# XAI in Agriculture

Submitted By

**Viraj J. Patel**

**22MCEC14**


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**INSTITUTE OF TECHNOLOGY**

**NIRMA UNIVERSITY**

**AHMEDABAD-382481**

**May 2024**

# XAI in Agriculture

**Major Project - II**

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (CSE)


Submitted By

**Viraj J. Patel**

**(22MCEC14)**

Guided By

**Dr. Jigna Patel**




**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**INSTITUTE OF TECHNOLOGY**

**NIRMA UNIVERSITY**

**AHMEDABAD-382481**


**May 2024**

# Certificate

This is to certify that the major project entitled **"XAI in Agriculture"** submitted by **Viraj J. Patel (Roll No: 22MCEC14)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (CSE) of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-I, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Jigna Patel
Guide & Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sudeep Tanwar
Professor,
Coordinator M.Tech - CSE (CSE)
Institute of Technology,
Nirma University, Ahmedabad

Dr. Madhuri Bhavsar
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Himanshu Soni
Director,
Institute of Technology,
Nirma University, Ahmedabad

# Statement of Originality

I, **Viraj J. Patel**, Roll. No. **22MCEC14**, give undertaking that the Major Project entitled "**XAI in Agriculture**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (CSE)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

V. J. Patel

Signature of Student

Date: 21-05-2024

Place: Ahmedabad

21/5/24

Endorsed by

Dr. Jigna Patel

(Signature of Guide)

# Acknowledgements

I am delighted to convey my sincere appreciation and gratitude to **Dr. Jigna Patel**, Assistant Professor in the Department of Computer Engineering at the Institute of Technology, Nirma University, Ahmedabad, for her invaluable counsel and support during the course of this endeavor. her sincere appreciation and unwavering support have served as tremendous inspiration for me to strive for greater accomplishments. Her counsel has stimulated and fostered my intellectual development, to which I shall perpetually be grateful.

It is with great sincerity that I extend my gratitude to **Dr. Madhuri Bhavsar**, Head of the Computer Science and Engineering Department at Nirma University's Institute of Technology in Ahmedabad, for her generous assistance, which included the provision of fundamental infrastructure and a conducive research environment.

A special thank you is expressed wholeheartedly to **Dr Himanshu Soni**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

Additionally, I would like to extend my gratitude to the Computer Engineering Department faculty of Nirma University, Ahmedabad, for their great guidance and helpful suggestions regarding the project.

**Viraj J. Patel**

**22MCEC14**

# Abstract

In order to make an accurate prediction regarding the quantity of cotton that will be harvested, we have compiled a specialized dataset consisting of environmental parameters as part of this particular research project. The purpose of the dataset collection was to collect data on environmental parameters from a variety of locations in Gujarat, and we have utilized geographic information systems in order to accomplish this. For the purpose of utilizing artificial intelligence that can be explained, we decided to concentrate on this particular application that is associated with agriculture. Within the scope of this investigation, we have employed machine learning and deep learning models to make projections regarding yield, and we have utilized XAI models to offer an explanation concerning a particular prediction.

# Abbreviations

| | |
|---|---|
| **XAI** | Explainable Artificial Intelligence. |
| **GIS** | Geographic Information Systems. |
| **MI** | Mutual Information. |
| **RF** | Random Forest |
| **DNN** | Deep Neural Network |
| **ML** | Machine Learning |
| **DNN** | Deep Learning |
| **LIME** | Local Interpretable Model-agnostic Explanations |
| **SHAP** | SHapley Additive exPlanations |

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Explainable Artificial Intelligence, also commonly referred to as XAI, refers to the evolution of machine learning models and algorithms that offer easily comprehensible insight into their decision-making process[2]. As opposed to black box artificial intelligence systems, XAI aims to unveil the intricate hidden algorithm process, allowing users to understand the 'how' and the 'why' behind specific decisions or predicted outcomes. Transparency offered by XAI is vital for fostering trust, holding people accountable, and promoting the use of artificial intelligence in various domains. It ensures that they can justify and reasoned decision from end users, stakeholders and regulators on what the intelligent system's output implies.

Explainable or intelligent artificial intelligence (XAI) is a concept that allows for more

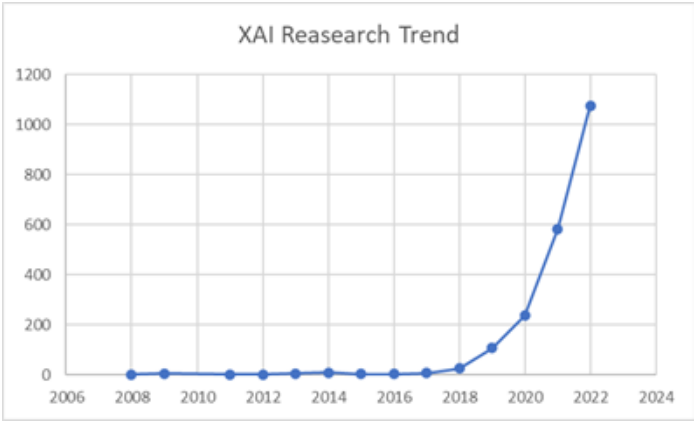


Figure 1.1: XAI Research Trend (in published research papers) [1]

advanced machine learning models that can explain why they made specific decisions in a transparent way. The global artificial intelligence (AI) software market is forecast to grow

rapidly in the coming years as you can see in figure 1.1, reaching around 126 billion U.S. dollars by 2025[3]. In the agricultural sector where precision and efficiency matter most, XAI explains the complex and often invisible elements of artificial intelligence algorithms. However, as the stakeholders and farmers increasingly rely on AI-based technologies to manage crops, check weather conditions and enhance the general agriculture productivity, the AI needs to be easy to comprehend and use. Explainable AI in Agriculture helps end-users to ensure fair decisions and trust modern solutions. This paradigm shift towards transparent AI solutions enhances the robustness and longevity of agricultural systems while giving stakeholders the information they require to take appropriate steps against evolving challenges.

Explainable artificial intelligence (XAI) in agriculture was chosen as a topic because

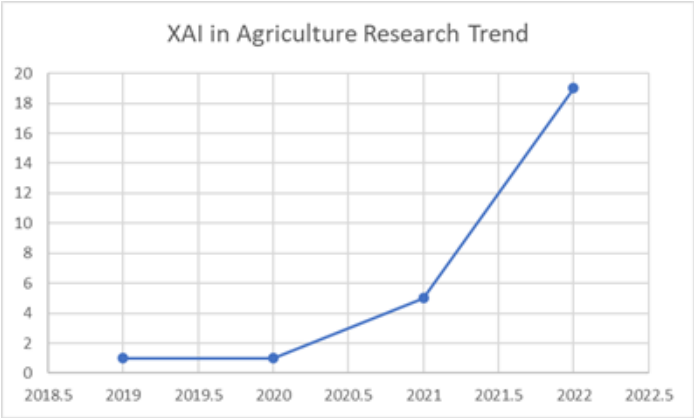


Figure 1.2: XAI Agriculture Trend (in published research papers)[1]

of the growing significance of artificial intelligence in transforming traditional farming practices as you can see in figure 1.2 which is having no. of research papers published on the X-axis and year of publishing is on the Y-axis. This trend is the driving force behind the selection of XAI in agriculture [4]. It is becoming increasingly important for these advanced systems to have decision-making processes that are both transparent and interpretable. This is because artificial intelligence technologies are becoming increasingly integrated into agricultural practices. For the purpose of maximizing crop yields, effectively distributing resources, and ensuring environmental sustainability, it is essential for those working in the agricultural industry to make decisions that are accurate and well-informed. The exploration of the realm of explainable AI in agriculture has a number of objectives, one of which is to gain an understanding of how these technologies can

be made more accessible and understandable to farmers, stakeholders, and policymakers. When it comes to addressing concerns regarding trust and accountability, as well as the seamless integration of artificial intelligence into the existing fabric of agriculture, the motivation lies in addressing these concerns. In light of the rapidly shifting landscape of the world, this will ultimately contribute to the development of farming practices that are more environmentally friendly and productive.

Yield prediction as an agriculture-related application have been selected and applied explainable AI models to it. GIS techniques have been integrated to collect the environmental parameters' data for yield prediction. Data were collected from only areas where the probability of cultivation of a particular crop is high. That is having much importance in collecting accurate (in terms of usefulness in predicting yield from it) environmental parameters' data rather using whole area's data (whole area in terms of non-cultivation area). After collecting data, ML/DL algorithms implemented on that and select the best one out of it. Then, explainable AI models implemented to the best ML/DL algorithms[5] for explaining why a particular number came up as a prediction for yield based on the data provided.

In a number of studies, the user is required to possess a particular set of skills in order to implement it in real time. The usability of this research is what makes it novel. after the models have been trained and tested, and after it has been finalized. It does not require a specific skill set for anyone to be able to successfully use it.

## 1.1 Objectives

- Collect relevant agricultural data using Geographic Information Systems.
  - Focus on target regions and crops to gather precise, reliable envirnomental data using satellite imagery and geospatial analysis.
- Develop high-performance yield prediction models
  - Implement machine/deep learning algorithms to forecast crop yield.
  - Optimize models for accuracy on collected agricultural data sets.
- Apply explainable Al techniques to agricultural machine learning models

– Understand internal workings of complex models when applied to agricultural data.

– Explain how models arrive at yield predictions based on environmental parameters.

