Nutrient Content Estimation for *Smart Farming* Using *Long Short Term Memory* (LSTM) Method

Iryna Vira Aulia Budiman
Department of Electrical Engineering
Padjadjaran University
Bandung, Indonesia
iryna20001@mail.unpad.ac.id

Lidia Marserlina
Department of Electrical Engineering
Padjadjaran University
Bandung, Indonesia
lidia20001@mail.unpad.ac.id

Khalisa Rofila Sundus
Department of Electrical Engineering
Padjadjaran University
Bandung, Indonesia
khalisa20002@mail.unpad.ac.id

Raafi Musa Fatahilah Kemas
Department of Electrical Engineering
Padjadjaran University
Bandung, Indonesia
raafi20001@mail.unpad.ac.id

Arjon Turnip
Department of Electrical Engineering
Padjadjaran University
Bandung, Indonesia
turnip@unpad.ac.id

Jonathan Given Hamonangan
Department of Electrical Engineering
Padjadjaran University
Bandung, Indonesia
jonathan20001@mail.unpad.ac.id

Abstract- Indonesia is known as an agricultural country with the majority of the population working as farmers. By 2023, 88.89% of the informal workforce will be in the agricultural sector. Conventional methods require direct monitoring of the land and the provision of nutrients that have not been optimized. This research focuses on farming using the Long-Short Term Memory (LSTM) method to provide optimal growth with appropriate nutrition. The research was conducted on dry and rainy beds with 6 polybags each outdoors, monitoring NPK parameters, soil pH, planting age, and soil moisture. Results showed the LSTM model successfully predicted nutrient requirements with 86.6% accuracy in dry conditions (24 ppm to 188 ppm) and 81% in rainy conditions (15 ppm to 53 ppm). Plant growth in rainy and dry beds with the LSTM method is more optimal compared to manual beds, as seen from the graphs of plant height, number of leaves, and number of flowers.

Keywords-Estimation, Nutrients, LSTM

I. INTRODUCTION

Indonesia is known as an agricultural country because most of the population's livelihoods are farmers. According to the Central Bureau of Statistics (BPS), by 2023, 88.89% of the informal workforce will be in the agricultural sector [1]. Agriculture is the backbone of the Indonesian economy with a very important role. As the population increases, the demand for basic needs increases, while strong economic growth can increase household income and consumer purchasing power. Changes in lifestyle and consumer preferences, such as the increasing awareness of the importance of healthy food, also affect demand.

Therefore, effective methods are needed in the agricultural sector to minimize inhibiting factors. Along with the development of technology, conventional farming systems have begun to shift to modern agriculture using sophisticated tools to simplify the work process [2-4]. This research focuses on eggplant farming, where farmers still use conventional systems in monitoring and controlling plants. Providing nutrients such as NPK often requires direct checking of soil conditions, which is less efficient especially in large areas [5-7]. Automation technology is needed that is able to reduce human labor, with the help of AI systems that are able to send data through networks without the help of computer devices and humans [8-10].

In smart farming, a crucial aspect that needs to be considered is the estimation of the nutrient levels of eggplant plants. Proper nutrient levels in the soil and growing environment are key factors in determining the quality and quantity of agricultural products [11][12]. This system is implemented on a solenoid valve integrated with AI to get an estimation of the right nutrient levels [13-15]. This design includes the creation of a system of sensors to measure environmental parameters, a data recording and display system, and an actuator activation system to maintain the environmental conditions of eggplant planting. The use of Long-Short Term Memory (LSTM) method can be used to predict eggplant plant nutrition based on factors such as NPK, soil pH, and soil moisture.

BPS data showed that eggplant consumption in Indonesia was 704,223 thousand tons in 2022, an increase of 4.12% from the previous year [16]. Eggplant responds quickly to environmental changes, making it a good research subject for climate change studies and crop adaptation to changing environmental conditions. This system allows real-time access to sensor data, helping to speed up and improve the accuracy of estimating nutrient levels in eggplant plants, providing convenience and comfort for farmers in monitoring and controlling farm conditions without having to be on site [17-20].

With this problem, it is necessary to estimate the nutritional needs of eggplant plants for optimal growth. Optimized eggplant plants have healthy roots, a large number of leaves, plant height, and healthy fruit. This optimal result can be seen by comparing beds that use system control with manual beds (manual nutrition once a month). One way to estimate the nutrient requirements of eggplant plants in the future is with the LSTM algorithm [21-22].

LSTM, a modification of Recurrent Neural Network (RNN), has a memory and three types of gates: forget gate, input gate, and output gate. LSTM is able to learn more than 1000 previous steps and overcome the vanishing gradient in RNN, allowing LSTM to learn and remember long-term information more effectively [23-28]. This research is expected to be a solution for farmers in providing nutrients to eggplant plants to be more efficient and optimize plant growth.

II. RESEARCH METHODS

A. Place of Research

In this study, researchers conducted a design in the laboratory of Building C, Center for Basic Science Services (PPBS) Padjadjaran University. The agricultural area consists of 12 plant polybags located in the rooftop area. The system is set up using automation technology using the Long-Short Term Memory (LSTM) method. There are plant objects that will be managed with the automation system. The object of this research is eggplant plants. There are 2 conditions, namely open conditions and closed conditions, each with 6 polybags per bed in the generative growth phase. The ideal condition of eggplant planting is influenced by several aspects, namely: NPK (Nitrogen, Phosphorus, Potassium), soil pH, planting age, and soil moisture.

The nutrient requirements of eggplant plants in the generative growth phase may vary depending on various factors, including soil type, farming practices, and plant variety. Based on the journals read, estimates for eggplant nutrient requirements in ppm (parts per million) or mg/L in season are Nitrogen (N) at 30 - 200 ppm, Phosphorus (P) at 15 - 80 ppm, Potassium (K) at 20 - 200 ppm, with an NPK ratio of 2:1:2. Nitrogen focuses on leaf and stem growth, phosphorus focuses on flowering and root development, while potassium supports healthy fruit development.

