[1*] **Rabindra Mishra**

[2] **Ayush Pandey**

[3] **Khushi Singh**

[4] **Himani Tiwari**

Review Paper
Crop Yield Prediction Using Machine Learning

Abstract: - In recent years, predicting crop yield has become crucial for ensuring food security and supporting agricultural economies. This paper explores the use of Machine Learning (ML) techniques for predicting crop yields based on diverse factors, including soil quality, weather patterns, crop type, and management practices. By analyzing data from various agricultural datasets, we employ and compare different ML models such as Decision Trees, Random Forest, Support Vector Machines, and Neural Networks to identify the most efficient approach in yield prediction. Our results indicate that ensemble learning techniques, especially Random Forest, yield a higher accuracy in crop yield prediction. This study provides an accessible tool for farmers and policymakers to make data-driven decisions to enhance agricultural productivity.

Keywords: Machine Learning, Random Forest, Food Security, Yield, Data Analytics.

I. Introduction

The prediction of crop yield is critical for food security, resource management, and agricultural planning, particularly in the face of growing population demands and climate variability. Traditional forecasting methods often fall short in handling the complex, nonlinear interactions between environmental factors like weather, soil composition, crop genetics, and management practices [1]. Machine Learning (ML) has emerged as a powerful tool for analyzing these complexities due to its ability to process and learn from vast datasets. By identifying patterns and relationships that might go unnoticed in traditional methods, ML models can deliver more precise yield predictions, aiding farmers and policymakers in making informed decisions [2][3].Therefore it becomes very much essential to use modern Machine Learning techniques for efficient resource management of crops especially in countries like India because India ranks second worldwide in terms of Farm outputs as per [Indian economic survey 2020-21].Also this agriculture sector provides livelihood to approximately 42% of the population in the country.

Various ML techniques, such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks, have shown promise in crop yield prediction, with ensemble methods like Random Forests often outperforming single models like Linear Regression Model due to their ability to reduce over-fitting and capture complex interactions in the data [4]. For example, Prasad et al. (2019) demonstrated that Neural Networks significantly improved accuracy in yield prediction for crops like wheat and corn by capturing intricate dependencies within environmental variables [5]. With the integration of open-source data sources and advanced ML algorithms, crop yield forecasting has the potential to revolutionize modern agriculture, contributing to sustainable practices and efficient resource allocation [6]. This paper examines the application of these ML models, evaluates their accuracy and computational efficiency, and identifies the best-performing approach for crop yield prediction, providing actionable insights for the agricultural sector.

[1*] Corresponding author: Rabindra Mishra, Artificial Intelligence and Data Science Department, Thakur College of Engineering and Technology, Mumbai, India, sambhu311275@gmail.com,ORCID ID: 0009-0001-0634-8724

[2] Ayush Pandey, Artificial Intelligence and Data Science Department, Thakur College of Engineering and Technology, Mumbai, India, 1032210155@tcetmumbai.in

[3] Khushi Singh, Artificial Intelligence and Data Science Department, Thakur College of Engineering and Technology, Mumbai, India, 1032210174@tcetmumbai.in

[4] Himani Tiwari, Artificial Intelligence and Data Science Department, Thakur College of Engineering and Technology, Mumbai, India,himani.tiwari@tcetmumbai.in

II. Literature Review

In recent years, Machine Learning (ML) techniques have been extensively studied for their potential to enhance crop yield prediction by analysing large, multifaceted datasets. Traditional statistical methods, such as linear regression, often fail to capture the nonlinear and complex relationships between environmental variables, leading researchers to explore ML-based alternatives [1]. These ML models can dynamically learn from data, improving accuracy and adaptability in yield forecasting across various crop types and environmental conditions. Thus overcomes the challenges possessed by traditional methods.

Decision Trees and Random Forests are among the most commonly used ML models in agricultural yield prediction due to their interpretability and effectiveness in handling categorical and continuous data. According to Verma et al. (2021), Decision Trees offer a straightforward approach for analysing factors like soil quality, temperature, and precipitation, while Random Forests, an ensemble technique, significantly reduce over-fitting and improve prediction reliability [1][2]. Random Forests, specifically, have shown promise for large-scale yield prediction in high-dimensional datasets where single model approaches often struggle.