# Chapter 2

# Literature Survey

Throughout the following paragraphs, each of them has a process or goal that is similar to the one that is being investigated, such as yield prediction, applying XAI models to ML/DL models, or processing GIS images. Even though the database is unique to this study, methods that have been used in similar studies in the past can also be used in this study with similar goals. Following are the summary of the papers which are reviewed you can find it in tabular form (see table 2.1)

Celik et al. (2023) [6] put forward an explainable boosting machine (EBM) approach for cotton yield forecasting, integrating satellite data, climate records, and soil attributes. Compared to SVM, random forests, XGBoost and LightGBM, benefits were model interpretability and accurate yield predictions while quantifying influential features without needing post-hoc techniques. Limitations were slower training than some models and lack of geographic/remote sensing inputs currently. Key drivers identified were precipitation, vegetation indices, and leaf area, while static soil properties contributed less. Future work involves adding geographic data and evaluating explanation methods like Grad-CAM.

Quach et al. (2023) [7] examined VGG16, ResNet50, MobileNet and other CNN architectures for plant disease recognition, utilizing gradient-weighted class activation maps to explain model reliability. A constraint was limited samples available presently. MobileNet achieved very high accuracy, though overlaying regions triggering predictions against expert disease knowledge showed EfficientNetV2 and Xception as most robust. Next steps entail deploying the model operationally via apps, and developing global explanation approaches.

Bandi et al. (2023) [8] designed an end-to-end pipeline combining YOLOv5 disease

detection followed by Vision Transformer models for fine-grained apple leaf damage severity quantification. Comparisons with and without background removal pre-processing boosted classifier performance. Expanding to more plant categories and disease types offers future direction.

Mehedi et al. (2023) [9] assessed leading deep CNN models including EfficientNetV2L, MobileNetV2, and ResNet152v2 for plant disease distinction, employing locally interpretable model-agnostic explanations to clarify model rationales, thereby increasing user trust. Almost 99.6% accuracy was attained, though larger annotated datasets are still wanting. Testing web/mobile deployment and devising more advanced explanation techniques provide next steps under limited resources.

Sun et al. (2019) [10] introduced a CNN-LSTM model fusing convolutional neural networks' prowess in extracting visual features from satellite imagery with long short-term memory networks' sequence modeling capabilities suited for temporal climate data to forecast soybean yields. Outperformance over individual CNNs and LSTMs was exhibited but currently restricted to few environmental inputs available. Incorporating more data modalities constitutes an advancement opportunity.

Arvind et al. (2021) [11] leveraged deep neural networks for plant disease categorization, followed by applying LIME and Grad-CAM methods to offer accompanying explanations for each prediction. Across CNN architectures evaluated, EfficientNet B5 proved optimal further enhanced via fine-tuning, with YOLOv4 object localization confirming explanation validity. Core limitations rest with significant computational burdens for such complex models. Analyzing diseased leaves showing multiple simultaneous infections provides next phase research.

Wolanin et al. (2020) [12] devised a deep learning pipeline with regression activation mapping to predict and shed light on wheat yield drivers across agricultural areas of India, outperforming baseline random forest and ridge regression approaches. Scope remains for extending to more geographic regions cultivating wheat to improve generalizability and better aid real-world decision making.

Viana et al. (2021) [13] used random forests, PFI, PDPs, and LIME to explain factors influencing agricultural land use. Key findings were that drainage, slope, and soil type strongly affect land use. A limitation was availability of variables related to political and cultural factors. Advantages included capturing complex system behaviors for land

suitability analysis.

Ryo (2022) [14] showcased interpretable ML methods like SHAP, variable importance plots, and partial dependence plots. Multiple tree-based, SVM, and neural network models were compared. A key finding was that no-tillage increased crop yield under certain conditions. A limitation was that analysis was specific to one dataset without causality testing.

Cartolano et al. (2022) [15] applied SHAP and LIME on crop recommendation data using XGBoost, SVM, and neural networks. Models struggled to distinguish some crops with certain features more influential. A limitation was reliability concerns with post-hoc explanation methods.

Lundberg (2017) [16] introduced SHAP values as a unified measure of feature importance. A contribution was defining an additive feature attribution method satisfying properties like consistency. There was no detailed discussion on limitations or models used.

Ribeiro et al. (2016) [17] present LIME for explaining individual predictions by locally approximating complex models. Tradeoffs between interpretability and local fidelity were highlighted. Benefits include model-agnostic and modular properties.

Letzgus et al. (2023) [18] proposed extending XAI techniques to handle regression problems using layer-wise relevance propagation. Proposed methods outperformed baselines without evaluating end-user benefits. A limitation was the computational expense of retraining complex models.

Dhaliwal et al. (2022) [19] compared algorithms like linear regression, random forests, and neural networks for predicting cotton yield and interpreting key determinants. Findings were that management practices were more influential than climate in increasing yields. A limitation was potential overfitting with the single-site dataset used.

Dieber (2020) [20] compared machine learning models such as decision trees, random forests, logistic regression, and XGBoost to explain rainfall predictions using the LIME technique. Findings were that while LIME improved interpretability, there were limitations around the completeness of local explanations, documentation gaps, and the potential for misinterpretation. A limitation was reliance on a single dataset. Recommendations included enhancing documentation, developing complementary global explanation tools, and benchmarking against alternative frameworks to address these issues.

| Title | Method/Approach | Limitations | Advantages | Disadvantages | Future Approach | Key Findings | Dataset | Models Used |
|---|---|---|---|---|---|---|---|---|
| Explainable Artificial Intelligence for Cotton Yield Prediction With Multi-source Data [6] | "Used explainable boosting machine (EBM) for cotton yield prediction; integrated remote sensing data, climate data and soil data" | Does not consider geographical features or SAR/MSI data for yield prediction | Interpretable and accurate for prediction; can quantify feature importances without post-hoc methods | Slower training than black box models | Incorporate geographical data and SAR/MSI data; evaluate Grad-CAM interpretability | "Key features driving model are precipitation, EVI and LAI; static features less important" | "Multisource dataset of satellite, climate and soil data" | EBM |
| Using Gradient-weighted Class Activation Mapping to Explain Deep Learning Models on Agricultural Dataset [7] | "Used VGG16, ResNet50, MobileNet for disease classification; used Grad-CAM for interpretability" | Limited dataset size; model not yet deployed for practical use | High accuracy for MobileNet; Grad-CAM shows model reliability | | Deploy model via computer/mobile apps; develop global vs. local XAI | MobileNet highly accurate but EfficientNetV2 and Xception more reliable for features | Plant disease image dataset | CNN models + Grad-CAM |
| Leaf disease severity classification with explainable artificial intelligence using transformer networks [8] | "Detected diseases with YOLOv5, classified severity with Vision Transformer" | Limited to apple leaf diseases | End-to-end pipeline from detection to classification/recommendation | | Include more plant species and diseases | Background removal improves ViT classifier performance | Images of diseased plant leaves | YOLOv5 + Vision Transformer |
| Plant Leaf Disease Detection using Transfer Learning and Explainable AI [9] | "Used EfficientNetV2L, MobileNetV2, ResNet152V2 for detection, LIME for interpretability" | Data and infrastructure limitations | High accuracy for disease detection; model transparency | | Include more diseases and larger dataset | EfficientNetV2L best performer with 99.63% accuracy | Images of diseased plant leaves | Transfer learning CNN models + LIME |
| County-Level Soybean Yield Prediction Using Deep CNN-LSTM Model [10] | "Proposed CNN-LSTM model using satellite, weather and soil data" | Limited environmental variables used | Integrates spatial and temporal features; outperforms CNN/LSTM | | Incorporate more environmental data; optimize model architecture | MODIS data more predictive than environmental data | "Satellite, weather and crop yield data" | CNN-LSTM |
| Deep Learning Based Plant Disease Classification With Explainable AI and Mitigation Recommendation [11] | "Used deep neural networks for plant disease classification, then applied LIME and Grad-CAM for explainable AI, and validated explanations using YOLOv4" | Not evaluated for multiple diseases per leaf | High accuracy with deep learning; Improved trust and explainability using XAI methods | Computationally expensive models | Test multi-model approach for leaves with multiple diseases | EfficientNet B5 gave best accuracy; Explainable AI enhances trust in predictions | Tomato plant disease images from PlantVillage dataset | "EfficientNet B5, LIME, Grad-CAM, YOLOv4" |
| Estimating and understanding crop yields with explainable deep learning in the Indian Wheat Belt [12] | Used CNN with regression activation mapping for wheat yield estimation in India | Focused only on wheat yield | Improved predictive performance over baseline models; Interpretability of complex DL models | | | Length of growing season and light/temperature conditions are key drivers of yield variability | "Meteorological, vegetation and crop yield data" | "CNN, Regression Activation Mapping" |
| Evaluation of factors explaining agricultural land use - ML & model-agnostic approach[13] | "Used random forest + PFI, PDPs and LIME for explaining factors influencing land use" | Limited variables available related to political/cultural factors | Captures complex behaviors of land use systems; Useful for land suitability analysis | | Test approach across different geographic contexts and crops | "Drainage, slope and soil type strongly influence land use" | "Biophysical, bioclimatic and agricultural socioeconomic data" | "Random forest, PFI, PDP, LIME" |
| "Explainable artificial intelligence and interpretable machine learning for agricultural data analysis"[14] | "Showcased various interpretable ML methods like SHAP, variable importance, partial dependence plots etc." | Specific to one dataset; Did not test causality | "Useful for discovering patterns, interactions; Enhances model trust" | Using post-hoc methods instead of inherently interpretable models can be limited | Communicate discoveries with domain experts to understand mechanisms | Identified under which conditions no-tillage can increase crop yield | Dataset on effect of no-tillage on crop yield | "Multiple tree-based, SVM, neural network models" |