B. Research stages

The sensors used for data collection are *capacitive soil moisture* sensor, RS485 soil sensor, DHT22 air temperature and humidity sensor, pH sensor, and DS18B20 soil temperature sensor. The component used to make the actuator is a *solenoid valve*. The microcontroller used is Node MCU ESP8266 to control irrigation automatically. Other required software is multiplexer, *soil moisture*, panel box, wifi router, and power supply which will be connected to the server to be forwarded to the *website* for remote control.

Literature studies used to find references are taken from books, articles, journals, and other sources related to monitoring systems on *smart farming. The* data seen on the *website* will be processed data from sensors sent from the ESP-8266 nodeMCU to the MySQL *database* server. The author designs tools and system designs that will be developed in the monitoring system.

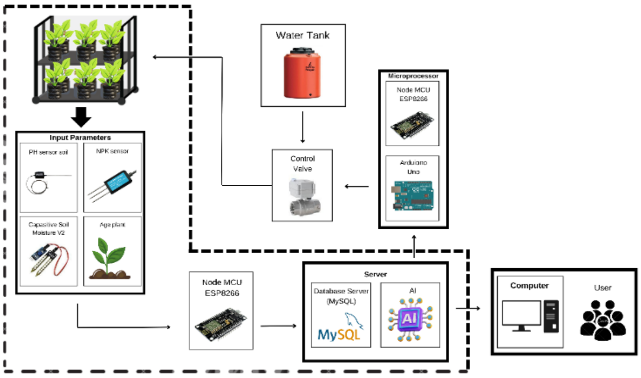


Fig. 1. *Smart Farming* System Design

The system design in the study can be seen in Fig. 1. The data collected by these sensors is then processed by the NodeMCU ESP8266, a microcontroller in charge of sending the information to the central server. The data received on the server will be stored in a MySQL database, and then analyzed by an artificial intelligence (AI)-based system to estimate future crop nutrient requirements. The results of this analysis will be displayed on the smart farming website for users to access.

After the nutrient estimation is completed, the system will send a command to the control valve that regulates the release of water and nutrients from the water tank. This valve will open the water and nutrient channels, which are then distributed to each polybag through a drip irrigation system according to the needs of each plant. All these processes are done automatically based on real-time data collected from sensors. The system is fully controlled by the NodeMCU ESP8266 operating at an input voltage of 5VDC, which is obtained from a 12VDC power supply. In addition, users can monitor the estimation results and plant conditions directly through a web interface connected to the server.

In the beds there are water channels and irrigation drip pipes from the pump, then there are *soil moisture* sensors, NPK, pH sensors in each bed equipped with *control valves* to be controlled. In open conditions, both disposal and water sources are more than one, sourced from rain and drip pipes or water droplets, so the evaporation output is also greater. Meanwhile, the closed condition will be covered by uv plastic, and the nutrient input is only from the drip pipe, so the evaporation output is also smaller. The manual treatment will be compared with both conditions, where the system must be able to produce better estimation results and can work in both conditions.

The soil treatment system in smart farming uses Capacitive Soil Moisture, NPK, pH sensors placed in the

Box Panel. In this system design, NodeMCU ESP8266, Power Supply, Multiplexer, and these sensors are used to monitor soil conditions in real-time. For water irrigation control, a panel box is designed that controls the valve. The system is designed to work automatically, where proportional valves are used to automatically adjust valve openings based on nutrient delivery commands and soil moisture conditions. The proportional valve is installed along the path of the field and connected to a water toren that delivers water to each bed using a drip pipe.

In this research, various sensors are used to collect soil environment data which are then connected to the ESP8266-based NodeMCU microcontroller for data processing and transmission. In Figure 2, the NPK (Nitrogen, Phosphorus, Potassium) sensor is connected to the NodeMCU through a MAX485E module that converts RS-485 signals into signals that can be read by the NodeMCU. The NodeMCU collects soil fertility data measured by the NPK sensor. Furthermore, the soil pH sensor is connected to the NodeMCU using an analog connection, allowing the measurement of soil acidity. A soil moisture sensor is also connected to the ESP8266 NodeMCU to measure soil moisture at several points.

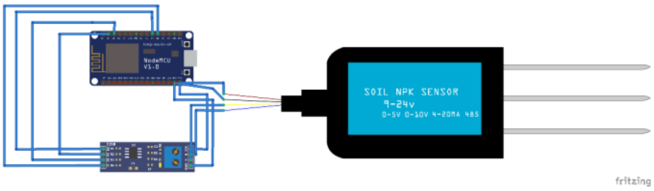


Fig. 2. Schematic of NPK Sensor Circuit

C. *Long Short Term Memory* (LSTM) algorithm

The method used in this *smart farming system* uses an *Artificial Intelligence* system. LSTM (*Long Short Term Memory*) has a chain layer of 3 gates, where the three gates have their respective functions and tasks in collecting, classifying, and processing data. LSTM has an *internal cell state* to store selected information from the previous unit. The LSTM structure as shown in Figure 3 consists of *input gate*, *forget gate*, and *output gate*. *Forget gate* is the gate in charge of forgetting some information that is irrelevant and no longer needed by the system. The *input gate is* tasked with entering information that has previously been selected first through the *forget gate*. Finally, the *output gate* can produce complete and actual data information.