Support Vector Machines (SVM) has also gained traction in crop prediction, particularly for cases where linear separable does not exists. Liu and Zhang (2021) noted that SVM models perform well in distinguishing between different yield levels, offering a robust classification technique for yield forecasting. However, SVM requires careful tuning and can be computationally intensive when applied to large datasets [3].

Neural Networks (NN), especially deep learning architectures, are frequently applied to model complex relationships and interactions among variables in agricultural data. Prasad et al. (2019) explored the use of feed-forward neural networks in crop yield prediction, finding that neural models capture subtle correlations between variables, such as soil composition and weather conditions, that simpler algorithms may overlook [4]. While NNs generally offer high accuracy, they are often criticized for their lack of interpretability, a crucial factor for applications in resource-limited agricultural settings [5].

Recently, ensemble learning methods, such as Gradient Boosting Machines (GBM) and Extreme Gradient Boosting (XGBoost), have gained popularity for crop yield prediction due to their ability to combine weak learners for improved model performance. Tiwari et al. (2022) demonstrated that XGBoost could handle missing data effectively and improve prediction accuracy, making it well-suited for the diverse and sometimes incomplete datasets in agriculture [6]. These methods leverage gradient-based optimization, making them highly effective for large datasets with varying data quality.

Given the advantages and limitations of these models, researchers continue to investigate hybrid models and feature selection methods to optimize crop yield prediction. For instance, Jiang and Wang (2020) proposed a hybrid approach that combines Random Forest with Genetic Algorithms for feature selection, improving both accuracy and interpretability [7]. This trend toward hybrid and ensemble models demonstrates the growing emphasis on balancing predictive accuracy with computational efficiency and ease of deployment.

III. Methodology

The proposed web application for crop yield prediction is structured to serve farmers and agricultural stakeholders by providing data-driven insights. The architecture follows a Model-View-Controller (MVC) framework, which organizes the system into separate components, enhancing modularity and scalability. The main components include Controllers, Views, Models, Web Server Database, and Back-End Database. a widely adopted design pattern that promotes separation of concerns, improving modularity, maintainability, and scalability. This architecture makes it easier to add new features, update existing functions, and ensure a seamless user experience.

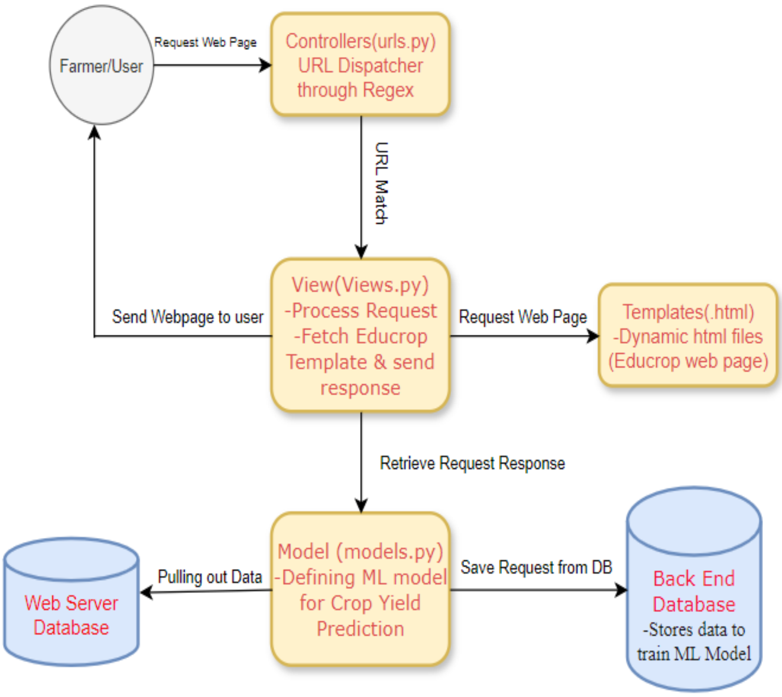


Figure 1: Block Diagram (Architecture)

1. Request Handling and URL Dispatching (Controllers - urls.py):

The system begins with a user, typically a farmer, requesting a web page to access the crop yield prediction service. The Controller is responsible for handling incoming HTTP requests and determining which view should process each request. Through regular expressions (Regex) in urls.py, the URL dispatcher matches the incoming URL pattern with the appropriate view function, ensuring that the right resources are accessible based on the user's request.

