An IoT Crop Recommendation System with k-NN and LoRa for Precision Farming

Mateus Cruz, Samuel Mafra and Eduardo Teixeira

***Abstract*— Choosing the planting site is a complex and decisive task for crop success, but data can help with this task. Wireless Sensor Networks can capture large volumes of data, but manual analysis may be impossible depending on the number of devices and sensors deployed. Furthermore, Machine Learning techniques are handy for processing data and detecting patterns and are widely used nowadays. The union of these two technologies is promising, presenting itself as a path to precision agriculture. This paper proposes a system based on Wireless Sensor Networks capable of detecting the best regions to for cultivating plants such as Kidney Beans, Pomegranate, and Apple. The system uses LoRa technology and Time Division Multiplexing for excellent coverage, various devices at the same channel, and local processing, eliminating the need for the Internet.**

***Keywords*— IoT, LoRa, Machine Learning, Edge Computing.**

I. Introduction

Agriculture is a significant factor that sustains human life on planet earth. It is through agriculture that much of the food consumed is produced today. However, the rapid expansion of cities and the necessary reduction of deforestation make the process of choice and use of a local for planting extremely important. Analogously, the local planting conditions are decisive variables in plant development and production. Moreover, agriculture is now a significant source of growth in the economy of many countries. At the same time, farmers continue to use traditional, less precise methods in their planting. So that efficiency, production, and fruit quality suffer. Nowadays, different technologies are researched and developed for agriculture, including Wireless Sensor Networks (WSN)[1].

WSN is an approach that helps farmers transform traditional agriculture into precision farming. This approach can be helpful in different aspects, allowing collect data periodically, real-time data processing and analysis, among other utilities. In WSN, sensors are connected and spread across the crop for data capture and transmission, and it is widely used for climate change monitoring. In addition, sensor networks are inexpensive and flexible in their implementation and can be applied to the most diverse scenarios and processes, including agriculture [2].

Most WSN applications in agriculture periodically collect crop-relevant data such as temperature, humidity, and minerals in the soil. All these data can be used to increase crop yield and quality. Therefore, choosing the ideal location for planting plays a crucial role in the outcome of the harvest and must be carefully chosen.

However, manually analyzing this large amount of data can become a challenge. Because of this, several Machine Learning (ML) applications have been developed and proposed for automatic data processing and classification. These applications can process the collected data and present relevant information to the farmer. Many agricultural applications have been developed for this purpose to automate data processing on a large scale [3].

This paper presents a WSN that uses Long Range communication (LoRa) and ML to classify and identify the best locations for planting specific plants. The application uses microcontrollers, microprocessors, LoRa modules, batteries, and sensors to perform all computation at the edge, eliminating the necessity of an Internet connection. This way, the application can be used in remote locations where mobile networks and the Internet are unavailable.

The remainder of this paper is organized as follows. Section II discusses the literature reviews and the main technologies used in IoT agriculture solutions. Section III describes the proposed IoT application architecture and gives a detailed workflow of the services provided. Section IV presents and analyses the results instead. Finally, Section V concludes the paper and suggests further future works.

II. Related works

In [4], several sensors are used in a WSN approach to collect data on the plantation autonomously. A mobile application gives the user access to the collected data in real-time. These local data can be used to increase the efficiency and quality of the crop. The authors main proposal is to offer the possibility of monitoring his crop from anywhere at any time.

WSN can play an essential role in data collection related to disease prediction. Authors in [5] propose a WSN application to predict crop diseases in agriculture. This application uses ML models running in the cloud, requiring mandatory Internet access for its implementation. Applications for greenhouses are also being developed. The control of temperature and humidity in a greenhouse is essential for developing the species cultivated there. Therefore, authors in [6] proposed an application capable of predicting the ambient temperature using Artificial Neural Networks (ANN). This way, greenhouses

Mateus Raimundo da Cruz, Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí-MG, e-mail: mateusraimundo@mtel.inatel.br; Samuel Baraldi Mafra, Departamento de Engenharia de Telecomunicações, Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí-MG, e-mail: samuelbmafra@inatel.br; Eduardo Henrique Teixeira,Instituto Nacional de Telecomunicações, Santa Rita do Sapucaí-MG, e-mail: eduardoteixeira@dtel.inatel.br.
This work is partially supported by RNP, with resources from MCTIC, Grant No. No 01245.010604/2020-14, under the Brazil 6G project of the Radiocommunication Reference Center (Centro de Referência em Radiocomunicações - CRR) of the National Institute of Telecommunications (Instituto Nacional de Telecomunicações - Inatel), Brazil, the National Council for Scientific and Technological Development-CNPq (403827/2021-3), FAPESP (2021/06946-0)

become more autonomous and precise regarding temperature control. Also, the use of data in the decision process is one pillar of precision agriculture.