The figure above illustrates the contents of the hidden layer of LSTM, namely memory cells. This memory cell in LSTM stores a value or state (*cell state*) for a short or long period of time. The gates in one LSTM memory cell are *input gate*, *forget gate*, and *output gate*. *The input gate* (i_t) has a role in taking the previous *output* and new *input*, as well as helping pass through the *sigmoid* layer. This gate returns a value of 0 or 1. The formula of (i_t) as follows.

$$i_t = \sigma(W_i S_{t-1} + W_i X_t)$$

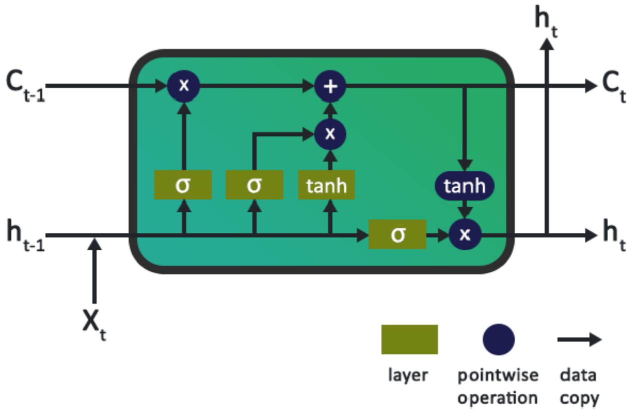


Fig. 3. LSTM architecture [24]

It is known that W_i is the weight of the *input gate*, S_{t-1} is the previous *state* or state at time t-1, X_t is the *input* at time t, and σ After that, the *input* gate value *is* multiplied by the *output of the* candidate layer ($\tilde{C}$).

$$\tilde{C} = tanh(W_c S_{t-1} + W_c X_t)$$
$$c_t = (i_t * \tilde{C}_t + f_t * c_{t-1})$$

It is known that $\tilde{C}$ is the *intermediate cell state*, W_c is the weight of the *cell state*, S_{t-1} is the previous *state* or the state at time t-1, and X_t is the *input* at time t. The previous *state is* multiplied by the *forget gate* and then added to the new candidate function allowed by the *output gate*. *Forget gate* (f_t) is a *sigmoid* layer that takes the *output* at time t-1 and the *input* at time t and combines them, and applies a *sigmoid* activation function, the *output of* this *gate* is 0 or 1. If $f_t = 0$ then the previous *state* will be forgotten, while if $f_t = 1$ the previous state does not change.

$$f_t = \sigma(W_f S_{t-1} + W_f X_t)$$

It is known that W_f is the weight of the *forget gate*, S_{t-1} is the previous *state* or state at time t-1, X_t is the *input* at time t, and σ This layer applies a hyperbolic tangent to the previous mixture of *inputs* and *outputs*. Returns a vector of candidates to be added to the *state. The output gate* (o_t) controls how much *state* passes to the *output* and works in the same way as the other *gates*, then finally produces a new *state cell* (h_t).

$$o_t = \sigma(W_o S_{t-1} + W_o X_t)$$
$$h_t = o_t * tanh(c_t)$$

It is known that W_o is the weight of the *output gate*, S_{t-1} is the previous *state* or state at time t-1, X_t is the *input* at time t, and σ *The* system will train the LSTM using data from manual measurements sourced from agricultural experts. At each time step in the input data sequence, the

LSTM receives the input from the previous time and the current input. The LSTM decides to what extent the old information in the memory cells should be retained or ignored with a *forget gate*. This is done by multiplying the previous elements of the memory cell by a value between 0 (forget completely) and 1 (retain completely) based on the current input and the previous input.

The LSTM then decides to what extent new information should be added to the memory cells with *input* gates. This is done by multiplying the current *input* by a value between 0 (ignore completely) and 1 (add completely) based on the current and previous inputs. The updated memory cells now contain the relevant information from the current and previous input data. Finally, the LSTM uses output gates to generate outputs from the updated memory cells, which can then be used as inputs for the next time step or as network *outputs*.

Data preparation, model building, and model evaluation are research activities that are conducted sequentially. Figure 4 illustrates the data preparation process consisting of data collection, preprocessing, representation, and data sorting (dataset sharing). Dataset preprocessing includes data cleaning and normalization. After the preprocessing stage is completed, the dataset is divided into two parts: training dataset and test dataset. The next stage is to build the system to achieve prediction results. This includes the training process and the prediction process. Since model building is time-consuming, the training process is performed separately from the prediction process.

It starts with model building and training of the LSTM Network, which results in an LSTM model stored in an HDF5 or .h5 file. The resulting model is then used in the prediction process by loading the model file. The last stage of the experiment is the process of denormalizing the test data to get the prediction result value and the evaluation value of the model performance. The benefit of separating the training and prediction processes is that when the prediction process is run without new data, there is no need to model the training data again; instead, the model can be loaded directly from the .h5 file, which speeds up the prediction process because there is no need to retrain the same data.

The estimation work in this research will involve the use of AI methods in the server to predict the nutritional needs of chili peppers. The first step is to collect relevant data, which can be sourced from sensors in the study and environmental parameters. The collected data then needs to be processed to remove noise and prepare it for analysis, including merging data from various sources into a dataset. Modeling will involve AI, namely the LSTM method. This model will then be trained using the previously processed data. After the model is applied, there is a need for testing and validation. The estimated results of the model will be compared with the actual data to measure how well the model works, as shown in Figure 3.

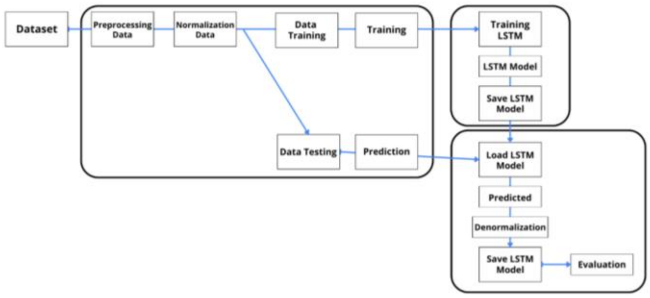


Fig. 4. Research Flow Diagram

The research also took a focus on measuring NPK (Nitrogen, Phosphorus, Potassium) directly in the soil, rather than in a nutrient reservoir. The measurement methods used include conventional measurement and sensor-based measurement, which will be a key component in the development of the estimation system. Thus, the results of this research are expected to improve agricultural efficiency and productivity and support the sustainable development of smart farming.

III. RESULTS AND DISCUSSION

A. Research Results

The research was carried out by designing hardware as discussed in the research method. Hardware design is done by making a system panel for data collection of capacitive soil moisture sensors and NPK sensors. The component used to distribute nutrients is a solenoid valve. The microcontroller used is Node MCU ESP8266 to control irrigation automatically. Other components used for hardware on the system panel are 5V power supply unit (PSU) and PCB. PCB circuit on this terminal block is used to connect the PSU and soil moisture sensor in the circuit. The header is also used to connect the ESP8266 to the circuit.

All of the above circuits are arranged in the system panel box as shown in Figure 5. This panel contains ESP8266 for data collection of NPK sensors and soil moisture sensors. Other required software are multiplexer, soil moisture sensor, NPK sensor, panel box, wifi router, and power supply. The devices will be connected to the server to be forwarded to the website for remote control. The data seen on the website will be processed data from sensors sent from the ESP-8266 nodeMCU to the MySQL database server. The use of a wifi router uses a wifi modem as the main link between the devices in the panel box and the internet network. With this WiFi router, data from the NPK sensor and soil moisture sensor connected to the ESP8266 can be sent in real-time to the server. The server will store the data into a mongoDb database, where the data can then be accessed and analyzed through a website.

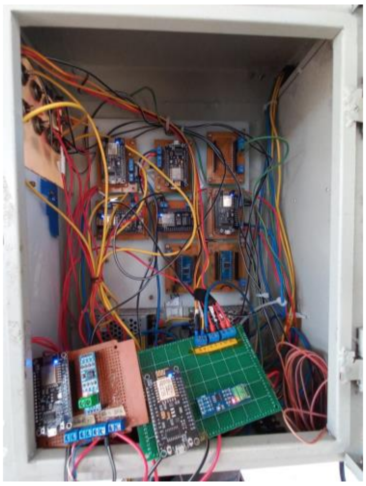


Fig. 5. System panel for sensor data capture

The sensors used in the research consist of 12 soil moisture sensors, 2 pH sensors, and 2 NPK sensors. The sensors are connected to the circuit using a 4-core cable. The distance between the polybag bed where the sensor is embedded and the system panel is 3 meters. In the first panel, there are 3 ESP8266 each connected to 6 soil moisture sensors and 2 ESP8266 connected to 2 NPK sensors, and 2 pH sensors. The use of 3 ESP8266 for soil moisture sensors shows 3 rows of beds with 3 different treatments (open, closed, manual). Meanwhile, 2 NPK sensors and 2 pH sensors were plugged into the open and closed bed treatments. This research process was carried out on the rooftop of the PPBS C building, Padjadjaran University as shown in Figure 6.

Fig. 6. Data Collection Process of smart farming research

The training process using the LSTM method is used to train and generate a model. This model is used to predict NPK, pH, and *soil moisture* which will be processed for plant nutrition needs in the form of a mixture of NPK and water in the future. In this study, *LSTM* from the *tensorflow.keras library was* used. The first stage is to prepare data consisting of data collection, preprocessing, representation, and data sorting (dataset division). The dataset is divided into 2 datasets to be performed on the LSTM model that has been created. These two datasets are an open treatment dataset and a closed treatment dataset, each consisting of input parameters *soil moisture* (SM), Nitrogen (N), Phosphorus (P), Potassium (K), pH, planting age (Age). Table 1 shows the input parameters used in each dataset.

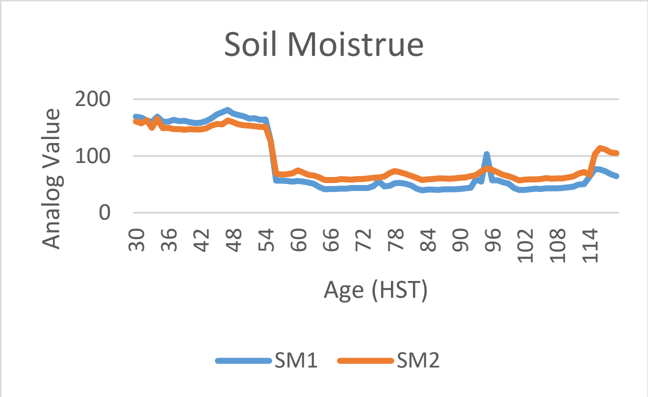
TABLE I. PARAMETERS USED AS INPUT

No.	Parameters	Unit
1	Soil moisture (SM)	Analog value (0-1023)
2	Nitrogen element (N)	Ppm (*parts per million*) or mg/L
3	Phosphorus element (P)	Ppm (*parts per million*) or mg/L
4	Potassium element (K)	Ppm (*parts per million*) or mg/L
5	pH	volt
5	Planting age (Age)	Day after planting (HST)

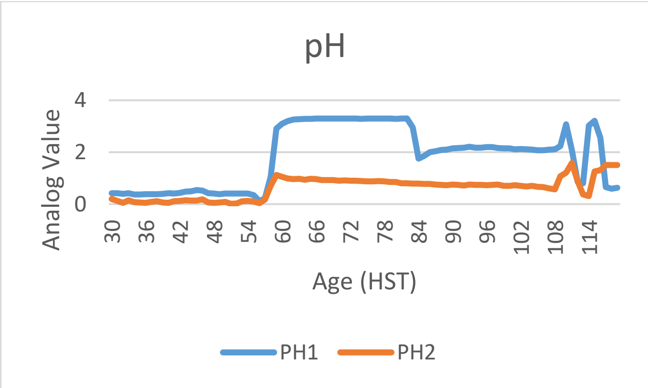
Dataset preprocessing includes data cleaning and normalization. After the preprocessing stage is completed, the dataset is divided into two parts: training dataset and test dataset. The next stage is to build the system to achieve prediction results. This includes the training process and the prediction process. Since model building takes a considerable amount of time, the training process is performed separately from the prediction process. The data is stored in a webserver database that can be accessed using the MongoDB program. The microcontroller in the data collection system circuit is set to be able to send data on soil moisture, pH, and NPK every day. Data collection to create the model was carried out for 3 months, every day starting from mid-January to April starting from Day After Planting (HST) 30 to HST 120.

Exponential Moving Average (EMA) filters with spans of 10 and 15 were used to tidy up the data, smoothing out insignificant fluctuations, thus facilitating the detection of important patterns. This filter was chosen for its ability to reduce noise without losing much historical information. The filtered data was organized in a table with parameters such as soil moisture, nitrogen, phosphorus, potassium, and pH. After filtering, the frequency of data capture was changed from 5 minutes to an average of 8 hours, resulting in 270 datapoints per sensor, as shown in Figure 7. The effect of this up and down graph also indicates the external influence of the natural evaporation process that occurs.

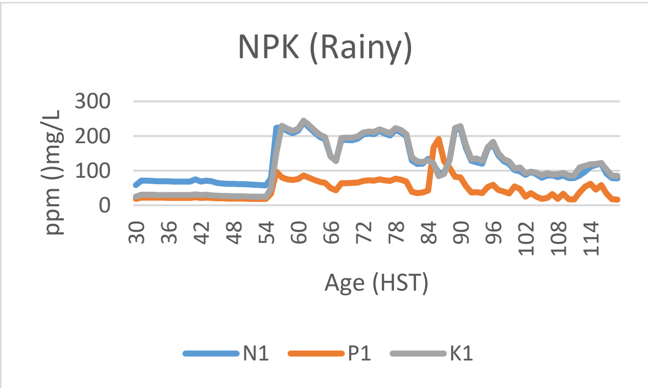
This is due to daytime or nighttime weather and when there is rain.



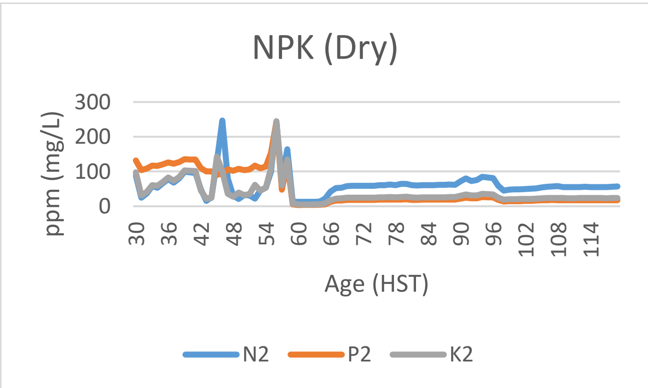
(a)



(b)



(c)



(d)

Fig. 7. Data of (a) Soil moisture (SM), (b) pH, and (c-d) NPK data taken from rainy and dry conditionss

In training modeling, there was an *overfitting* problem in the context of predictive models where the data was too specific to the training dataset. This happens when the model has a complex structure. Therefore, *tensorflow.keras.dropout is* used to overcome this problem. *Dropout* is an effective computational technique to reduce *overfitting* in neural networks. The way the *dropout layer* works is to assign a value of 0 to certain nodes in the training process by removing them from the network. Thus, the node has no influence on prediction and *backpropagation*. In this model, a *dropout rate of* 0.2 is used. Therefore, by using the *dropout* layer the neurons can no longer depend on the nodes from the previous layer and reduce the loss function quickly.

Furthermore, the use of *tuning* on the LSTM model is also used as a hyperparameter adjustment process to find the best performance configuration on the given data. Hyperparameters are parameters that are set before the training process begins and are not updated during training. Hyperparameters in this model include the number of units in the LSTM layer, *dropout rate*, *learning rate*, and the number of LSTM layers. This model uses Keras Tuner as a *library* that helps in finding the best hyperparameters for the Keras model. This tuning process using the *build_model* function defines the LSTM model architecture and defines the hyperparameter search space. Keras Tuner is initialized using *RandomSearch* by randomly selecting a hyperparameter combination from the predefined parameter space. Then, it searches for the best hyperparameters by running multiple trials ('max_trials') where each trial involves training a model with a different hyperparameter combination. The best model is then evaluated using test data to calculate RMSE, MSE, MAE, MAPE metrics. Plots of the predicted results are compared with the actual data which can be used to visualize the model performance.

It starts with model building and training of the LSTM Network, which results in an LSTM model stored in an HDF5 or .h5 file. The resulting model is then used in the prediction process by loading the model file. The last stage of the experiment is the process of denormalizing the test data to get the prediction result value and the evaluation value of the model performance. The benefit of separating the training and prediction processes is that when the prediction process is run without new data, there is no need to model the training data again; instead, the model can be loaded directly from the .h5 file, which speeds up the prediction process because there is no need to retrain the same data.

In Fig 7, after about 40 epochs, the *training loss* and *validation loss* values reach a stable point and begin to level off, indicating that the model has reached convergence. At this point, the model has learned the relevant patterns from the data and no longer improves performance significantly with additional epochs.

