LabelEncoder()
scaler = MinMaxScaler()
dt["district"] = le.fit_transform(dt["District_Name"])
dt['season'] = le.fit_transform(dt["Season"])
#dt["area"] = scaler.fit_transform(dt[["Area"]])
dt["state"] = le.fit_transform(dt["State_Name"])
print("Data after preprocessing")
print(dt.head(10))
print()
print()
#_____________________Datasplitting_____________________
_____
X = dt[["Area", "district", "season", "state"]]
y = dt["Production"]
split=int(len(dt)*0.85)
#k=dt["Area"].values
#k=k[split:]
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.15)
print()
print()
print("printing the training data")
print(X_train.head())
print()
```

```
print()
print(y_train.head())
print()
print()
print("printing the testing data")
print(X_test.head())
print()
print()
print(y_test.head())
print()
print()

#_____________________________RandomForest_____________________
model = RandomForestRegressor()
model.fit(X_train, y_train)
print("Score of Random forest regressor:",model.score(X_test,
y_test))
print("Prediting the production by taking the test data")
y_predict=model.predict(X_test)
print()
print()
print("predicted values",y_predict[:10])
print()
print()
print("Actual values",y_test[:10])
print()
```

```
print()
#_____Plotting the actual production with area_____
plt.figure(figsize=(14, 10))
sns.regplot(Name["Area"], Name["Production"]).set(title='Area vs Actual production')
plt.show()
#_____Plotting the predicted production with area_____
h1=X_test['Area'].values
h1=h1.reshape((-1,1))
plt.figure(figsize=(14, 10))
sns.regplot(h1,y_pre).set(title='Area vs Predicted production')
plt.show()
```

CHAPTER 5

EXECUTION PROCEDURE AND TESTING

5.1 EXECUTION PROCEDURE

Step-1 Import the modules/packages that are required. The modules Numpy, Pandas, Sklearn, Seaborn have been imported.

Step-2 Using the read csv method the dataset is read and data frame object is created.

Step-3 Displaying the top 10 rows of the data frame, so that we can see the columns and data present in the data frame.

Step-4 Checking for the null values and dropping the null values from the data frame.

Step-5 Filtering the data so that only the crops which has data more than 1500 rows will be taken out from the whole data.

Step-6 Displaying the list of crop names that are present in the data.

Step-7 Entering the crop name as input for which we need to predict the production/yield, and displaying the selected crop data.

Step-8 Transforming the data using the label encoder object, so that all the strings in the data are transformed in to integers.

Step-9 Splitting the data in to test data and train data , so that train data will be 85% and test data will be 15% of the whole data.

Step-10 By passing the training data in to the fit method of the algorithm object the model is trained.

Step-11 By passing the data for which the prediction is needed in to the predict method of the algorithm object we can get the predicted values as a result.

Step-12 The evaluation of the model is done by using the score method and mse function.

Step-13 Finally the Plots are drawn using the regplot method.

CHAPTER 6

RESULTS AND PERFORMANCE EVALUATION

6.1 Data Selection

The data is selected from the below dataset,

df - DataFrame

Index	State_Name	District_Name	Crop_Year	Season	Crop	Area	Production
0	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Arecanut	1254	2000
1	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Other Kharif pulses	2	1
2	Andaman and Nicobar Islands	NICOBARS	2000	Kharif	Rice	102	321
3	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Banana	176	641
5	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Coconut	18168	6.51e+07
6	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Dry ginger	36	100
7	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sugarcane	1	2
8	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Sweet potato	5	15
9	Andaman and Nicobar Islands	NICOBARS	2000	Whole Year	Tapioca	40	169
10	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Arecanut	1254	2061
11	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Other Kharif pulses	2	1
12	Andaman and Nicobar Islands	NICOBARS	2001	Kharif	Rice	83	300

FIGURE 7.1: DATA SELECTION

6.2 Data Preprocessing

Missing values are removed and the count of missing values becomes 0.

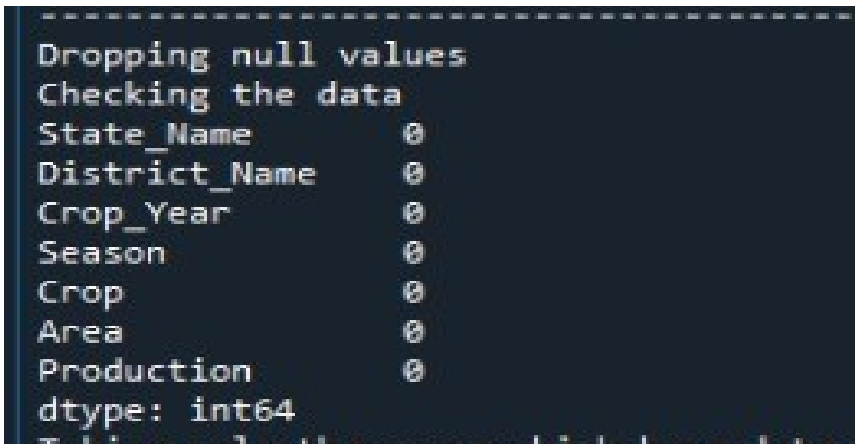


FIGURE 6.2.1: MISSING VALUES

6.3 Data splitting

Data is split into two sets namely training and testing data. In this we have x_train data, y_train data, x_test data and y_test data.

X_train - DataFrame

Index	Area	district	season	state
91681	35128	318	1	14
25676	119234	537	5	3
134556	8300	456	3	17
18628	3560	205	3	3
207625	10155	119	1	30
5926	41191	445	1	1
156737	1218	390	3	22
101072	2695	421	3	15
141889	85	615	0	19
20069	44836	270	3	3
68447	167149	287	1	10
152959	22000	260	0	22
122479	5128	516	1	16
124610	37722	588	4	16
181067	21709	277	1	27

FIGURE 6.3.1: X_TRAIN DATA

y_train - Series

Index	Production
91681	119948
25676	163235
134556	21000
18628	5973
207625	18208
5926	124809
156737	2704
101072	8925
141889	181
20069	97006
68447	610000
152959	6000
122479	3942
124610	26630
181067	82523

FIGURE 6.3.2: Y_TRAIN DATA

X_test - DataFrame

Index	Area	district	season	state
204272	176864	55	1	30
103709	2739	63	1	16
61457	87100	301	1	9
56973	29124	416	1	8
170337	31214	220	1	25
218548	2881	302	3	30
158351	31000	451	3	22
189773	49868	600	1	27
148244	59120	60	3	22
226110	139590	467	1	30
43854	28217	527	3	4
21560	55600	285	5	3
98033	10396	10	5	15
76558	5605.83	422	0	13
95486	16441	515	3	14

FIGURE 6.3.3: X_TEST DATA

y_test - Series

Index	Production
204272	420936
103709	874
61457	201300
56973	76689
170337	59881
218548	5906
158351	70000
189773	190899
148244	141439
226110	274294
43854	34215
21560	122895
98033	31207
76558	3968.67
95486	52677

FIGURE 6.3.4: Y_TEST DATA

6.4 Random Forest Algorithm