| Title | Method/Approach | Limitations | Advantages | Disadvantages | Future Approach | Key Findings | Dataset | Models Used |
|---|---|---|---|---|---|---|---|---|
| "Explainable AI at Work! What Can It Do for Smart Agriculture?" [15] | Applied SHAP and LIME on crop recommendation dataset | Some reliability concerns with SHAP/LIME | Visual interpretations to understand model behavior | | Improve approach with more XAI methods | Models tend to confuse certain crops; Some features more influential | Crop recommendation dataset | "XGBoost, SVM, neural networks" |
| A Unified Approach to Interpreting Model Predictions [16] | Introduces SHAP values as a unified measure of feature importance that various methods approximate | Model-agnostic; faithful to models; consistent with human intuition based on experiments | Not discussed | Develop faster model-specific estimation methods; integrate work on estimating interaction effects; define new explanation model classes | "There is a unique additive feature attribution method that satisfies several desirable properties (consistency, local accuracy, missingness)" | Introduced SHAP model | Not clearly specified | |
| "Why Should I Trust You?: Explaining the Predictions of Any Classifier" [17] | LIME: Locally approximating models to explain individual predictions in an interpretable manner. Compared against interpreting gradients directly and global approximation with Parzen windows | Model-agnostic; modular & extensible | Explanations have a tradeoff between interpretability and local fidelity | Explore different explanation families; further experiments in speech/video/medical domains; explore computational optimizations | "Explanations are useful for range of models and tasks like improving trust, detecting data issues, model selection" | Introduced LIME model | | "Random forests, neural networks, SVMs, etc." |
| Toward Explainable AI for Regression Models[18] | Layer-wise Relevance Propagation (LRP) Extending XAI techniques designed for classification to handle regression problems | Did not evaluate benefit to end user | Preserves units of measurement in explanations | computationally expensive to retrain complex models | Incorporate model uncertainty into explanations | Proposed methods outperform baselines for explaining regression models | Low-dimensional synthetic datasets and real-world image and chemical datasets | Neural networks |
| Predicting and interpreting cotton yield and its determinants under long-term conservation management practices using machine learning[19] | " Compared performance of algorithms(Linear regression, ridge regression, lasso regression, random forest, XGBoost, artificial neural network) for predicting historical cotton lint yield and identifying key determinants" | Single site data constrains model's prediction domain; needs wider range of training data | Captured complex variable relationships; useful insights for soil health management | Risk of overfitting with limited data; difficult to interpret complex models like RF and ANN | Include multi-site data covering wider output and predictor ranges to improve model robustness | Management practices more influential than climate; optimized N rate and legume cover crop can increase cotton yield | "32-years of yield, management, climate and soil data from a long-term cotton experiment" | "Random forest, XGBoost (best performance)" |
| Why model why? Assessing the strengths and limitations of LIME [20] | Train models on rainfall data; Use LIME for local explanations; Evaluate via user study and ISO standdss | Local explanations incomplete; Lack of documentation | Provides local feature importance; Fast explanation | Global analysis requires manual effort; Potential for misinterpretation | Improve documentation and user experience; Develop global analysis tools; Benchmark against other frameworks | LIME improves interpretability but usability needs enhancement | Rainfall prediction data | "Decision tree, random forest, logistic regression, XGBoost" |

Table 2.1: Literature Review Table

# Chapter 3

# Methodology

This chapter provides information about methodology. As we can see in figure 3.1, it has been divided into three parts, with the green part giving information about the data collection process, for which information is available in section 3.2, the blue part showing an overview of the yield prediction method; more details are present in sections 3.3 and 3.4 and the orange part giving information about explanation.

## 3.1 Proposed Architecture

## 3.2 Data Collection

General understanding of GIS data, There are spatial coordinates in it that show where the features are. The metadata that goes with these data sets is stored in scientific data formats that people who study Earth science use[21]. The authentication of collected data is given below 3.1 for every parameter it uses the metadata from related spatial co-ordinates. The dataset that is used for the collection of data for that particular parameter is provided.

Authors have generated own dataset using geo-spatial data and process using GEE. The significance of data is the data is collected from the locations on which cultivation is going on. for extraction of that area, we have got data by Global Food-and-Water Security-support Analysis Data [22].

### 3.2.1 Selection of Region Of Interest

For collecting data we have to first get the region of interest based on past data. We have particularly selected cotton crop to do yield prediction.Why we have selected some

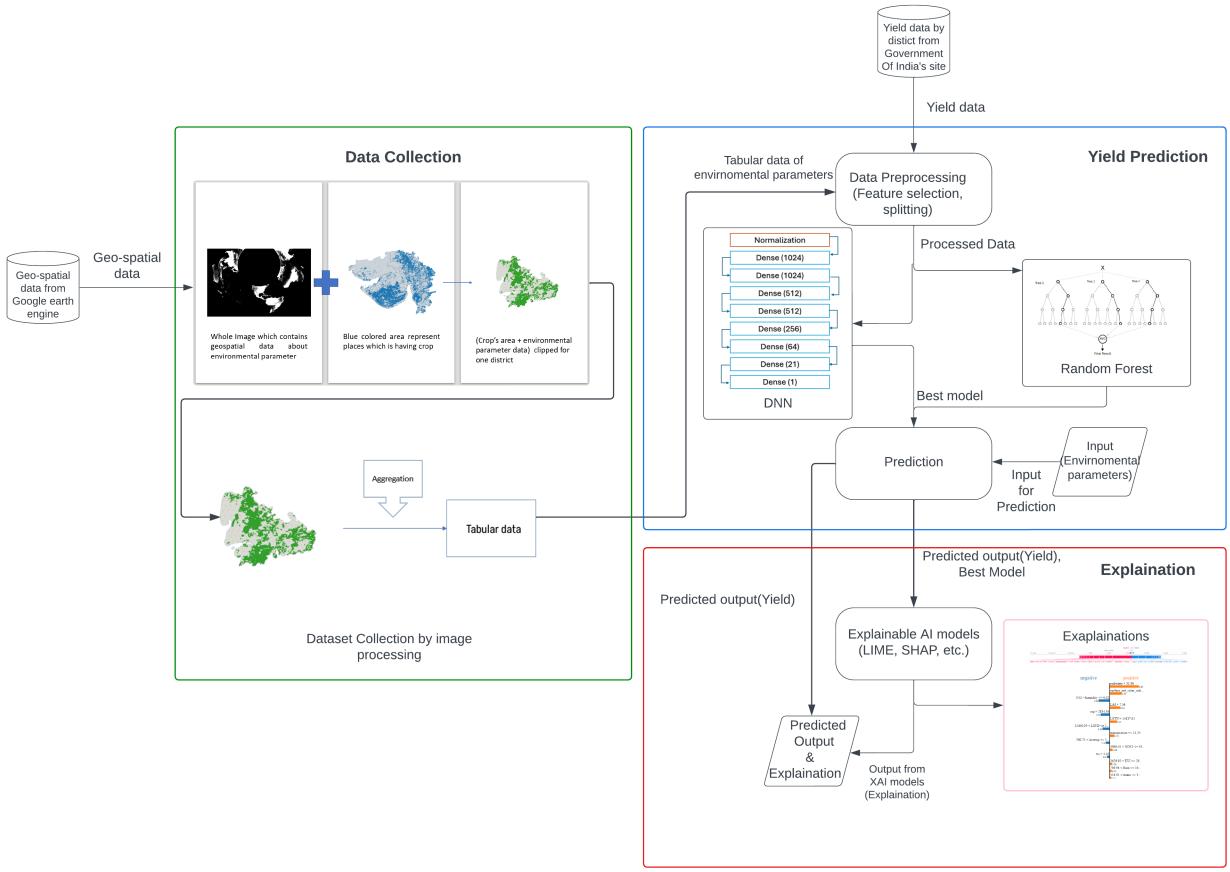


Figure 3.1: Overview of method

areas of major cotton producing states as our region of interest is because it is one of the major producer of cotton as per provided government document[23]. So, we have got yield data by districts of major cot using web-site from government of India [24]. Some areas of Gujarat is considered major farming zone of cotton which is needs to be sorted. So we removed low production districts from region of interest(which is displayed in 3.3 on right side, red part showing ROI).

### 3.2.2 Processing satellite images

Google earth engine have been used to process image data and extract data into required format. To extract data that we have got the image data from dataset as input data. We need to get the required band of the image from that image we have masked in our ROI (Cotton cultivation area)(figure 3.3(on right)) which is present in the middle of the figure 3.4. after masking on ROI, this ROI image (which is having parameter's data on top of that) clipped to one district.