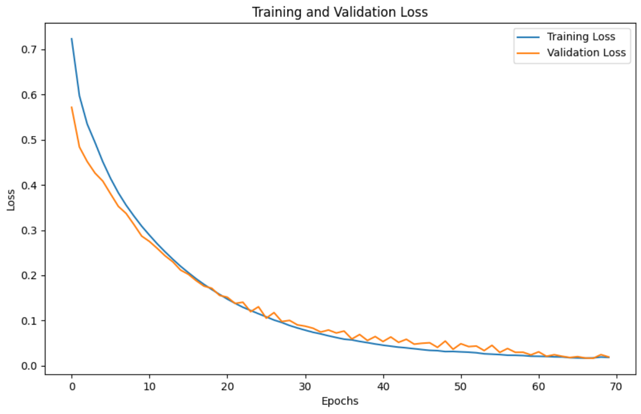
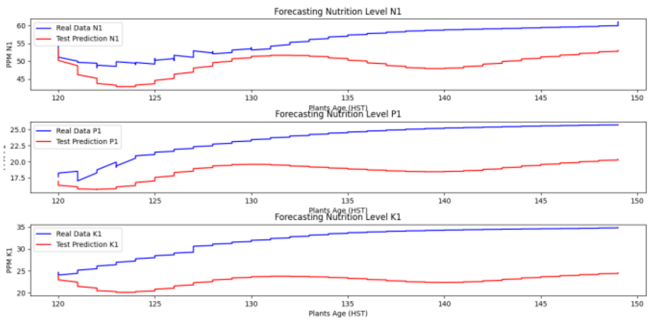


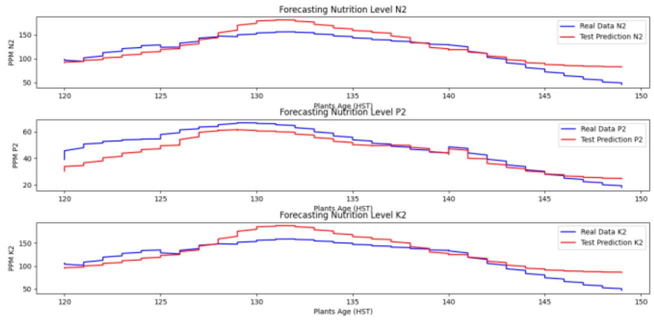
Fig. 8. Plot of *validation loss* in the open treatment and closed treatment models

Under rainy conditions, the graph shows a gradual decrease in accuracy from the first week to the fourth week. This decrease is due to the LSTM model relying more and more on previous prediction data, which potentially increases the error over time. In addition, more dynamic environmental conditions such as changes in soil moisture and nutrient concentrations occur more frequently during the rainy season. As for the dry conditions, the increase in accuracy in week 3 could be due to better data stability and lack of noise or significant fluctuations in the data in that week.

The performance evaluation of offline testing in April for predicting nutrient requirements in dry conditions showed excellent results with an RMSE value of 14.5, MAE of 11.1, and MAPE of 13.4%. In Figure 9, this shows that the model has a very high prediction accuracy of 86.6% in dry conditions. Meanwhile, for rainy conditions, the model shows a slightly lower performance with an RMSE value of 7.4, MAE of 6.7, and MAPE of 19%. The prediction accuracy was 81% under rainy conditions. This decrease in accuracy is due to larger variations and external factors affecting rainy conditions, such as irregular rainfall and larger variations in soil moisture, compared to the more stable dry conditions. The prediction results in Figure 4.10 show the estimated nutrients every 8 hours for rainy and dry conditions. This range of prediction values shows that the model is able to predict well, but the values differ due to variations in environmental conditions.



(a)

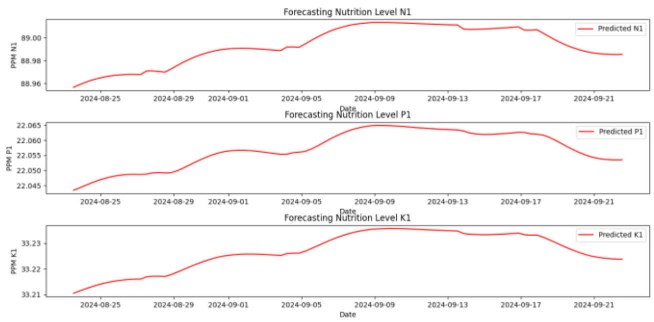


(b)

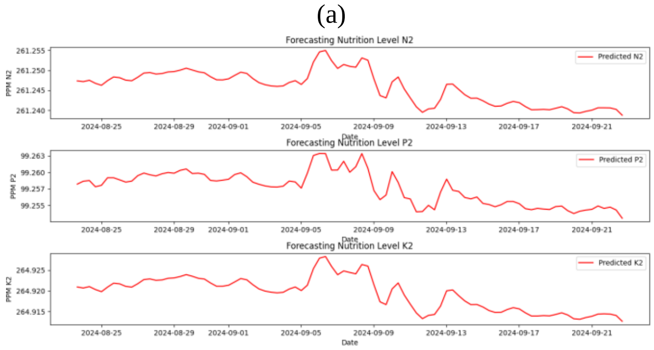
Fig. 9. Offline test result plot of forecasting comparison for the next 30 days in (a) rainy model; (b) dry model

Realtime testing was conducted to monitor crop nutrient requirements based on different weather conditions, namely in the rainy season (NPK1) and dry season (NPK2). The prediction process is carried out every 8 hours to project nutrient needs for the next month. The prediction results are then displayed on the smartfarmingunpad web platform, allowing users to monitor nutrient requirements continuously and make better decisions regarding nutrient management.

Data from MongoDB is used as input into the prediction model, where the latest data is retrieved periodically to ensure accurate and relevant prediction results. The prediction results are saved back into MongoDB and visualized in the form of a graph that displays the change in nutrient requirements (PPM) over time as shown in Figure 10. This graph helps in understanding the pattern of changes in nutrition and taking appropriate steps.



(a)



(b)

Fig. 10. Test graph of realtime forecasting of nutrient requirements in (a) rainy and (b) dry conditions

Figure 11 shows a view of the nutrient prediction page available on the https://sf2.ctailab.com/ platform. This page is designed to provide up-to-date information on crop nutrient requirements based on the model predictions that have been applied. The information presented includes the prediction time and prediction values for nitrogen, phosphorus and potassium nutrients.