In [7] an application for agriculture based on WSN that encompasses ML and computer vision is proposed. However, most of the processing is not done locally, requiring the user to have a reliable Internet connection for the system to work. At the same time, the solution integrates technologies such as LoRa and 4G, thus broadening the applicable scenarios. Through LoRa communication, it is possible to reach significant distances, and 4G has a wide diffusion all over the globe. The application aims to use the collected images and data to improve planting efficiency, thus achieving better results.

ML is also used for disaster prediction, and the authors in [8] developed a WSN solution for flood prediction. Flooding is one of the most common disasters and can cause significant damage to crops. The proposed solution uses WSN to collect data such as humidity, pressure, and water level. Afterward, it injects this data into the input of an ANN responsible for predicting the location's flooding. This way, the farmer ends up being able to mitigate the possible damage caused by the disaster.

The proposed system makes use of a WSN for monitoring variables and LoRa technology for communication between devices. A machine learning model is also installed to indicate the best planting options for a given region.

III. Proposed WSN and ML Model

The main objective of the WSN system is to present what crop type can benefit from the local conditions. The system is composed of two primary devices named Sensor and Collector nodes. For that, the Sensor node periodically captures local data and sends it through LoRa to the Collector node. Then, the collector node real-time analyzes the captured data and presents what of the 22 crop varieties can take more advantage of the presenting variables. With this approach, farmers can monitor various locations simultaneously and choose better locations to plant. Figure 1 shows an overview of the entire system.

To capture metrics of different and distant locations simultaneously, all the two devices use LoRa to exchange data in a Machine-to-Machine approach. LoRa is a low-power Radio Frequency (RF) communication generally used in low-transmission rates and long-distance scenarios. The technology is an excellent option for IoT devices and sensor network applications [9]. Time Division Multiplexing (TDM) is the technique used to allow multiple sensor nodes to work within a single LoRa channel, so each sensor node has a set period allowed to transmit data. The receiver side always waits for messages and uploads all the information to the Internet through the Message Queuing Telemetry Transport (MQTT) protocol. The MQTT is a lightweight publish-subscribe-based messaging protocol extensively used in IoT applications [10]. The protocol is developed to connect devices in a remote location with limited network bandwidth. The MQTT in the application sends the data to the NodeRED [11] tool, which is responsible for inserting the data into the database, dashboard, and ML model.

The sensing device comprises Arduino Uno R3, batteries, LoRa module, and sensors. The prototyping board contains a microcontroller responsible for reading, pre-processing, and controlling information flow in the sensor node. Batteries are necessary to allow the implementation in remote zones, the LoRa module for transmitting all the captured data, and sensors are responsible for collecting the local data. The sensors used are DHT11 (humidity and temperature sensor) and hygrometer (soil moisture sensor). The pre-processing done by the Sensor node is in structuring the data in JavaScript Object Notation (JSON) structure. This format is later transformed into a JSON in the Collector node and injected into the input of the ML model.

The Collector node comprises Raspberry Pi 4B, LoRa module, and batteries. The board is responsible for receiving, processing, and uploading the data to the cloud. The entire system offers an online and offline database and dashboard, allowing heterogeneous implementation scenarios. LoRa module is necessary to guarantee a long-range connection to receive the sensing nodes data. Once received, all data in string format is transformed into a JSON file and sent to the ML model for real-time sensor data classification. The model output is stored in a cloud and offline time-series database called InfluxDB. Afterward, a cloud dashboard captures the data and shows the results for the final user.

K-Nearest Neighbors (k-NN) were chosen among the various possible techniques to classify the collected data. The technique was chosen because it allows clustering of the data, trying to identify patterns in their behavior. k-NN follows instance-based learning, where all computations are postponed until the time of classification. The classification takes place by considering the majority vote of its neighbors, basing the classification on the same class as its closest neighbors. During the training period, N clusters with similar characteristics are created. When inserted an unknown sample into the model, all the distances between the samples inserted in training and the new observation can be computed [13]. Some proceedings are followed to develop all these ML capabilities: (I) Data mining, (II) Model training, and (IV) Deployment.

