eFarm-Lab: Edge AI-IoT Framework for Agronomic Labs Experiments

Youssouph Gueye and Maïssa Mbaye(✉)

LANI (Laboratoire D'Analyse Numérique et Informatique), Université Gaston Berger de Saint-Louis, Saint-Louis, Senegal
{gueye.youssouph1,maissa.mbaye}@ugb.edu.sn

Abstract. Agronomists deal with challenges to determinate ideal parameters (e.g., soil moisture, temperature, etc.) to grow each variety of plants according to the nature of soil and climate zone. Traditional method consists in having experimental farms in which different conditions are created to discover which environmental and chemical conditions enable maximizing yield for each variety of seed. This process is fastidious and accuracy of results is difficult to evaluate. In this paper, we propose an Edge AI Internet of Things (IoT) framework for agronomic experimentations and will the solution be cost efficient, easy to deploy, low maintenance, and robust, which makes it very appealing in the African context. Our proposal is composed of three segments: experimental farm zone (Lab) where sensors and actuators network are deployed, a set of data collection and processing gateways called Edge AI-IoT Nodes which implements Edge Machine Learning Models, and Cloud and Fog segment that provides a social network and services for agronomic experts. Social network is an interface for agronomic experts that allow them to follow data collected from experimental farms and for cross validation of results around the world. For the purpose of illustration two use cases are presented: plant leaf disease detection using machine learning; and smart automated irrigation with IoT framework.

Keywords: AI-IoT · Edge computing · Smart-agriculture · Smart-irrigation · Machine learning · Experimental farms · Plant Leaf Disease Detection · ICT4D

1 Introduction

Internet of Things (IoT) based Smart-Agriculture is a fast-emerging research and development field with wide range of applications. It consists in using in farming sensors or Unmanned Aerial Vehicles (UAV) to collect data on farm's physical environment (soil moisture, pH., temperature, wind speed, electrical conductivity, etc.) and actuators connected to communication system. The result can be a decision-support systems (such as proper amount of nitrogen, phosphorus, potassium, etc.), optimization system of farming resources (water, fertilizers,

Y. Faye et al. (Eds.): CNRIA 2021, LNICST 400, pp. 101–112, 2021.
https://doi.org/10.1007/978-3-030-90556-9_9

insecticides, etc.) [5], farming monitoring systems (such as detecting plant stress, wheat diseases, pests, and weeds), automated irrigation system.

However, the optimizing tasks and early agronomic research are less addressed by IoT in specific areas in Africa. Indeed, African agronomic researchers and engineers have less opportunities to experiment with a large variety of Farms-Labs environments a large variety of Farms-Labs environments. IoT and AI tools on the edge can be very valuable [13,17]. For instance, an IoT based automated irrigation system wouldn't be efficient without taking into account threshold of dry and moisture that can be supported by plants in the field. Agronomic Engineers might need systems that assist them to monitor their testing farms and provide support in analyzing produced data. This is even more relevant in the African context where there is a lack of agronomists experts and an inefficiency due to outdated and/or out of context data.

Traditional IoT architectures composed of IoT Core network and cloud computing resources are not suitable in the Sub-Saharan Africa area. The main reasons include the followings: firstly these architectures require centralization in a Cloud as well as good network coverage in the experimental fields [14]. Secondly, rural areas in Sub-Saharan Africa suffers from low network coverage and available bandwidth. This makes it very difficult to consider developing centralized architecture. Finally, national agronomic research structures do not have much means to support large scale tests over a long period of time.

In this paper we propose an Edge AI-IoT framework for experimental agriculture that we call eFarm-Lab. Basically, the use of IoT, Edge, and AI in agriculture is not new [7,16]. However, in our knowledge, using a framework for studying agronomic conditions in experimental farms is something new as far as we know. The general principle is that the framework is designed to allow, on the fly, machine learning modelling and deployment of models on Edge Nodes to assist local agronomic researchers in their experimental labs. So the outcome of this proposal targets experimentation farms, not production ones. For instance, to study growth phases of a plant, sensors (cameras, humidity sensors, etc.) can be deployed to monitor the height of the plant and other agronomic parameters, and use machine learning models to better know the needs of the plant.

eFarm-Lab is composed of three segments : Simple IoT sensors and actuators network; a set of Edge-AI-IoT nodes implementing machine learning models for experimentation; and finally Cloud architecture. A social network of agronomic experts as oracles can help labelling data and enhance quality of learning. Social network is an interface for agronomic experts that allow them to follow plants evolution using pictures captured by platform and for cross validation of results by agronomist community.