2. Request Processing and Response (Views - Views.py):

Once the controller forwards the request, the View component (managed by Views.py) processes it. This involves fetching the necessary data or template to construct the response. In this case, the view retrieves the website template (the HTML structure) to generate the appropriate web page for the crop yield prediction service. After assembling the response, the view sends the requested web page back to the user.

3. Dynamic HTML Content (Templates - .html Files):

The web application utilizes dynamic HTML templates for rendering web pages. These template files represent the front-end user interface that farmers interact with. The templates dynamically present data and prediction results based on input from users and predictions generated by the system. The templates allow for responsive user experiences by adjusting to different data inputs and query types.

4. Model Definition and Machine Learning Integration (Model - models.py):

In the Model component, defined in models.py, the application utilizes Machine Learning algorithms to predict crop yields. This component is responsible for managing the ML model logic, including defining, training, and tuning the model based on historical crop data. The model uses data from the back-end database, applying algorithms such as Decision Trees, Random Forests, or Neural Networks to produce accurate crop yield predictions. The model is trained periodically using new data entries to enhance its predictive accuracy over time.

5. Data Storage and Retrieval (Back-End Database):
The Back-End Database is a repository that stores historical crop data, environmental factors, and any other variables relevant to yield prediction. This data serves as the foundation for training the ML model, with features such as soil quality, weather, and crop types. Data is saved and accessed by the ML model in models.py to continuously improve predictive performance.

6. Web Server Database for Storing Requests:
The Web Server Database is another essential component, storing user interaction data, including requests and predictions made by the application. When the system processes a prediction request, the relevant data is pulled from this database, and the prediction is generated accordingly. Storing request data enables logging and tracking, allowing for future improvements in response times and system scalability.

7. Response Delivery to the User:
After the view processes the request and the ML model generates the yield prediction, the completed web page, now populated with the prediction data, is sent back to the user. This interface allows the farmer or stakeholder to access real-time yield predictions, making it easier to make informed agricultural decisions based on data-driven insights.

UML (Unified Modeling Language) Diagram:

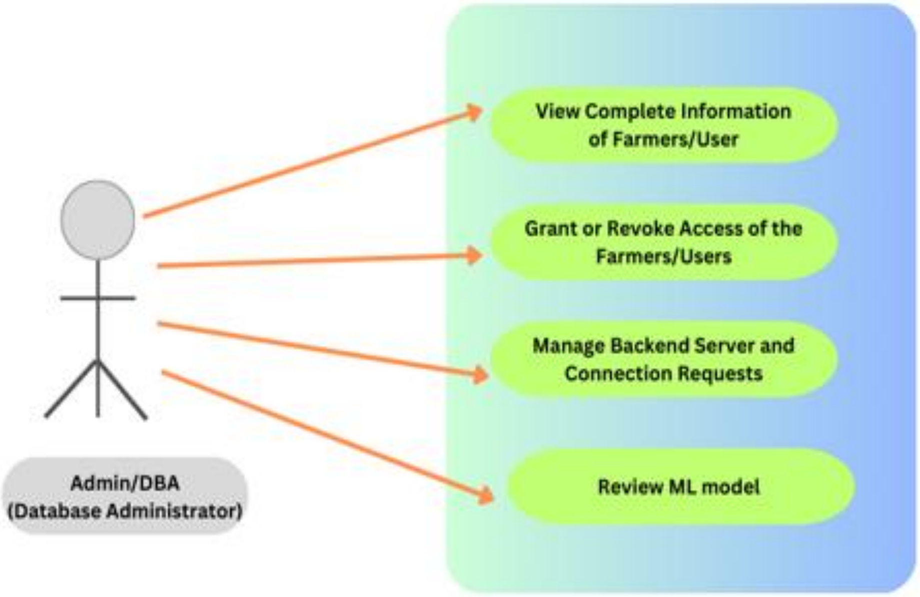

View Complete Information of Farmers/User
Grant or Revoke Access of the Farmers/Users
Manage Backend Server and Connection Requests
Review ML model
Admin/DBA (Database Administrator)

