**­­Detecting the Source of Leaf Wetness with the MLP-LSTM Hybridized Machine Learning Model**

Harshitha Paladugu 1, a) and Prasanna Venkatesan Theerthagiri2, b)

*Department of Computer Science and Engineering, GITAM Deemed to be University, Bengaluru-562163, India*

a)[*hpaladug@gitam.in*](mailto:hpaladug@gitam.in)

*b)[vprasann@gitam.edu](mailto:vprasann@gitam.edu)*

**Abstract.** The increasing importance of machine learning and IoT in agriculture has focused on gaps in classifying and detecting ambient parameters like leaf wetness sources using hybridization techniques. Our research focused on hybridization, a practice where the strength of individual approaches is combined to resemble a single model to accomplish complex tasks. Current methods use isolated machine learning models to detect leaf wetness sources, which are inadequate for complex predictions. If the model is not isolated but robust, and the dataset is simple, then the accuracy achieved by the model will be likely high. On the other, hand if the model is simple and the dataset is complex, the model’s accuracy will be likely low. In our current study, we employed hybridization by combining the strengths of Multilayer Perceptron and Long Short-Term Memory (MLP-LSTM). This study examined whether the hybridization elevated the accuracy score by employing a simple dataset. The results exhibit an accuracy of 99.731% for MLP-LSTM, which is greater than the accuracy of the isolated MLP model employed with numerical data [21], which is 92.86%. This technique is useful not only in agriculture but also useful in other domains where machine learning and deep learning algorithms are employed.

1. **INTRODUCTION**

Machine learning and IoT add technology to agriculture and lead to advancements such as precise monitoring of environmental conditions, systematic resource management, disease management, etc. The IoT-enabled systems furnished with advanced machine learning models overcome challenges like irrigation optimization, crop health surveillance, and environmental sensing. For example, IoT-enabled devices and wireless sensor networks are extensively used to gather environmental data, and machine learning models analyze the collected environmental data to identify actionable insights. Source identification is important for agriculture because it predicts the conditions in which plant diseases and fungal growth are commonly triggered by moisture on the leaves. Conventional approaches rely on sensor data, but environmental conditions make it difficult to distinguish the source of the wetness: dew, rain, or irrigation. This research puts forward a hybridized model that classifies leaf-wetness sources based on data acquired from leaf-wetness sensors, including moisture and temperature.

A Multilayer Perceptron model was used to predict the diseases related to fungus based on atmospheric and soil sensor data. Their study illustrated the productiveness of the neural network for the environmental data [3]. Motivated by the high accuracy of 98% achieved by their model, our model uses the MLP-LSTM model to detect time-dependent patterns of wetness, which improves the robustness in the detection of the source of leaf wetness. The ensemble methods and deep Learning are used to forecast the susceptibility of floods, exhibiting high accuracy through combining models like deep learning and random forest for sturdy environmental predictions [8]. This approach stimulated the use of LSTM to increase classification accuracy in recognizing leaf wetness sources. The isolated model using Multilayer Perceptron alone achieved 92.86% accuracy with numerical data [21]. By employing Long Short-Term Memory in our proposed model, the accuracy is increased to 99.731%.

The LSTM networks employed to forecast plant diseases through analyzing the temporal relationships in the sensor data [11] motivated to addition of LSTM layers to the multilayer perceptron model to focus on time dependent changes in the wetness of the leaf. The integration of LSTM in the Multilayer Perceptron allows the incorporation of patterns that are time-dependent and are important in agriculture. With the growth of IoT in agriculture and smart systems, the necessity of detailed classification for applications in real-time is crucial for strategies of disease management and optimizing irrigation.

1. **LITERATURE SURVEY**

The advancements in IoT and machine learning for agriculture mainly deal with environmental monitoring, plant health, and disease prediction. The review of methodologies and results focuses on IoT-based systems, machine learning algorithms, and hybrid techniques for precision agriculture. A Social IoT (SIoT) system integrates smartphone images with garden sensors to track plant health and support disease prediction, highlighting IoT's potential in sustainable agriculture through deep learning [1]. An IoT-based Wireless Sensor Network (WSN) combined with fuzzy logic and machine learning (KNN, MLP, and Random Forest) enables effective pest prediction and environmental data collection for precision agriculture [2]. A flexible substrate IoT sensor tracks the duration of leaf wetness. The light and sensitive sensor proves the potential of IoT in real-time tracking of environmental parameters, which focused on the identification of leaf wetness source [3].

An IoT-based smart irrigation system that uses RF energy harvesting for power supports sustainable, real-time irrigation control, which emphasizes the need for accurate environmental sensing [4]. MLP models predict fungal diseases based on soil and atmospheric sensor data with over 98% accuracy, thus demonstrating the effectiveness of neural networks in classifying plant health issues [5]. Genetic algorithms optimize CNN architectures for agricultural image classification tasks and have improved classification accuracy, thus showing the potential of combining optimization techniques with deep learning [6]. A study on climate effects on plant-pathogen interactions emphasizes the need for accurate monitoring of environmental parameters, which is in line with identifying leaf wetness origins for disease control [7]. Ensemble deep learning with Random Forest achieves high accuracy for flood prediction, validating the robustness of combining deep learning and ensemble techniques for agricultural data [8].

CNN-LSTM and CNN-GRU models perform well on sequential data; therefore, hybrid models can be used for complex predictions in agriculture [9]. The ensemble approaches like bagging, stacking, and boosting, improve the prediction robustness, which can be applied to select Random Forest, Bagged SVM, and XGBoost for the proposed model [10]. LSTM networks process time-series agricultural data with high accuracy, making the inclusion of LSTM layers within MLP models appropriate to extract time-related patterns [11]. Hybrid models outperform traditional models in predicting leaf wetness duration, aiding disease management systems [13]. Nonlinear regression models, such as XGBoost, are well suited for capturing complex relationships between relative humidity and leaf wetness duration [14]. Favorable temperatures combined with prolonged wetness duration increase disease risk, highlighting the importance of precise wetness source detection for disease control strategies [15].

Incorporating temperature data in leaf wetness models optimizes predictions and improves disease prediction accuracy [16]. CNN architectures like DenseNet-121 and ResNet-50 achieve high accuracy in plant disease detection, underscoring deep learning's capability for complex classification tasks. Their study justifies the use of MLP-LSTM in our model for better leaf wetness classification [17]. CNNs for Plant Disease Diagnosis: CNNs effectively diagnose a wide range of plant diseases, validating deep learning's application for precise classification [18]. Ensemble models combining decision trees, random forests, and naive Bayes improve accuracy and stability in environmental forecasting [19]. A voting ensemble achieves 100% accuracy in predictive modeling for improved generalization for robust predictions [20].

1. **METHODOLOGY**

The dataset collected through the leaf wetness sensor is cleaned using data preprocessing techniques. Cleaning the data is not only sufficient before training the model, but also the balance in the dataset needs to be checked. It is done using SMOTE analysis by adding synthetic data points to balance the dataset. The hybridized model MLP-LSTM is developed by adding an LSTM layer at the end of the MLP layers. Before training the developed model, splitting of the dataset is done to training, testing, and validation in the ratio 80:10:10. After the model got trained on the pre-processed dataset with required regularization techniques, the performance of MLP-LSTM model is visualized using the plots shown in the Fig. 13 and Fig. 14. The sequence of steps followed in the methodology of our research is represented in the Fig. 1.

Start

Data Collection

Data Preprocessing

SMOTE Analysis

Model Developent

Train, Validation, and Test Split

Hybridization

Performance Visualization

End

**FIGURE 1.** *Flowchart of the Methodology*

The block diagram of the complete process in detecting the leaf wetness source using the hybridization technique involving deep learning algorithms is shown in the Fig. 2. The leaf moisture sensor fixed in the field of the chili plantation collects the temperature and moisture. This data, collected for 5 months from May to September, at an interval of every 20 minutes, is stored in the cloud. The data is fetched from the cloud and cleaned using data preprocessing techniques mentioned in the section 3.2 and 3.3. The hybrid model using MLP-LSTM is developed. The sections 3.4, 3.5, and 3.6 explain about splitting of the dataset for training the developed model and its performance evaluation and visualization.

The leaf moisture sensor used in our study to record the leaf moisture and leaf temperature of the chili plant for 5 months, is shown in the Fig 3. The data is collected through the leaf moisture sensor from May to September, for every 20 minutes. The data collected at a frequency of every 20 minutes for those 5 months led to the formation of a dataset containing 6919 data points.

SMOTE Analysis

Data Preprocessing

MLP-LSTM Model

Model Development

Ratio of Preprocessed dataset split; 80:10:10

Train, Validation and Test Dataset

Training Accuracy and Validation Accuracy versus Epochs

Performance Visualization

Accuracy and

Classification Report

Performance Evaluation



Gateway

****

Chili crop

Leaf Moisture Sensor



Cloud

*Figure 2. Block Diagram*

**L**

*igure 2.Block Diagram*

**FIGURE 2***.Block Diagram*

****

**FIGURE 3.** *Leaf Moisture Sensor model LMS01-LB*

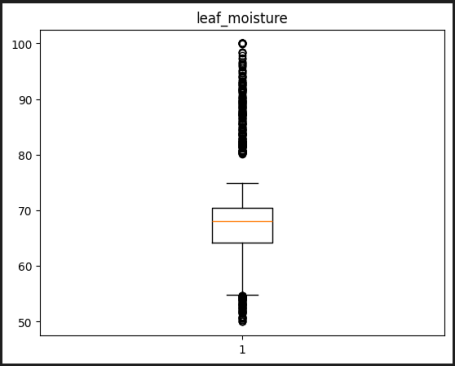
**3.1. Data Collection**

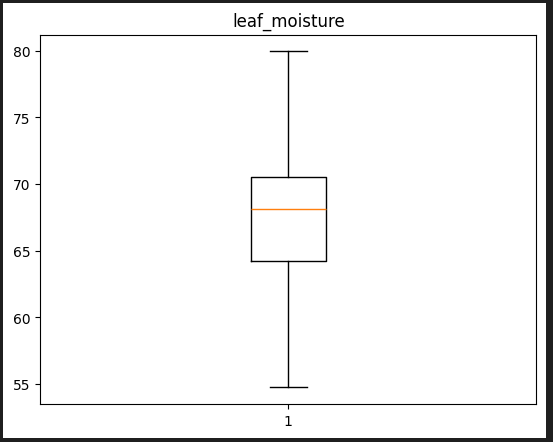
The chili crop is chosen for this study to detect leaf wetness sensors using hybridization. The dataset of size 6919 data points is collected through sensors established in the field. The dataset has the column names: Date, Time, Device Name, bat, leaf\_moisture, leaf\_temperature, rssi, snr, and spreading\_factor. The parameters other than leaf moisture and leaf temperature monitor the health status of the leaf wetness sensor, so the parameters that are not useful to detect leaf wetness are removed to avoid unnecessary data for further analysis. A column with the name ‘class’ is added to the dataset to classify the source of leaf wetness. To classify the leaf wetness source of chili plants using the parameters leaf\_moisture and leaf\_temperature, the conditions used are, If leaf\_temperature < 25 and 60 <= leaf\_moisture <= 75, then the source of leaf wetness is ‘dew.’, If 20 <= leaf\_temperature <= 30 and leaf\_moisture > 80, then the source of leaf wetness is ‘rainfall.’, If 25 <= leaf\_temperature <= 35 and 50 <= leaf\_moisture <= 70, the source of leaf wetness is ‘irrigation.’ The count of NULL values in each column is found to handle them in the data preprocessing step.

**3.2. Data Preprocessing in Exploratory Data Analysis**

The size of the dataset may decrease after data preprocessing depending on the presence of missing values and unnecessary data. In this study, after data preprocessing, the size of the dataset decreased from 6919 to 3304 data points. Due to the absence of NULL values in the dataset, the step to handle NULL values is skipped. The handling of NULL values in the dataset depends upon the type of data in each column. If the data type is categorical, the NULL values are handled using the statistical measure called MODE to replace the NULL values with the MODE value of the data in that respective column. This is called MODE imputation. If the data type is numerical, a statistical measure called MEAN or MEDIAN is used to replace the NULL values with the mean or median value of the data in that respective column. This is called MEAN imputation or MEDIAN imputation. The data points that do not lie within the range of the upper and lower boundary of the dataset in each column are called outliers.

The performance of a model gets affected by the outliers in a dataset. So, outliers in each column are handled using the interquartile range (IQR). Figure 4, the black spots in the form of small circles are outliers present in the leaf\_moisture parameter. These data points are clipped within the range of upper bound and lower bound values of the leaf\_moisture column.. The 25th percentile of the data falls below the first quartile, represented as Q1. Similarly, the 50th quartile is Q2, and the 75th quartile is Q3, which is represented in equation. Using the quartile values, interquartile range is calculated for leaf\_moisture column.The outliers present in the leaf\_moisture column are represented in the form of circles on one another as in the boxplot shown in Fig. 4. Figure 5 shows the boxplot of the leaf\_moisture parameter after removing outliers, that is, after handling outliers. The outliers are handled using the formulae mentioned for calculating the quartile Range (IQR). The distribution of each column is visualized to know if data is distributed normally over the mean of the data of each column. If the data in any column of the dataset is not normally distributed, then the data of that column needs to be handled to make the data normally distributed; this is possible through the skewness metric.





**FIGURE 4.***‘leaf\_moisture’ before handling outliers* **FIGURE 5.** *‘leaf\_moisture’ after* *handling outlier.*

Figure 7 plot shows the positive skewness in the leaf\_moisture parameter. The skewness values in the range of -0.5 and 0.5 show the distribution is normal or symmetric. Here the skewness is 1.469, which shows data in that column is highly skewed. Boxcox transformation is applied to handle the skewness. Boxcox is applied for strictly positive skewed data.

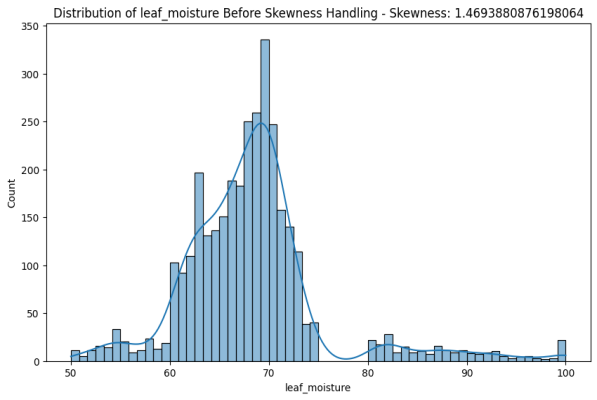
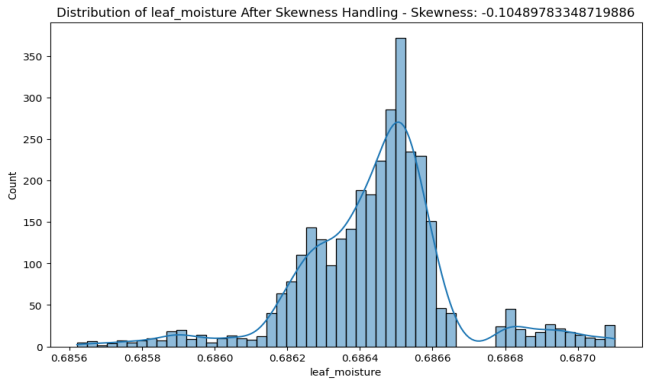
**FIGURE 7.***Distribution of leaf\_moisture column before handling skewness*

Figure 8 is obtained after applying boxcox transformation on the leaf\_moisture parameter to get the normal distribution. This plot has a skewness value of -0.104, which lies in the range of normal distribution, which is -0.5 to 0.5.

**FIGURE 8.** *Distribution of leaf\_moisture column after handling skewness*

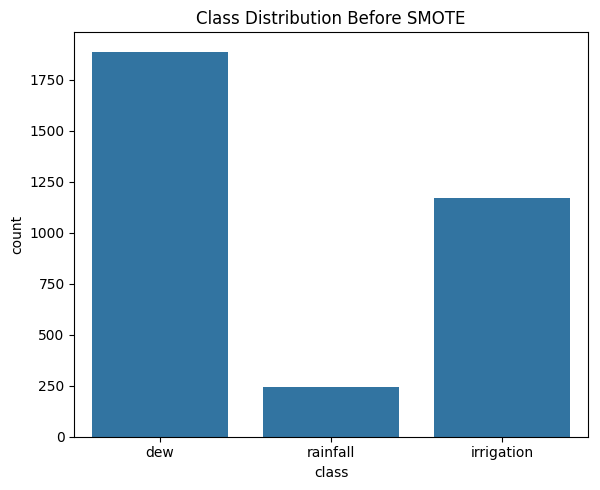
* 1. **Synthetic Minority Oversampling Technique Analysis**

**3.3 Synthetic Minority Oversampling Technique Analysis**

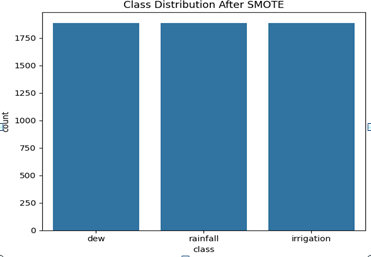
To balance the dataset Synthetic Minority Oversampling Technique Analysis is used, it is represented as SMOTE. Imbalance in the dataset is caused when the data points belonging to the respective class labels are not nearly equal. To convert the imbalanced dataset to a balanced one, SMOTE analysis is performed by generating synthetic samples. These synthetic samples can be generated using three ways,

1. Standard SMOTE: generates synthetic samples to make some data points of majority and minority classes equal.
2. Minority SMOTE: generates synthetic samples only to balance underrepresented classes.
3. Custom/Dictionary-based SMOTE: generates synthetic samples based on a user-specified sample size of each class.

To avoid overfitting in a model, Standard SMOTE is not recommended. In this study, custom-based SMOTE is used to balance the dataset to avoid overfitting of machine learning models. The distribution of data points to each class before SMOTE analysis is represented in Fig. 9, 1888 data points belong to the class dew, 1171 data points belong to the class irrigation, and 245 data points belong to the class rainfall before SMOTE analysis. The distribution of data points to each class after SMOTE analysis is, 1307 data points belong to class rainfall, 1267 data points belong to class irrigation, and 1146 data points belong to class dew.



**FIGURE 9.***Class Distribution before SMOTE analysis*



**FIGURE 10.** *Class distribution after SMOTE analysis*

**3.4 Train, Validation, and Test split**

After splitting the dataset in the ratio 80:10:10, resultant training, validation, and testing datasets were saved in the form of pickle files, which were used by the developed model for learning, validation, and evaluation. The pickle files thus obtained were given as input to the MLP-LSTM (Multilayer Perceptron-Long Short Term) for training, validating, and evaluating the model

* 1. **Hybridization**

The hybridization technique is employed in our study using Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM) model. The model architecture is shown in the Fig. 11. The input features leaf\_temperature and leaf\_moisture are given as input to the MLP-LSTM model. X1 and X2 represent the input features given as input to the hybridized model MLP-LSTM. The count of hidden layers and count of neurons in each layer is mentioned in the Fig. 11. The dotted lines in the MLP-LSTM architecture represent dropout layer. Reshape layer emits the shape expected by the LSTM

Dew

Irrigation

Hidden Layer 2

[32 neurons]

Output Layer

[3 neurons]

Reshape

LSTM

Rainfall

X1

X2

Hidden Layer 1

[64 neurons]

**FIGURE 11.***MLP-LSTM Architecture*

. The patterns or sequential dependencies are captured by the Long Short Term Memory. The activation function used by hidden layer is Rectified Linear Unit and output layer is softmax. The output layer emits the probabilities of three classes as output.

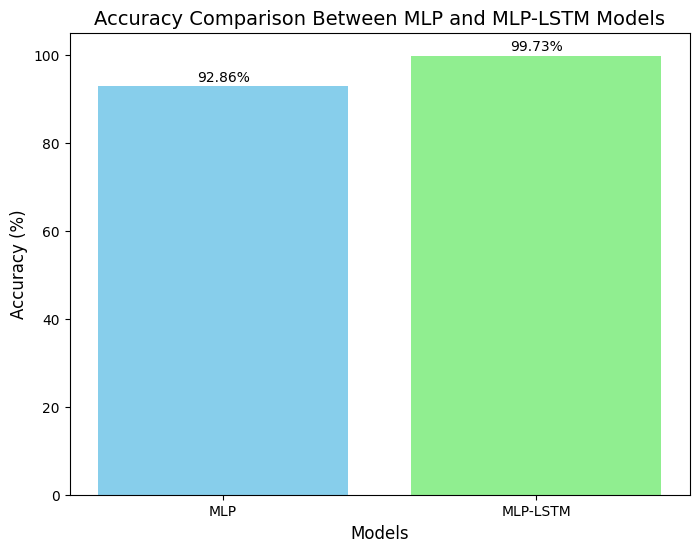
Input Layer

The processing of the data takes place at the hidden layers. The final layer which gives the output is called the output layer. Each neuron in the MLP has an activation function except the neurons in the input layer. Long Short-Term Memory is one of the types of Recurrent Neural Network. It is good at handling time series data and avoiding vanishing gradient problems. The dataset used in this study is collected over a time period of 5 months, that is from May to September. As LSTM is good at handling time series data, adding LSTM to the MLP enhances the performance of the model. L2 regularization and Dropout are two effective techniques to avoid overfitting, and mostly used when the dataset is imbalanced or of limited size. These techniques ensure not only good performance on the training dataset but also on validation and testing datasets that are unseen. Dropout prevents the model from memorizing the patterns while training the model by introducing randomness during the training phase. This nature of the dropout technique helps to overcome overfitting that occurs due to a high number of neurons per layer, and number of layers.

‘Dropout’ regularization technique is implemented by adding dropout layers after each hidden layer to avoid overfitting. It works by deactivating a random fraction of neurons. The output obtained from the second dense layer is given as input to the LSTM model after reshaping. The reshaped dimension is given as input to the LSTM layer. The L2 regularization technique with 0.01 regularization strength is applied in the LSTM layer to avoid overfitting by penalizing the large weights. The output obtained from the LSTM layer is given as input to the output layer containing 3 neurons, and softmax activation function, used for multiclass classification. The result given by the output layer is the probabilities of 3 classes.

**3.6 Performance Visualization**

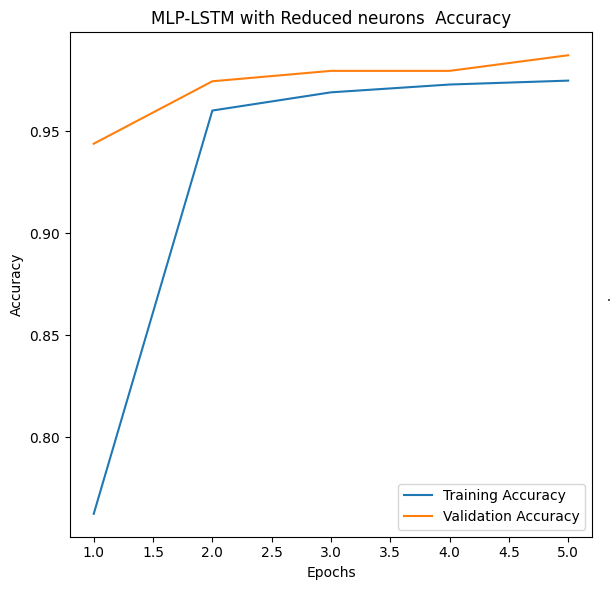
The proposed model’s performance is visualized using the plots represented in the Fig. 13 and Fig.14. These plots help us to know how the performance of the model is when predicting unseen data through the accuracy metric and loss metric. The curves in the plot tell whether the model is overfitted or not. Figure 13 shows the hybridized machine learning model with the MLP-LSTM architecture achieved higher accuracy which is 99.731% compared to the isolated MLP model which is 92.86%.



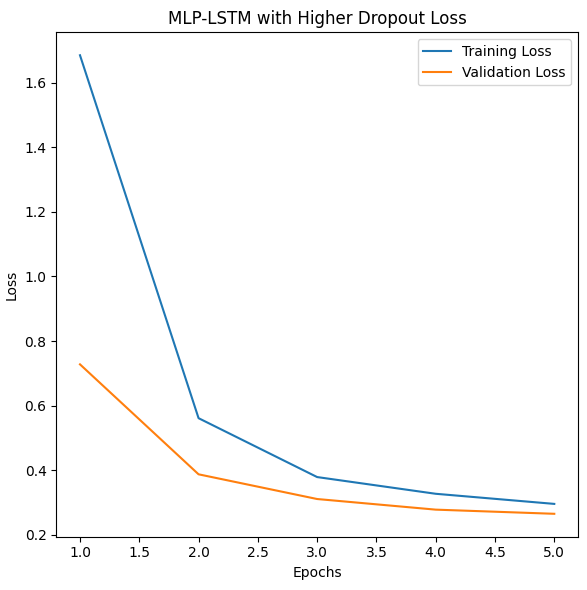
**FIGURE 12.***Accuracy comparison between isolated MLP model and hybridized MLP-LSTM Model*

1. **RESULT**

Figure 13 shows the training accuracy and validation accuracy versus epochs. Regularization techniques such as L2, Dropout, and Early stopping are employed during the training of the model to prevent overfitting by constraining the model’s capacity. This led to a slightly higher accuracy of the validation dataset when compared to the training dataset accuracy. Due to this reason, the plot in the Fig. 13 shows the training accuracy curve below the validation accuracy curve. The non-declining behavior of validation accuracy and inclining behavior of training accuracy after a few numbers of epochs show the absence of overfitting in the model and show good capability of generalization. Figure 14 show that training loss is greater than validation loss, which shows a well-regularized model that is not overfitting. The constant lowering of validation loss without increasing at any epoch shows good generalization. The higher training loss compared to validation loss exhibits the model is memorizing training data or is not over-optimized. The constraints introduced by the regularization techniques such as L2 regularization and Dropout in the training process penalize the few parameter updates, which results in increasing training loss to prevent the model from being overfitted.

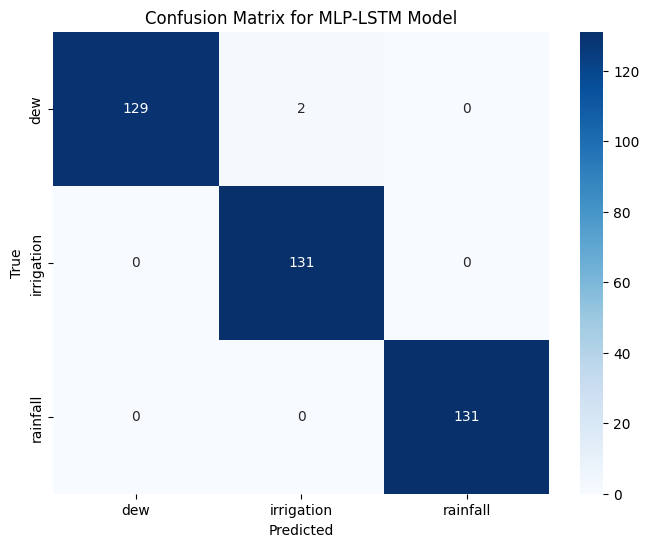


**FIGURE 13***.The Training Accuracy and Validation Accuracy versus Epochs*



**FIGURE 14.***The Training Loss and Validation Loss versus Epochs*

The confusion matrix of MLP-LSTM model is represented in the Fig. 15. The elements present in the diagonal indicate the samples that were correctly predicted. 129 samples were accurately classified as dew, similarly 131 samples as irrigation, and 131 samples as rainfall. 2 of the samples were misclassified as irrigation for dew. The diagonal elements of the confusion matrix showed the model did not favor any individual class. This shows the model is not overfitted.fig

****

**FIGURE 15.***Confusion Matrix*

From table 1, the precision values of each class are close to 1.00 shows that instances of each class are correctly predicted. The recall value of 0.98 for rainfall shows the existence of slight variation in prediction, whereas the other classes dew and irrigation is 1.00. The F1-score value of 0.99 or 1.00 shows an excellent balance between precision and recall. The support value shows that each class has 131 samples, resembling a balanced dataset.

**TABLE 1***. Classification Report of Hybridized MLP-LSTM Model*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class** | **Precision** | **Recall** | **F1-score** | **Support** |
| Rainfall (0) | 1.00 | 0.98 | 0.99 | 131 |
| Dew (1) | 0.98 | 1.00 | 0.99 | 131 |
| Irrigation (2) | 1.00 | 1.00 | 1.00 | 131 |

1. **CONCLUSION**

This work showed hybridization technique increased performance of the model, compared to isolated model. Hybridization is achieved by combining two or more models as a single model, this technique increases the strength of the resultant hybrid model as it merges the strengths of two models MLP (Multilayer Perceptron) and LSTM (Long Short Term Memory). The resultant model MLP-LSTM scored 99.731% accuracy which is higher than the accuracy scored by isolated MLP to find the source of leaf wetness using the numerical data [21], that is 92.86% accuracy. To show the model is not overfitted, the plots between training accuracy and validation accuracy versus the number of epochs, and training loss and validation loss versus the number of epochs are plotted; through these plots, it is confirmed that the MLP-LSTM model is not overfitted. It is concluded that the hybridization technique enhanced the performance of classification models, so this technique can, not only be used in the domain of agriculture but also be used in other domains to score high-level accuracy.

1. **ACKNOWLEDGEMENTS**

We are grateful to the GITAM School of Technology, Bangalore, for the infrastructure and support for carrying out this research. We are thankful to our mentors and professors for guiding and encouraging us in this study.

1. **FUTURE WORK**

The hybridized machine learning model increased the accuracy of classification models when compared to isolated machine learning models. This research can be extended further by focusing on other environmental conditions to detect the primary source of leaf wetness. Advanced deep learning architectures like transformers or attention mechanisms can be used to improve the recognition of temporal patterns in time series data. For real-time implementation, IoT devices and edge computing could enhance the deployment scalability and efficiency.

1. **REFERENCE**
2. G. Delnevo, et al., "A deep learning and social IoT approach for plants disease prediction toward a sustainable agriculture," IEEE Internet of Things Journal 9, 7243–7250 (2021).
3. R. P. Sharma, et al., "IoT-enabled IEEE 802.15.4 WSN monitoring infrastructure-driven fuzzy-logic-based crop pest prediction," IEEE Internet of Things Journal 9, 3037–3045 (2021).
4. K. S. Patle, et al., "IoT enabled, leaf wetness sensor on the flexible substrates for in-situ plant disease management," IEEE Sensors Journal 21, 19481–19491 (2021).
5. A.D. Boursianis, et al., "Smart irrigation system for precision agriculture—The AREThOU5A IoT platform," IEEE Sensors Journal 21, 17539–17547 (2020).
6. M. Kumar, A. Kumar, and V. S. Palaparthy, "Soil sensors-based prediction system for plant diseases using exploratory data analysis and machine learning," IEEE Sensors Journal 21, 17455–17468 (2020).
7. Y. Sun, et al., "Automatically designing CNN architectures using the genetic algorithm for image classification," IEEE Transactions on Cybernetics 50, 3840–3854 (2020).
8. C. Velásquez, C. D. M. Castroverde, and S. Y. He, "Plant–pathogen warfare under changing climate conditions," Current Biology 28, R619–R634 (2018).
9. Y. Li and H. Hong, "Modelling flood susceptibility based on deep learning coupling with ensemble learning models," Journal of Environmental Management 325, 116450 (2023).
10. V. B. Semwal, A. Gupta, and P. Lalwani, "An optimized hybrid deep learning model using ensemble learning approach for human walking activities recognition," The Journal of Supercomputing 77, 12256–12279 (2021).
11. N. Rane, S. P. Choudhary, and J. Rane, "Ensemble deep learning and machine learning: applications, opportunities, challenges, and future directions," Studies in Medical and Health Sciences 1, 18–41 (2024).
12. K. S. Patle, et al., "Field evaluation of smart sensor system for plant disease prediction using LSTM network," IEEE Sensors Journal 22, 3715–3725 (2021).
13. M. Solís and V. Rojas-Herrera, "Approaches for the prediction of leaf wetness duration with machine learning," Biomimetics 6, 29 (2021).
14. B. Gama, et al., "Evaluation of a multi-model approach to estimate leaf wetness duration: an essential input for disease alert systems," Theoretical and Applied Climatology 149, 83–99 (2022).
15. K. Alsafadi, et al., "Prediction of daily leaf wetness duration using multi-step machine learning," Computers and Electronics in Agriculture 224, 109131 (2024).
16. L. L. Marcuzzo and D. Füchter, "Influence of temperature and daily leaf wetness duration on the severity of bacterial leaf blight of garlic," Summa Phytopathologica 47, 180–182 (2021).
17. S. Zito, et al., "Optimization of a leaf wetness duration model," Agricultural and Forest Meteorology 291, 108087 (2020).
18. J. Eunice, et al., "Deep learning-based leaf disease detection in crops using images for agricultural applications," Agronomy 12, 2395 (2022).
19. M. V. Mallikarjuna Nandi, "Plant leaf disease detection using convolutional neural network," International Journal of Science and Research (IJSR) 13, 1545–1548 (2024).
20. Y. Resti, et al., "Ensemble of Naive Bayes, Decision Tree, and Random Forest to predict air quality," IAES International Journal of Artificial Intelligence 13, 3039–3051 (2024).
21. N. Buslim, "Ensemble learning techniques to improve the accuracy of predictive model performance in the scholarship selection process," Journal of Applied Data Sciences 4, 264–275 (2023).
22. R. Saini, P. Garg, N. K. Chaudhary, M. V. Joshi, V. S. Palaparthy, and A. Kumar, "Identifying the source of water on plant using the leaf wetness sensor and via deep learning-based ensemble method," IEEE Sensors Journal 24, 7009–7017 (2024).