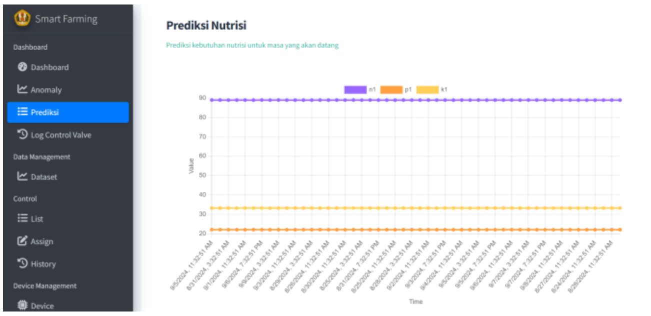


Fig. 11. Nutrient requirement prediction display on the webserver

Data collection on eggplant plants was taken in three beds, namely rain, dry, and manual beds. Growth was measured from 150th to 210th HST in June to July. Plants in each bed amounted to 6 plants each. Eggplant plants were monitored for growth every week, in terms of plant height, number of leaves, and number of flowers. Eggplant plants in the dry and rainy beds were controlled by the system by being given sufficient nutrients according to their needs every day. Meanwhile, eggplant plants in manual beds are given nutrients by direct watering using conventional methods by giving them once a month at 100 to 150 ppm by mixing nitrogen, phosphorus, and potassium nutrients in a ratio of 2:1:2. Data on plant height was taken weekly by measuring the distance from ground level to the top of the highest leaf using a ruler. Results showed that plant height growth in the dry bed reached 107.92%, higher than the manual bed. Growth in the rainy bed of 97.92% was also higher than the manual bed, but slightly lower than the dry bed. The graph can be seen in Figure 12.

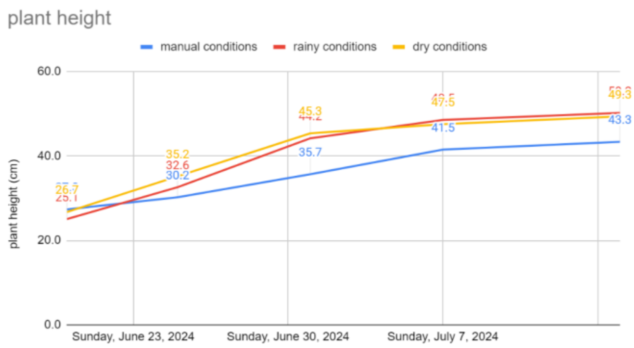


Fig. 12. Growth chart of plant height

The number of leaves is counted manually every week, paying attention to leaves that have opened and not budded. Counting starts from zero, although external factors such as broken or dried leaves may cause a decrease in the number. Based on the observation, the growth of leaf number in the dry bed reached 119.47%, higher than the manual bed. The growth in the rainy bed of 105.31% was also higher than the manual bed, but slightly lower than the dry bed. The graph can be seen in Fig 13.

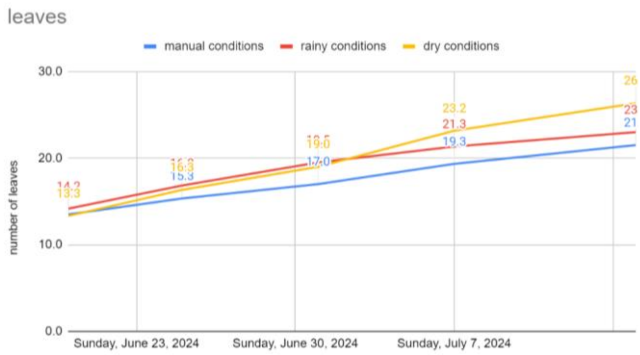


Fig. 13. Growth Chart of Number of Leaves

The number of flowers is also counted manually, paying attention to those that have opened and are colored. Counting starts from a value of zero, although external factors such as broken or dried flowers may cause a decrease in the number. Results showed that flower count growth in the dry bed reached 250%, much higher than the manual bed. The growth in the rainy bed of 150% was also higher than the manual bed, but lower than the dry bed. The graph can be seen in Fig 14.

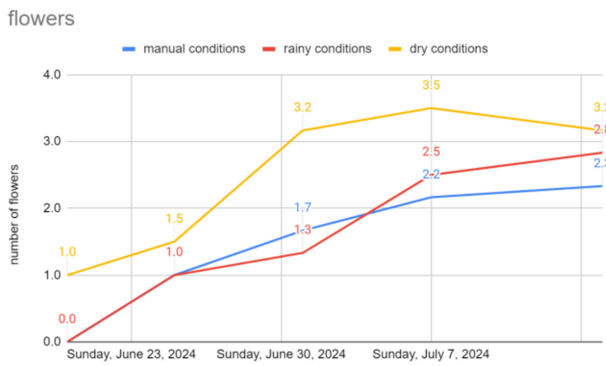


Fig. 14. Growth chart of flower number

Overall, the results showed that the growth of eggplant plants in dry and rainy beds was more optimal compared to manual beds. The dry bed showed the highest growth for all measured parameters, followed by the rain bed, and finally the manual bed. This indicates that automatically controlled nutrient application as needed every day is more effective compared to conventional methods that only provide nutrients once a month.

IV. CONCLUSIONS

Based on the design and training of the LSTM model to predict the nutritional needs of eggplant plants, it is concluded that this model is effective with a prediction accuracy of 86.6% for dry conditions and 81% for rainy conditions. Performance evaluation shows better RMSE and MAE values in rainy conditions compared to dry conditions. Plant growth using LSTM is more optimal than manual methods, and nutrient delivery is more targeted with a range of 15-53 ppm for rainy and 24-188 ppm for dry.

ACKNOWLEDGMENTS

This research was supported by the Department of Electrical Engineering, Padjadjaran University, Indonesia.

REFERENCES

[1] Budi, A. S., & Susilo, P. H. (2021). Implementation of SYM Method to Predict Rice Crop Yield. *JOUTICA*, *6*(1).

