Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 406–417

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Crop Yield Prediction using Machine Learning and Deep Learning Techniques

Kavita Jhajharia[a], Pratistha Mathur[a*], Sanchit Jain[a], Sukriti Nijhawan[a]

[a]*Manipal University Jaipur, Dehmi Kalan, Jaipur, India*

Abstract

Agriculture is a significant contributor to India's economic growth. The rising population of country and constantly changing climatic conditions have an impact on crop production and food security. A variety of factors influence crop selection, including market price, production rate, soil type, rainfall, temperature, government policies, etc. Many changes are required in the agricultural sector in order to enhance the Indian economy. In this research work authors have implemented various machine learning techniques to estimate the crop yield in Rajasthan state of India on five identified crops. The results indicate that among all the applied algorithms; Random Forest, SVM, Gradient Descent, long short-term memory, and Lasso regression techniques; the random forest performed better than others with 0.963 R^2, 0.035 RMSE, and 0.0251 MAE. The results were validated using R^2, root mean squared error, and the mean absolute error to cross-validation techniques. This paper intends to put the crop selection method into practice to help farmers solve crop yield problems.

Keywords: *Deep learning, machine learning, crop yield prediction.*

1. Introduction

Agriculture is extremely important to the global economy. Understanding global crop yield is critical for resolving food security issues and mitigating the effects of climate change as the human population continues to grow. Crop yield forecasting is a significant agricultural problem. Weather conditions (rain, temperature, etc.) and pesticides have a great impact on agricultural yield. It is important to have accurate knowledge about crop yield history while making decisions about agricultural risk management and yield forecasting [1]. Crop yield prediction is a challenge for decision-makers at all levels, including global and local levels. Farmers may adopt a good crop yield prediction model to decide what to plant and when to plant it. Crop yield forecasting may be done in several ways [2] [3].

[*] Corresponding author. Tel.: +91 9680321224.
E-mail address: Pratistha.mathur@jaipur.manipal.edu

10.1016/j.procs.2023.01.023

Machine learning is a realistic method that can provide better yield prediction based on many attributes. It is a subdivision of Artificial Intelligence (AI) that focuses on learning. Machine learning (ML) can discover information from datasets by identifying patterns and correlations. The models must be trained using datasets that represent prior experience-based outcomes [4]. The predictive model is built using a range of characteristics, and the parameters are calculated using previous data throughout the training phase. Some of the historical data that was not utilized for preparation is used to measure results throughout the testing procedure. An ML model can be descriptive or predictive, depending on the research topic and questions. Predictive models use past knowledge to predict what will happen in the future. Descriptive templates, on the other hand, help to describe how things are now or what happened in the past. Machine learning could help predict agricultural yields and decide which crops to sow and what to do during the growing season. Several machine learning algorithms were deployed to enhance the agricultural yield forecast investigation. Crop yields have lately been predicted using machine learning approaches such as multivariate regression, decision trees, association rule mining, and artificial neural networks [5] [6].

Machine learning models assume the output (crop yield) to be a non-linear function of the input variables (area and environmental factors). Deep learning models for crop yield prediction have recently gained popularity [7]. Deep learning is a form of machine learning that can predict outcomes from a variety of raw data arrangements. Deep learning algorithms, for example, can build a probability model from ten years of field data and provide insights into crop output under various climatic conditions. Deep learning uses artificial neural networks (ANNs) to replicate how people think and learn. Artificial neural networks with several layers drive deep learning. DNNs (Deep Neural Networks) are multilayer networks that can execute complicated operations such as representation and abstraction to grasp pictures, sound, and text [8] [9] [10]. Neural networks are built up of layers of nodes, just like the human brain is made up of neurons. Nodes in one layer are linked to nodes in another layer. The network's depth is indicated by the number of layers. In an artificial neural network, signals travel between nodes and are assigned weights. The objective of this research is to give a certain impetuous to more data-driven decision-making even to the agricultural section of our country. However, the experience of an individual in this field is what matters the most, our aim is to aid this thinking with our data-driven research and forecasting. In this research work, the authors predicted crop yield in Rajasthan state's 33 districts. The authors implemented four machine learning and one deep learning algorithm to compare the performance of models. The organization of the paper is as follows: first section of the article provides the introduction of the research; the second section describes methods implemented in the article along with the dataset used to obtain the results; third section contains result and discussion on the results; and finally, the work has been concluded in last section of the article.

2. Materials and Methods

2.1. Data Set

The data collection procedure refers to the programmer acquiring, and quantifying information based on variables relevant to the research. The data used was obtained from a variety of sources, including the official website of the Rajasthan Government. The data of the state's most harvested crops, which include wheat, rapeseed & mustard, barley, bajra, jowar onion, and maize, have been gathered and recorded from the state's 33 districts (Table 1). The data from 1997 to 2019 was obtained from the official Rajasthan Government website.

Table 1. Crop data in the dataset.

CROP	DATA AVAILABLE IN THE DATASET
Rapeseed and Mustard	748
Wheat	748
Barley	747
Bajra	742
Jowar	739
Onion	723
Maize	705

According to preliminary research, there is insufficient data for the crops - onion and maize - as well as for the Pratapgarh district. As a result, there is no relevant data to forecast for these specific entities, resulting in lower accuracy when applied to the relevant models. Hence, the entities have been dropped from the dataset to overcome these discrepancies. After removing all invalid and null data, the dataset's final shape is 3664 rows and 7 columns. Yield is now added as an independent variable. To obtain a more scientific result, a few more independent features are added to the dataset, including the soil type found in that district and the amount of rainfall that has occurred in these districts over the years. The reason for selecting these specific factors is that first and foremost, the soil is one of the most important components of crop yield because it provides the crop with the necessary nutrients, water, and oxygen, while rainfall patterns help in determining the amount of natural water that will be provided for crop growth, replenishment, and production.