2 Related Works

There are several works in the field of smart agriculture based on IoT eventually with AI. Topics covers from Smart Irrigation systems [3,9,12], Monitoring and information collection systems [9,11], Crops Protection systems and data analysis [15] and plan disease detection [4]. Main network technologies are WiFi, WiMAX, LR-WPAN, GSM-Based, Bluetooth, LoRa, SigFox, NB-IoT.

Bu et al. [6] use deep Reinforcement learning in IoT network for Smart agriculture. In this work, computations are centralized on the cloud.

Angelopoulos et al. in [2] propose an Edge computing architecture to reduce the traffic between the IoT network and cloud.

Ahmed Imteaj et al. proposed a system that is able to detect the appropriate time to water the field according to the soil moisture and the intensity of light. The system can also monitor irrigation level to prevent accumulation of water around tree roots and send a text message to the farmer in case of lack of water [12]. In [3], authors presents different technologies that can be used in the implementation of an automatic irrigation system for saving water using the Internet of Things. In this article, authors use Zigbee for communication between the sensors and the actuator. Authors of paper [9] designed a basic system based on the Internet and the cloud technologies. LI-FI technology is used to provide communication between the sensors and the data collection server. It is used to collect all the information on the field and to send on the cloud using GPRS or WIMAX as a transmission medium.

[1] has proposed smart farming using automation and IoT technology. The authors have implemented a GPS-based remote-controlled vehicle that will perform several tasks in the field and in the warehouse. Her tasks include scaring birds and animals, detecting soil moisture, spraying fertilizers and pesticides, weeding, detecting soil moisture, and so on.

If we sum up, all these papers about smart agriculture try to enhance agriculture inside production farms. However, before automating irrigation, or detecting plant disease, thresholds must be tested out by agronomist engineers.

Our objective in this paper is to design an Edge AI-IoT framework for experimenting farm conditions for development of varieties of plants in an uncontrolled environment. This point is very relevant for Sub-Saharan Africa since there is not enough agronomist experts to realise this kind of experimentation.

A second aspect of our proposal is it includes a social network of experts in agronomy in order to test different conditions of farming and remotely validated the best ones.

3 Our Proposition

3.1 General Architecture of Proposed Framework

eFarmLab is composed of three segments (Fig. 1): Experimentation farm domain that contains sensors and actuators network, Edge AI-IoT Nodes for small AI training and model deployment and Cloud/Fog for larger machine learning models training and dataset storage.

The experimentation farm area contains a set of plant squares, each corresponding to a specific agronomic experience (seed selection, disease study, plant need, etc.). The different squares can reproduce the same conditions or environment for experimenting and/or monitoring plant evolution (Fig. 2). These area contains plants and network of end nodes (sensors and actuators network). Sensor and actuator network is a set of nodes that embed sensors for collecting

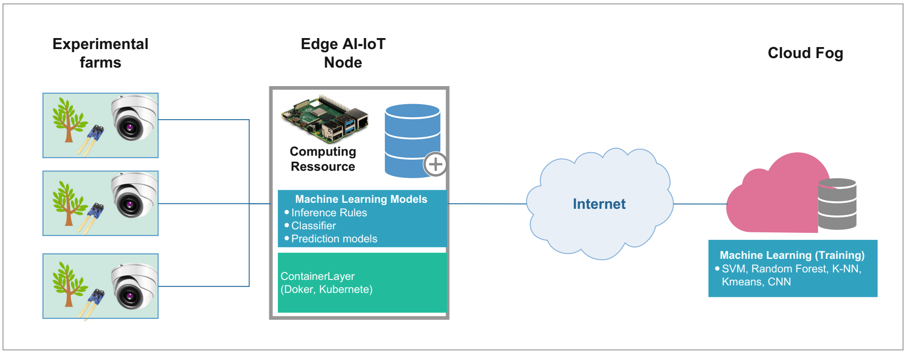

Fig. 1. General architecture for Edge AI-IoT framework

data or modifying environmental conditions via actuator to create different conditions in experimental field (such as starting watering). This network collects data about plant environment parameters like thresholds, soil moisture, plant appearance with camera, and communicate with Edge AI-IoT Nodes.

These sensor-actuator-nodes can manage multiple sensors depending on the need of monitoring. More concretely, if expert wants to explore the stress level of the plant according to the aridity conditions on the maps, it would be possible to use one or more cameras to monitor the general appearance of the plant.

The Edge AI-IoT Nodes (Gateway) act as interface between the sensor and actuator network, and the agronomic experts web based interface through the Internet. These nodes have the role of hosting the intelligence of the network. Intelligence is represented by training lightweight machine learning models but also receiving the deployment of models that come from the fog computing part.

This can be implemented by existing Edge AI platforms (Raspberry Pi, Nvidia Jetson Nano, etc.) with or without a GPU. Tiny Machine Learning models can be trained directly on Edge AI-IoT Node for ROI (Region of Interest) detection (KNN, KMeans, etc.). We will provide an example of directly trained machine learning model on Edge Node.

To allow the deployment of more complex models these nodes host environment containers so that they do not have compatibility issues in running or deployed models. This enables hot deployment and programming of the AI-IoT Edge node.

Finally the Fog/Cloud segment has more computing and storage resources to store larger datasets and train more complex/greedy machine learning algorithms such as CNN. The output models can be deployed on the Edge AI-IoT Nodes.

Agronomists experts use web based interface (social network) to evaluate the result of the machine learning services or enhancing them. With this platform, it could be allowed expert to participate in experimentations by comparing aspects of the plants at different moment. In this way, they can indicate to platform if it is doing well or not. Access devices could be tablets, smartphones and computers.

Fig. 2. Example of experimental farm captured at Saint-Louis/Senegal

The aim of this overall architecture is to automate testing and monitoring for agronomist while giving them possibility to participate on model enhancement.

3.2 Machine Learning Deployment on the Edge IoT

Machine learning tools are deployed in different places in the network. On the Cloud-Computing/fog part where there are more storage and computing resources available, machine learning algorithms are trained on large data sets in order to produce the best models according to what the agronomist expert seeks to study. For example, if the objective of the platform is to detect the presence or absence of disease of a plant from leaves in the Fog part, we will have a dataset of leaves of diseased or healthy plants. These models are created using well-known machine learning algorithms such as SVM, KNN, KMeans, CNN Machine learning models are mainly represented as classifiers, decisions trees, equations, inference rules, etc.

Once the model is validated, it can be deployed to any Edge node it has enough resources. Deployment can be done using containers instead of traditional virtual machines because they are lighter to deploy and consume less computing and storage resources.

This functionality makes it possible to deploy machine models in any equipment, knowing that the context. In the next session we will illustrate use cases of the deployment.

3.3 AI on the Edge for Experimental Fields

Edge KMeans Kernel-Learning for Plant Leaf Disease Detection. Plant diseases result in an alteration of the plant which modifies or interrupts its vital functions such as photosynthesis, transpiration, pollination, fertilization, germination, etc. Manifestations of the disease are usually seen on the leaves, fruits and stems of the plant. This can have a very big impact on the yield of the plant. In this use case we consider the context of an agronomist who wants to study a disease that manifests itself in the leaves automatically.

Edge AI-IoT Node, in this case embed a KMeans Kernel-Learning model to assist agronomic expert for plant leaf disease detection after a short number of interactions with the system. The principle of KMeans Kernel Learning consists in creating KMeans models trained with selected images (Kernel Images). The clusters resulting from these Kernel Images are called Kernel Clusters and are then labelled diseased zones or healthy zones [8]. This would help the expert extracting diseased area even if it is almost visible (Fig. 3).

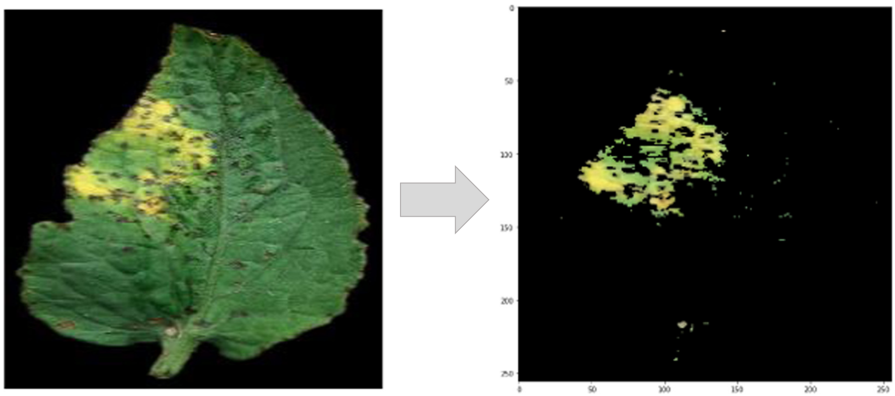

Fig. 3. KMeans Kernel learning clustering

More formally, considering I_{k0} a Kernel Image chosen we have the Eq. (1):

$$KMeans(I_{k0}) = \left\{\varphi_{k0}, \Omega_{k0}\right\} \tag{1}$$

Where φ_{k0} is the KMeans Kernel Model clustering model based on the I_{k0} kernel image and Ω_{k0} is the set of cluster centroids and their labels as a *healthy* or *disease* regions of the plant leaf. Ω_{k0} is defined by Eq. (2):

$$\Omega_{k0} = \left\{(\omega_{i,k0}, label)/i \in [0-3], label \in \left\{health, diseased\right\}\right\} \tag{2}$$

Where $\omega_{i,k0}$ is the centroid of the cluster number i (related to I_{k0}) and the label indicates if the cluster formed from this centroid belongs to a *diseased* region or *healthy*. We make the assumption that by taking the one cluster that contains most significant disease region we can make decision about the health of the plant leave. So only one cluster is labelled diseased and we always refer to it by $\omega_{2,k0}$. The framework uses Kernel Image I_{k0} which is supposed to have representative features of a diseased plant leaf. This Kernel Image is used to build a KMean Kernel Model φ_{k0} and Kernel Clusters $\omega_{i,k0}$; i $\in [0, 3]$. Each cluster can be labelled *healthy* or *diseased*. In our context we orient KMeans algorithm so the cluster that contains most of diseased region is always named $\omega_{2,k0}$. KMean Kernel Models are just classifiers based on KMean that have been trained with data $\mathbb{R}^3$ composed by Kernel Image pixels components (Fig. 4). We limited the number of clusters to 4 because we observed that the number of empty clusters increases when $K \geq 4$.

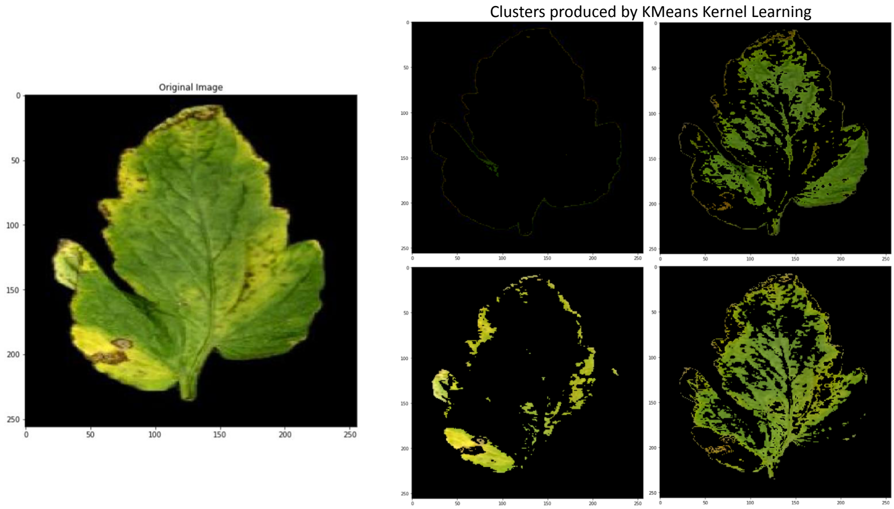

Fig. 4. Segmented plant leaf with four clusters

The KMean algorithm identifies the cluster containing the largest diseased part. Agronomic experts can at this stage tag a few clusters of a few plants to indicate which ones represent a diseased part. The system can present in the social network clusters such as in Fig. 4 so they can retag them if necessary.