[2] Y. Mercy. B. (2022). Preparing for Indonesian Agricultural Transformation in The Society Era 5.0. Vol. 24, No. 32-36.

[3] G. Dankan, et al. (2021). Smart Agriculture and Smart Farming using IOT Techonology. Vol. 21.

[4] Putra, B. T. W. (2020). Socialization of Precision Farming for Monitoring Plantation and Horticultural Crops in Jember Regency. *Warta Pengabdian*, *14*(4), 231. https://doi.org/10.19184/wrtp.v14i4.16704

[5] M. Aditya Eka. (2022). The Potential of Smart Farming IoT Implementation for Coffee Farming in Indonesia. Vol. 2(2). No. 53-70.

[6] B. T. W. Putra, "Socialization of Precision Farming for Monitoring Plantation and Horticultural Crops in Jember Regency," *War. Pengabdi.*, vol. 14, no. 4, p. 231, 2020, doi: 10.19184/wrtp.v14i4.16704.

[7] S. Budi and P. H. Susilo, "Implementation of SYM Method to Predict Rice Crop Yield," *JOUTICA*, vol. 6, no. 1, 2021.

[8] T. H. Simanjuntak, W. F. Mahmudy, and Sutrisno Sutrisno, "Implementation of Modified K-Nearest Neighbor with K Value Automation in Soybean Plant Disease Classification," *Pengemb. Technol. Inf. and Comput. Science*, vol. 1, no. 2, pp. 75-79, 2017, [Online].

[9] Jhon S. Arie, "Implementation of KNN Algorithm in Prediction of Rainfall and Rice Plant Temperature," *Pros. Semin. Ilm. System. Information and Technol. Inf.*, vol. VIII, no. 1, pp. 11-20, 2019.

[10] O. Folorunso. (2023). "Exploring Machine Learning Models for Soil Nutrient Properties Prediction: A Systemic Review" *Perspectives*, vol. 18, no. 2, p. 91, 2020, doi: 10.21082/psp.v18n2.2019.91-103.

[11] Ambarish. G., & Lenka, S. K. (2016). Neural network pattern classification and weather dependent fuzzy logic model for irrigation control in WSN based precision agriculture. Procedia Computer Science, 78, 499-506.

[12] Kenneth. D. (2022). "Smart Farming Prediction Models for Precision Agriculture: A Comprehensive Survey". https://doi.org/10.1007/s10462-022-10266-6

[13] Central Bureau of Statistics (BPS) 2020. National Chili Production 2016- 2020.

[14] Betul, E. B., & Aksoy, U. T. (2011). Prediction of building energy needs in early stage of design by using ANFIS. Expert Systems with Applications, 38(5), 5352-5358.

[15] Bhattacharya, R., Shrivastava, S., Ojha, A. (2019). IoT Based Plant Monitoring System using Node MCU. Inre Revolution in Computer Science Communication Engineering. Vol 5 Issue: 7. ISSN: 2452-4248.

[16] P. Bamurigire, A. Vodacek, K. Jayavel and S. R. Ngonga, "A Decision-making Module for Fertilization and Irrigation Control System in Rice Farming Using Markov Chain Process and SARSA Algorithm," *WCSE 2021*, 2021.

[17] Components, "*Soil moisture* Sensor Module", April 16, 2021. [Online]. Avalible: https://components101.com/modules/soil-moisture-sensor-module. [accessed November 5, 2021].

[18] Dalenogare, L. S., Benitez, G. B., Ayala, N. F., & Frank, A. G. (2018). The expected contribution of Industry 4.0 technologies for industrial performance. International Journal of Production Economics, 204, 383-394.

[19] Perdana, D., Renaldi, L., Alinursafa, I. (2020). Performance Analysis Of *Soil moisture* Monitoring Based On Internet Of Things With Lora Communications. Journal of Southwest Jiaotong University. Vol. 55 No 5. ISSN: 0258-2724. DOI：10.35741/issn.0258-2724.55.5.31.

[20] Gracia, L., Parra, L., Jimenez, J. M., Lloret, J., Lorentz, P. (2020). IoT-Based Smart Irrigation Systems: An Overview on the Recent Trends on Sensors

[21] K. Shamsa, et al. (2022). Predicting Risk Through Artificial Intelligence Based on Machine Learning Algorithms: A Case of Pakistani Nonfinancial Firms. Vol. 22.

[22] A. Mahmuda, et al. (2021). Traffic Congestion Predistion Using Artificial Intelligence. Vol. 21.

[23] M. Mohammad, et al. (2022). Cardiovascular Events Prediction Using Artificial Intelligence Models and Heart Rate Variability. Vol. 22. No. 231-238.

[24] J. Brownlee. (2022). *Long Short-Term Memory Networks With Python: Develop Sequence Prediction Models With Deep Learning*, v1.0.

[25] J. Wu and Z. Wang. (2023). A Hybrid Model for Water Quality Prediction Based on an Artificial Neural Network, Wavelet Transform, and Long Short-Term Memory," *Water*, vol. 14, no. 4, p. 610.

[26] Y. Liu et al. (2023). "A Long Short-Term Memory-Based Model for Greenhouse Climate Prediction," *Int. J. Intell. Syst.*, vol. 36, no. 7, pp.

[27] [1] C. Munthali, R. Kinoshita, K. Onishi, A. Rakotondrafara, K. Mikami, M. Koike, M. Tani, J. Palta, and D. Aiuchi. (2024). "A Model Nutrition Control System in Potato Tissue Culture and Its Influence on Plant Elemental Composition," *Plants*, vol. 11, no. 20.

[28] A. F. Soomro, S. Tunio, M. I. Keerio, I. Rajper, Q. Chachar, and M. Y. Arain. (2024). "Effect of inorganic

NPK fertilizers under different proportions on growth, yield and juice quality of sugarcane (Saccharum officinarum L)," Pure Appl. Biol., vol. 3, no. 1, pp. 10-18.