The type of soil that is unique to a particular district was identified. About 27 different types of soil have been identified in Rajasthan, and the soil in each of the current 32 districts is a mixture of any of these 27 types of soil. As one can see, each district has its unique combination of soil, and this information is presented in the form of categorical data. As a result, in order to perform prediction over the dataset, the soil data must be encoded, resulting in the creation of dummy variables. The data frames were then combined, and the "Soil Type" column was removed from the original data frame. A concurrent change in the format in which the names of the districts have been represented is also required for a clear understanding of the dataset at hand. Following that, rainfall data was added to the dataset. The rainfall data were available for each district of Rajasthan from 1901 to 2002 and 2004 to 2010. The rainfall data corresponding to 2003 was filled by using the mean of the data available, and data for the years 2011 to 2017, was extracted from the official documentation done by the Rajasthan Government. Following that, rainfall data for the years 2018 and 2019 were added using the mean of the data from 1901 to 2017. Because the main dataset contains crop production information from 1997 to 2019, the final dataset contains monthly rainfall data for each district for the same period. Finally, the rainfall dataset is merged with the main dataset.

Each crop has a distinct season in which it is harvested. As a result, considering the rainfall pattern for the entire year for that crop makes no sense. So, in order to calculate the correct amount of rainfall that the crop would have received, the mean value of rainfall for the months corresponding to those seasons is used. Thus, rainfall received during July, August, September, and October is considered for crops produced during the Kharif season, while rainfall received during November, December, January, February, and March is considered for crops produced during the Rabi season. The columns from January to Annual Total are then deleted. The column "State" is also removed because only Rajasthan is being considered, and the different districts act as distinguishing entities, so the presence of the "State" variable is unnecessary.

Table 2 shows the data and source of the data.

Table 2. Data and their sources.

DATA	VARIABLE (S)	TIME COVERAGE	SOURCE
Rajasthan Crop data	State, District, Area, and Production	1997 to 2010	[17]
Rajasthan Crop data	State, District, Area, and Production	2011 to 2019	[18]
Rainfall data	Jan, Feb, Mar, Apr, May, Jun, Jul, Sept, Oct, Nov, Dec, and Annual_Total	1901 to 2002	[19]
Rainfall data	Jan, Feb, Mar, Apr, May, Jun, Jul, Sept, Oct, Nov, Dec, and Annual_Total	2004 to 2010	[20]
Rainfall data	Jan, Feb, Mar, Apr, May, Jun, Jul, Sept, Oct, Nov, Dec, and Annual_Total	2011 to 2019	[21]
Soil data	District, and Soil_Type		[22]

2.2. Methodology

- Data Preprocessing

Data preprocessing is a technique for transforming unprocessed data into a flawless data set. At the end of the day, whenever data is gathered from various sources, it is gathered in a raw or crude form that cannot be analyzed by machine learning or deep learning methodologies.

- Data Encoding

  There are six categorical columns in the final data frame: State, District, Season, Crop, and Soil Type. As a result, a dataset can be divided into two types of variables: continuous variables and categorical variables. A categorical variable takes one value from a limited set of values and is not quantifiable. As a result, to apply any algorithm, such variables must be encoded, because many machine learning algorithms cannot work directly on labelled data, so all input must be in the form of numeric values. There are several methods for encoding and handling such variables. LabelEncoder, OneHotEncoder, and other methods fall into this category. Dummy variables were created from all the categorical data in the given dataset. Creating dummy variables gives flexibility while performing regression analysis and hence suits well for the given dataset.

- Standardization Of Features

  The DataFrame consists of 71 features. To get the desired output, standardization of these features is important as it helps bring all the features into a single scale for much more accurate calculations. The StandarScaler method from the SciKit Learn library has been applied to the given dataset. This helps standardize features by removing mean and scaling to unit variance. Standardization of any dataset is common practice and a necessity at times for a lot of machine learning and deep learning methodologies.

- Splitting Dataset into Testing and Training Sets

The final step of data preprocessing is testing and training the data. The division between the training and testing size of the data is not done equally. A larger training set is required as compared to the testing size, since, for prediction, the model needs to be trained over as many data points as possible. For this, the ScikitLearn library and import of the train_test_split module is used. The train_test_split method has been used to split the data, with the test set being 2% of the total dataset and the random state set to 71. These values give the most accurate possible result, due to the skewness of the dataset as well as its enormity of it.