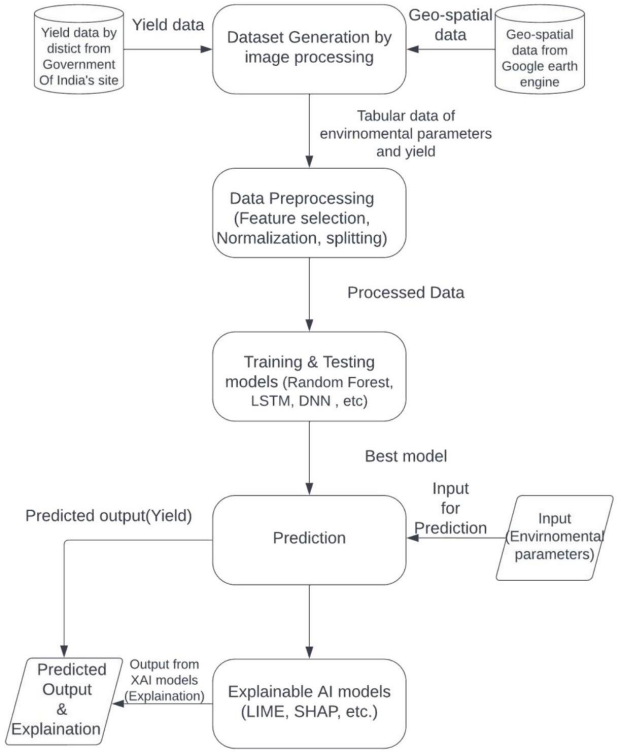


Figure 3.2: Proposed Architecture

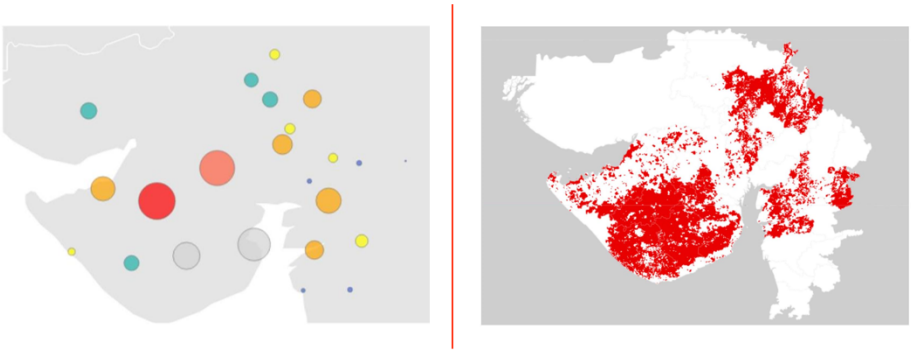

Figure 3.3: Left: Plot of cotton production per district bigger the circle larger the yield | Right: ROI from filtered districts

### 3.2.3 Data conversion process: Processed satellite image to tabular format data

After clipping it for a district, every pixel has its coordinates and parameter's value. We fetched the parameter's value by applying an aggregation technique (by using reducer method from google earth engine) and extracted the mean of that parameter's value from the image we have clipped as shown in output of figure 3.2. We have done this

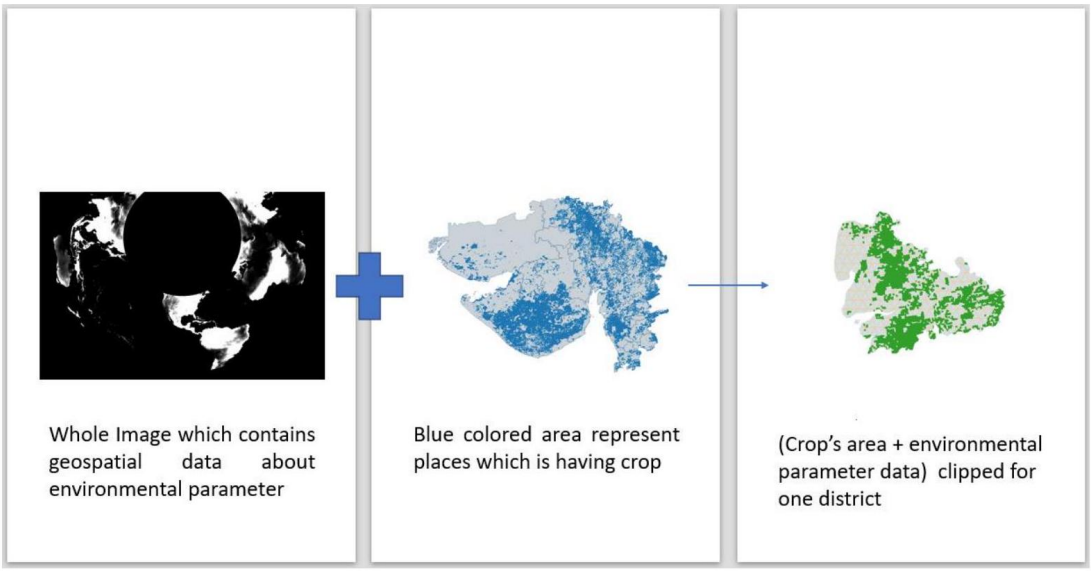


Figure 3.4: Processing satellite images

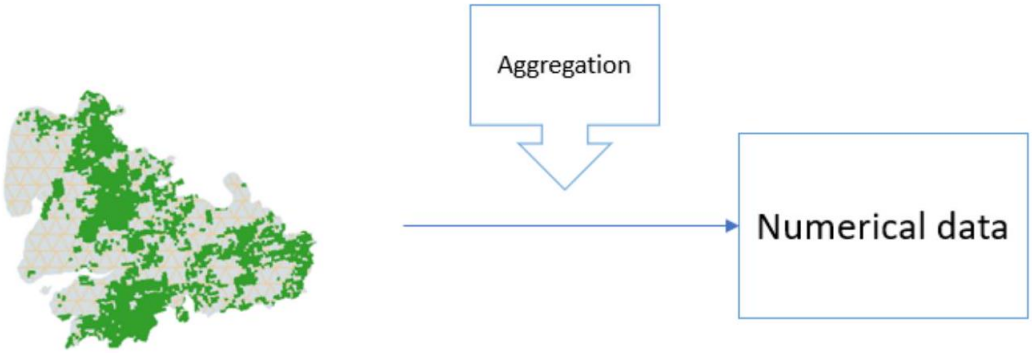


Figure 3.5: Processing satellite images

process for every district we are interested in for 22 years' data for every parameter mentioned in table 3.1 which is mentioned in the above point.

### 3.2.4 Parameters of dataset

| Parameters | Dataset |
|---|---|
| Soil Moisture | GLDAS-2.1: Global Land Data Assimilation System |
| Wind Speed | GLDAS-2.1: Global Land Data Assimilation System |
| Humidity | GLDAS-2.1: Global Land Data Assimilation System |
| Transpiration | GLDAS-2.1: Global Land Data Assimilation System |
| Rainfall | CHIRPS Daily: Climate Hazards Group InfraRed Precipitation with Station Data (Version 2.0 Final) |
| surface net solar radiation | ERA5-Land Hourly - ECMWF Climate Reanalysis |
| Temperature | ERA5-Land Hourly - ECMWF Climate Reanalysis |
| Soil pH | OpenLandMap Soil pH in H2O |
| Leaf Area Index | MOD15A2H V6.1 MODIS combined Leaf Area Index (LAI) and Fraction of Photosynthetically Active Radiation (FPAR) |
| Fraction of Photosynthetically Active Radiation | MOD15A2H V6.1 MODIS combined Leaf Area Index (LAI) and Fraction of Photosynthetically Active Radiation (FPAR) |
| Vapor pressure deficit | TerraClimate is a dataset of monthly climate and climatic water balance for global terrestrial surfaces. |
| Vapor pressure | TerraClimate is a dataset of monthly climate and climatic water balance for global terrestrial surfaces. |
| Maximum temperature | TerraClimate is a dataset of monthly climate and climatic water balance for global terrestrial surfaces. |
| Minimum temperature | TerraClimate is a dataset of monthly climate and climatic water balance for global terrestrial surfaces. |
| Runoff | TerraClimate is a dataset of monthly climate and climatic water balance for global terrestrial surfaces. |
| Day land surface temperature | MOD11A2 V6.1 product provides an average 8-day land surface temperature (LST) |
| Night land surface temperature | MOD11A2 V6.1 product provides an average 8-day land surface temperature (LST) |
| NDVI | MOD13A2 V6.1 product provides two Vegetation Indices (VI): the Normalized Difference Vegetation Index (NDVI) and the Enhanced Vegetation Index (EVI) |
| EVI | MOD13A2 V6.1 product provides two Vegetation Indices (VI): the Normalized Difference Vegetation Index (NDVI) and the Enhanced Vegetation Index (EVI) |
| NDWI | MODIS Terra Daily NDWI |

Table 3.1: Parameters

## 3.3 Data Preprocessing

### 3.3.1 Feature selection

We have used Mutual Information (MI) [25] to extract features from dataset. It is a method used to assess the statistical dependence between features and the target vari-

able in a dataset. It measures the degree to which knowing the value of one variable reduces uncertainty about the other, thereby quantifying the amount of information one variable provides about another. Mutual information is calculated for every feature in the feature extraction process in relation to the target variable. While features with low mutual information might not be as relevant to the prediction task, those with high mutual information are thought to be informative. This method is especially useful for identifying features that are highly correlated with the target, which aids in the identification of significant predictors and enhances the effectiveness and interpretability of machine learning models.[26] For continuous random variables, the summation is replaced with integration:

where:

$$I(X;Y) = \iint p(x,y)\log(p(x,y))/(p(x)p(y))\Big)\,dxdy$$

- $p(x,y)$ is the joint probability density function of $X$ and $Y$.
- $p(x)$ and $p(y)$ are the marginal probability density functions of $X$ and $Y$, respectively.

```
Algorithm: Mutual Information
1.  X←{temperature, soilph,
    LSTD,LSTN,transpiration,soilmoisture,windspeed,vpa,vpd,fpar,lai,maxtemp,mintemp,runo
    ff,NDVI,EVI,NDWI,rainfall,radition,humidity}
2.  Y={yield}
3.  Miscore ←{}
4.  function MI(X,Y)
5.      Initialize I(X;Y) = 0
6.          for each possible value x of X
7.                  for each possible value y of Y
8.                          p(x,y) = probability P(X=x, Y=y) from the joint distribution
9.                          p(x) = probability P(X=x) from the marginal distribution of X
10.                         p(y) = probability P(Y=y) from the marginal distribution of Y
11.                         if p(x,y) > 0
12.                                 I(X;Y) += p(x,y)*log(p(x,y)/(p(x)*p(y)))
13.                         end if
14.                 end for
15.         end for
16.   return I(X;Y)
17. end function
18. for each x of X
19.     Miscore←add output of function(x,Y) into array.
```