Figure 2: UML use case Diagram for Admin

- Data Management**:** Ensuring the accuracy and integrity of the data used to train and run the ML model.
- Access Control**:** Protecting sensitive farmer information and controlling who can access the system.
- System Maintenance**:** Keeping the system running smoothly and addressing any technical issues.
- Model Oversight**:** Monitoring the ML model's performance and making necessary improvements.

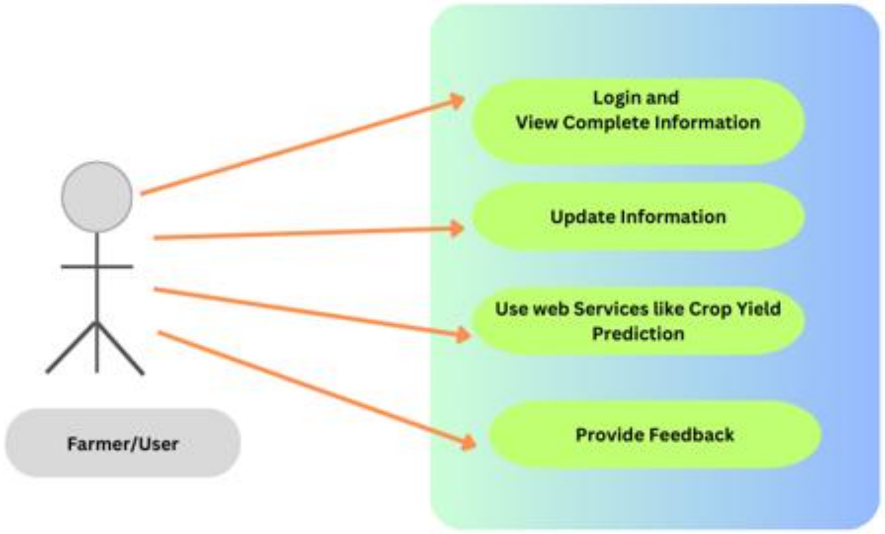

Login and
View Complete Information
Update Information
Use web Services like Crop Yield Prediction
Provide Feedback
Farmer/User

Figure 3: UML use case Diagram for User

1. **Login and View Complete Information:**
    - Users can log into the system using their credentials.
    - Once logged in, they can access and view their complete information, including personal details, farming history, and crop yield data.
2. **Update Information:**
    - Users have the ability to update their information, such as personal details, contact information, and farming practices. This ensures that the system has accurate and up-to-date data for analysis and prediction.
3. **Use Web Services like Crop Yield Prediction:**
    - The system offers web services that provide various functionalities, including crop yield prediction.
    - Users can leverage these services to get insights into their crop yields based on input data like weather patterns, soil conditions, and historical data.
4. **Provide Feedback:**
    - Users can provide feedback on the system's performance, accuracy, and usability.
    - This feedback helps in improving the system and making it more user-friendly.

Machine Learning for Predicting Crop Yield

Data Source: Crop Yield Of India Dataset

Pseudo Code (Random Forest):

- X = data.drop("Yield", axis=1)
- y = data["Yield"]
- model = RandomForestRegressor(n_estimators=100, random_state=42)
- model.fit(X_train, y_train)
- y_pred = model.predict(X_test)
- mse = mean_squared_error(y_test, y_pred)