Pseudocode:

```
Import the required libraries – Pandas, Numpy and SciKit Learn
SET df as pd.read_excel('Rajasthan_Crop_Final.xlsx')
Dropping Onion, Maize and Pratapgarh district from the DataFram
Calculate Yield as Production / Area
Assigning Soil Type to the Districts
SET Soil as Empty List
for x in District Name
    IF x IS "BARMER" add "Desert soils and sand dunes aeolian soil, coarse sand in texture some places calcareous" to soil
    ELSE IF x IS "GANGANAGAR" or "HANUMANGARH" add "Alluvial deposites calcareous, high soluble salts & exchangeable sodium" to the soil
    ELSE IF x IS "BIKANER" or "JAISALMER" add "Desert soils and sand dunes aeolian soil, loamycoarse in texture & calcareous" to soil
    ELSE IF x IS "NAGAUR" or "SIKAR" or "JHUNJHUNU" add "Sandy loam, sallow depth red soils in depressions" to the soil
    ELSE IF x IS "JALORE" or "PALI" add "Red desert soils" to the soil
    ELSE IF x IS "JAIPUR" or "AJMER" or "DAUSA" or "TONK" add "Sierozens, eastern part alluvial, west-north-west lithosols, foothills, brown soils" to soil
    ELSE IF x IS "ALWAR" or "DHOLPUR" or "BHARATPUR" or "KARAULI" or "SAWAI MADHOPUR" add "Alluvial prone to water logging, nature of recently alluvial calcareous has been observed" to soil
    ELSE IF x IS "BHILWARA" or "RAJSAMAND" add "Soil is lithosolsat foot hills & alluvials in plains" to the soil
    ELSE IF x IS "DUNGARPUR" or "BANSWARA" add "Predominantly reddish medium texture, well-drained calcareous, shallow on hills, deep soils in valleys" to the soil
    ELSE IF x IS "KOTA" or "JHALAWAR" or "BUNDI" or "BARAN" add "Black of alluvial origin, clay loam, groundwater salinity" to the soil
```

xviii. ELSE IF x IS "JODHPUR" add "Desert soils and sand dunes aeolian soil, coarse sand in texture some places calcareous, Red desert soils" to soil
xix. ELSE IF x IS "CHURU" add "Desert soils and sand dunes aeolian soil, loamycoarse in texture & calcareous, Sandy loam, sallow depth red soils in depressions" to soil
xx. ELSE IF x IS "SIROHI" add "Red desert soils, Soil are lithosolsat foot hills & alluvials in plains" to the soil
xxi. ELSE IF x IS "UDAIPUR" or "CHITTORGARH" add "Soil are lithosolsat foot hills & alluvials in plains, Predominantly reddish medium texture, well-drained calcareous, shallow on hills, deep soils in valleys" to soil
xxii. Add the Soil Type column to the DataFrame
xxiii. Create Dummies of the Soil Type Column
xxiv. Merge the DataFrame with dummies DataFrame
xxv. Remove Soil Type column
xxvi. Clean the District Name, check for spelling errors or uppercases
xxvii. Import the Rainfall Data
xxviii. Clean the District Name in the Rainfall Data
xxix. Remove the unwanted column
xxx. Add the Rainfall Data based on District Name
xxxi. Merge Rainfall Data with initial DataFrame
xxxii. Fill Median in case of any Null Values on Merging both DataFrames
xxxiii. Standardize Features using StandardScaler()
xxxiv. Split DataFrame in Testing and Training Sets

2.3. Models

- Random Forest

Random forest is a very famous machine learning algorithm that felicitates in cases of both classification and regression issues [11]. This algorithm works on the notion of ensemble learning, which works on the principle of merging several classifiers to give the solution for any complex problem and improve the precision and performance of the applied model. Random Forest is a classifier that uses numerous decision trees on subsets of a dataset and takes the average into account to increase the dataset's prediction accuracy. In other words, rather than depending on a single decision tree, this algorithm considers forecasts from each tree and predicts the ultimate result based on the majority of votes.

- Support Vector Machine (SVM)

The goal of the support vector machine algorithm is to discover a hyperplane in N-dimensional space (N refers to the number of features present in the dataset) that classifies the data points very distinctly. To detach the two classes of information points, various possible hyperplanes could be chosen. To find a plane that has the best edge, i.e., the highest distance between data points of the two classes is dissected [12]. Extending the edge distance gives some help so future data points can be organized with more assurance. To increase and maximize the edge between the data points and the hyperplane, the Hinge function is utilized:

$$c(x, y, f(x)) = \begin{Bmatrix} 0, \ if\ y * f(x) \geq 1 \\ 1 - y * f(x), \qquad else \end{Bmatrix} \tag{1}$$

Hinge loss function

The cost comes out to be 0 if the predicted value and the actual value are of the same sign, but if not, then the loss value is calculated. To balance the regularized limits, the regularization function is added to the Hinge loss function:

$$min_{\omega}\lambda\ ||\omega||^2 + \sum_{i=1}^{n}(1 - \ y_i(x_i, \omega))_+ \tag{2}$$

The loss function for SVM

As the loss function has now been defined, concerning weights need to be partially differentiated to find the gradients, and by using this gradient the weights are updated. After finding the gradients regularization parameter is added so that there exists no misclassification. If there still exists any misclassification, the model will make mistakes when performing predictions of the class of the data points, thus the loss is included with the regularization parameter to perform the final gradient update.

$$\frac{\delta}{\delta \omega_k} \lambda \, ||\omega||^2 = 2\lambda\omega_k \qquad \textbf{(3)}$$

$$\frac{\delta}{\delta \omega_k}(1 - \ y_i(x_i, \omega))_+ = \begin{Bmatrix} 0, \ if \ y * f(x) \geq 1 \\ -y_i x_{ik}, \qquad\qquad else \end{Bmatrix} \qquad \textbf{(4)}$$

Gradients

$$\omega = \ \omega - \ \alpha 2\lambda\omega \qquad \textbf{(5)}$$

Gradient Update – No misclassification

$$\omega = \ \omega + \ \alpha(y_i x_i - 2\lambda\omega) \qquad \textbf{(6)}$$

Gradient Update – Misclassification

- Gradient Descent

Gradient descent is a well-known optimization algorithm that is frequently used in machine learning and deep learning. It finds the coefficients that minimize the cost function as far as possible by identifying a local minimum of the differentiable function. Gradient descent begins by characterizing the initial parameter values and then uses analytics and calculus to iteratively change the values so that they limit the given cost function. Gradient descent can also be defined as the slope of any given function [13]. The greater the gradient, the greater the slope, and thus the faster the model's learning rate. When the slope reaches zero, the model stops learning. Hence, one can argue that a gradient is simply the partial derivate concerning the introduced values. The following equation describes what the given algorithm does:

$$b = a - \ \gamma \nabla f(a) \qquad \textbf{(7)}$$

b = next position, a = current position

The negative sign denotes the minimization component of gradient descent, whereas the gamma denotes the waiting feature and the gradient term (Δf(a)) denotes the direction of the sharpest decrease.

In a machine or deep learning model on applying the algorithm to minimize the cost function J (ω, b) and reaching the local minimum by tweaking the parameter it is observed that the gradient descent function is convex, but there do exist non-convex examples as well, all depending upon the dataset

- Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that is capable of long-term dependence. Neural networks are a collection of algorithms designed to mimic the human brain and find patterns. RNN is simply a generalized feedforward neural network with internal memory [14]. RNN is recurrent in nature, as the name implies, and so performs the exact function for all data inputs, although the output of the current input is significantly dependent on prior calculations. Unlike other standard feedforward neural organizations, LSTM has feedback connections [15]. It cannot only cycle single information points that are the data points, for example, pictures, but also the whole groupings of information, for example, discourse or video. Such a network is used to process and predict problems involving time series and other complex problems such as this one. LSTM, being a modification of

RNN, helps resolve one of the major problems faced on any RNN network, which is the vanishing gradient problem. It trains any model by back-propagation.

- Lasso Regression

The Least Absolute Shrinkage and Selection Operator or in short Lasso regression is a kind of linear regression that utilizes shrinkage, i.e., data points are shrunk towards a mid-point, as done when finding the mean. This model is particularly useful and suits well for models that show high levels of multicollinearity as is the case in this particular dataset. This algorithm executes L1 regularization, which adds a penalty that is equivalent to the absolute value of the degree of the coefficients [16]. Lasso solutions are quadratic programming problems, that use the following formula:

$$\sum_{i=1}^{n}(y_i - \sum_j x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \quad \textbf{(8)}$$

The intensity of the L1 penalty is controlled by a tuning parameter. Which is essentially the amount of shrinking. If it is zero, then all features are taken into account, and it is equivalent to linear regression, in which just the residual sum of squares is used to form a predictive model. If it is infinity, it means that no features are taken into account. The bias increases while the variance decreases with an increase in λ, and vice-versa.

3. Results and discussion

According to the analysis, even though there hasn't been a consistent rainfall pattern over the years (figure 1), crop production has steadily increased, except Wheat and Jowar from 2016 to 2019, where there has been a sharp decrease in rainfall patterns. This indicates a growing reliance on modern irrigation techniques for most crops, while water-intensive crops like wheat and jowar continue to rely on natural water sources, primarily rain.

After examining each crop separately, it is possible to conclude: For Bajra, the area under irrigation used to produce this crop has more or less, hasn’t changed erratically (figure 2), i.e., from 1997-2019, and has been constant, with a certain dip between the years 2000-2003 due to reasons unknown, but the yield of this crop has seen a constant rise barring the year 2003 and 2009 (figure 3). This can be factored in due to better seasonal rainfall as well as the availability of modern and ever-improving irrigation methods as well as better fertilizers, pesticides, and production techniques. For Wheat, the area under irrigation and production when plotted against the years, one can observe that while there’s a drop in the area used from 1997-2003, the area started increasing gradually from there on till 2019 (figure 4). Considering the yield, it has remained consistent over the years, with a certain boom in 2018 but a dip in 2019 for the state (figure 5). There can be several reasons for such an observation. One of them can be the growing need to produce for other parts of the country as well. For Rapeseed & Mustard, the area under irrigation used to produce the given crop has remained pretty much constant as shown by the graph, with a dip during 2000-2003 and a high increase in from 2003-2005 and then becoming constant (figure 6). In the case of yield, over the years there has been a gradual increase in the yield (figure 7) of this crop even though the area has been pretty much constant. This can be factored with the increase in demand and modernization of production and irrigation techniques of farming. For Barley, again as observed in the case of other crops, the area under production hasn’t increased as such (figure 8) but the yield over the years has been rising in this factor over the years (figure 9). For Jowar, the area has remained almost the same over the years (figure 10), which has been the case for most of the crops, but the production has yet again increased over the years (figure 11) as furthermore observed in the case with all the crops that have been grown in the state of Rajasthan.

From these observations, the conclusion drawn is that area is something that has remained pretty much similar over the years, indicating that area is a very limited resource that cannot be increased since land is required for several other purposes. But due to the increase in population, the demand has also been constantly increasing, so to meet these demands, production needs to be increased, which has been the case over the years. But, since the area hasn't altered much for most crops, the rise in the yield implies that the modern methods of irrigation and production and the use of better fertilizers and other modernized methods have resulted in a better yield of these particular crops.