The dataset used [16] in the application contains 2200 observations, 8 variables and 22 plant types and their respective optimal planting conditions. The dataset is already fully labeled, only mining and adapting the dataset to use only variables collected by sensors. The data mining process aims to improve the dataset quality for model training, eliminating all possible outliers and gaps in the data. First, a dataset scan is done to ensure that Not a Number (NaN) and outliers values are not present on the dataset. Afterward, variables that are not captured by sensors are also taken from the dataset, leaving only temperature and humidity data in the dataset. Also, a plot of each crop's temperature and humidity median and the standard deviation is used to understand the crop's ideal condition of each variable in the model classification. Figure 2 and Figure 3 illustrates these information.

Removing features from the dataset can cause the model metrics to decay. Therefore, a training run was also performed using all the variables presented in the dataset to verify the impact caused on the model's performance due to their

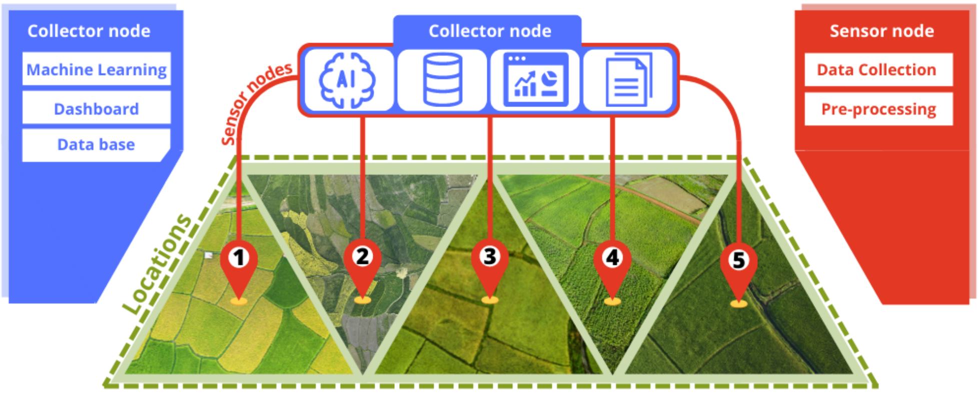

Fig. 1. The sensor node captures the data locally and transmits the data to the collector node, which is responsible for processing, storing, and displaying the data.

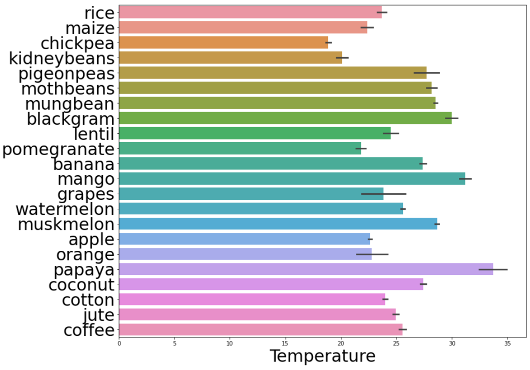

Fig. 2. Respective crops and their temperature profiles.

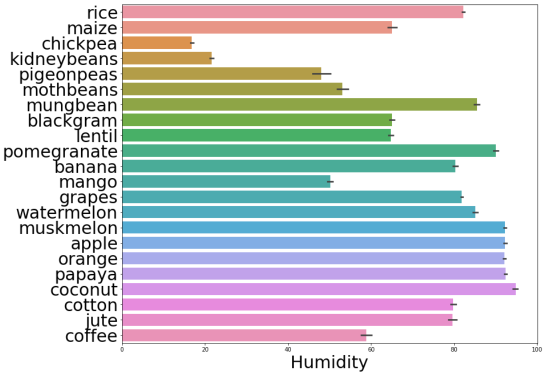

Fig. 3. Respective crops and their humidity profiles.

removal in the previous process. The results of both model training are presented in section IV, while the conclusion on the differences is discussed in section V.

The second phase is model training. The training is done through the Scikit Learn library on a local computer. After, the final model is exported and installed with Scikit learn and NodeRED tools on the Raspberry Pi board. The dataset is split into 80% for training purposes and 20% for model testing. This approach helps generate performance model metrics and configure hyperparameters, ensuring a better model score. The model developed uses the following hyperparameters: 22 Neighbors, Uniform weights, Leaf Size of 10, and Euclidean distance for Minkowski metric.