As experimentation we use Plant Village DataSet [10] wich is composed of plant leaf images that are segmented. The aim was to design a model for plant leaf disease detection based on Kernel KMeans. We selected 1474 images of diseased plant leaves and 1129 images of healthy plant leaves. For the training/testing split we used 80%/20%. For implementation of Kernel KMeans we used popular Sci-kitLearn, Pandas, openCV and matplotlib.

The results of the test on multiple samples of plant leaf images is presented by the following table.

The model is fast and accurate without much help from experts. The precision is about 95% while accuracy is 93%. It exists machine learning models that are

	Precision	Recall	F1-score	Support
Diseased leaves	0.93	0.95	0.94	1474
Healthy leaves	0.93	0.90	0.91	1129

more accurate and the purpose was not to compete in term of accuracy. However it can be a decision support and monitoring tool for agronomic expert.

3.4 Use Case: Smart Irrigation Experimental Farm

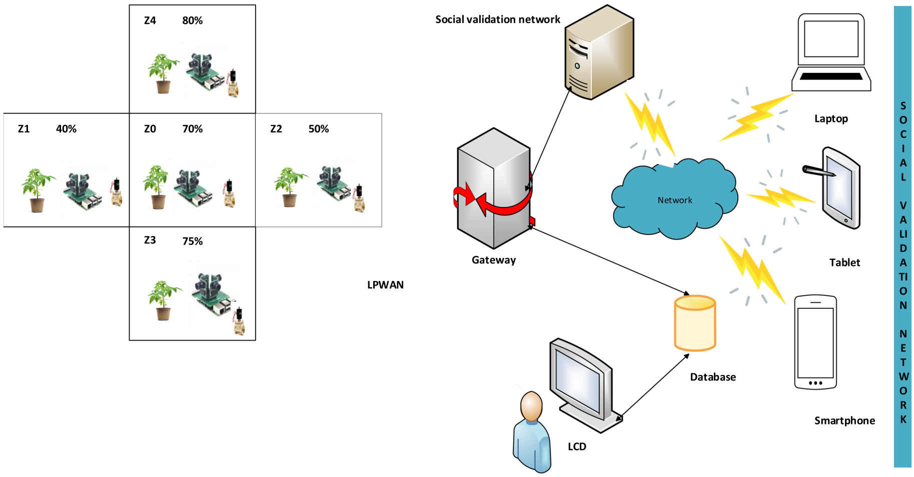

Fig. 5. Architecture for Edge AI-IoT network smart irrigation experimentation

Each plant has its ideal environmental conditions for instance for tomatoes soil moisture should be between 60% and 80%. To discover this kind of information, agronomic tests are done in specialized experimentation farms where different environment conditions can reproduced. Challenging task is to reproduce results for a large number of plant varieties in uncontrolled outdoor environment. To address this problem, as a second use case we propose that experimentation fields are divided into five numbered zones (Fig. 5): Z_0 to Z_4. In each zone, we have a sensors and actuators network that help to learn thresholds for ideal conditions. All zones are connected to one gateway. With this layout, system should learn four thresholds (Fig. 5):

– Z_0: this is the reference zone of our field with a soil moisture threshold that can ensure a good development of the plant. We will therefore compare this zone (Z_0) with the other zones (Z_1, Z_2, Z_3, Z_4) to see which zones have a humidity that favors or alters the appearance of the plant;

– Z_1: the minimum threshold that negatively affects the appearance of the plant with a humidity of 40% compared to Z_0;
– Z_2: in Z_2 the minimum moisture threshold that retains the same appearance of the plant as that found in the reference zone;
– Z_3: the limit threshold which makes it look better than that of the reference plant;
– Z_4: the optimal threshold which gives a very good appearance and a better qualitative transformation of the plant compared to Z_0;

Figure 6 below illustrates the result that our algorithm should provide after the experimental phase of studying our plant. The experimentation is done almost remotely.

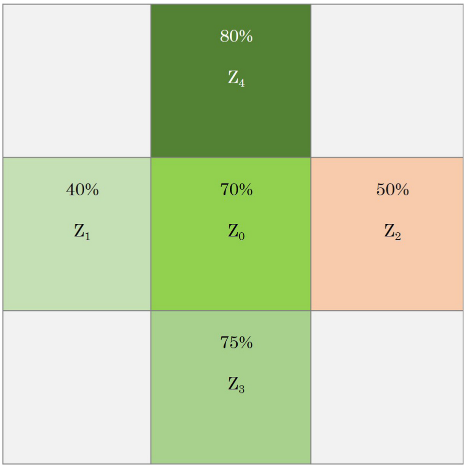

Fig. 6. Example of expected result in the case of tomato

In this use case, model is an algorithm executed by Edge node and that have double inputs: one from sensor network, other from social network of agronomic experts. The algorithm collects data from sensor network and makes decisions according to feedback events from experts in social network part. Indeed, when sensor network does an action (start watering for instance), after a while, it needs to get feedback from agronomic experts which are considered as oracles to tell if this action has positives effect or not. Algorithm 1 executes a main loop.

Algorithm 1: Gateway's Automated Irrigation System Control Algorithm

```
Data:
feedback : feedback of the social network about plant
Zone_i : Concerned Zone inside expermation zone
socialNetworkServerAddress : Address of Expert FrontEnd Server
h : local soil humidity/moisture
Result: Thresholds in Z1, Z2, Z3, Z4
initialization;
while true do
    picture ← getPicture(Zone_i);
    send(picture, Zone_i, socialNetworkServerAddress);
    feedback ← getFeedback(Zone_i, socialNetworkServerAddress);
    if feedback == state1 then
        if (Zone_i == Z1) or (Zone_i == Z2) then
            stopWatering(Zone_i);
        end
        if (Zone_i == Z3) or (Zone_i == Z4) then
            startWatering(Zone_i);
            wait(1 hour);
        end
    end
    if feedback == state2 then
        if Zone_i == Z1 then
            h ← h − 5%;
            sendToSocialNework(Zone_i, h);
        end
        if Zone_i == Z2 then
            stopWatering(Zone_i);
            h ← h − 5%;
            sendToSocialNework(Zone_i, h);
        end
        if (Zone_i == Z3)or(Zone_i == Z4) then
            sendToSocialNework(Zone_i , FAILURE);
            stopWatering(Zone_i);
        end
    end
    if feedback == state3 then
        if Zone_i == Z3 then
            sendToSocialNework(Zone_i , h);
        end
        if Zone_i == Z4 then
            stopWatering(Zone_i);
            wait(1 hour);
        end
    end
    if feedback == state4 then
        h ← h + 5%;
        sendToSocialNework(Zone_i , h);
    end
end
```

First step is the IoT node takes a picture of plant and send it to social network of experts. Agronomic experts give a feedback that is actually of an appreciation of the action of gateway regarding to plant development. The possible feedback events are:

– $State_1$- plant with the same appearance as the reference Z_0
– $State_2$- plant in a state of degradation in comparison to Z_0
– $State_3$- better appearance of the plant in comparison to Z_0
– $State_4$- Substantial improvement in the appearance of the plant in comparison to Z_0.

As an example, when gateway sends a picture of plants in all zones to experts' social network. Experts answer with feedbacks that are represented by Z_i and notifies to the gateway that all plants have the same aspect in all zones ($state_1$). This is to say that the plants have the same appearance in comparison to Z_0 and thus watering is stopped in zones Z_1 and Z_2, but continues Z_3 and Z_4.

4 Conclusion and Future Works

In this paper we proposed an Edge AI-IoT framework for experimental agriculture that we call eFarm-Lab. The general principle is that the framework is designed to allow, on the fly machine learning model training and deployment of models on Edge nodes to asssist agronomic experts in their experimental labs. eFarm-Lab is composed of three kind of nodes: IoT (sensors and actuators) network; a set of Edge-AI-IoT nodes implementing machine learning models; and finally Cloud architecture. A social network of agronomic experts as oracles can help labelling data and enhance quality of learning. We exhibited two use cases where this framework can be deployed for agronomic experimentation. The first is related to plant leaf disease detection and we implemented it to show the proof of concept. And finally the smart irrigation use case to illustrate how the social network of expert can be used to enhance remote testing.

The next step is to deploy a real testbed to see the behavior of deploying an Edge node with controllers encapsulating machine learning models.

References

1. Amandeep et al.: Smart farming using IOT. In: 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, pp. 278–280 (2017). https://doi.org/10.1109/IEMCON.2017.8117219
2. Angelopoulos, C.M., Filios, G., Nikoletseas, S., Raptis, T.P.: Keeping data at the edge of smart irrigation networks: a case study in strawberry greenhouses. Comput. Netw., 107039 (2019). https://doi.org/10.1016/j.comnet.2019.107039
3. Anjana, S., Sahana, M.N., Ankith, S., Natarajan, K., Shobha, K.R.: An IoT based 6LoWPAN enabled experiment for water management. In: IEEE ANTS 2015 1570192963, Bangalore, India, pp. 1–6 (2015)

4. Ashok, S., Kishore, G., Rajesh, V., Suchitra, S., Sophia, S.G.G., Pavithra, B.: Tomato Leaf disease detection using deep learning techniques. In: 2020 5th IEEE, International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 979–983 (2020)
5. Boursianis, A.D., Papadopoulou, M.S., et al.: Internet of Things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: a comprehensive review. Internet Things, 100187 (2020). https://doi.org/10.1016/j.iot.2020.100187
6. Bu, F., Wang, X.: A smart agriculture IoT system based on deep reinforcement learning. Futur. Gener. Comput. Syst. (2019). https://doi.org/10.1016/j.future.2019.04.041
7. Calo, S.B., Touna, M., Verma, D.C., Cullen, A.: Edge computing architecture for applying AI to IoT. In: 2017 IEEE International Conference on Big Data (Big Data) (2017). https://doi.org/10.1109/bigdata.2017.8258272
8. Gueye, Y., Mbaye, M., et al.: KMeans Kernel-learning based AI-IoT framework for plant leaf disease detection. In: Hacid, H. (ed.) ICSOC 2020. LNCS, vol. 12632, pp. 549–563. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-76352-7_49
9. Mekala, M.S., Viswanathan, P.: A novel technology for smart agriculture based on IoT with cloud computing. In: 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, pp. 75–82 (2017). https://doi.org/10.1109/I-SMAC.2017.8058280
10. Pandian, J.A., Geetharamani, G.: Data for: identification of plant leaf diseases using a 9-layer deep convolutional neural network. Mendeley Data, V1 (2019). https://doi.org/10.17632/tywbtsjrjv.1
11. Prathibha, SR., Hongal, A.: IoT based monitoring system in smart agriculture. In: 2017 International Conference on Recent Advances in Electronics and Communication Technology, pp. 81–84 (2017)
12. Ahmed, I.T., Rahman, M.K.: IoT based autonomous percipient irrigation system using Raspberry Pi. In: 19th International Conference on Computer and Information Technology, North South University, Dhaka, Bangladesh, 18–20 December 2016, pp. 563–568 (2016)
13. Ran, X., Chen, H., Zhu, X., Liu, Z., Chen, J.: DeepDecision: a mobile deep learning framework for edge video analytics. In: 2018 IEEE Conference on Computer Communications (INFOCOM 2018), pp. 1421–1429 (2018)
14. Roopaei, M., Rad, P., Choo, K.-K.R.: Cloud of things in smart agriculture: intelligent irrigation monitoring by thermal imaging. IEEE Cloud Comput. **4**(1), 10–15 (2017). https://doi.org/10.1109/mcc.2017.5
15. Sarjerao, R.K.: a low cost smart irrigation system using MQTT protocol. In: 2017 IEEE Region 10 Symposium (TENSYMP) (2017)
16. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016). https://doi.org/10.1109/jiot.2016.2579198
17. Huang, Y., Ma, X., Fan, X., et al.: When deep learning meets edge computing. In: IEEE 25th International Conference on Network Protocols (ICNP 2017), pp. 1–2 (2017)