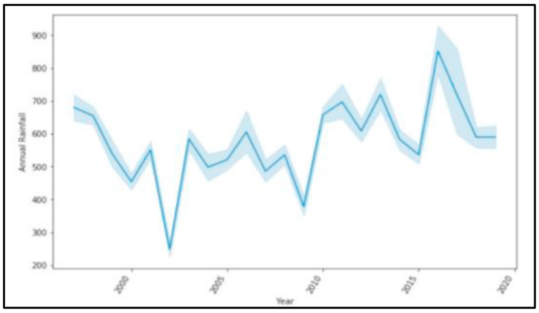

Fig. 1. Annual rainfall vs. Year.

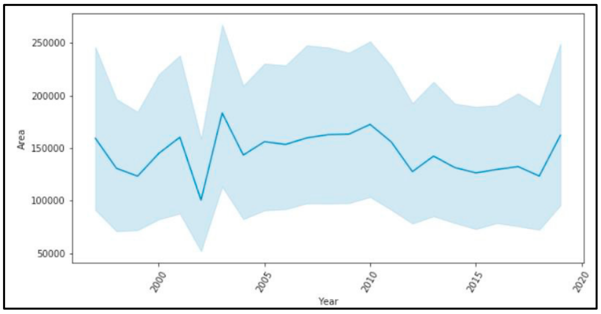

Fig. 2. Area under irrigation vs Year (Bajra).

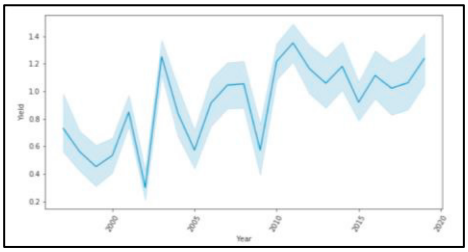

Fig. 3 Yield vs Year (Bajra).

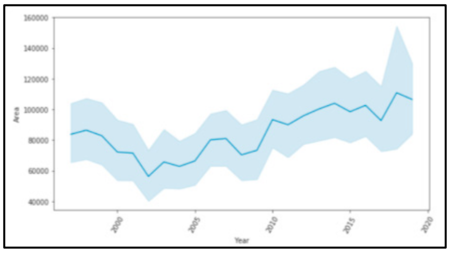

Fig. 4. Area under irrigation vs. Year (Wheat).

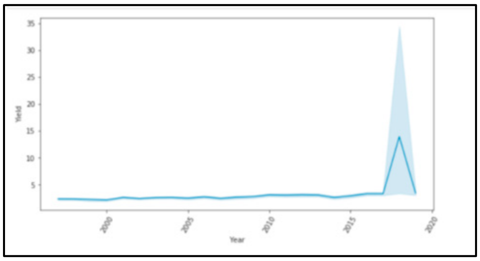

Fig. 5 Yield vs Year (Wheat).

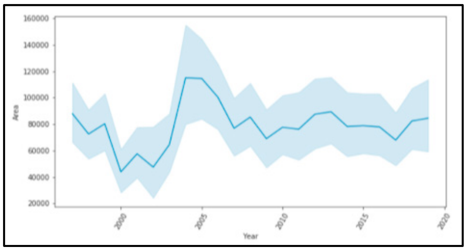

Fig. 6. Area under irrigation vs Year (Rapeseed & Mustard.

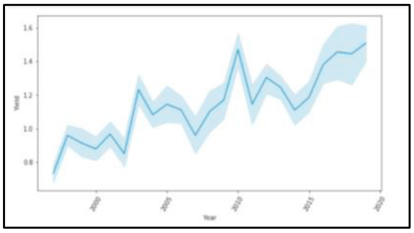

Fig. 7 Yield vs Year (Rapeseed & Mustard).

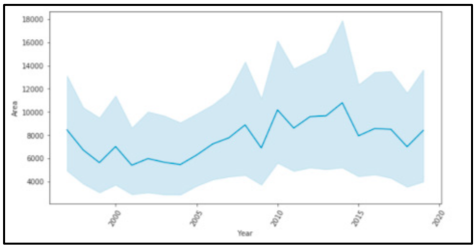

Fig. 8. Area under irrigation vs Year (Barley).

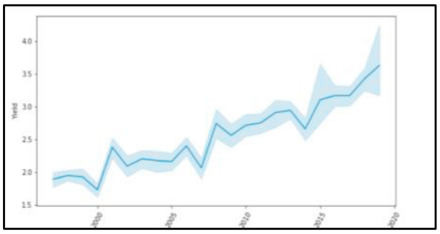

Fig. 9 Yield vs Year (Barley).

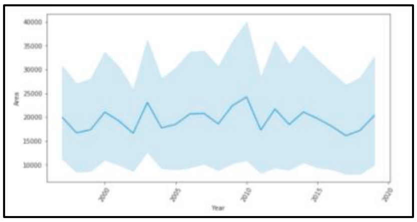

Fig. 10. Area under irrigation vs Year (Jowar).

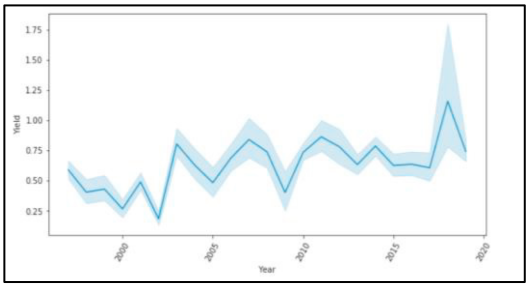

Fig. 11. Yield vs Year (Jowar).