```
Score of Random forest regressor: 0.942578661518394
Prediting the production by taking the test data


predicted values [116865.4821 195235.3      3672.24   417087.22   143606.99    46046.57
 266878.48    96734.33   314188.25    10390.2782]


Actual values 16918     135106.00
197678    193760.00
102174      3551.95
201245    303911.00
185280    159610.00
75045      38470.00
67686     230000.00
45317     149839.00
94522     346950.00
73291       7682.00
Name: Production, dtype: float64
```

FIGURE 6.4: ACCURACY_SCORE

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

This system was proposed for the detection of the crop yield in an effective way using the machine learning algorithms like Random Forest algorithm. The analysis of the experimental results have shown that the proposed model produces an efficient and effective results when compared to the results obtained from the reinforcement learning algorithm.

8.REFERENCES

[1] D. Elavarasan and P. M. D. Vincent, "Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications," in *IEEE Access*, vol. 8, pp. 86886-86901, 2020.

[2] P. Nevavuori, N. Narra, and T. Lipping, ‘‘Crop yield prediction with deep convolutional neural networks,’’ Comput. Electron. Agricult., vol. 163, Aug. 2019, Art. no. 104859.

[3] Q. Yang, L. Shi, J. Han, Y. Zha, and P. Zhu, ‘‘Deep convolutional neural networks for rice grain yield estimation.