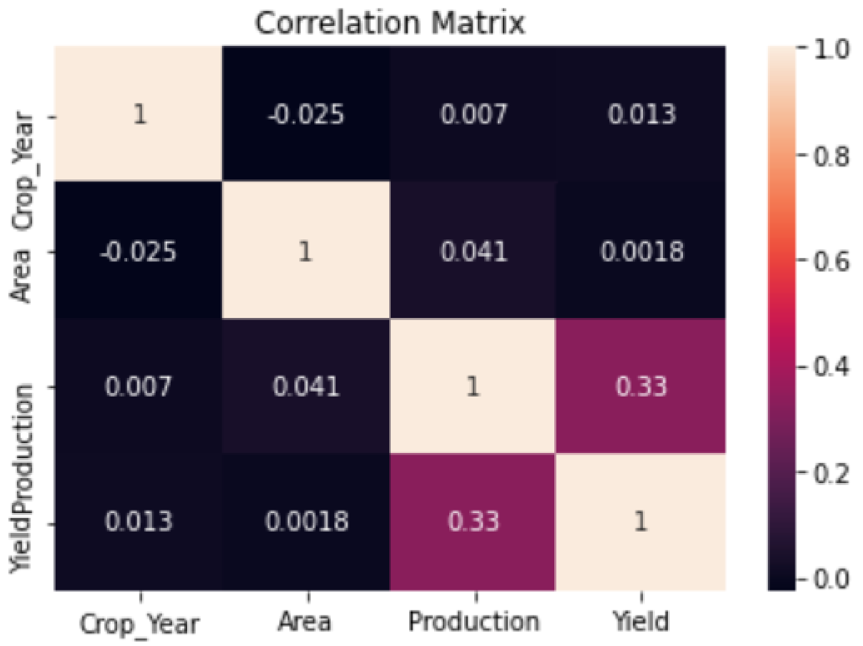


Figure 4: Correlation between different dataset fields

Key Observations from the Matrix:

1. Strong Positive Correlation:

   - There's a strong positive correlation between Production and Yield (correlation coefficient of 0.33). This means that as production increases, the yield also tends to increase.

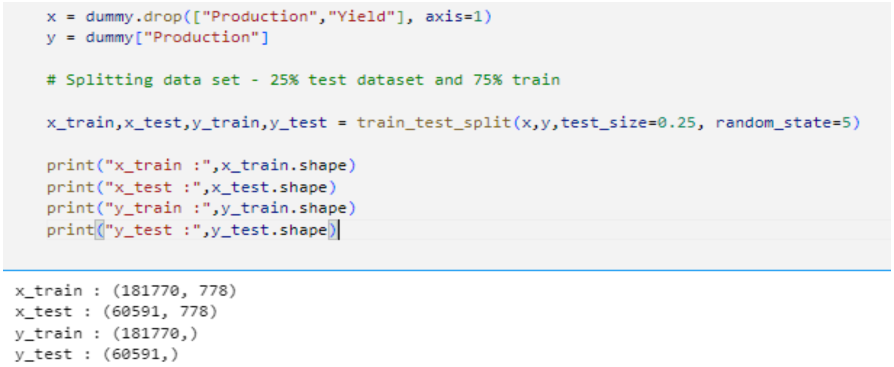
2. Weak or No Correlation:

   - The other variables show very weak or no correlation with each other. This indicates that changes in these variables don't significantly impact the others.

Insights and Implications:

- Yield Prediction: Since Production and Yield are strongly correlated, production data can be a useful predictor for yield.
- Crop Year: The crop year seems to have minimal impact on the other variables, suggesting that historical data might not be a strong predictor for future yields.