The r2 score, MAE (Mean Average Error), and RMSE (Root Mean Square Error) have been chosen to assess the accuracy of the models deployed.

Except for LSTM, the dataset has been divided into a ratio of 98 percent training and 2% testing, with a random state set at 71. For LSTM, a ratio of 99 percent training set to 1% test set with the same value for the random state was used. Two separate analyses were carried out, the first using only area, production, year, and district as independent features for yield prediction, and the second including the mean seasonal rainfall as well as the soil type of the specific district to understand the effect of these variables on yield. Following an analysis of the results for each model individually, the following observations were made:

In the case of the Random Forest algorithm, the R2 score is the highest, thus, it has higher accuracy when compared to other models for both cases, and that is about 97% in the case of selected parameters and about 96.3% in case of all parameters (figure 12). The RMSE for the selected parameters is about 3.2% and the MAE score is about 2.1%, in the case of all parameters, the RMSE is 3.5% and MAE is 2.5%. In the case of the Support Vector Machine or commonly referred to as the SVM algorithm, the R2 score is 90.3% when considering the selected factors (figure 13), and for all factors considered the score is about 89.8%. For the selected parameters the RMSE is 5.8% and MAE is 4.74%, in the case of all parameters the RMSE is 5.9% and MAE is 4.78%. In the case of the Lasso Regression algorithm, the R2 score is 79.2% in the case of selected parameters and in the case of all parameters it is 81.4% (figure 14). The RMSE is 8.5% and MAE is 6.2% in the case of selected parameters while RMSE is 8.06% and MAE is 5.8% in the case of all parameters. In the case of the Gradient Descent algorithm, the R2 score is 67.2% for selected parameters, while on the dataset with all the parameters the score is 73.7% (figure 15). For the RMSE and MAE values for the selected parameters dataset, the values are 10.7% and 8.8% respectively. In the case of the dataset with all the parameters, the values are RMSE as 9.6% and MAE as 7.9%. In the case of the Long Short-Term Memory or commonly known as the LSTM algorithms, when this model is applied to both types of the dataset, the R2 score is 76.05%, RMSE is 49.8% and MAE is 41.2% for the dataset with selected parameters (figure 16). While in the case of all parameters the R2 score is 75.7%, RMSE is 50.1% and MAE is 41.6%. Same parameters have been used for all the models while splitting into training and testing sets to get the best possible accuracy, but in the case of LSTM since the algorithm in general works best and requires a larger database than what is being provided, the testing set has been reduced by 1%. This provided a higher accuracy when applied to the dataset.

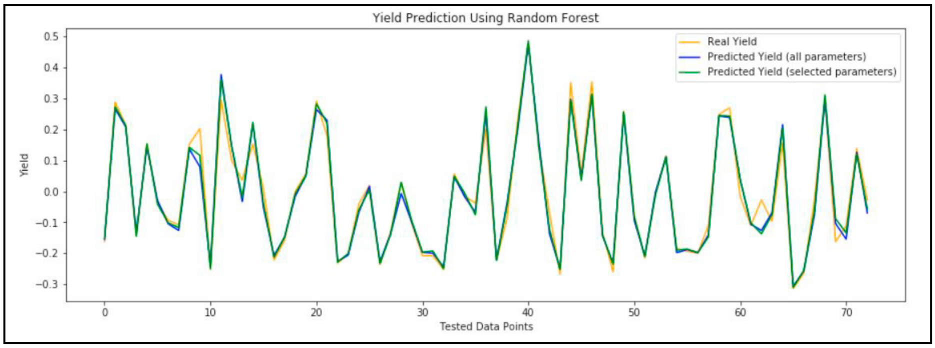

Fig. 12. Model performance of Random Forest.

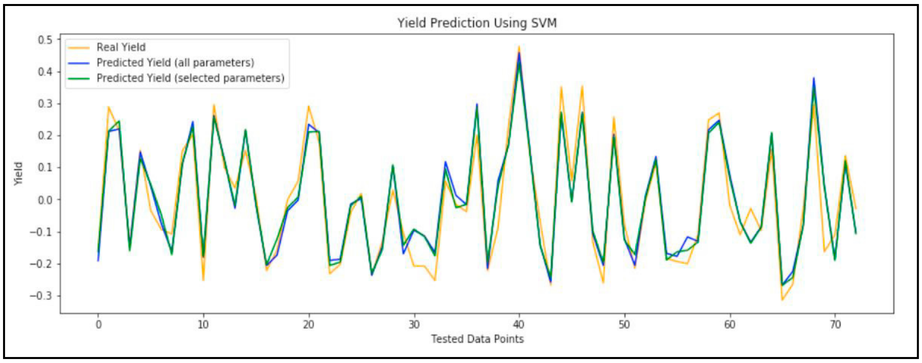

Fig. 13. Model performance of SVM.