The number of classes is generally related to the number of neighbors used K-NN model. In this case, the number of neighbors equals the number of classes. The weight parameter defines the weight function used by the model for computing the distance of the observation. The model uses the Uniform approach. In this case, the model makes the predictions considering all neighbors weighted equally. The Leaf Size hyperparameter defines the complexity of the model, affecting the speed of the construction, query, and the memory required for storage. The default value is 30, but the optimal value depends on the nature of the problem, and in this case, the value of 10 is the minimum found that does not affect the model performance. Finally, the Minkowski metric is the distance metric used for the tree [14].

The model is deployed on a Raspberry Pi board in the final step. Some packages are installed in the Raspberry Pi operational system for real-time classifying of the various nodes information. Although the main packages used are Scikit Learn and NodeRED. Scikit Learn is an open-source ML library that supports supervised and unsupervised learning [15]. In this case, a supervised multi-label classification is utilized. The model constructs the clusters during the training phase based on the relations between observations (temperature and humidity data) and their respective labels (crop types). The new observation classifications are done by computing the distance between the new observation and the already created clusters. The multi-label technique can classify the local condition into 22 different crop varieties.

NodeRED runs locally and connects the inputs and outputs of services as illustrated in Figure 4, making it possible to use various tools and services in parallel. The final classification presented by the model is sent directly to the local databases and cloud through an Application Programming Interface (API). The database used is InfluxDB, a time-series-based database. InfluxDB enables local, and cloud databases and integration with NodeRED and Grafana [12] tools, both of which were implemented in the application. The simultaneous implementation of the databases ensures redundancy in the application and ensures that the application runs entirely

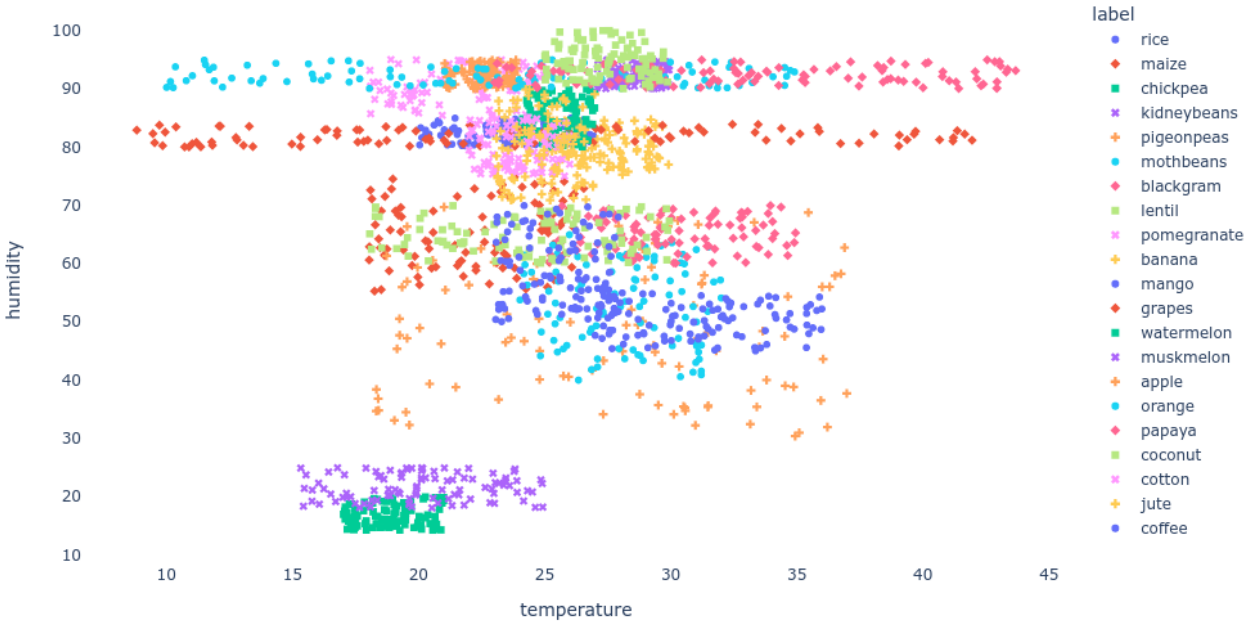

Fig. 4. Ilustration of the humidity and temperature samples.

offline. The dashboard used is built on top of the Grafana service. Another API retrieves the database data and shows it in the dashboard.