Figure 3.6: Mutual Information Algorithm

As you can see algorithm in 3.6, We first define X as the input variable, representing a set of features like temperature, soil pH etc. Y is defined as the output variable, in this

case the crop yield. The MI() function calculates the mutual information score between any given variable X and the output Y. We initialize the MI score I(X;Y) to 0. Then, we iterate through every possible value x that X can take in its distribution. For each x, we iterate through every possible value y that Y can take. We calculate the joint probability p(x,y) of X=x and Y=y occurring together based on the joint distribution. Additionally, we calculate the marginal probabilities p(x) and p(y) of the individual variables X and Y independently from their marginal distributions. For each combination of x and y that has a non-zero joint probability, we calculate an MI contribution score of p(x,y)*log(p(x,y)/(p(x)*p(y))) using the standard MI equation. We add this to the overall MI score. After iterating over all combinations of x and y and summing their MI contributions, we get the final MI score between that variable X and output Y. We repeat this whole process for each variable in the input feature set X. The higher the MI score, the more information X contains about predicting or explaining Y. As a result (see the figure 3.7), in this dataset we are getting postive MI score for every feature. So, we can say that every feature is independent from each other and not dependent on other features. Also, every feature contains information about yield.

Why Mutual information (MI)? It is chosen as a feature extraction technique for its versatility and effectiveness in capturing both linear and non-linear relationships between variables. Unlike some linear methods, MI makes no assumptions about the data distribution and is non-parametric, allowing it to handle diverse datasets without relying on specific mathematical models. Its information-theoretic foundation provides a principled way to quantify the information shared between features and the target variable, making it suitable for understanding the information content of features. MI is particularly valuable for variable selection, helping to identify the most informative features for a prediction task and assess redundancy among features. Its insensitivity to scaling, model-agnostic nature, and robustness to different units or scales make it a flexible choice applicable to various machine learning scenarios. While MI is a powerful technique, its selection depends on the specific characteristics of the dataset and the analytical goals, often complementing other feature selection methods for a comprehensive approach. for comparison of mi to other feature selection techniques see table 3.2.

| Technique | Key Differences from Mutual Information |
|---|---|
| Pearson Correlation | Only captures linear relationships |
| Uni-square test | Specific to tree ensemble models, less Reneralizable |
| Feature Importance | Faster approximation but less accurate than mutual information |
| Variance Threshold | Simply removes low variance features, no pairwise evaluations |
| Exhaustive Feature Selection | Tries all possible feature combinations, computationally infeasible |
| Recursive Feature Elimination | Model-specific, computationally intensive |
| Principal Component Analysis | Sensitive to relative scaling of features |
| Statistical Similarity Measures | Specific similarity measures less grounded in information theory |
| Genetic Algorithm Search | Computationally expensive, requires wrapper |
| Neural Network Feature Extraction | Requires training a full neural network |
| Hybrid Filter-Wrapper Method | More complex implementation |

Table 3.2: comparison of mi to other feature selection techniques

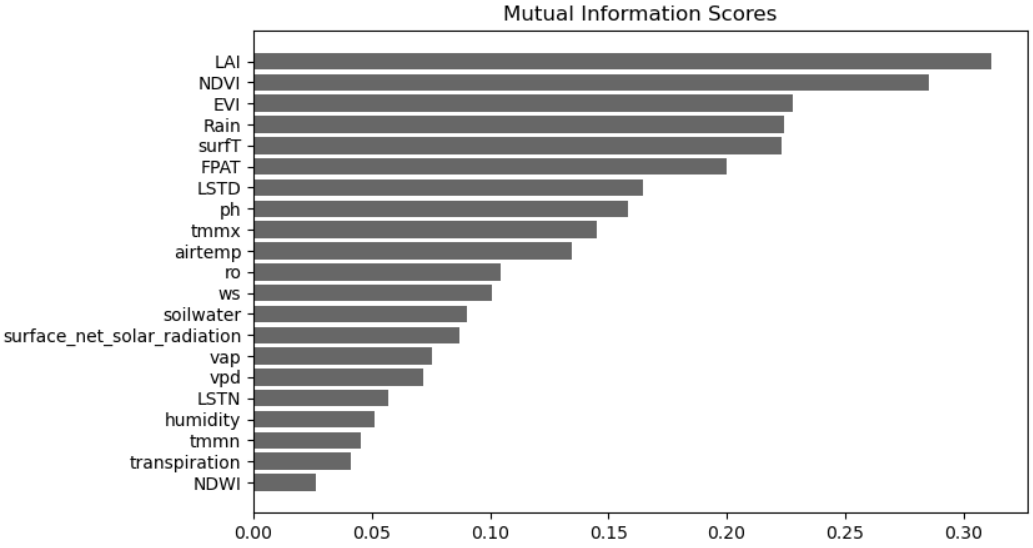


Figure 3.7: MI score

## 3.4 Models Used For Prediction (ML/DL models)

In existing researches about yield prediction, most have good accuracy from Random forest and deep neural networks for tabular data. for image data, they have good accuracy by using CNN, CNN-LSTM, etc. here, tabular data were used as mentioned in section

3.2. So, Random forest and deep neural networks were used

### 3.4.1 Random Forest

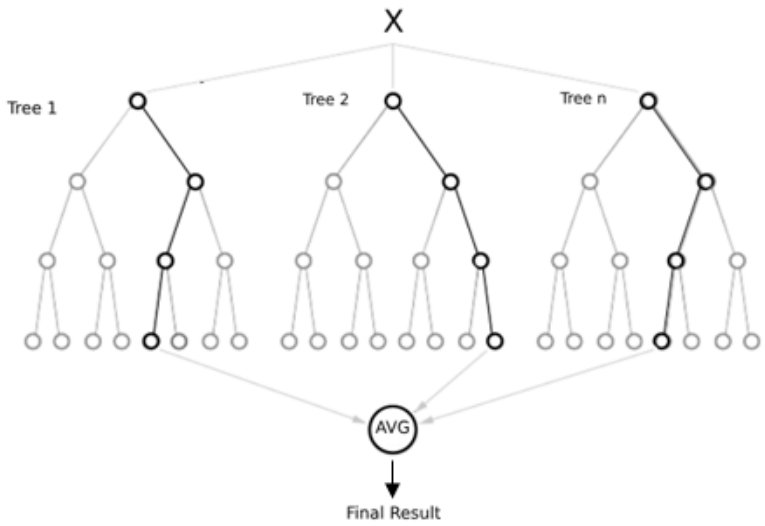


Figure 3.8: Random Forest architecture

**Algorithm** : Random Forest

**Precondition:** A training set $S := (x_1, y_1), \ldots, (x_n, y_n)$, features $F$, and number of trees in forest $B$.

```
1  function RandomForest(S, F)
2      H ← ∅
3      for i ∈ 1, ..., B do
4          S^(i) ← A bootstrap sample from S
5          h_i ← RandomizedTreeLearn(S^(i), F)
6          H ← H ∪ {h_i}
7      end for
8      return H
9  end function
10 function RandomizedTreeLearn(S, F)
11     At each node:
12         f ← very small subset of F
13         Split on best feature in f
14     return The learned tree
15 end function
```

Figure 3.9: Random Forest algorithm

For machine learning models, we have used Random Forest regression [27] which is

less sensitive to scale of input data. The reason for this insensitivity lies in the nature of decision trees, the base learners used in a Random Forest. Decision trees make splits in the data based on feature thresholds. These splits are determined by comparing feature values to certain thresholds, and the decision to split is not influenced by the scale of the features. Therefore, whether a feature is on a small or large scale doesn't affect the decision-making process of individual trees [28]. A random forest regressor is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the mean prediction of the individual trees. Here is an overview of how it works:

- Bootstrap Samples: The training dataset is sampled randomly with replacement to create multiple bootstrap training sets. Each set will have same number of instances as original dataset, but some instances will be repeated.

- Decision Trees: A decision tree regressor is trained on each bootstrap sample. When splitting nodes during tree creation, rather than searching for the best split among all features, a random subset of features is searched at each node. This results in a greater tree diversity, since different trees will use different features.

- Ensemble Prediction: At prediction time, unseen samples are pushed down each decision tree to obtain individual tree predictions. These predictions are then averaged to get an ensemble prediction for that sample.

- Reduce Overfitting: By creating random variations in the tree creation and then ensembling, the correlation between individual trees decreases. This results in reduced variance and guards against overfitting compared to using a single tree.

  The randomness injected into the model aims to de-correlate the trees so that the averaging process can reduce variance and improve generalizability. The number and depth of trees are important tuning parameters. Overall, random forests balance accuracy and robustness well.