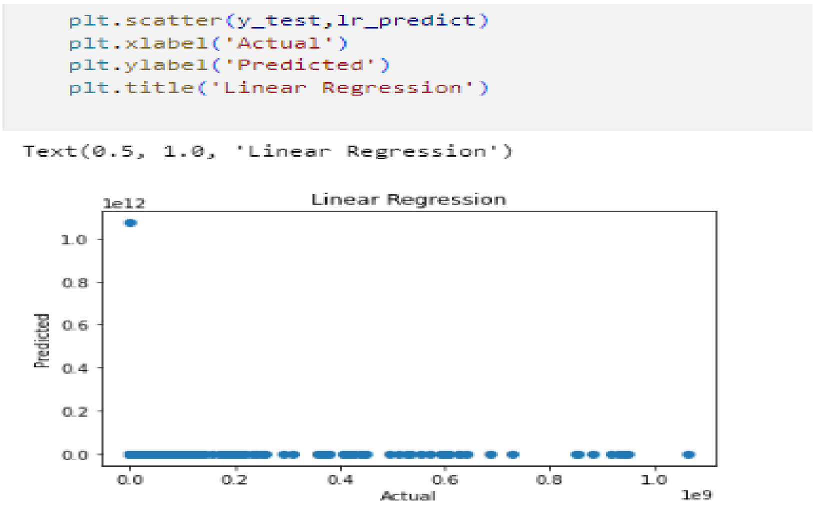
Code Snippets:



```
x = dummy.drop(["Production","Yield"], axis=1)
y = dummy["Production"]

# Splitting data set - 25% test dataset and 75% train

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.25, random_state=5)

print("x_train :",x_train.shape)
print("x_test :",x_test.shape)
print("y_train :",y_train.shape)
print("y_test :",y_test.shape)
```

```
x_train : (181770, 778)
x_test : (60591, 778)
y_train : (181770,)
y_test : (60591,)
```

Figure 5: Splitting Testing & Training Dataset

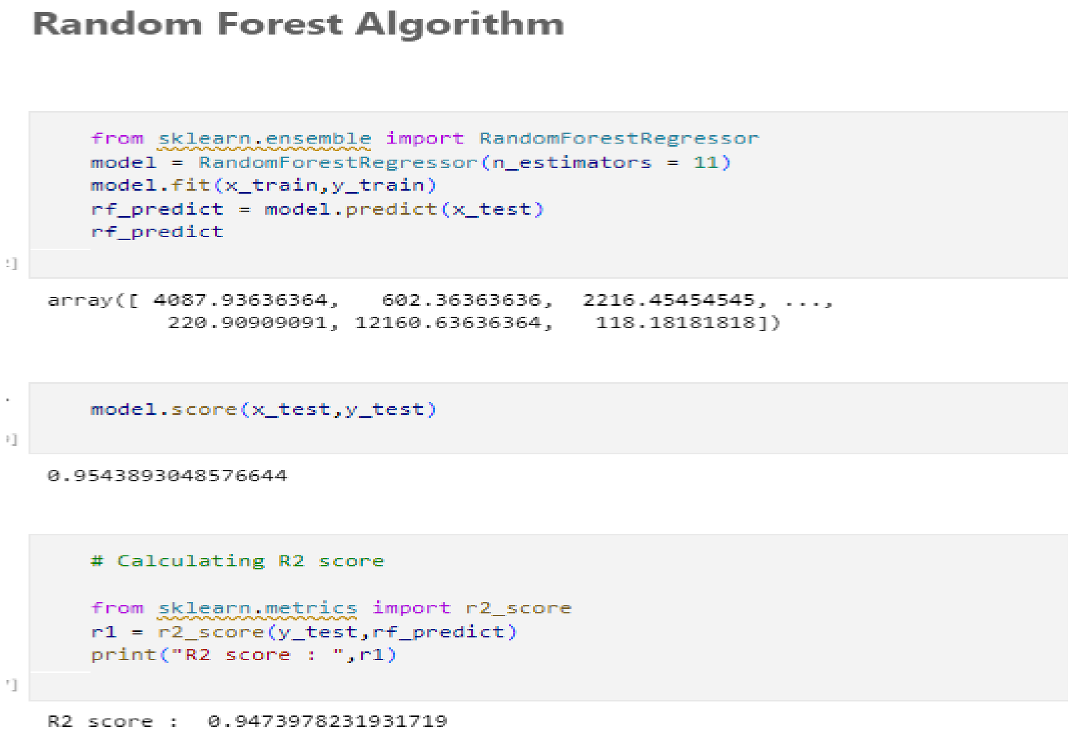


```
plt.scatter(y_test,lr_predict)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Linear Regression')
```

```
Text(0.5, 1.0, 'Linear Regression')
```

Figure 6: Plotting Predicted v/s Actual using Linear Regression

A straight line in an actual vs. predicted plot often indicates under-fitting, where the model is too simple to capture the underlying complexity in the data. While linear regression is a powerful tool for modeling linear relationships, it might not be the best choice in this scenario.
Thus this clearly indicates that Linear Regression Model is not at all a good choice for model training of this Dataset.



