Hello World

*

Meet Mehta
Department of Computer Engineering
D J Sanghvi College of Engineering
Mumbai, India
mehtameetoff2802@gmail.com

Abhijit Saha
Department of Computer Engineering
D J Sanghvi College of Engineering
Mumbai, India
abhijits1802@gmail.com

Siddharth Fulia
Department of Computer Engineering
D J Sanghvi College of Engineering
Mumbai, India
siddharthfulia18092002@gmail.com

Tushar Jain
Department of Computer Engineering
D J Sanghvi College of Engineering
Mumbai, India
tusharrahuljain@gmail.com

Dr. Narendra M. Shekokar
Department of Computer Engineering
D J Sanghvi College of Engineering
Mumbai, India
narendra.shekokar@djsce.ac.in

***Abstract*—In India, a very small percentage of farmers do soil testing, with one of the major reasons being the time taken to get soil test reports from government labs and the fact that private soil testing is not affordable to the farmers. Except for this, many farmers find it difficult to interpret the soil testing report. Also, the soil testing report does not provide farmers with suggestions as to what crop they should grow. To overcome this problem, a system has been suggested in this paper that will use multiple sensors to collect soil information and send the information to the mobile app, which will suggest the best crop a farmer should grow based on the soil conditions. To provide crop suggestions, the system uses a RandomForest machine learning model, which is trained on a custom soil nutrients dataset. The dataset has many different parameters, such as nitrogen, phosphorus, rainfall, temperature, iron, sulphur, zinc, and ph value. The machine-learning model has an accuracy of about 92**

***Index Terms*—crop, crop recommendation, nitrogen, phosphorus, potassium, zinc, sulphur, iron, internet of things**

I. INTRODUCTION

India is a farming nation; agriculture is the primary source of livelihood for about 58% of India's population, but till today only a small percent of farmers do soil testing. Multiple government as well as private models have been developed to solve this problem, but none have been able to increase the percentage of farmers doing soil testing to very great extents. Currently, farmers have to either get the testing done at government facilities or at private labs. Getting testing done at government labs is a very time-consuming process as many labs are not well equipped and the ones that are well equipped are overloaded. Due to this, the testing and result delivery take time, which may cause farmers to miss the right time for sowing seeds. If farmers use private labs for testing, then the cost of testing is very high, making it unaffordable for them. Except for this, all the labs just provide a report telling the quality and amount of nutrients in the soil; none of them tell which crop should be grown based on the quality of the soil. Many farmers are unable to understand the report, which makes it difficult for them to select the right crop to grow. Recently, some startups have entered the farming industry. Their products help farmers get the soil testing done at a low cost and much faster, but they have very little coverage and also do not tell the best crop that should be grown.

As a solution, we will be making an ML model trained on a custom dataset that has multiple parameters, including nitrogen, phosphorus, rainfall, temperature, iron, sulphur, zinc, and ph value. The ML model will help farmers predict the right crop to grow based on the nutrients in the soil. First, using sensors and other techniques, all soil information such as NPK values, micronutrient content, and rainfall in that area will be collected. This information will then be entered into the ML model, which will then predict the crop that the farmer should grow. The model will be deployed using an API, and farmers will be able to access it using our mobile application. Using a mobile app will ensure that a vast number of farmers can use the crop recommendation service without having to buy or learn much about computers. The mobile application app will also provide information about different schemes that could benefit farmers, and farmers will also be able to monitor the history of soil testing.

Identify applicable funding agency here. If none, delete this.

II. LITERATURE REVIEW

"Soil Fertility Prediction," "Crop Suggestions," and "Crop Yield Prediction" are the titles of the three sections of the papers by Jagdeep Yadav et al. Different datasets were employed for each section. The soil dataset with 15 properties was employed for soil analysis, the crop dataset with 4 attributes (temperature, humidity, pH, and rainfall) was utilised for crop recommendation, and the yield dataset with 6 attributes was utilised for crop yield prediction. Several types of models, including J48, SVM, Random Forest, and ANN, were utilised for each component. ANN had the best accuracy across all three datasets, followed by Random Forest.

Karan Mehta et al. in their paper suggested predicting soil fertility and did crop prediction based on the soil nutrient readings. In the event that the soil fertility is insufficient to support the growth of a specific crop that the farmer wanted to produce, they also suggested offering a list of crop fertilisers. The pH value of the soil is measured as part of the soil analysis utilising an Arduino UNO, pH metre, and Wi-fi module (ESP8266). The back-end server side maps this data to NPK values, and the soil is categorised as Low, Medium, or High class depending on the NPK levels. This was implemented via a website that displays the mapped pH and NPK data, as well as the predicted crop and crop fertiliser list, in order to enhance the current nutrient value of the soil.

A crop recommender system was put forth by S. Bangaru Kamatchi et al. and trained on a 47-attribute weather dataset from the previous three years obtained from UCI Data Repository. Feature selection, data preprocessing, data prediction, data recommendation, and data classification made up their technique to forecast crops depending on the selected weather characteristics. A time series function is created to get the accurate weather conditions from the dataset, and this function factors in on the recommendation of the crop later in the ANN. The hybrid recommendation system generates a list of ranked crop suggestions by their weights using ANN for crop prediction and classification. Case Based Reasoning (CBR) is utilised to boost the success ratio of the ANN model.

In contrast to the literature that has previously been proposed, when analysing soil and recommending crops, our method takes into account a number of factors, including micronutrients and other essential requirements. A mobile application is utilised to initiate soil testing and retrieve the results from our deployed model.

III. Proposed Method

A. Dataset Creation

Our model does crop recommendation using multiple parameters which include nitrogen, phosphorus, potassium, sulphur, iron, zinc, temperature, rainfall, and ph. No dataset was publicly available which covered all these parameters so a custom dataset Modified Crop Dataset (MC dataset) was created. To create CropDataset a base dataset Crop Recommendation Dataset (CR dataset) published on Kaggle was referenced. The CR dataset consists of 22 crops and the following attributes: nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, pH, and rainfall. Out of the 22 crops, 12 unique crops were identified for the MC dataset, which are rice, maize, chickpea, lentil, pomegranate, banana, mango, grapes, apple, orange, cotton and coffee. For all these crops, the attributes N, P, K, pH, rainfall and temperature were taken from the CR dataset and for the rest of the attributes for MR dataset which are zinc, sulphur and iron, various resources which included websites and published documentations were used. From all this collected information, a dataset of 12 crops of 100 rows each was created. Since the dataset created was small, a new dataset was created taking the ranges of each attribute of every crop. So, the new dataset consisted of 5000 rows for every crop and was created by randomising the values between the ranges that was obtained for each crop's attribute. This dataset didn't consist of any noise, so using Condition Tabular Generative Adversarial Network (CTGAN) some noisy data was introduced and the final MC dataset of the same size was created.

B. Model Training

For the recommendation system, multiple models were tested which included Random Forest (rf) and multiple combinations of Random Forest and Convolution Neural Network (CNN). Very first model was a simple rf model which gave an accuracy of about 92%. Second model used was a rf+CNN2 model, this model was a combination of rf and CNN where CNN included 2 dense layers. Accuracy of rf+CNN2 was about 90%. The third model used was rf+CNN4, this model was similar to rf+CNN2 but had 4 dense layers and had accuracy of about 91%. The last model was rf+CNN6 which was similar to rf+CNN4 and rf+CNN2 but had 6 dense and accuracy was about 88%. The most accurate of all was the rf model which was finally implemented and is being used in the system.

Model	Accuracy
rf	0.9186
rf + cnn2	0.9032
rf + cnn4	0.9052
rf + cnn6	0.8792
svm	0.8923

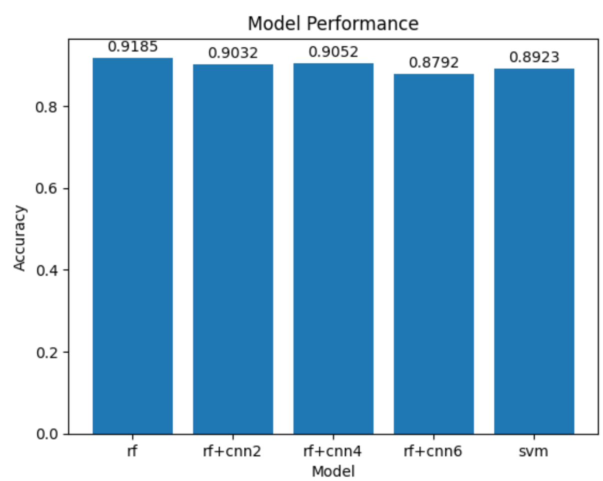

Fig. 1. Event Performance

C. Hardware

The hardware used in the system includes a DS18B20 temperature sensor, a Ph sensor, an NPK sensor, an Arduino UNO, a RS-485 to TTL converter, and an ESP32 WiFi module. The DS18B20 temperature sensor is attached to pin 5 of the Arduino UNO, while the NPK sensor and the Ph sensor both use the RS-485 model to transfer data to the Arduino UNO. In the end, all of the data is transmitted to the ESP32 WiFi Module, which relays the information to the server. The Arduino's 5V and ground pins are used to connect the WiFi module to the board. For the purposes of communication, the Wifi module is attached to the Arduino nano through the TX and RX pins, which correspond to pins 1 and 0 on the Arduino UNO, respectively. Ph and NPK sensors both have their input voltages connected to an external source, and both the RS-485 and the external adapter are used as grounding points. Both the NPK and the Ph sensors, which are coupled to the RS-485 module, have a modbus A pin and a modbus B pin. The yellow pin is the modbus A pin, while the blue pin is the modbus B pin. The input voltage for the temperature sensor is 3.3V, and the data wire is coupled to a resistor with a 4.7K ohm value.

IV. Implementation

A. Hardware Implementation

The data is first gathered from the NPK sensor and converted using an RS-485 module, then from the DS18B20 temperature sensor, and finally from the Ph sensor, which is gathered and converted once more using an RS-485 module. Using the ESP32 WiFi module, all the data is combined and then uploaded to the server. The app then gathers the information from the server.

B. Frontend Architecture

The ReactNative mobile app is the project's front end. It uses API calls to the Firebase to get data, a hosted model to recommend crops, and a ThingSpeak server to get data from sensors. The app has a standard screen for logging in, and new users can sign up for an account by going to the sign-up screen. Once the user enters their information, they are sent back to the login screen so they can use the app. After logging in, the user will be taken to the landing page, which has a dashboard that shows crop suggestions from the past. When a user clicks on a card, they will be taken to a detailed view of that card. This view has a lot of information, such as the NPK values, temperature, and other factors that were used to make the recommendation for that crop, as well as the date of the recommendation.

The user can use the left drawer navigator to get to the testing and recommendations screen.

The screen has fields for nitrogen, phosphorus, potassium, pH, temperature, rainfall (in mm), sodium, iron, and zinc. These are the things that are needed to make a recommendation. The user can auto-fill data for nitrogen, phosphorus, potassium, pH, and temperature by clicking the 'Get Sensor Data' button on the top, which will fetch the data measured by soil testing sensors. The rainfall data can be found by clicking the 'Go to IMD website' link and entering the rainfall data for the appropriate region in the app. Soil testing data for sodium, iron, and zinc has to be entered manually by the user. The crop recommendation result can be found by clicking on the 'Get Crop Recommendation' button.

There is also a section for looking at government programmes for farmers. You can find out more about these programmes by clicking on the link in the application.

The user interface is made by using the built-in core components of React Native and the following packages: icons by "react-native-vector-icons," navigation by "React Navigation."

C. Backend Architecture

To access the ml model an API has been created using flask. The API has been hosted on a free to use deployment server pythonanywhere. The ml model can be accessed using the API an carry out crop recommendation the nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, pH, rainfall, zinc, sulphur and iron values have to be provided as an input to the API which then access the ml model and provides the value to it, based on the input received a crop is recommended and provided as the output which the API gives in the form a json list. To access the ml model an API has been created using flask. The API has been hosted on a free to use deployment server pythonanywhere. The ml model can be accessed using the API an carry out crop recommendation the nitrogen (N), phosphorus (P), potassium (K), temperature, humidity, pH, rainfall, zinc, sulphur and iron values have to be provided as an input to the API which then access the ml model and provides the value to it, based on the input received a crop is recommended and provided as the output which the API gives in the form a json list.

V. Future Scope

1) Model Retraining: Currently, the model is trained on a small dataset of about 60000 rows and 12 unique crops. Over time, as more data is collected from farmers, the model could be retrained so as to increase accuracy and cover a much larger range of unique crops for recommendation.
2) Multiple Crop Recommendation: Multiple Crop Recommendation: Currently, the model just suggests a single crop, but as more data is collected, new features like multiple crop recommendation can be provided, giving the farmer the choice to select which crop he wants to grow based on other factors like return on investment.
3) Crop Mapping: Many times it happens that farmers select what to grow based on which crop gave the best return on investment the previous year, and it may lead to multiple farmers growing the same crop, and overall the market price may go down. So a feature can be implemented that shows farmers the demand for a particular crop and what crops are being grown in their particular area, making it easier to select a crop that would give them the best return on investment.

VI. References

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

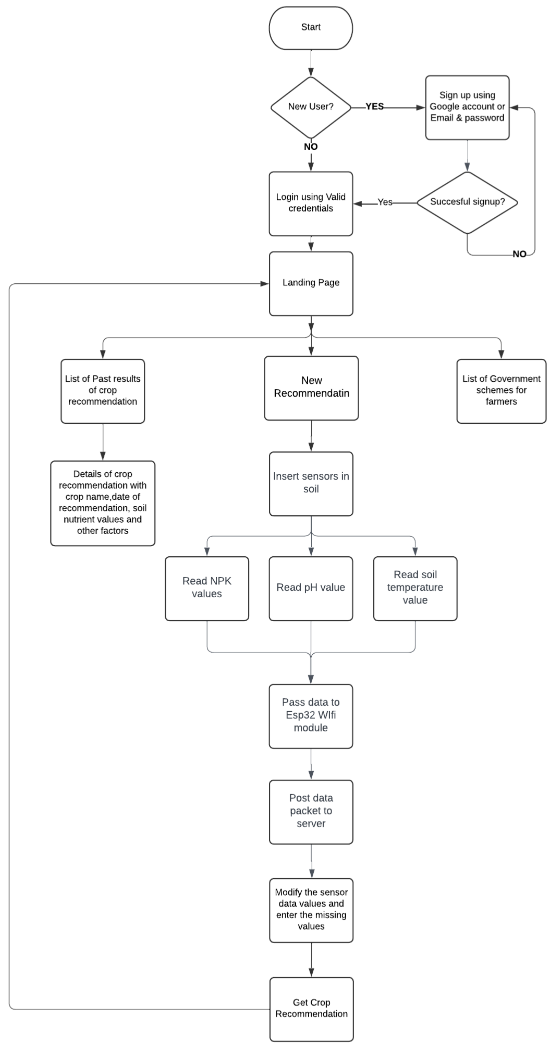

Fig. 2. Event Performance

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.