Few reasons why random forest regression[29] would be a good choice for modeling:

- Non-linear relationships: Variables like transpiration, LAI etc. can have complex non-linear relationships with the predictors. The ensemble of decision trees in random forests is able to model such non-linearities.

- No need for scaling/transformations: Numerical features in the data have very different scales. But tree-based models do not require scaling or normalization of features. Random forests can handle this raw data.

- Avoid overfitting: With 25 potential predictors for modeling yield, overfitting could be a concern. The ensemble approach of random forests along with random feature selection naturally avoids overfitting.

- Interpretability: Compared to black box models like neural networks, random forest feature importance scores provide some model interpretability. We can identify which variables are most influential.

### 3.4.2 Deep Neural Networks

Agricultural systems present intricate connections between dynamic factors impacting crop growth and yield. Modeling these complex data relationships requires flexible function approximators [30]. Deep Neural Networks (DNNs) provide a powerful modeling approach through hierarchical feature learning. The time-series data reflects the influence of weather, soil conditions, plant indicators etc. on yield across the growing cycle. DNNs can implicitly learn appropriate intermediate representations from raw data in end-to-end fashion without hand-engineered inputs. Their deep architecture of multiple neural layers enables capturing sophisticated data patterns like nonlinear variable interactions. Sophisticated generalization can prevent overfitting the training data. Weights across millions of parameters are tuned automatically via backpropagation and gradient descent optimization.[31] Daily measurements can be provided sequentially as inputs or aggregated into a feature vector per growing season. Both convolutional and fully-connected DNN architectures can map these inputs to an accurate yield forecast. The representation depth of DNNs suits the intricacies of agricultural systems, providing high accuracy without restrictive assumptions. DNN modeling empowers data-centric precision agriculture to improve decision making and productivity.
A deep neural network (DNN) has several layers that work together to make predictions from data.

- Input Layer: The first layer is the Input Layer. It receives the raw data. The number of nodes here matches the number of features in the input data. This layer does not perform any calculations. It just passes the data to the next layers.

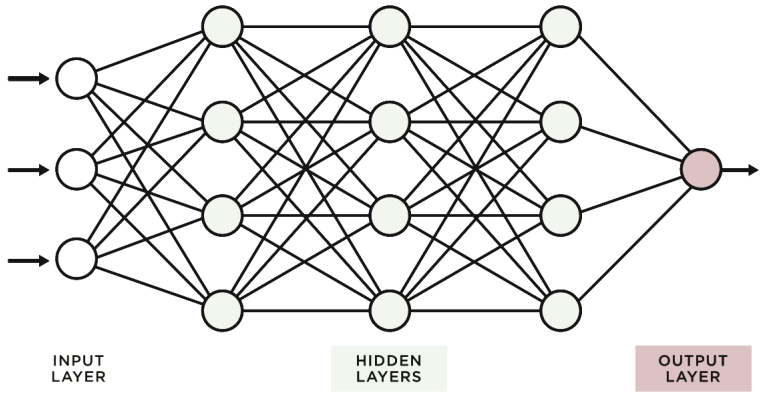


Figure 3.10: General Architecture about Deep Neural Network

- Hidden Layer: Then there are one or more Hidden Layers. These are fully connected layers that transform the input data into meaningful patterns. The more hidden layers, the more complex patterns the DNN can find. The number of hidden layers and nodes in each layer can be changed so the DNN learns better. Hidden layers use activation functions like ReLU to introduce non-linearity.

- Output Layer: The last layer is the Output Layer. This makes the final predictions or conclusions from the data patterns. it is regression, a linear activation gives the number prediction.

- Loss Function: the Loss Function measures how far the predictions are from the true targets. The optimizer improves the DNN by updating connection weights to reduce the loss. Stochastic gradient descent is commonly used. Techniques like dropout also enhance performance. In simple terms, raw data enters the DNN, hidden layers extract meaningful patterns, output layer makes predictions, loss function compares them to truth, and optimizer iteratively improves the DNN. The depth and automated learning let DNNs excel at many predictive tasks.

When training deep learning models, it is very useful to apply normalization to each input feature in the data. This process is called feature-wise normalization. What this means is that we standardize the values going into the model by adjusting the mean and variance. For example, a feature like temperature initially may have a range from 20 to 35. After normalization, that feature will have an average value close to 0 and similar variance around 0. This puts different features onto a common baseline. There is a convenient

Normalization Layer we can include before the main neural network layers to automatically carry out this input feature normalization. The layer does the work of scaling the mean and variance of each input. Performing this input-level normalization provides important benefits - it enables much faster training of deep models through stabilization and helping optimization algorithms like gradient descent converge properly. The specifics like type of normalization and direction to apply it depends on the problem. But in general, feeding normalized features to neural networks unlocks more representational power through smoother training. The models can learn richer underlying patterns.

Here are some key reasons why using a Deep Neural Network (DNN) model would be a good choice for available data:

- Captures Complex Relationships: There are likely complex nonlinear relationships between various weather, soil, and plant growth parameters that collectively impact crop yield over time. The representation depth of a DNN would help model these intricate data patterns.

- Temporal Data Patterns: The time series nature of data can display long-term and short-term trends useful for prediction. A deep architecture can extract relevant temporal features efficiently.

- Multivariate Dependencies: Multiple parameters interact in a sophisticated way to influence yield. Dense neural connections can capture higher-order multivariate dependencies.

- Avoid Overfitting: Lots of input variables leads to chance of overfitting training data. Combination of depth and regularization mechanisms in DNNs guards against overfitting.

- End-to-End Learning: The lack of need for complex data preprocessing or feature crafting makes DNN modeling easier to apply and tune on this data.

A summary of the deep neural network architecture used in this research is defined using a sequential approach. It consists of the following layers, which are displayed in figure 3.11:

- A normalization layer that applies feature-wise normalization to the 21 input features.

- A series of dense (fully connected) layers with decreasing numbers of units: 1024 → 1024 → 512 → 512 → 256 → 64 → 21 →1. These layers extract higher-level features from the input data.

- A final dense layer with a single unit, for predicting a continuous target variable (Yield).

The deep architecture with multiple hidden layers allows the network to learn complex nonlinear relationships between the input features and the target variable. The normalization layer helps with faster and more stable training, while the dense layers progressively extract relevant representations from the data.

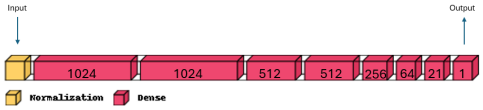


Figure 3.11: Architecture of Deep Neural Network

In summary, the intricacies in agricultural systems, presence of temporal data, and number of variables make using an adaptable non-linear model like a Deep Neural Network well-suited for the task of yield forecasting using this dataset.

## 3.5 Explainable AI

Explainable AI models are used to explain the predictions given by the DNN model.

### 3.5.1 SHAP

Shapley Additive exPlanations (SHAP) [16] is a unified model interpretation approach developed in 2017 to explain individual predictions from any machine learning model by attributing impact of features on the prediction through a connection to game theory.[32] Specifically, SHAP treats the ML model as a function representing a cooperative game between input features that contributes to producing the output prediction. Shapley values from game theory are then leveraged to assign each feature an importance value for a particular prediction. These SHAP values quantify how much influence each feature had in moving the prediction away from the average model output over the entire training set, known as the base value. By adhering to principles of fairness in cooperation from game theory, SHAP provides consistent, intuitive feature contribution explanations. As a

model-agnostic method that encapsulates feature interactions, SHAP has become widely used in practice for inspecting models and debugging to improve transparency and trust. Tailored computational methods exist for different model types with TreeSHAP being a popular variant. Overall, the key innovation in SHAP is grounding explanation of prediction attribution in mature game theoretic concepts to better understand both global and local behaviour of complex modern machine learning models.A high-level overview is provided in figure 3.12. A high-level overview of how SHAP technically works:

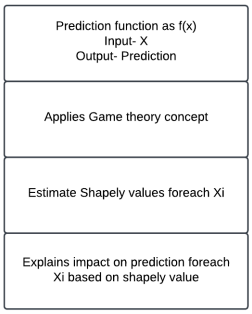


Figure 3.12: Overview of SHAP

1. Treat the machine learning model as a function $f$ that takes input features $x$ and outputs a prediction f(x).

2. Model the prediction $f(x)$ as if it is a "payout" in a cooperative game where each feature $x_i$ cooperates to produce the final output f(x).

3. Estimate the Shapley values $\phi_i$ for each feature $x_i$ using sampling or model-specific approximation methods. These Shapley values represent each feature's fair contribution to the prediction.

4. The SHAP equation then models the prediction as:

$$f(x) = b + \sum \phi i$$

Where $b$ is the base value or expected value of $f(x)$ across the dataset.

5. To explain an individual prediction, the SHAP values $\phi$ i quantify the impact each feature value $xi$ had on moving the prediction from the base value $b$. The sum of SHAP values equals the difference between $f(x)$ and $b$.

6. Features with positive $\phi$i increased the prediction from b, negative values decreased it. Absolute SHAP magnitude measures the importance of that feature's attribution.

So in summary, SHAP technically leverages the model's function to estimate Shapley values for a sample, attributing how much prediction movement each feature caused. This connection to ideas of fairness from game theory is what provides the foundation for consistent, stable feature attributions.

The details of estimating SHAP values $\phi i$ vary per model type, using sampling or approximations. But the high level approach remains framing prediction explanation as a cooperative game.