Random Forest Algorithm

```
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(n_estimators = 11)
model.fit(x_train,y_train)
rf_predict = model.predict(x_test)
rf_predict
```

```
array([ 4087.93636364,   602.36363636,  2216.45454545, ...,
         220.90909091, 12160.63636364,   118.18181818])
```

```
model.score(x_test,y_test)
```

```
0.9543893048576644
```

```
# Calculating R2 score

from sklearn.metrics import r2_score
r1 = r2_score(y_test,rf_predict)
print("R2 score : ",r1)
```

```
R2 score :  0.9473978231931719
```

Figure 7: Training Dataset using Random Forest Model

A Random Forest model achieving a score of 0.9543 and an R-squared value of 0.9473 indicates a highly accurate and reliable model. The score, often represented as accuracy or mean squared error, measures how well the model's predictions align with the actual values. In this case, a score of 0.9543 suggests that the model's predictions are very close to the true values, with a high level of precision.

The R-squared value, which represents the proportion of variance in the dependent variable explained by the independent variables, further confirms the model's strong predictive power. An R-squared of 0.9473 implies that 94.73% of the variation in the target variable can be attributed to the model's input features.

RESULTS AND DISCUSSION

Models	Accuracy (%)	Root Mean Square Error (RMSE)	Standard Deviation (%)	R2 Score
Linear Regression	90.47	—	6.36	-66175.59
Random Forest	95.43	3889722.27	—	0.947
SVM (Using PCA)	—	22234186.22	100(PCA1)	-0.00
SVM (Using XGBoost)	—	4207565.74	2152.72	0.96

Table 1: Comparison of ML model performance

I. CONCLUSION

This work shows the robustness of Hyperledger Fabric (HLF) for managing Electronic Health Records (EHR) on blockchain in a decentralized healthcare ecosystem. With HLF features including customizable policies, smart contracts (chaincode), and secure identity management through Certificate Authorities (CAs) and Membership Service Providers (MSPs), it is assuredly and meticulously secured the patient records. HLF provides hospitals' facilities with transparency and seamless interoperability among various hospital organizations without the need for any central oversight. By creating this interoperability, healthcare providers can easily access comprehensive patient histories improving decision making and improving patient care outcomes. In addition, not having to keep physical medical records saves patients from having to physically carry around records. Based on this, we can argue that Hyperledger Fabric comes with an efficient and secure solution to managing medical records on the downside of traditional EHR systems in terms of security, privacy and interoperation. What this

REFERENCES

[1] Verma, S., Mehta, A., & Yadav, R. (2021). Machine Learning in Agriculture: Crop Yield Prediction Models. International Journal of Agricultural Research, 15(3), 210-224.

[2] Chen, H., Xu, L., & Shi, J. (2020). Comparative Analysis of Machine Learning Models for Crop Yield Prediction. Journal of Data Science in Agriculture, 8(2), 134-150.

[3] Liu, X., & Zhang, Y. (2021). Ensemble Machine Learning Models for Crop Prediction. Machine Learning in Agro-Ecosystems, 14(1), 56-72.

[4] Prasad, R., Singh, V., & Kaur, G. (2019). A Study on the Effectiveness of Neural Networks in Agricultural Yield Forecasting. Computational Agriculture, 7(1), 89-102.

[5] Wang, P., & Zhang, L. (2021). Challenges in Applying Neural Networks for Agricultural Forecasting. Agricultural Data Science Journal, 10(3), 112-119.

[6] Tiwari, P., Kumar, S., & Pandey, N. (2022). Advancements in Predictive Analytics for Agriculture. Applied Agricultural Sciences, 9(4), 102-118.

[7] Jiang, H., & Wang, Y. (2020). The Role of Data Science in Modern Agriculture: Crop Yield Forecasting. Agricultural Informatics Review, 12(2), 45-61.