The result is displayed in a bar displaying the model's classification in different colors. It makes it intuitive for the user to identify how long that place has been in ideal conditions for each plant. In addition, several nodes can be visualized simultaneously. Grafana offers several tools for customizing the graph's colors, making it possible to use other visualization forms.

IV. Results

The testing step can be separated into two stages: (I) Testing the ML model and (II) Testing communication and application. The first relates to obtaining metrics such as the model accuracy, Recall, and F1-Score. The second aims to test the LoRa communication and the application's operation.

The model is tested during the training phase. The Scikit Learn library provides the user with several functions capable of capturing and displaying metrics about model performance to the developer. These tools allow measuring each class results in isolation, getting a deeper and more detailed view of the model's capability. The metrics utilized to test the model classify capacity are: (I) Precision, (II) Recall, and (III) F1-Score. All individual results presented by the model can be seen in Table I.

Each metric evaluates different aspects of the model. For computing each of them, four parameters are used to evaluate the model performance: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative(FN). True Positives determine that the model's predicted and actual ranking are the same. Similarly, True Negatives indicate that the actual and predicted values are the same but are now negative. False Positives, in turn, indicate an error made by the model in its classification, so the actual and predicted values are not the same. Finally, the False Negatives also indicate that the predicted and actual values are not the same, but now negative [15].

TABLE I

Isolated score of each plant in the dataset.

	Precision	Recall	F1-Score
Rice	0.67	0.91	0.77
Maize	0.29	0.28	0.29
Chickpea	0.76	0.86	0.81
Kidneybeans	0.88	1.00	0.94
Pigeonpeas	0.80	0.67	0.73
Mothbeans	0.43	0.53	0.47
Mungbean	0.50	0.45	0.48
Blackgram	1.00	0.59	0.74
Lentil	0.46	0.24	0.32
Pomegranate	1.00	0.90	0.95
Banana	0.38	0.56	0.45
Mango	0.50	0.30	0.37
Grapes	0.50	0.94	0.65
Watermelon	0.75	0.38	0.50
Muskmelon	0.83	0.79	0.81
Apple	0.78	0.96	0.86
Orange	0.78	0.47	0.58
Papaya	0.67	0.57	0.62
Coconut	1.00	0.21	0.35
Cotton	0.75	0.52	0.62
Jute	0.17	0.46	0.25
Coffee	0.35	0.63	0.45

Precision measures how much of all data classified as positive is positive.

$$Precision = \frac{TruePositives(TP)}{TruePositives(TP) + FalsePositives(FP))}$$

The Recall is an important metric when we do not have a representative dataset for all included classes. Recall tells the percentage of data classified as favorable compared to the number of positives in the sample.

$$Recall = \frac{TruePositives(TP)}{TruePositives(TP) + FalseNegatives(FN)}$$

F1-Score, in turn, metric unites Precision and recall to bring a single number that determines the overall quality of our model.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

The model presents promising results for some plants like kidney beans, pomegranate, apple, grapes, chickpea, and muskmelon that achieve an F1-Score above 80%. In contrast, the model has difficulty identifying the ideal condition for jute, maize, lentil, and other plants. Therefore, the applied dataset does not contain all variables to identify ideal plant locations. A plot of the temperature and humidity data was made to better understand the clusters created, as presented in Figure 4.

The model misclassifications are due to the similarity of the observations, with several plants having the same set of ideal planting temperature and humidity. Because of this, the user can choose to cultivate plants with the same temperature and humidity characteristics or add other sensor types to the application to create more accurate recommendations. In summary, although each plant has the exact ideal temperature and humidity conditions at planting, variables such as Nitrogen, Potassium, and Calcium can differentiate and indicate more precise locations. In the tests performed using these variables, the model achieved more promising results like overall Precision of 96.74%, Recall of 92.62%, and F1-Score of 95.46%.

V. Conclusion

This paper proposed a system capable of detecting the best planting locations for 22 different plants. The system uses ML and LoRa for classification and long-range communication between nodes, respectively. The tested system uses only the local humidity and temperature. However, due to the similarities between the plants, it is necessary to include more soil and environmental information to allow a more accurate model. The system works entirely with no necessity of the Internet, and this approach can ensure greater privacy and security of the data collected. However, the same functionalities, such as dashboard and database, are available in the cloud. Thus, the farmer also has the option to activate the features and have access to the collected data from any place and time. New models should be tested and implemented, and new sensor sets should also be integrated into future studies. Also, the power consumption of each component in the network, packet loss during transmission, packet transmission delay, and error rate should be investigated in future work. The Scikit Learn library offers several other classification models like Decision Trees and Random Forests that can be applied for the same function and guarantee better results. Also, other sensors such as Nitrogen and Potassium can be integrated into the application to ensure better accuracy and Precision of the model in its classifications.