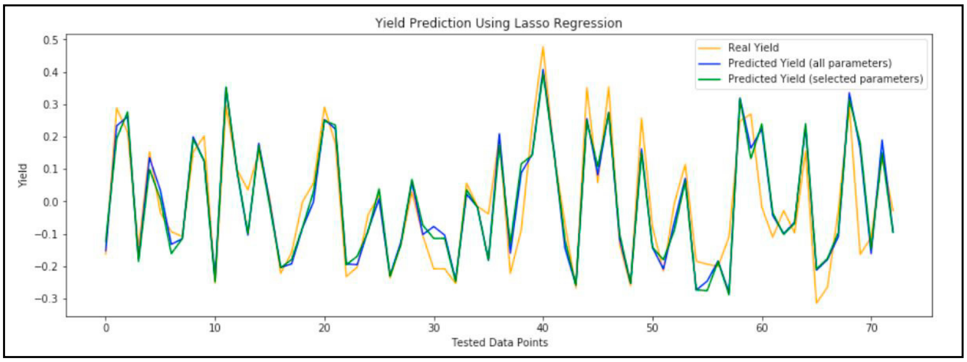

Fig. 14. Model performance of Lasso Regression.

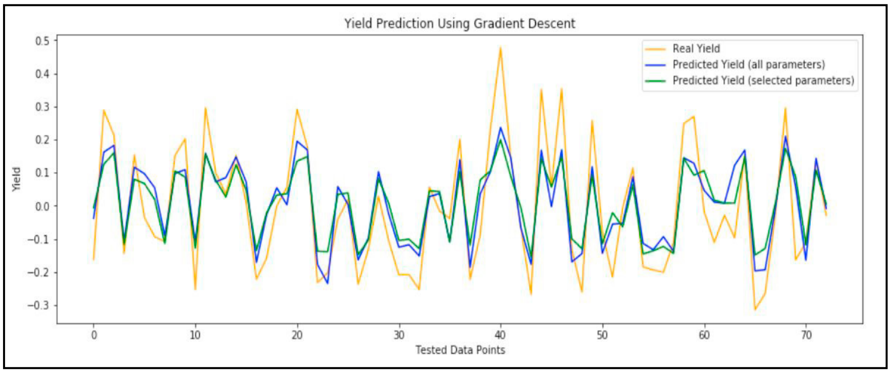

Fig. 15. Model performance of Gradient Descent.

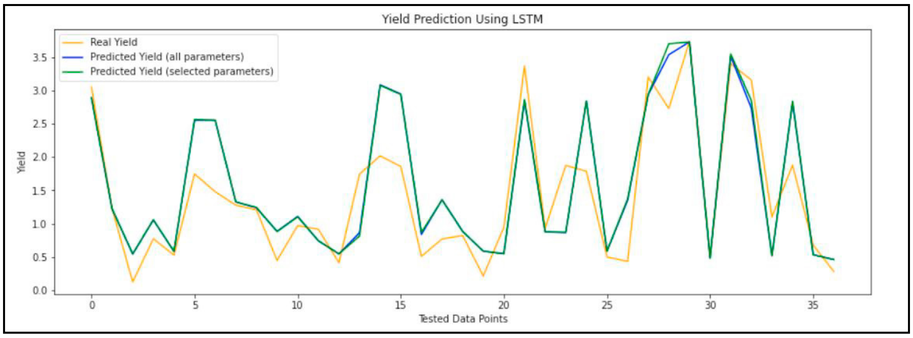

Fig. 16. Model performance of LSTM.

Table 3 shows the comparison of all models using only area and production parameters whereas in Table 4 models are compared using all the parameters.

Table 3. Model performance using only area and production as parameters.

MODELS	R2 SCORE	RMSE	MAE
RANDOM FOREST	0.9706683562457551	0.03210438555877871	0.02171092742447838
SVM	0.9033044480088029	0.0582907187068775	0.04744273781182322
LASSO REGRESSION	0.7929212370102475	0.08530292881366584	0.06279053471454211
GRADIENT DESCENT	0.6721446803333357	0.10733399123627024	0.08833348922241419
LSTM	0.7605970571904346	0.4989149684766394	0.4128063470375437

Table 4. Model performance using all parameters.

MODELS	R2 SCORE	RMSE	MAE
RANDOM FOREST	0.9638436515081396	0.03564416484020771	0.025122868659073126
SVM	0.8982433161230687	0.059796757425198625	0.047852253033345137
LASSO REGRESSION	0.8146824212217434	0.08069645755381685	0.0588093331954562
GRADIENT DESCENT	0.737347767385024	0.09606976093705671	0.07906984158002965
LSTM	0.75781934584411	0.50180099201939	0.4164927767185484

Based on observations, most models have nearly identical accuracy in the case of both types of datasets. Most models' accuracy decreases when the scientific parameters of soil and mean rainfall are added, but here the scenario is the opposite in models like Lasso Regression and Gradient Descent. Although the STM model has a higher R2 score than the Gradient Descent algorithm, the RMSE and MAE values are also very high, which is not a good sign for any predictive model. Since the dataset is based on actual data as provided by the official Government of Rajasthan website, the accuracy based on tested data points, shows that creating a data-driven predictive model can boost and maximize the yield for almost all the crops. This can help in the growth of the agricultural economy when mixed with the know-how and experience of the farmers.

4. Conclusion

Based on all of the discussions and analyses, it is clear that the machine learning models used - Random Forest, Support Vector Machine (SVM), and Lasso Regression - outperform the deep learning models used - Gradient Descent and Long Short-Term Memory (LSTM) - in terms of accuracy. This could be because, when compared to other models, models like LSTM require a larger quantum of data for a better predictive analysis. Furthermore, based on the observations, most of the models perform better on the specified parameters, whereas models such as Gradient Descent and Lasso Regression perform better when applied to the dataset with all of the characteristics. While soil and rainfall quantity are important in crop production and general farming, it can be concluded that a deeper investigation of these elements, as well as a larger database, is required for real-life research of such elements using prediction models. Finally, it can be concluded that the Random Forest algorithm outperforms all other models when applied to any of the datasets.

The current research can be extended into performing further analysis and forecasting the factors that influence crop yield. A larger dataset and more historically accurate data about the environment and weather during each crop year is required to identify best performing model between deep learning and machine learning models. To find the best-performing technique, more deep learning models need to be tested on the dataset. In the field of crop yield prediction, remote sensing data could be merged with the district-level statistical data to improve the model's performance.

References

[1] Kavita, Ms, and Pratistha Mathur. (2020) "Crop Yield Estimation in India Using Machine Learning." In 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), 220–224. doi:10.1109/ICCCA49541.2020.9250915.

[2] Fan, Wu, Chen Chong, Guo Xiaoling, Yu Hua, and Wang Juyun. (2015) "Prediction of Crop Yield Using Big Data." In 2015 8th International Symposium on Computational Intelligence and Design (ISCID), 1:255–

260. doi:10.1109/ISCID.2015.191.

[3] Kamath, Pallavi, Pallavi Patil, Shrilatha S, Sushma, and Sowmya S. (2021) "Crop Yield Forecasting Using Data Mining." Global Transitions Proceedings, International Conference on Computing System and its Applications (ICCSA- 2021), 2 (2): 402–407. doi:10.1016/j.gltp.2021.08.008.

[4] Wigh, Daniel S., Jonathan M. Goodman, and Alexei A. Lapkin. "A Review of Molecular Representation in the Age of Machine Learning." WIREs Computational Molecular Science n/a (n/a): e1603. doi:10.1002/wcms.1603.

[5] Kavita, and Pratistha Mathur. (2021) "Satellite-Based Crop Yield Prediction Using Machine Learning Algorithm." In 2021 Asian Conference on Innovation in Technology (ASIANCON), 1–5. doi:10.1109/ASIANCON51346.2021.9544562.

[6] Bali, Nishu, and Anshu Singla. (2022) "Emerging Trends in Machine Learning to Predict Crop Yield and Study Its Influential Factors: A Survey." Archives of Computational Methods in Engineering 29 (1): 95–112. doi:10.1007/s11831-021-09569-8.

[7] van Klompenburg, Thomas, Ayalew Kassahun, and Cagatay Catal. (2020) "Crop Yield Prediction Using Machine Learning: A Systematic Literature Review." Computers and Electronics in Agriculture 177 (October): 105709. doi:10.1016/j.compag.2020.105709.

[8] Yalta, Nelson, Kazuhiro Nakadai, and Tetsuya Ogata. (2017) "Sound Source Localization Using Deep Learning Models." Journal of Robotics and Mechatronics 29 (1): 37–48. doi:10.20965/jrm.2017.p0037.

[9] Shen, Dinggang, Guorong Wu, and Heung-Il Suk. (2017) "Deep Learning in Medical Image Analysis." Annual Review of Biomedical Engineering 19 (1): 221–248. doi:10.1146/annurev-bioeng-071516-044442.

[10] Minaee, Shervin, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. (2021) "Deep Learning--Based Text Classification: A Comprehensive Review." ACM Computing Surveys 54 (3): 1–40. doi:10.1145/3439726.

[11] Belgiu, Mariana, and Lucian Drăguţ. (2016) "Random Forest in Remote Sensing: A Review of Applications and Future Directions." ISPRS Journal of Photogrammetry and Remote Sensing 114 (April): 24–31. doi:10.1016/j.isprsjprs.2016.01.011.

[12] Suthaharan, Shan. (2016) "Support Vector Machine." In Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning, edited by Shan Suthaharan, 207–235. Integrated Series in Information Systems. Boston, MA: Springer US. doi:10.1007/978-1-4899-7641-3_9.

[13] Hochreiter, Sepp, A. Steven Younger, and Peter R. Conwell. (2001) "Learning to Learn Using Gradient Descent." In Artificial Neural Networks — ICANN 2001, edited by Georg Dorffner, Horst Bischof, and Kurt Hornik, 87–94. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. doi:10.1007/3-540-44668-0_13.

[14] Sherstinsky, Alex. (2020) "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network." Physica D: Nonlinear Phenomena 404 (March): 132306. doi:10.1016/j.physd.2019.132306.

[15] Yu, Yong, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. (2019) "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures." Neural Computation 31 (7): 1235–1270. doi:10.1162/neco_a_01199.

[16] Ranstam, J, and J A Cook. (2018) "LASSO Regression." British Journal of Surgery 105 (10): 1348. doi:10.1002/bjs.10895.

[17] https://data.world/thatzprem/agriculture-india

[18] https://agriculture.rajasthan.gov.in/content/agriculture/en/Agriculture-Department-dep/agriculture-statistics.html

[19] https://www.indiawaterportal.org/met_data

[20] https://water.rajasthan.gov.in/content/water/en/waterresourcesdepartment/WaterManagement/IWRM/annualrainfall.html#

[21] https://agriculture.rajasthan.gov.in/content/agriculture/en/Agriculture-Department-dep/agriculture-statistics.html

[22] https://agriculture.rajasthan.gov.in/content/agriculture/en/Agriculture-Department-dep/Departmental-Introduction/Agro-Climatic-Zones.html