**Why to use SHAP?**

Here are some of the key reasons to use SHAP for explaining:

- Model Agnostic: SHAP is a model-agnostic approach that can explain predictions from any machine learning model, including deep neural networks, tree models, kernels and more. This flexibility is useful when working with different algorithms.
- Local Explanations: SHAP provides local, instance-level explanations by attributing how much each feature impacted an individual prediction, not just global feature importance. This helps understand specific cases.
- Interactions: SHAP values reflect interaction effects between features, identifying when combinations of features together impacted a prediction. This provides richer explanations.
- Theoretical Foundations: SHAP is grounded in coherent concepts from game theory and local explanation methods. This provides mathematically sound attribution values.
- Intuitive: By explaining models in terms of feature contributions, SHAP produces intuitive explanations that are easier for people to understand. Plots like the force plot visualize these intuitively.
- Consistency: Explanations are consistent between similar models and instances, improving trust and stability.

In summary, properties like model flexibility, faithfulness to real feature interactions, and intuitive visual representation through approaches like force plots have made SHAP a popular and reliable approach to improve transparency and debugability of complex machine learning models. The connections to solid theoretical foundations also distinguish SHAP from other interpretation methods.

### 3.5.2 LIME

LIME (Local Interpretable Model-agnostic Explanations) is an approach developed in 2016 to explain individual black box model predictions by approximating the complex model locally with an interpretable one to attribute feature importance. The key idea is that while complex models may be too opaque globally, they can be approximated well in small local regions. So LIME focuses on explaining individual predictions by sampling input data near the instance of interest, gathering predictions using the complex model on those perturbations, and fitting a highly interpretable surrogate model like a linear model or small decision tree on that neighbourhood data. This simple transparent surrogate acts as the local explanation, revealing which features were most important for the behaviour of the complex model in that region for that particular prediction. Unlike methods requiring model access, LIME treats it as a black box, lending model flexibility. By providing intuitive feature contributions for specific predictions via local approximation, LIME improves trust in model fairness and enables better errors analysis. The fidelity arguments rest on the idea that transparent models can sufficiently explain complex model behaviour for individual instances, even if different globally. A high-level overview is provided in figure 3.13. A high-level overview of how LIME technically works


Figure 3.13: Overview of LIME

to explain individual predictions from complex models:

1. Select an individual prediction to explain from the complex black-box model (e.g. a neural network).

2. Sample perturbed datasets in the neighbourhood of the selected instance by randomly permuting its features.

3. Get predictions on the perturbed datasets using the original complex model.

4. Weight the perturbed datasets according to their proximity to the original instance.

5. Train a simple linear model or decision tree on the perturbed dataset and weights to act as a local surrogate model.

6. Interpret the weights or importance values assigned to features in the simple model to determine which features were most relevant for the prediction from the complex model in that local area.

In essence, LIME uses the complex model as a black box to sample localized data around the prediction of interest. It then trains a transparent, interpretable model on this data to approximate complex model behaviour in the locality. The feature importance from the simple model explains which inputs matter most to the complex model locally. This provides local fidelity while leveraging easy-to-understand models. The core insight is that simple models can estimate local predictions even if different globally, affording intuitive explanations.

**Why to use LIME?**

Here are some of the key reasons to use LIME for explaining black box model predictions:

1. Model Agnostic: LIME can explain predictions from any machine learning model (neural nets, SVM, ensemble models, etc.) without access to model internals or parameters. This flexibility is important for complex black box models.

2. Local Fidelity: LIME focuses on local behaviour and provides fidelity to model predictions in the region of the specific instance being explained rather than globally. This local accuracy is more important for explanations.

3. Intuitive Explanations: By using simple linear models or small decision trees, LIME provides intuitive feature importance explanations that are easy for people to understand.

4. Human Interpretable: Humans have an easier time understanding explanations from transparent models compared to complex models, even if their global behaviour differs. LIME generates human interpretable local explanations.

5. Model Checking: Important to detect individual cases where the model is failing for the end users. LIME enables users to audit suspicious model behaviour.

6. Trust: By explaining specific predictions, LIME provides visibility into model behaviour that reassures users and improves trust in otherwise black box models.

In summary, the model agnostic nature combined with local fidelity explanations from interpretable models make LIME suitable for explaining a prediction from any black box algorithm in an intuitive manner to improve transparency.

# Chapter 4

# Results & Conclusion

## 4.1 ML/DL Results

The Random Forest model was developed using 21 potential predictor variables related to soil, weather, and plant growth conditions to predict crop yield. The model was trained on 22 years of historical data. On evaluating model performance, the Random Forest achieved an R-squared value of 0.51 on the test data for now. A Deep Neural Network (DNN) model was developed for predicting crop yield using time-series data spanning several years across parameters like weather, soil moisture, vegetation indices, etc. The DNN architecture comprised 5 hidden layers, ReLU activation functions, and a linear output layer for regression to crop yield. The DNN model achieved an R-squared of 0.8 . This indicates that the DNN can account for roughly 80% accuracy in predicting crop yield. Further, tuning the models can improve the accuracy of both models.

The decision to use the DNN in the implementation is well-justified, as it outperformed the Random Forest model in terms of the R-squared metric. In general, models with higher R-squared values are preferred, as they provide a better fit to the data and are more reliable for making predictions or understanding the relationship between the independent and dependent variables.We have tried using a variety of combinations of hidden layers and the number of nodes in each layer, but we got a lower R-squared value (between [0.72-0.77]) (as you can see in table 4.1) than the explained model.

| Layers (Each number represents the number of nodes in one layer) | R-squared value |
|---|---|
| 512− > 512− > 256− > 64− > 21− > 1 | 0.72 |
| 512− > 256− > 64− > 21− > 1 | 0.77 |
| 1024− > 512− > 256− > 64− > 21− > 1 | 0.76 |
| 256− > 64− > 21− > 1 | 0.75 |
| 512− > 256− > 128− > 64− > 21− > 1 | 0.76 |
| 1024− > 512− > 256− > 128− > 64− > 21− > 1 | 0.72 |
| Explained model (see section 3.4.2) | 0.8 |

Table 4.1: comparison of r-squared between DNN architecture

## 4.2 XAI Results

**LIME Result**

As in the figure 4.1, LIME gives output as features and its impact on the predicted value. For the particular instance in which the explanation is given, soilwater is having the most positive impact on the yield prediction, with a LIME score of 0.84, followed by solar radiation with LIME score of 0.35, Leaf Area Index with LIME score of 0.31, LSTN, transpiration and after that some of the parameters are having lower but positive impact on yield. On the other hand, humidity has the most negative impact on yield, with a score of -0.32, followed by vap (vapour pressure), LSTD, air temperature, wind speed and others which are having negligible impact on yield.

**SHAP**

As in figure 4.2, SHAP gives output as features and its impact on the predicted value. for particular instance on which the explanation is given soilwater is having most positive impact on the yield prediction, with shapley score of 1.01. We can see that after soil water, LAI (Leaf Area Index) is also having a positive impact on the prediction of yield, followed by tmmx, NDVI, transpiration, and other 12 parameters. Other than that, we can see here that vap (vapor pressure) has a shapely score of -0.38, which is having most negative impact on prediction of yield, followed by ws (wind speed), humidity, and LSTD (land surface temperature for day). We can get an analysis of the most impactful parameters for yield, which are negligible, so we can ignore small variations in those parameters.

As discussed above, LIME and SHAP results can provide us with a good analysis of

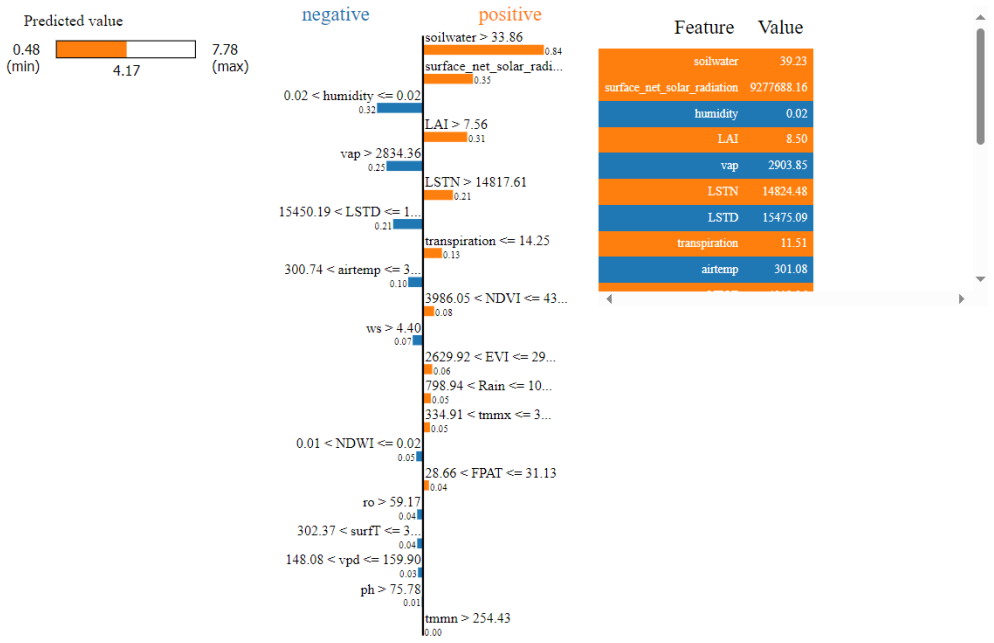


Figure 4.1: Lime Output

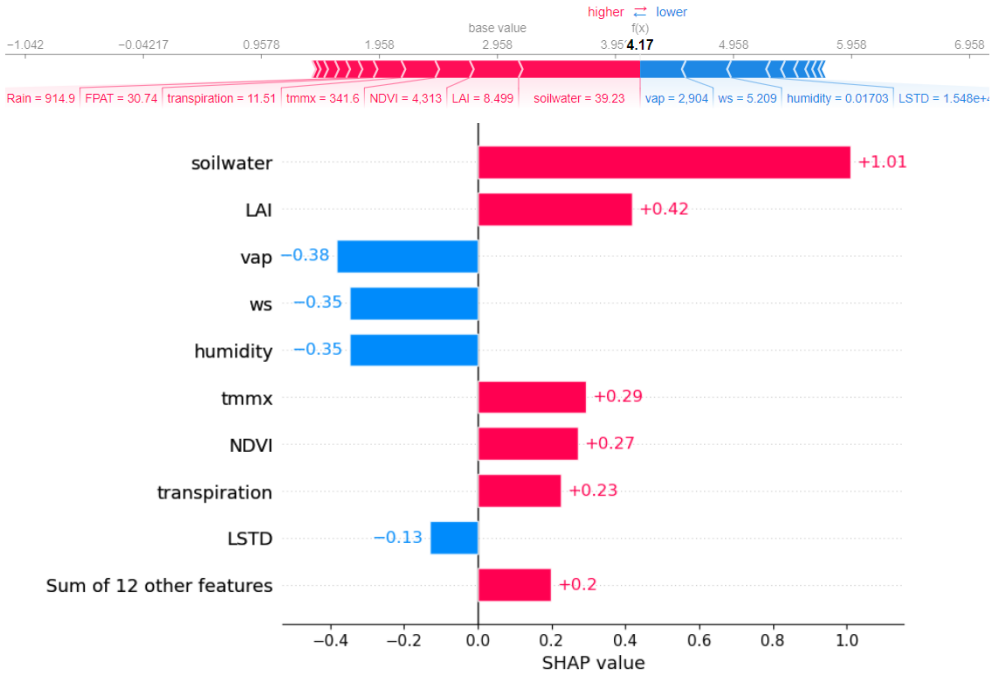


Figure 4.2: SHAP Output

parameters that are having an impact on yield. So, we can get insights about how and why a particular number came as a yield. Farmers and traders can have insights about yield so that they can make decision with detailed reasons.

# Chapter 5

# Future work

In this study, we employed an explainable machine learning approach to understand the factors influencing crop yield predictions for a specific crop in a particular region. The model was trained on 20 years' worth of data, which, while limited in time-span, provided valuable insights into the key parameters impacting the target variable, i.e., crop yield.Through the use of techniques like LIME (Local Interpretable Model-agnostic Explanations) or SHAP (Shapley Additive Explanations), we were able to obtain explanations for individual predictions and also for overall predictions made by the model. These explanations highlighted the input features or parameters that had a significant positive or negative impact on the predicted yield values. However, it's important to note that the accuracy of the model and the reliability of the explanations were constrained by the limited availability of yield data, which is the most crucial parameter in this study. To improve the model's performance and obtain more robust explanations, it would be beneficial to incorporate additional relevant parameters into the dataset. One such parameter that could potentially enhance the model's predictive capabilities is soil nutrient levels, specifically the levels of nitrogen (N), phosphorus (P), and potassium (K). These nutrients play a crucial role in plant growth and development, and their availability in the soil can significantly influence crop yield. Also, one can integrate data that is collected from the ground, like whether a crop is having issues like pests or diseases that will impact its yield.

# Bibliography


[1] Elsevier, "Scopus database (https://www.scopus.com/)," 2023.

[2] M. Van Lent, W. Fisher, and M. Mancuso, "An explainable artificial intelligence system for small-unit tactical behavior," in *Proceedings of the national conference on artificial intelligence*, pp. 900–907, Citeseer, 2004.

[3] Statista, "Revenues from the artificial intelligence (ai) market worldwide from 2016 to 2025," June 2018.

[4] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[5] A. Shastry, H. Sanjay, and E. Bhanusree, "Prediction of crop yield using regression techniques," *International Journal of Soft Computing*, vol. 12, no. 2, pp. 96–102, 2017.

[6] M. F. Celik, M. S. Isik, G. Taskin, E. Erten, and G. Camps-Valls, "Explainable artificial intelligence for cotton yield prediction with multisource data," *IEEE Geoscience and Remote Sensing Letters*, 2023.

[7] L.-D. Quach, K. N. Quoc, A. N. Quynh, N. Thai-Nghe, and T. G. Nguyen, "Explainable deep learning models with gradient-weighted class activation mapping for smart agriculture," *IEEE Access*, vol. 11, pp. 83752–83762, 2023.

[8] R. Bandi, S. Swamy, and C. Arvind, "Leaf disease severity classification with explainable artificial intelligence using transformer networks," *International Journal of Advanced Technology and Engineering Exploration*, vol. 10, no. 100, p. 278, 2023.

[9] M. H. K. Mehedi, A. S. Hosain, S. Ahmed, S. T. Promita, R. K. Muna, M. Hasan, and M. T. Reza, "Plant leaf disease detection using transfer learning and explainable

ai," in *2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0166–0170, IEEE, 2022.

[10] J. Sun, L. Di, Z. Sun, Y. Shen, and Z. Lai, "County-level soybean yield prediction using deep cnn-lstm model," *Sensors*, vol. 19, no. 20, p. 4363, 2019.

[11] C. Arvind, A. Totla, T. Jain, N. Sinha, R. Jyothi, K. Aditya, M. Farhan, G. Sumukh, G. Ak, *et al.*, "Deep learning based plant disease classification with explainable ai and mitigation recommendation," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 01–08, IEEE, 2021.

[12] A. Wolanin, G. Mateo-García, G. Camps-Valls, L. Gómez-Chova, M. Meroni, G. Duveiller, Y. Liangzhi, and L. Guanter, "Estimating and understanding crop yields with explainable deep learning in the indian wheat belt," *Environmental research letters*, vol. 15, no. 2, p. 024019, 2020.

[13] C. M. Viana, M. Santos, D. Freire, P. Abrantes, and J. Rocha, "Evaluation of the factors explaining the use of agricultural land: A machine learning and model-agnostic approach," *Ecological Indicators*, vol. 131, p. 108200, 2021.

[14] M. Ryo, "Explainable artificial intelligence and interpretable machine learning for agricultural data analysis," *Artificial Intelligence in Agriculture*, vol. 6, pp. 257–265, 2022.

[15] A. Cartolano, A. Cuzzocrea, G. Pilato, and G. M. Grasso, "Explainable ai at work! what can it do for smart agriculture?," in *2022 IEEE Eighth International Conference on Multimedia Big Data (BigMM)*, pp. 87–93, IEEE, 2022.

[16] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[17] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[18] S. Letzgus, P. Wagner, J. Lederer, W. Samek, K.-R. Muller, and G. Montavon, "Toward explainable artificial intelligence for regression models: A methodological perspective," *IEEE Signal Processing Magazine*, vol. 39, p. 40–58, July 2022.

[19] J. K. Dhaliwal, D. Panday, D. Saha, J. Lee, S. Jagadamma, S. Schaeffer, and A. Mengistu, "Predicting and interpreting cotton yield and its determinants under long-term conservation management practices using machine learning," *Computers and Electronics in Agriculture*, vol. 199, p. 107107, 2022.

[20] J. Dieber and S. Kirrane, "Why model why? assessing the strengths and limitations of lime," *arXiv preprint arXiv:2012.00093*, 2020.

[21] NASA, "Geographic information systems (gis) (https://www.earthdata.nasa.gov/learn/gis)," 2023.

[22] P. Teluguntla, P. Thenkabail, J. Xiong, M. Gumma, C. Giri, C. Milesi, M. Ozdogan, R. Congalton, J. Tilton, T. Sankey, R. Massey, A. Phalke, and K. Yadav, "Global cropland area database (GCAD) derived from remote sensing in support of food security in the twenty-first century: Current achievements and future possibilities," in *Land Resources: Monitoring, Modelling, and Mapping* (P. S. Thenkabail, ed.), vol. II of *Remote Sensing Handbook*, ch. 7, In Press, 2014.

[23] "Revolution in indian cotton (https://ncipm.icar.gov.in/ncipmpdfs/folders/revolutioninindiancotton.pdf)," *Department of Agriculture Cooperation Ministry of Agriculture, Govt of India*.

[24] "Yield data of india by government of india (https://aps.dac.gov.in/apy/public_report1.aspx)," *Yield data of India*, 2023.

[25] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.

[26] W. Li, "Mutual information functions versus correlation functions," *Journal of statistical physics*, vol. 60, pp. 823–837, 1990.

[27] A. Liaw and M. Wiener, "Classification and regression by randomforest," *Forest*, vol. 23, 11 2001.

[28] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[29] N. Prasad, N. Patel, and A. Danodia, "Crop yield prediction in cotton for regional level using random forest approach," *spatial information research*, vol. 29, pp. 195–206, 2021.

[30] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[31] S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud, "A comprehensive analysis of deep regression," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 9, pp. 2065–2081, 2019.

[32] K. Roshan and A. Zafar, "Using kernel shap xai method to optimize the network anomaly detection model," in *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 74–80, IEEE, 2022.