References

[1] A. Balafoutis et al., "Precision Agriculture Technologies Positively Contributing to GHG Emissions Mitigation, Farm Productivity and Economics," Sustainability, vol. 9, no. 8. MDPI AG, p. 1339, Jul. 31, 2017. doi: 10.3390/su9081339.

[2] A. Lanzolla and M. Spadavecchia, “Wireless Sensor Networks for Environmental Monitoring,” Sensors, vol. 21, no. 4. MDPI AG, p. 1172, Feb. 07, 2021. doi: 10.3390/s21041172.

[3] A. Cravero, S. Pardo, S. Sepúlveda, and L. Muñoz, “Challenges to Use Machine Learning in Agricultural Big Data: A Systematic Literature Review,” Agronomy, vol. 12, no. 3. MDPI AG, p. 748, Mar. 21, 2022. doi: 10.3390/agronomy12030748.

[4] D. D. K. Rathinam, D. Surendran, A. Shilpa, A. S. Grace and J. Sherin, "Modern Agriculture Using Wireless Sensor Network (WSN)," 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS), 2019, pp. 515-519, doi: 10.1109/ICACCS.2019.8728284.

[5] H. Wani and N. Ashtankar, "An appropriate model predicting pest/diseases of crops using machine learning algorithms," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017, pp. 1-4, doi: 10.1109/ICACCS.2017.8014714.

[6] G. Codeluppi, A. Cilfone, L. Davoli and G. Ferrari, "AI at the Edge: a Smart Gateway for Greenhouse Air Temperature Forecasting," 2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), 2020, pp. 348-353, doi: 10.1109/MetroAgriFor50201.2020.9277553.

[7] G. Kakamoukas et al., "A Multi-collective, IoT-enabled, Adaptive Smart Farming Architecture," 2019 IEEE International Conference on Imaging Systems and Techniques (IST), 2019, pp. 1-6, doi: 10.1109/IST48021.2019.9010236.

[8] P. Mitra et al., "Flood forecasting using Internet of things and artificial neural networks," 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016, pp. 1-5, doi: 10.1109/IEMCON.2016.7746363.

[9] S. Devalal and A. Karthikeyan, "LoRa Technology - An Overview," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2018, pp. 284-290, doi: 10.1109/ICECA.2018.8474715.

[10] C. R. M. Silva and F. A. C. M. Silva, "An IoT Gateway for Modbus and MQTT Integration," 2019 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), 2019, pp. 1-3, doi: 10.1109/IMOC43827.2019.9317637.

[11] G. Tricomi, Z. Benomar, F. Aragona, G. Merlino, F. Longo and A. Puliafito, "A NodeRED-based dashboard to deploy pipelines on top of IoT infrastructure," 2020 IEEE International Conference on Smart Computing (SMARTCOMP), 2020, pp. 122-129, doi: 10.1109/SMARTCOMP50058.2020.00036.

[12] G. Suciu, C. Istrate and M. Diţu, "Secure smart agriculture monitoring technique through isolation," 2019 Global IoT Summit (GIoTS), 2019, pp. 1-5, doi: 10.1109/GIOTS.2019.8766433.

[13] L. Dey, S. Chakraborty, A. Biswas, B. Bose, and S. Tiwari, "Sentiment Analysis of Review Datasets Using Naïve Bayes’ and K-NN Classifier," International Journal of Information Engineering and Electronic Business, vol. 8, no. 4, pp. 54–62, Jul. 2016, doi: 10.5815/ijieeb.2016.04.07.

[14] A. Kolte, B. Mahitha and N. V. G. Raju, "Stratification of Parkinson Disease using python scikit-learn ML library," 2019 International Conference on Emerging Trends in Science and Engineering (ICESE), 2019, pp. 1-4, doi: 10.1109/ICESE46178.2019.9194627.

[15] S. K. Vishwakarma, Akash and D. S. Yadav, "Analysis of lane detection techniques using openCV," 2015 Annual IEEE India Conference (INDICON), 2015, pp. 1-4, doi: 10.1109/INDICON.2015.7443166.

[16] A. Ingle, 2020, "Crop Recommendation Dataset," V1, Retrivied 04/2022 from https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset