Received: 25 February 2023 | Accepted: 14 February 2024

DOI: 10.1002/spe.3324

RESEARCH ARTICLE

WILEY

Fly: Femtolet-based edge-cloud framework for crop yield prediction using bidirectional long short-term memory

Tanushree Dey[1] | **Somnath Bera**[1] | **Bachchu Paul**[2] | **Debashis De**[1] | **Anwesha Mukherjee**[3] | **Rajkumar Buyya**[4]

[1]Centre of Mobile Cloud Computing, Department of Computer Science & Engineering, Maulana Abul Kalam Azad University of Technology, West Bengal, Nadia, West Bengal, India

[2]Department of Computer Science, Vidyasagar University, Midnapore, West Bengal, India

[3]Department of Computer Science, Mahishadal Raj College, Purba Medinipur, West Bengal, India

[4]Cloud Computing and Distributed Systems (CLOUDS) Lab, School of Computing and Information Systems, The University of Melbourne, Australia

Correspondence
Anwesha Mukherjee, Department of Computer Science, Mahishadal Raj College, Mahishadal, Purba Medinipur, West Bengal-721628, India.
Email: anweshamukherjee2011@gmail.com

Abstract

Crop yield prediction is a crucial area in agriculture that has a large impact on the economy of a country. This article proposes a crop yield prediction framework based on Internet of Things and edge computing. We have used a fifth generation network device referred to as femtolet as the edge device. The femtolet is a small cell base station that has high storage and high processing ability. The sensor nodes collect the soil and environmental data, and then the collected data is sent to the femtolet through the microcontrollers. The femtolet retrieves the weather-related data from the cloud, and then processes the sensor data and weather-related data using Bi-LSTM. The femtolet after processing the data sends the generated results to the cloud. The user can access the results from the cloud to predict the suitable crop for his/her land. This is observed that the suggested framework provides better accuracy, precision, recall, and F1-score compared to the state-of-the-art crop yield prediction frameworks. This is also demonstrated that the use of femtolet reduces the latency by ∼25% than the conventional edge-cloud framework.

KEYWORDS

Bi-LSTM, crop yield prediction, Femtolet, internet of agricultural things, low latency services

1 | INTRODUCTION

Crop yield prediction of land is one of the promising research aspects that can have a vital impact on the agricultural growth. The use of the Internet of Things (IoT) in smart agriculture has gained high attention of the researchers. The IoT is a collection of various devices, which interact with one another and their internal and external states by communicating, sensing, and acting on those states using embedded technology. In contrast to the traditional wireless sensor networks, IoT and cloud computing make it possible to apply data analytics to the massive amount of data generated by sensors. Smart agriculture system implements full automation by harnessing popular technologies such as IoT, artificial intelligence (AI), and big data to enhance the quality of life for farmers.[1] To store the large amount of sensor data, cloud computing

Abbreviations: 5G, fifth generation; ANN, artificial neural network; Bi-LSTM, Bidirectional Long Short-Term Memory; CNN, convolutional neural network; DBN, deep belief network; DT, Decision tree; EVI, Enhanced Vegetation Index; Fly, *Femtolet*-based edge-cloud framework for crop yield prediction; GRU, Gated recurrent unit; IoAT, Internet of Agricultural Things; IoT, Internet of Things; KNN, K-Nearest Neighbour; LSTM, Long Short-Term Memory; MLP, multilayer perceptron; MLR, multiple linear regression; ms, Milliseconds; RF, Random forest; SCceNB, small cell cloud enhanced eNodeB; s, Seconds; SVM, support vector machine; XGBoost, extreme gradient boosting.

is generally used. However, the use of the remote cloud for analyzing and storing the entire data enhances the service provisioning latency, network traffic, and computation overhead on the cloud.[2] Moreover, the entire data storage over the cloud raises a major concern in data security. Edge and fog computing can provide solutions to reduce the propagation and data transmission latencies.[2] Small cells with high computation ability such as small cell cloud enhanced eNodeB (SCceNB), femtolet can also be used for data processing.[3] Femtolet is a fifth generation (5G) network device that provides communication and computation services simultaneously.[3] The architecture of femtolet was proposed along with its working principle in Reference 3. Femtolet is an integration of femtocell and cloudlet.[3] Femtocell is a small cell base station, and cloudlet is a computer or cluster or computers that contains cache copies of data stored inside the cloud and works as an agent between the end user and cloud.[3] Femtolet contains the components for femtocell to provide communication service like femtocell, as well as has the storage and processing ability like cloudlet.[3] Thus, femtolet is a small cell base station with storage and high processing ability.[3] The femtolet was used for smart home monitoring.[4] The femtolet was used for retail centers.[5] As discussed in the existing literature, femtolet can process the sensor data, user data registered under the femtolet, and store the data.[3–6] The existing works presented that the use of femtolet reduces the latency than only cloud-based systems.[3–6] Though the use of femtolet was discussed for various IoT applications such as home monitoring, smart retail and so forth, in the existing literature,[4,5] its use in agriculture has not been focused. In our proposed framework, we use femtolet for providing low latency data processing for crop yield prediction.

Modernization in the agricultural field can be due to the integration of traditional practices with leading technologies such as IoT, cloud computing, machine learning and so forth. Agricultural yields are influenced by organic, economic, and seasonal factors. The environment, climate, soil, water, temperature, rainfall, vegetative index, nutrients and so forth. have a significant impact on crop production.[7,8] For the economic benefits of the farmers, they need to be aware of the suitable crops for their land for a given season as well as if any special safety measure is required based on the soil and environmental conditions. Traditionally, data gathering in agricultural fields heavily relies on human labor, and can only produce a small number of data points with poor resolution and accuracy. In this case, IoT emerges as a crucial technology along with cloud, edge, fog computing, and machine learning for continuous monitoring and control. The use of IoT in agriculture has introduced Internet of Agricultural Things (IoAT).[9]

AI is a key component of smart computing, that includes deep learning to simulate the structure and function of the human brain.[10] The neural network serves as the foundation for deep learning and is processed using hidden layers for learning enhancement. Deep learning expands its capabilities in the field of harvesting.[11] Bidirectional Long Short-Term Memory is one of the most powerful deep learning methods for data analysis. Based on the Long Short-Term Memory (LSTM) network,[12] Bidirectional LSTM (Bi-LSTM) was introduced.[13] Our proposed approach combines IoT, 5G network, and Bi-LSTM, so that crop yield prediction can be performed with high accuracy and at low latency. In our proposed framework we use Bi-LSTM for data analysis because Bi-LSTM model records and examines sequential data, which makes it ideal for simulating the intricate temporal dynamics involved in crop yield production. The dependencies and patterns in historical crop data can be efficiently captured by Bi-LSTM model by taking into account both past and future information. In order to make more accurate projections, the model is able to incorporate non-linear relationships between weather patterns, soil characteristics, and historical yield data. The Bi-LSTM model is briefly discussed as follows.

Bi-LSTM model for data analysis: Bi-LSTM models are often more effective when dealing with contextual information because their output at a time is dependent on both previous and next segments, as opposed to unidirectional LSTM models. LSTM can remember information for long time periods that is, can learn long term dependencies. Cell state is the key element in LSTM. LSTM has formed in a chain like structure with four neural networks called gates and different memory blocks called cells. One shortcoming of standard LSTM is the ignoring of the future context, which turns out to be crucial for classifying long-period occurrences. A Bi-LSTM model might process information in both directions, forward and backward, using two separate LSTM layers to comprehend past and future data.[13,14] It includes past and future data by performing data processing in both directions with two distinct hidden layers, which are subsequently fed to the same output layers.

LSTM consists of a sigmoid function whose output numbers are between zero and one. Value one means it will pass everything which it has at the input while a value of zero means nothing to through that has at the input. First, in LSTM, we have to choose what information to remember and what not to remember. This decision is made by a sigmoid function applied with the combined value of input and hidden state of the previous time step. This is the formation of forget gate, and the equation of the forget gate is mentioned in Equation (1) as follows:

$$f_t = \sigma\left(w_t\, h_{t-1} + w_t\, x_t + b_f\right), \tag{1}$$

where w_t is weight at time t, h_{t-1} defines hidden state at time (t-1), x_t defines current input at time t, and b_f denotes bias of the forget gate.

The subsequent step is to determine what new data will be added to the cell state. This is done in two parts. Firstly, an input gate (i_t) in a sigmoid layer decides which value to change. Secondly, a $\tan h$ function makes a vector of new candidate gate value c_t. The state will then be updated by adding these two values. The equations are as follows:

$$i_t = \sigma\ (w_i\ h_{t-1} + w_i\ x_t + b_i), \tag{2}$$

$$c_t = \tan h(w_c\ h_{t-1} + w_c\ x_t + b_c), \tag{3}$$

where the weight of the input gate is denoted as w_i, b_i denotes the bias of the forget gate, w_c is mentioned as the weight of the candidate gate, and b_c denotes the bias of the candidate gate.

Now, the old cell state (C_{t-1}) is updated to the new cell state (C_t) by the calculated value of f_t, i_t, and c_t. The new cell state is presented by Equation (4) as follows:

$$C_t = (i_t * c_t) + (f_t * C_{t-1}). \tag{4}$$

Finally, the output is determined. The output will depend on the cell state and will be filtered. The output gate decides the next hidden state. Firstly, we have to run a sigmoid function that tells the parts of the cell state going to the output. Then, the new cell state is put through $\tan h$ function and multiplied by the output of the sigmoid function to decide the information that the hidden state will carry on. Then, the new cell state and the hidden state are carried forward to the next time step. The equations of O_t and h_t are presented by Equations (5) and (6) respectively, as follows:

$$O_t = \sigma\ (h_{t-1} + w_O\ x_t + b_O\ ), \tag{5}$$

$$h_t = O_t * \tan h\ (C_t), \tag{6}$$

where x_t defines current input at time t, h_t defines hidden state at time t, h_{t-1} defines hidden state at the previous time (t-1), b_O is the bias, C_t defines memory or cell state, w_O is the initial weight, and O_t defines output gate. Weight values (w_i, w_c, w_O) are updated during the training of LSTM.

The output Z_t of Bi-LSTM is now a function of the hidden state for the forward pass s_f and for the backward pass s_b along with the corresponding weights (forward weight w_f and backward weight w_b) and bias (b_h), presented by Equation (7) as follows:

$$Z_t = \sigma\left(w_f s_f + w_b s_b + b_h\right). \tag{7}$$

In the proposed framework, using Bi-LSTM, the crop yield of a land is predicted based on the sensor data and weather-related data.

1.1 | Motivations and contributions

Crop yield prediction of land is significant for the economic benefits of the farmers. Cloud computing and machine learning were largely used for crop yield prediction. However, the use of the remote cloud enhances the latency, network traffic, and computation overhead on the cloud. Moreover, the storage and processing of the entire data inside the remote cloud may not be secure. The use of edge computing though dealt with these issues by bringing the resources to the network edge, good Internet connectivity is still a major concern in rural areas mostly containing the harvesting lands. The motivation of this work is to develop an IoAT framework for crop yield prediction at lower latency and high accuracy. To fulfill the objective, the contributions of this article are:

1. An IoT and edge computing-based crop yield prediction framework is proposed, where femtolet is used the edge device. Femtolet is a 5G network device that can provide communication and computation services simultaneously. Femtolet is a small cell base station having high processing ability and storage. The proposed framework is referred

to as *Fly* (*F*emto*l*et-based edge-cloud framework for crop *y*ield prediction). The IoAT sensors are used to collect the soil and environmental data. The sensor data is sent to the femtolet through microcontrollers. The femtolet retrieves weather-related data from the cloud. The femtolet analyses the soil data, environmental data, and weather-related data. After the analysis, the femtolet sends the results to the cloud. The users can access the results from the cloud anytime, anywhere using user credentials.

2. For data analysis, we use Bi-LSTM, and the prediction accuracy, precision, recall, and F1-score are measured. We observe that by using Bi-LSTM the proposed framework outperforms the existing strategies.
3. The latency of the femtolet-based proposed framework is determined and compared with the conventional edge-cloud-based and cloud-only frameworks. The results present that using femtolet the proposed framework outperforms the existing frameworks in terms of latency.

1.2 | Organization of the article

The rest of the article is organized as follows. Section 2 describes the existing research works along with a comparison with the proposed framework. Section 3 presents the femtolet-based proposed framework for crop yield prediction. Section 4 presents the performance analysis of the proposed framework. We conclude the work in Section 5.

2 | RELATED WORK

LSTM neural network was used in place of a backpropagation system to predict the time series in smart agriculture.[15] For precision farming, an intelligent irrigation system using IoT and deep learning neural networks was proposed.[16] This method used an LSTM network to forecast the volumetric soil moisture content, irrigation time, and spatial distribution of water necessary to irrigate arable land one day in advance. By controlling the functioning of the irrigation scheduler, this work focused on the critical needs of agriculture, such as the quantity of water saved and the irrigation period. Along with the LSTM to disclose phenological properties, convolutional neural network (CNN) was also used for exploring the spatial features to develop a crop recommendation system.[17] LSTM and Conv1D were used for the classification of the summer crops by using the Landsat Enhanced Vegetation Index (EVI) time series.[12] Large-scale rice crops were identified using Bi-LSTM to provide the best results among other models.[18] Bi-LSTM can learn time-series data from forward and backward directions, and can achieve good performance in crop classification that gives LSTM a global perspective. Along with several deep learning models, deep reinforcement learning models that is, Deep Recurrent Q Network models were used for proper forecasting of the crop yield.[8] The agent was then awarded an aggregate score for the efforts taken to minimize error and maximize forecast accuracy.

Cloud computing is often used to analyse a large amount of sensor data. To reduce the network traffic and delay, edge and fog computing are used.[2,19,20] By giving farmers access to modern tools and technology that help to maximize output quality and quantity while reducing agricultural costs, smart farming benefits both scientists and agronomists as well as cultivators.[20,21] In fog computing the intermediary fog nodes take part in data processing. Fog computing thereby minimizes network traffic and processing overhead on the cloud. Edge computing has brought resources at the network edge and reduced the latency.[2] An edge computing and IoT data sensing architecture for the entire crop lifespan was established.[22] The suggested method could significantly decrease the amount of time needed to collect data and the amount of energy used. The application of edge computing in agriculture was highlighted.[22,23] The application of IoT was investigated for smart farming.[1,24,25] Though, in the work[25] edge computing and offloading were discussed for smart farming, the data analysis was not highlighted. The agricultural data analysis is a vital issue for crop harvesting because planting the right crop and using the land properly are crucial for farmers' economic success.

A developing area of research in computing science is dew computing that uses end nodes and cloud computing together to store data locally and give users access to it even when they are not online.[26,27] Agricultural fields are typically found in isolated locations in developing nations, where network access may be unavailable frequently. In this scenario, dew computing can offer local storage and enable offline access to the data. However, for large-scale data processing cloud servers are required to be used.

Smart agriculture[28,29] system was proposed for collecting and monitoring temperatures, humidity, and soil moisture data, stored on the cloud for analysis. An expert system was developed with the help of IoT, to identify various weeds, pests, and insects attacking cotton crop.[30] To develop nations and rural areas to utilize the available infrastructure, such as the Internet and renewable resources, an existing work[31] sought to construct an IoT-based system for agricultural activity operation that was scalable and affordable. The economically beneficial crop that will grow throughout the year can be planned using the acquired data, which can then be further assessed on the environmental and climatic circumstances.

Comparison of proposed and existing crop yield prediction frameworks: A comparison of the proposed and existing crop yield prediction frameworks is presented in Table 1. From Table 1 we observe that the proposed approach is novel compared to the existing works. We perform multi-parametric multi-crop data analysis using latency-aware femtolet-based framework and Bi-LSTM. The use of femtolet is a unique feature of the proposed framework. The femtolets are network devices offering communication and computation services simultaneously. In our framework, femtolets are allocated inside the fields for exclusively processing the agricultural data. Latency is a significant parameter for real-time applications. From Table 1, we observe that the latency was not determined in the existing works. In our framework, we have calculated the latency. In our framework, instead of storing the entire data inside the cloud, only the results are sent to the cloud. This in turn helps to protect data privacy, and can reduce the latency, data traffic, and storage overhead on the cloud. The users can access the results from the cloud using user credentials such as login ID and password. We use Bi-LSTM for processing the agricultural data. We use edge computing to reduce the latency. We perform multi-parametric data analysis considering a multicrop dataset to evaluate the performance of the proposed framework more precisely.

3 | FLY: FEMTOLET-BASED EDGE-CLOUD INTEGRATED FRAMEWORK FOR CROP YIELD PREDICTION

In this section, we discuss the proposed framework for predicting the crop yield of a land. After that, the latency of the proposed framework is calculated.

3.1 | Femtolet-based framework for crop yield prediction

In the proposed framework, IoAT sensors, microcontrollers, femtolet, and cloud are used. Figure 1 pictorially presents the proposed framework. The principal components of the proposed framework are mathematically defined as follows.

- Sensor: A sensor (θ) is defined as a three tuple as follows: $\theta = < \theta_{id}, \theta_m, \theta_o >$, where θ_{id}, θ_m, and θ_o denotes the sensor node ID, mode of the sensor that is, active or idle, and the respective object category (humidity, temperature, etc.), respectively.
- Microcontroller: A microcontroller (η) is defined as a two tuple as follows: $\eta = < \eta_{id}, \eta_m, \eta_{sp} >$, where η_{id}, η_m, and η_{sp} denotes the microcontroller ID, its mode that is, active or idle, and its configuration (memory, processor, etc.), respectively.
- Femtolet: A femtolet (ζ) is defined as a three tuple as follows: $\zeta = < \zeta_{id}, \zeta_m, \zeta_{sp} >$, where ζ_{id}, ζ_m, and ζ_o denotes femtolet ID, its mode that is, active or idle, and its configuration (memory, processor, storage, etc.), respectively.
- Cloud servers: To define a cloud computing instance, a two tuple can be used as follows: $\gamma = < \gamma_{id}, \gamma_{pr} >$, where γ_{id} and γ_{pr} denotes the cloud computing instance ID and a set that consists of processing unit IDs of its required cloud servers, respectively.

A sensor node collects the respective object status and sends it to the microcontroller with which it is connected. A microcontroller receives data from the connected sensors and sends the data to the edge device with which it is connected. The femtolet is used as the edge device in the proposed framework. Most of the agricultural fields are in the rural regions. In rural regions, the network connectivity with cellular base stations is not usually good and WiFi access is usually not available. In such a scenario, transmission of the sensor data to the cloud is a challenge. Though edge computing has brought the network resources to the edge, if the connectivity with the base station suffers, accessing the edge server will be also affected in the conventional edge-cloud paradigm, because the edge servers are attached with the base stations.[2]

To solve this problem, we are using one or more femtolets in an agricultural field based on the harvesting land area size, and these femtolets are exclusively allotted for analyzing the soil, environmental, and weather-related data respective to the field. Femtolet is a small cell base station with storage and processing abilities.[3–6] The femtolet provides computation and communication services simultaneously, and can cover an area of 10–20 m. In our system, we are using IoAT

TABLE 1 A comparison of proposed and existing crop yield prediction frameworks.

Work	Used classifier	Data processing in the intermediate node (edge/ fog/ femtolet/ SCceNB)	Multi-parametric dataset-based analysis/ Multi-spectral data analysis	Multi-crop dataset-based analysis	Latency calculation (including data transmission and processing)
Mathew et al.[32]	Random forest (RF)	×	✓	✓	×
Nischitha et al.[33]	Decision tree (DT)	×	✓	✓	×
Thilakarathne et al.[34]	K-nearest neighbor (KNN), DT, RF, extreme gradient boosting (XGBoost), support vector machine (SVM)	×	✓	✓	×
Bakthavatchalam et al.[35]	JRip, decision table, multilayer perceptron (MLP)	×	✓	✓	×
Escorcia-Gutierrez et al.[36]	Gated recurrent unit (GRU), deep belief network (DBN), and Bi-LSTM	×	✓	✓	×
Cruz et al.[37]	KNN	✓	✓	✓	×
Gopal et al.[38]	Hybrid multiple linear regression (MLR)-artificial neural network (ANN)	×	✓	×	×
Nevavuori et al.[39]	CNN	×	✓	✓	×
Chen et al.[14]	Bi-LSTM	×	✓	✓	×
Gopi et al.[40]	LSTM, Bi-LSTM, GRU	×	✓	✓	×
Fly (proposed framework)	Bi-LSTM	✓	✓	✓	✓

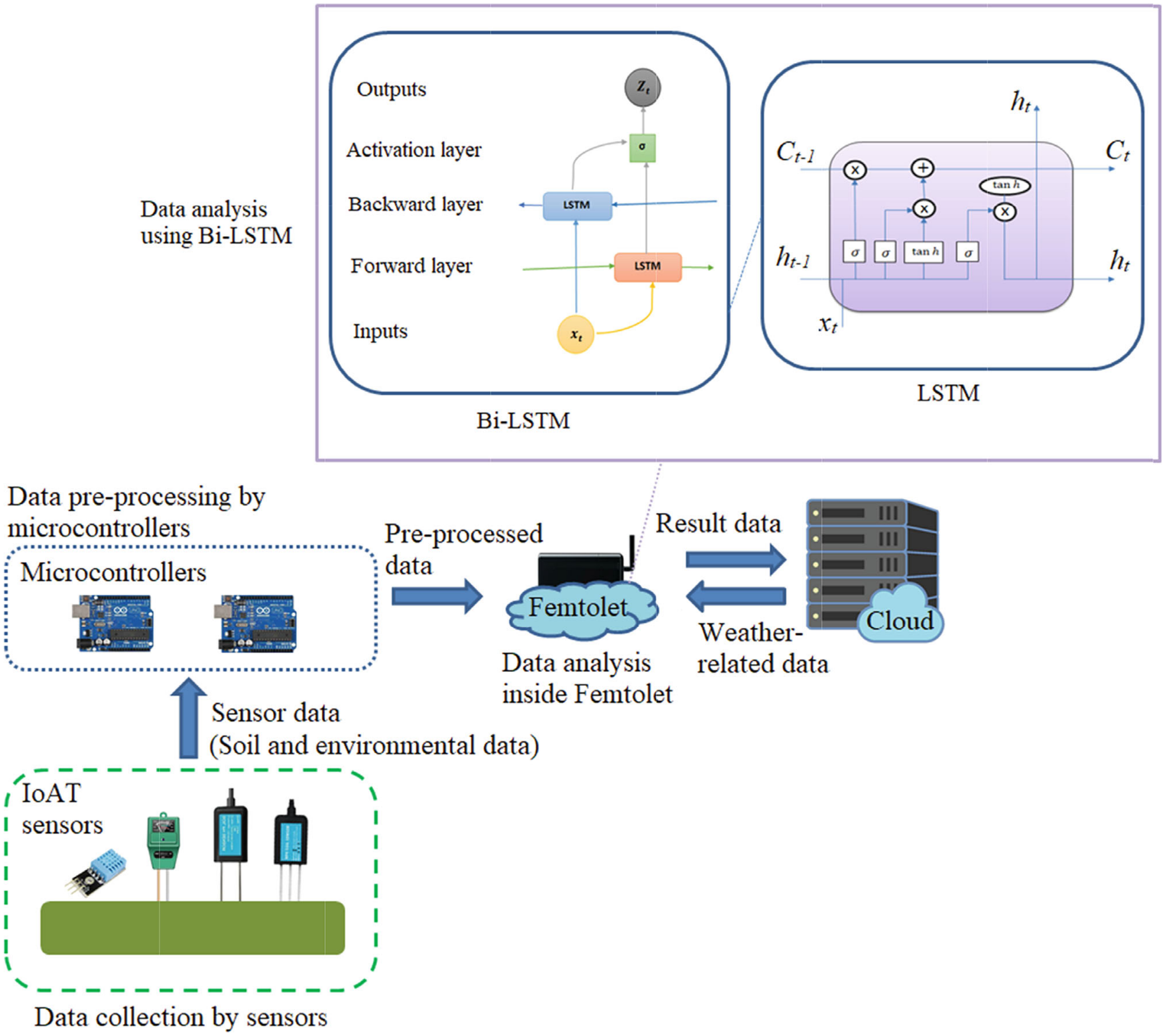


FIGURE 1 Proposed femtolet-based framework for crop yield prediction.

sensors, such as pH, NPK sensors (for collecting pH, Nitrogen (N), Phosphorous (P), and Potassium (K) levels of the soil), temperature and humidity sensors (for collecting environmental temperature and humidity). The sensors send the data to the microcontrollers, which are connected to the femtolet. In the proposed framework, the microcontrollers after receiving data from the IoAT sensors (soil and environmental data), pre-process the data and send it to the femtolet. The femtolet retrieves weather-related data from the cloud such as rainfall data. The femtolet analyses the soil data, environmental data, and weather-related data using Bi-LSTM, and then sends the result to the cloud. A user can access the prediction-related results from the cloud by using proper user credentials (login ID, password). As the results are stored in the cloud, a user can remotely access the results that will help him/her to predict which crop is suitable for his/her land. The Bi-LSTM model used for data analysis has been discussed in Section 1.

3.2 | Latency in the femtolet-based framework for crop yield prediction

Let, the sensor data collection latency is δ_{cl}, the latency in data transmission from the sensors to the microcontrollers is δ_{smi}, the latency involved in queuing and pre-processing the sensor data inside the microcontrollers is δ_{qpmi}, the latency in data transmission from the microcontrollers to the femtolet is δ_{mif}, the latency in weather-related data retrieval from the cloud is δ_{rcf}, the queuing latency of the femtolet is δ_{qf}, the data processing latency inside the femtolet is δ_{pf}, and the latency in result transmission from the femtolet to the cloud is δ_{fc}.

The latency in pre-processing sensor data (soil and environmental data) inside a microcontroller is calculated as, $\frac{data_{\eta}}{speed_{\eta}}$, where $data_{\eta}$ denotes the amount of sensor data and $speed_{\eta}$ denotes the processing speed of the microcontroller.

The latency in soil and environmental data transmission from a microcontroller to the femtolet is calculated as, $\left(1+fail_{mif}\right)\frac{data_{mif}}{rate_{mif}}$, where $fail_{mif}$, $data_{mif}$, and $rate_{mif}$ present the link failure rate, transmitted data amount, and data transmission rate from the microcontroller to the femtolet, respectively.

If more than one microcontroller is there, then δ_{mif} is determined using Equation (8) as follows:

$$\delta_{mif} = \max\left(\delta_{mif1}, \delta_{mif2}, \dots , \delta_{mifN_{mi}}\right), \tag{8}$$

where N_{mi} denotes the number of microcontrollers, δ_{mif1} denotes the data transmission latency from microcontroller η_1 to the femtolet, δ_{mif2} denotes the data transmission latency from microcontroller η_2 to the femtolet, and so on.

The latency in weather-related data transmission from the cloud to the femtolet is presented in Equation (9) as follows:

$$\delta_{rcf} = \left(1+fail_{rcf}\right)\frac{data_{rcf}}{rate_{rcf}}, \tag{9}$$

where $fail_{rcf}$, $data_{rcf}$, and $rate_{rcf}$ present the link failure rate, transmitted data amount, and data transmission rate from the cloud to the femtolet, respectively.

The data processing latency inside the femtolet is presented in Equation (10) as follows:

$$\delta_{pf} = \frac{data_{pf}}{speed_{pf}}, \tag{10}$$

where $data_{pf}$ denotes the amount of data (soil, environmental, and weather-related data) and $speed_{pf}$ denotes the processing speed of the femtolet.

The latency in result transmission from the femtolet to the cloud is presented in Equation (11) as follows:

$$\delta_{fc} = \left(1+fail_{fc}\right)\frac{data_{fc}}{rate_{fc}}, \tag{11}$$

where $fail_{fc}$, $data_{fc}$, and $rate_{fc}$ present the link failure rate, transmitted result data amount, and data transmission rate from the femtolet to the cloud, respectively.

The total latency in the femtolet-based edge-cloud framework is determined using Equation (12) as follows:

$$\delta = \delta_{cl} + \delta_{smi} + \delta_{qpmi} + \delta_{mif} + \delta_{rcf} + \delta_{qf} + \delta_{pf} + \delta_{fc}. \tag{12}$$

The sensor data collection latency depends on the interval set for collecting the data, the number of samples to be considered in data analysis and so forth.

In a conventional edge-cloud framework, the edge server processes the data. However, the distance to the edge server is comparatively higher as edge servers are attached with the cellular base stations.[2] If the latency in data transmission from the microcontrollers to the edge server is δ_{miE}, the latency in weather-related data retrieval from the cloud is δ_{rcE}, the queuing latency of the edge server is δ_{qE}, the data processing latency inside the edge server is δ_{pE}, and the latency in result transmission from the edge server to the cloud is δ_{Ec}, the total latency in conventional edge-cloud framework is determined by Equation (13) as follows:

$$\delta_{EC} = \delta_{cl} + \delta_{smi} + \delta_{qpmi} + \delta_{miE} + \delta_{rcE} + \delta_{qE} + \delta_{pE} + \delta_{Ec}. \tag{13}$$

The latency in soil and environmental data transmission from a microcontroller to the edge server is calculated as, $\left(1+fail_{miE}\right)\frac{data_{miE}}{rate_{miE}}$, where $fail_{miE}$, $data_{miE}$, and $rate_{miE}$ present the link failure rate, transmitted data amount, and data transmission rate from the microcontroller to the edge server, respectively.

If more than one microcontroller is there, then δ_{miE} is determined using Equation (14) as follows:

$$\delta_{miE} = \max\left(\delta_{miE1}, \delta_{miE2}, \ldots, \delta_{miEN_{mi}}\right), \tag{14}$$

where N_{mi} denotes the number of microcontrollers, δ_{miE1} denotes the data transmission latency from microcontroller η_1 to the edge server, δ_{miE2} denotes the data transmission latency from microcontroller η_2 to the edge server, and so on.

The latency in weather-related data transmission from the cloud to the edge server is determined by Equation (15) as follows:

$$\delta_{rcE} = \left(1 + fail_{rcE}\right)\frac{data_{rcE}}{rate_{rcE}}, \tag{15}$$

where $fail_{rcE}$, $data_{rcE}$, and $rate_{rcE}$ present the link failure rate, transmitted data amount, and data transmission rate from the cloud to the edge server, respectively.

The data processing latency inside the edge server is presented in Equation (16) as follows:

$$\delta_{pE} = \frac{data_{pE}}{speed_{pE}}, \tag{16}$$

where $data_{pE}$ denotes the amount of data (soil, environmental, and weather-related data) and $speed_{pE}$ denotes the processing speed of the edge server.

The latency in result transmission from the edge server to the cloud is presented in Equation (17) as follows:

$$\delta_{Ec} = \left(1 + fail_{Ec}\right)\frac{data_{Ec}}{rate_{Ec}}, \tag{17}$$

where $fail_{Ec}$, $data_{Ec}$, and $rate_{Ec}$ present the link failure rate, transmitted result data amount, and data transmission rate from the edge server to the cloud, respectively.

In a conventional cloud-only (sensor-cloud) framework, the data processing takes place inside the cloud. If the latency in data transmission from the microcontrollers to the cloud is δ_{miC}, the queuing latency of the cloud is δ_{qC}, and the data processing latency inside the cloud is δ_{pC}, the total latency in conventional cloud-only framework is calculated by Equation (18) as follows:

$$\delta_{EC} = \delta_{cl} + \delta_{smi} + \delta_{qpmi} + \delta_{miC} + \delta_{qC} + \delta_{pC}, \tag{18}$$

The latency in soil and environmental data transmission from a microcontroller to the cloud is calculated as, $\left(1 + fail_{miC}\right)\frac{data_{miC}}{rate_{miC}}$, where $fail_{miC}$, $data_{miC}$, and $rate_{miC}$ present the link failure rate, transmitted data amount, and data transmission rate from the microcontroller to the cloud, respectively.

If more than one microcontroller is there, then δ_{miC} is presented by Equation (19) as follows:

$$\delta_{miC} = \max\left(\delta_{miC1}, \delta_{miC2}, \ldots, \delta_{miCN_{mi}}\right), \tag{19}$$

where N_{mi} denotes the number of microcontrollers, δ_{miC1} denotes the data transmission latency from microcontroller η_1 to the cloud, δ_{miC2} denotes the data transmission latency from microcontroller η_2 to the cloud, and so on.

The data processing latency inside the cloud is calculated by Equation (20) as follows:

$$\delta_{pC} = \frac{data_{pC}}{speed_{pC}}, \tag{20}$$

where $data_{pC}$ denotes the amount of data (soil, environmental, and weather-related data) and $speed_{pC}$ denotes the processing speed of the cloud.

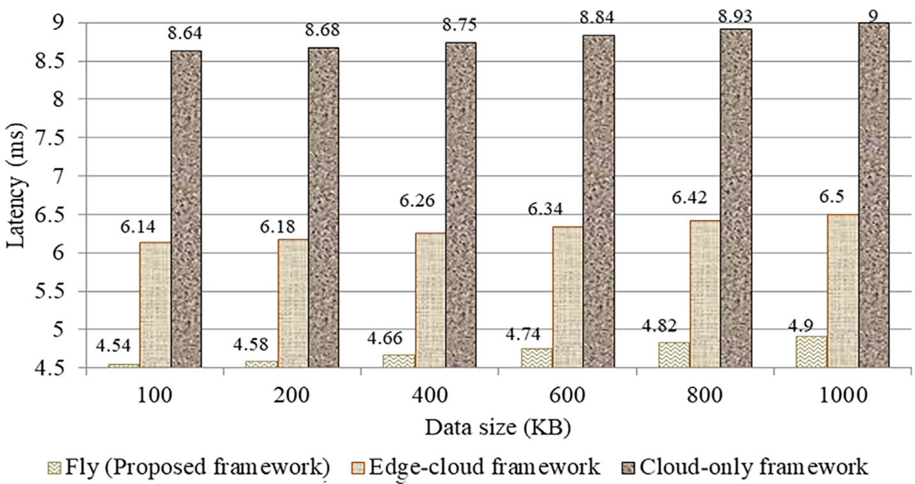
In Section 4, we have performed an analytical comparison among the latencies of the proposed, conventional edge-cloud, and cloud-only frameworks.

4 | PERFORMANCE EVALUATION

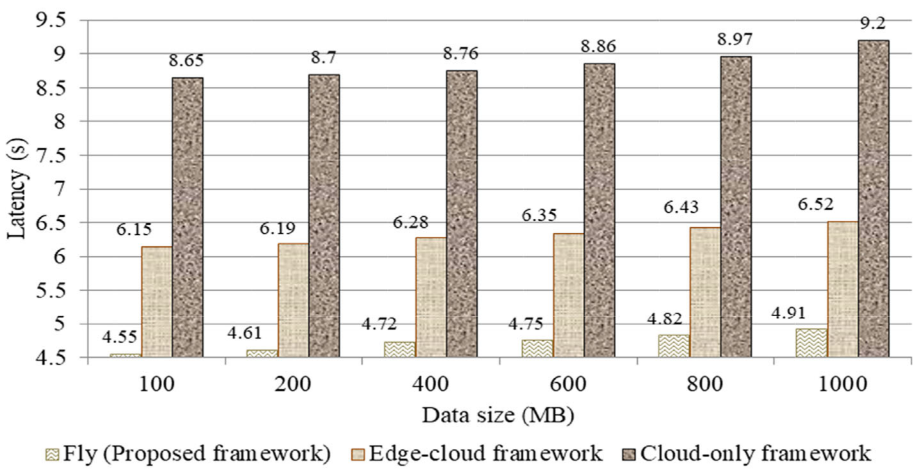
We carry out a theoretical analysis to compare the latencies of the proposed framework (Fly), conventional edge-cloud framework, and cloud-only framework. Then, we simulate the proposed framework using iFogSim and measure the application loop delays. Finally, we perform a data analysis and compare the proposed Bi-LSTM-based framework with the existing crop yield prediction frameworks.

4.1 | Theoretical analysis of latency

For theoretical analysis, we have used MATLABR2022b. The latency of the femtolet-based edge-cloud framework (Fly) is calculated and compared with the conventional edge-cloud-based framework and cloud-only framework for crop yield prediction. The average uplink data transmission rate is assumed 55 Mbps and the average downlink data transmission rate is assumed 110 Mbps. The minimum and maximum uplink data transmission rate is assumed 10 and 100 Mbps respectively. The minimum and maximum downlink data transmission rate is assumed 20 and 200 Mbps respectively. For better performance analysis we have considered two case scenarios: (i) Scenario 1 where data size is considered 100–1000 KB, and (ii) Scenario 2 where data size is considered 100–1000 MB. Figure 2 presents a graphical comparison of the latency of the proposed and conventional frameworks.



(A) Latency for scenario 1



(B) Latency for scenario 2

FIGURE 2 Latency in proposed and conventional frameworks.

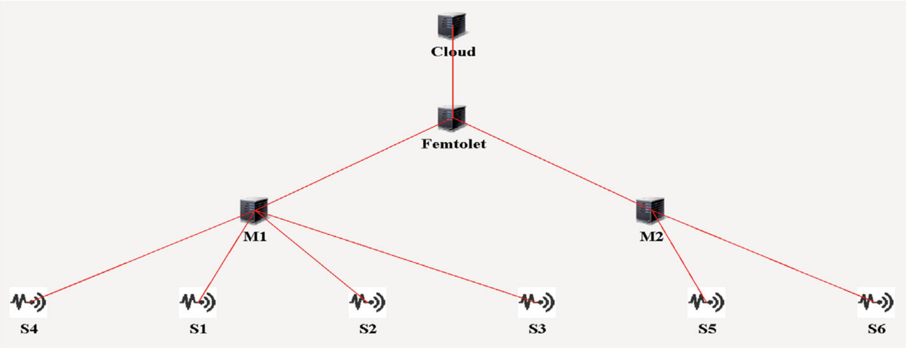


FIGURE 3 Topology of femtolet-based edge-cloud framework.

TABLE 2 Simulation parameters details.

RAM size of each microcontroller	100 KB
CPU length of each microcontroller	1000 MPIS
RAM size of Femtolet	8 GB
CPU length of Femtolet	40,000 MIPS
RAM size of Cloud virtual machine (VM)	8 GB
CPU length of Cloud VM	44,800 MIPS
Uplink bandwidth	20–100 Mbps
Downlink bandwidth	40–200 Mbps

We observe that the proposed framework reduces the latency by approximately 25% and 45% compared to the conventional edge-cloud framework and cloud-only framework respectively. In conventional cloud-only system, the microcontrollers send the sensor data to the cloud. However, most of harvesting lands are located at the rural regions, where the Internet connectivity is poor. Hence, the sensor data transmission to the cloud suffers from connection interruption that increases data transmission latency. As a result, the total latency is increased. In our framework, the femtolets are deployed inside the fields and used exclusively for soil, environmental, and weather-related data analysis for crop yield prediction. Thus, the data transmission latency is reduced. As a result, the total latency is reduced in Fly compared to the conventional cloud-only framework for crop yield prediction. In a conventional edge-cloud framework, the connectivity with the edge servers may get affected as the rural regions are mainly used for crop harvesting. Further, the edge servers are not exclusively allotted for agricultural data analysis. In the proposed approach, the femtolets are in the fields; thus, close to the sensors. Further, the femtolets having high processing ability and storage, are exclusively allotted for analyzing the soil, environmental, and weather-related data for crop yield prediction. As a result, the latency is low in our framework compared to the conventional edge-cloud framework.

4.2 | Simulation of the femtolet-based edge-cloud framework using iFogSim

To model and simulate the femtolet-based edge-cloud framework, we have used iFogSim.[41,42] Figure 3 presents the created topology in iFogSim. Here, we have used two microcontrollers (M1 and M2). With M1, four sensors (S1, S2, S3, and S4) are connected. With M2, two sensors (S5 and S6) are connected. S1, S2, and S3 sensors collect N (value: 0–140), P (value: 5–145), and K (value: 5–205) levels of the soil, respectively. S4 collects pH (value: 3–10) level of the soil. S5 and S6 sensors collect temperature (value: 8–44) and humidity (value: 14–100) of the environment, respectively. Based on the real dataset considered in the analysis (refer to Section 4.3), the maximum and minimum sensor values are taken as input. M1 and M2 are connected to the Femtolet, and the Femtolet is connected to the Cloud. Here, Femtolet is used as the edge device. The simulation parameter details are provided in Table 2. Five cases we have considered based on the uplink and downlink bandwidth, as shown in Figure 4. In case 1, case 2, case 3, case 4, and case 5, the uplink bandwidth are considered 20, 40,

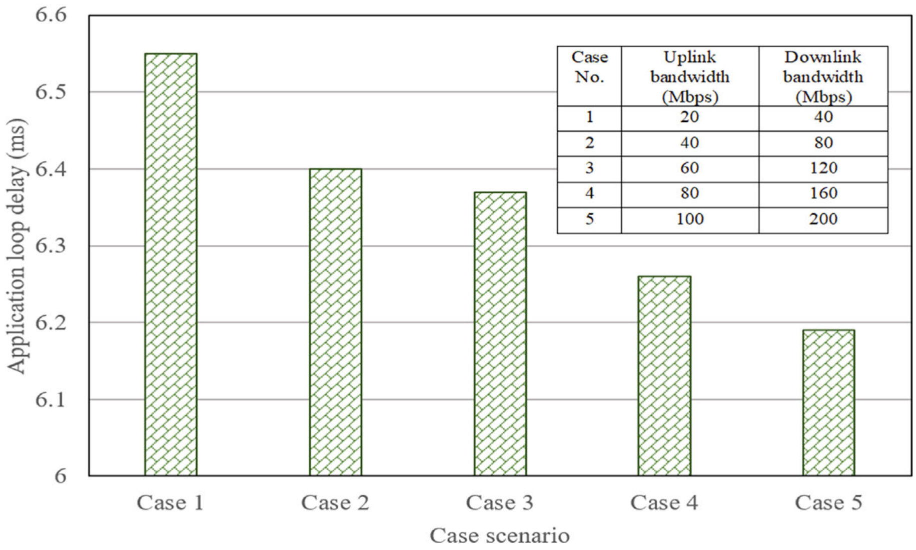


FIGURE 4 Application loop delay in the proposed framework.

TABLE 3 Statistical summary of the dataset containing 2200 samples.

Parameters	Minimum	Maximum	Mean	SD
N	0	140	50.55	36.91
P	5	145	53.36	32.98
K	5	205	48.14	50.64
pH	3.5	9.93	6.47	0.77
Temperature	8.82	43.67	25.62	5.06
Humidity	14.25	99.98	71.48	22.26
Rainfall	20.21	298.56	103.46	54.95

60, 80, and 100 Mbps, respectively. In case 1, case 2, case 3, case 4, and case 5, the downlink bandwidth are considered 40, 80, 120, 160, and 200 Mbps, respectively. The application loop delays in these five cases are presented in Figure 4. The application loop delay refers to the delay during the execution of an application from the time of request to the serving node. As observed from the results the application loop delay is 6.1–6.6 ms in our framework.

4.3 | Data analysis using Bi-LSTM

We have used a standard deep learning algorithm Bi-LSTM to predict crop yields from the input dataset.[43] The soil, environment, and weather factors (levels of pH, N, P, K in the soil, humidity, temperature, rainfall), as well as the crop, are considered in the input dataset containing 2200 samples. The number of entries of each parameter is 2200. Table 3 presents the minimum, maximum, mean, and standard deviation (SD) of the parameters in the dataset.

The accuracy, precision, recall, and F1-score are considered as the performance metrics. The classification accuracy is the proportion of correctly predicted crop yields to all the crop yield predictions made by the classifier, determined as follows: $Accuracy = \frac{\tau\rho+\tau\nu}{\tau\rho+\tau\nu+\phi\rho+\varphi\nu}$, where $\tau\rho$ denotes true positive, $\tau\nu$ denotes true negative, $\phi\rho$ denotes false positive, and $\varphi\nu$ denotes false negative. The F1-score is calculated based on the precision and recall of the model. Considering the harmonic mean of the precision and recall scores, the standard F1-score is produced. An F1 score of 1 corresponds to an ideal model. The F1-score is calculated as follows: $F1_{score} = \frac{2*(precision*recall)}{(precision+recall)}$, where $precision = \tau\rho/(\tau\rho + \varphi\rho)$ and $recall = \tau\rho/(\tau\rho + \varphi\nu)$. The accuracy level of the classification for the dataset is measured by comparing the actual class labels with the predicted class labels. According to the classifiers, the prediction accuracy of twenty-two (22) different classes

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	18	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
4	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	19	0	0	1	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0
21	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	18	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20

FIGURE 5 Confusion matrix for 2200 samples.

(1. Jute, 2. Rice, 3. Coffee, 4. Cotton, 5. Coconut, 6. Papaya, 7. Orange, 8. Banana, 9. Mango, 10. Grapes, 11. Watermelon, 12. Muskmelon, 13. Apple, 14. Pomegranate, 15. Lentil, 16. Blackgram, 17. Mungbean, 18. Mothbeans, 19. Pigeonpeas, 20. Kidneybeans, 21. Chickpea, and 22. Maize) are measured. In the real data set, we have 2200 samples. We have considered 550 samples (data size: 36 KB), 1100 samples (data size: 71 KB), 1650 samples (data size: 107 KB), and then the entire dataset containing 2200 samples (data size: 142 KB). The confusion matrix for 2200 samples is presented in Figure 5. Each entry in a confusion matrix shows how many predictions the model has done correctly or incorrectly.

Figure 6 presents the accuracy for case study 1 (550 samples), case study 2 (1100 samples), case study 3 (1650 samples), and case study 4 (2200 samples). This is observed that the accuracy for case studies 1, 2, 3, and 4 are 1 (100%), 0.98 (98%), 0.9818 (98.18%), and 0.9864 (98.64%), respectively. Figure 7 presents the F1-score for case study 1, case study 2, case study 3, and case study 4. As observed, the F1-score for case studies 1, 2, 3, and 4 are 0.99 (99%), 0.986 (98.6%), 0.985 (98.5%), and 0.9868 (98.68%), respectively. We have also generated a synthetic dataset of 55,000 samples (3.7 MB) for analyzing the performance of the model for a large dataset. This is observed that the Bi-LSTM classifier obtains an accuracy of 99.38% for analyzing the large dataset containing 55,000 samples.

Comparison of the proposed and existing models on crop yield prediction: In Table 4 and Figure 8, the proposed model is compared with the existing works on crop yield prediction using the dataset.[43] The complete dataset containing 2200 samples has been used. The accuracy, precision, recall, and F1-score using the proposed and existing approaches are presented in Table 4 and Figure 8.

As observed from Table 4, the existing work[37] used KNN and achieved 92.62% accuracy. In another existing work,[34] a comparative analysis among RF, KNN, DT, XGBoost, and SVM was presented based on their performance, and RF achieved the highest accuracy of 97.18%. In the existing work,[35] MLP, JRip, and decision table were used, and MLP achieved the highest accuracy of 98.23%. In the existing work,[40] LSTM, Bi-LSTM, and GRU were used, and achieved an accuracy of 98.45%. In our proposed work Fly, we have used Bi-LSTM and achieved an accuracy of 98.64%, which is higher compared to the existing approaches.[34,35,37,40] The precision, recall, and F1-score metrics are also higher in Fly, as observed from Table 4 and Figure 8. We also wish to highlight that the existing works[34,37,40] did not measure the time to build model. However, the existing work[35] measured the time to build model. For MLP the time to build model is 8.78 s but for decision table and JRip the time to build model are comparatively low. MLP achieved highest accuracy of 98.23%, where decision table and JRip achieved the accuracy of 88.59% and 96.23%, respectively.[35] Though the time to build model in JRip and decision table are lower, the accuracy is less than the MLP. We observed that in our framework Bi-LSTM has achieved

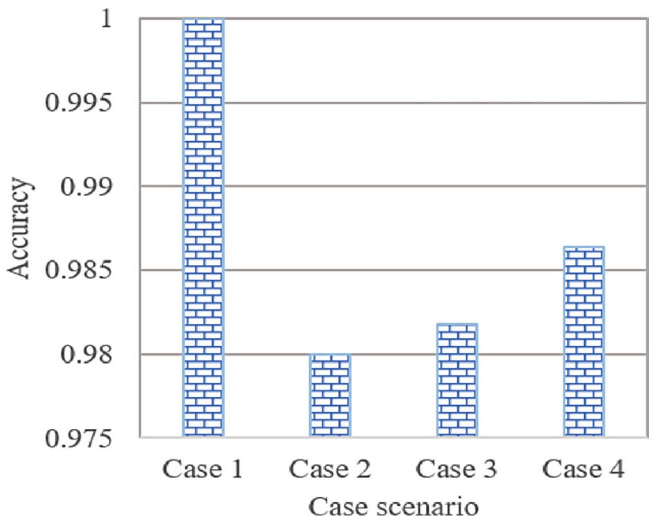


FIGURE 6 Accuracy for four considered case studies (case 1: 550 samples, case 2: 1100 samples, case 3: 1650 samples, case 4: 2200 samples).

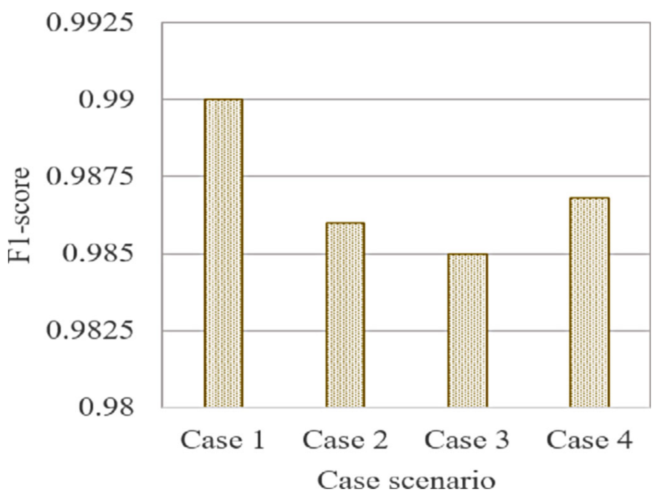


FIGURE 7 F1-score for four considered case studies (case 1: 550 samples, case 2: 1100 samples, case 3: 1650 samples, case 4: 2200 samples).

accuracy of 98.64% which is higher than the existing work.[35] In Fly the time to build model is 0.53 s, which is much low than the MLP.[35] Thus, if accuracy and time both are considered, our framework outperforms the existing work.[35]

We have observed in Section 4.1 that the use of femtolet in our work reduces the total latency compared to the conventional cloud-only and edge-cloud systems. As the data processing takes place locally in femtolet, the data privacy is maintained. Hence, finally we can conclude that the proposed approach is better than the existing approaches.

5 | CONCLUSIONS AND FUTURE WORK

We proposed an edge-cloud framework (Fly) for crop yield prediction using a 5G network device femtolet. The IoAT sensor nodes collect the soil (pH, N, P, and K levels) and environmental (temperature and humidity) data. The microcontrollers receive data from the sensors. After pre-processing the received data, the microcontrollers send it to the femtolet that serves as the edge device. The femtolet has high processing ability and storage, and can analyse a large amount of data. The femtolet receives the soil and environmental data from the microcontrollers, retrieves the weather-related data from the cloud, and then processes the soil, environmental, and weather-related data using Bi-LSTM. After processing the data, femtolet sends the generated results to the cloud. Using user credentials, the user can access the prediction results from

TABLE 4 Comparison of the proposed and existing approaches using the dataset.[43]

Work	Considered classifier	Accuracy	Precision	Recall	F1-score	Time to build model
Cruz et al.[37]	KNN	92.62%	96.74%	92.62%	95.46%	Not measured
Thilakarathne et al.[34]	RF (highest accuracy) KNN, DT, XGBoost, SVM	97.18%	97%	97%	97%	Not measured
Bakthavatchalam et al.[35]	MLP (highest accuracy), JRip, Decision Table	98.23%	98.4%	98.23%	98.2%	MLP: 8.78 s, JRip: 0.15 s, decision table: 0.24 s
Gopi et al.[40]	LSTM, Bi-LSTM, GRU	98.45%	98.51%	98.45%	98.46%	Not measured
Fly (proposed work)	Bi-LSTM	98.64%	98.68%	98.64%	98.68%	0.53 s

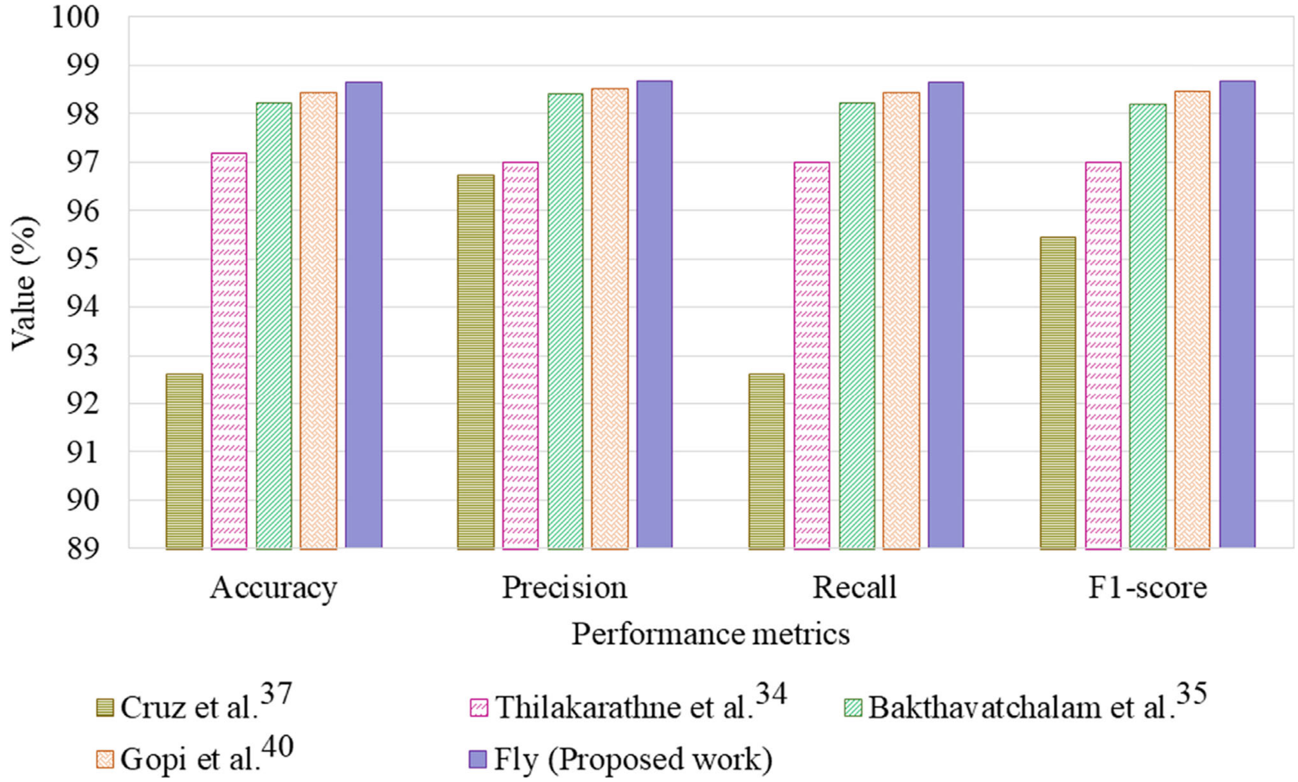


FIGURE 8 Comparison of proposed and existing works with respect to accuracy, precision, recall, and F1-score.

the cloud which will help to find the suitable crop for his/her land. We observe that the use of femtolet reduces the latency by ~25% and ~45% than the conventional edge-cloud and cloud-only frameworks respectively. The data analysis presents that the proposed framework has achieved an accuracy of 98.64%, and outperforms the existing crop yield prediction frameworks in terms of accuracy, precision, recall, and F1-score.

As part of our future work, we plan to investigate the correlation of different soil parameters for soil health monitoring for better crop productivity. The use of federated learning in crop yield prediction is another future research scope of this work.

AUTHOR CONTRIBUTIONS

Tanushree Dey: Writing the draft paper, methodology, experiment. **Somnath Bera:** Writing the draft paper, methodology, experiment. **Bachchu Paul:** Methodology, experiment. **Debashis De:** Manuscript review and editing, supervision.

Anwesha Mukherjee: Methodology, Experiment, Manuscript review and editing, supervision. **Rajkumar Buyya:** Manuscript review and editing, supervision.

FUNDING INFORMATION

There is no funding for this research.

DATA AVAILABILITY STATEMENT

The link of the dataset used for the analysis is mentioned in the paper.

ORCID

Debashis De https://orcid.org/0000-0002-9688-9806
Anwesha Mukherjee https://orcid.org/0000-0001-9110-8591
Rajkumar Buyya https://orcid.org/0000-0001-9754-6496

REFERENCES

1. Farooq MS, Riaz S, Abid A, Abid K, Naeem MA. A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *IEEE Access*. 2019;7:156237-156271.
2. Mukherjee A, De D, Ghosh SK, Buyya R. Introduction to mobile edge computing. *Mobile Edge Computing*. Springer; 2021:3-19.
3. Mukherjee A, De D. Femtolet: A novel fifth generation network device for green mobile cloud computing. *Simul Model Pract Theory*. 2016;62:68-87.
4. Deb P, Mukherjee A, De D. Design of green smart room using fifth generation network device femtolet. *Wirel Pers Commun*. 2019;104:1037-1064.
5. Mukherjee A, De D, Buyya R. E2R-F2N: Energy-efficient retailing using a femtolet-based fog network. *Softw Pract Experience*. 2019;49(3):498-523.
6. Mukherjee A, Deb P, De D, Buyya R. IoT-F2N: An energy-efficient architectural model for IoT using Femtolet-based fog network. *J Supercomput*. 2019;75(11):7125-7146.
7. Van Klompenburg T, Kassahun A, Catal C. Crop yield prediction using machine learning: A systematic literature review. *Comput Electron Agric*. 2020;177:105709.
8. Elavarasan D, Vincent PD. Crop yield prediction using deep reinforcement learning model for sustainable agrarian applications. *IEEE Access*. 2020;8:86886-86901.
9. Javaid N. Integration of context awareness in Internet of Agricultural Things. *ICT Exp*. 2023;9(2):189-196.
10. Bashar A. Survey on evolving deep learning neural network architectures. *J Artif Intell*. 2019;1(2):73-82.
11. Santos L, Santos FN, Oliveira PM, Shinde P. *Deep learning applications in agriculture: A short review. Iberian Robotics conference*. Springer; 2019.
12. Zhong L, Hu L, Zhou H. Deep learning based multi-temporal crop classification. *Remote Sens Environ*. 2019;221:430-443.
13. Nguyen TT, Hoang TD, Pham MT, et al. Monitoring agriculture areas with satellite images and deep learning. *Appl Soft Comput*. 2020;95:106565.
14. Chen B, Zheng H, Wang L, et al. A joint learning Im-BiLSTM model for incomplete time-series Sentinel-2A data imputation and crop classification. *Int J Appl Earth Obs Geoinf*. 2022;108:102762.
15. Suryo PSBC, Mustika IW, Wahyunggoro O, Wasisto HS. Improved time series prediction using LSTM neural network for smart agriculture application. *2019 5th International Conference on Science and Technology (ICST)*. IEEE; 2019.
16. Kashyap PK, Kumar S, Jaiswal A, Prasad M, Gandomi AH. Towards precision agriculture: IoT-enabled intelligent irrigation systems using deep learning neural network. *IEEE Sensors J*. 2021;21(16):17479-17491.
17. Sun J, Di L, Sun Z, Shen Y, Lai Z. County-level soybean yield prediction using deep CNN-LSTM model. *Sensors*. 2019;19(20):4363.
18. de Castro C, Filho H, de Carvalho A, et al. Rice crop detection using LSTM, Bi-LSTM, and machine learning models from sentinel-1 time series. *Remote Sens*. 2020;12(16):2655.
19. Mansouri Y, Babar MA. A review of edge computing: Features and resource virtualization. *J Parallel Distrib Comput*. 2021;150:155-183.
20. Kalyani Y, Collier R. A systematic survey on the role of cloud, fog, and edge computing combination in smart agriculture. *Sensors*. 2021;21(17):5922.
21. O'Grady M, Langton D, O'Hare G. Edge computing: A tractable model for smart agriculture? *Artificial Intell Agri*. 2019;3:42-51.
22. Zhang R, Li X. Edge Computing Driven Data Sensing Strategy in the Entire Crop Lifecycle for Smart Agriculture. *Sensors*. 2021;21(22):7502.
23. Guillén MA, Llanes A, Imbernón B, et al. Performance evaluation of edge-computing platforms for the prediction of low temperatures in agriculture using deep learning. *J Supercomput*. 2021;77(1):818-840.
24. Ahmed N, De D, Hussain I. Internet of Things (IoT) for smart precision agriculture and farming in rural areas. *IEEE Internet Things J*. 2018;5(6):4890-4899.
25. Zhang J, Li X, Zhang X, Xue Y, Srivastava G, Dou W. Service offloading oriented edge server placement in smart farming. *Softw Pract Experience*. 2021;51(12):2540-2557.

26. Hati S, De D, Mukherjee A. DewBCity: blockchain network-based dew-cloud modeling for distributed and decentralized smart cities. *J Supercomput*. 2022;78(6):8977-8997.
27. Javadzadeh G, Rahmani AM, Kamarposhti MS. Mathematical model for the scheduling of real-time applications in IoT using Dew computing. *J Supercomput*. 2022;78(5):7464-7488.
28. Yang J, Sharma A, Kumar R. IoT-based framework for smart agriculture. *Int J Agri Environ Informat Syst*. 2021;12(2):1-14.
29. Gondchawar N, Kawitkar R. IoT based smart agriculture. *Int J Adv Res Comput Commun Eng*. 2016;5(6):838-842.
30. Shahzadi R, Ferzund J, Tausif M, Suryani MA. Internet of things based expert system for smart agriculture. *Int J Adv Comput Sci Appl*. 2016;7(9):341-350.
31. Jaiswal SP, Bhadoria VS, Agrawal A, Ahuja H. Internet of Things (IoT) for smart agriculture and farming in developing nations. *Int J Sci Technol Res*. 2019;8(12):1049-1056.
32. Mathew J, Joy A, Sasi D, Jiji J, John J. Crop prediction and Plant Disease Detection using IoT and Machine learning. *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE; 2022.
33. Nischitha K, Vishwakarma D, Ashwini MM, Manjuraju MR. Cropprediction using machine learning approaches. *Int J Eng Res Technol*. 2020;9(8):23-26.
34. Thilakarathne NN, Bakar MSA, Abas PE, Yassin H. A Cloud Enabled Crop Recommendation Platform for Machine Learning-Driven Precision Farming. *Sensors*. 2022;22(16):6299.
35. Bakthavatchalam K, Karthik B, Thiruvengadam V, et al. IoT framework for measurement and precision agriculture: predicting the crop using machine learning algorithms. *Technologies*. 2022;10(1):13.
36. Escorcia-Gutierrez J, Gamarra M, Soto-Diaz R, Pérez M, Madera N, Mansour RF. Intelligent agricultural modelling of soil nutrients and ph classification using ensemble deep learning techniques. *Agriculture*. 2022;12(7):977.
37. Cruz M, Mafra S, Teixeira E. *An IoT Crop Recommendation System with k-NN and LoRa for Precision Farming*. Sociedade Brasileira de Telecomunicações; 2022.
38. Gopal PM, Bhargavi R. A novel approach for efficient crop yield prediction. *Comput Electron Agric*. 2019;165:104968.
39. Nevavuori P, Narra N, Lipping T. Crop yield prediction with deep convolutional neural networks. *Comput Electron Agric*. 2019;163:104859.
40. Gopi PSS, Karthikeyan M. Red fox optimization with ensemble recurrent neural network for crop recommendation and yield prediction model. *Multimed Tools Appl*. 2024;83:13159-13179.
41. Gupta H, Vahid Dastjerdi A, Ghosh SK, Buyya R. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Softw Pract Experience*. 2017;47(9):1275-1296.
42. Mahmud R, Buyya R. Modelling and simulation of fog and edge computing environments using iFogSim toolkit. *Fog and Edge Computing: Principles and Paradigms*. WILEY; 2019:1-35.
43. Ingle A. Crop Recommendation Dataset, V1. 2020 https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset

How to cite this article: Dey T, Bera S, Paul B, De D, Mukherjee A, Buyya R. Fly: Femtolet-based edge-cloud framework for crop yield prediction using bidirectional long short-term memory. *Softw: Pract Exper*. 2024;54(8):1361-1377. doi: 10.1002/spe.3324