



**HAL**  
open science

# **Precision Agriculture Deep Neural Network Driven Multi-hop Plant Image Noisy Data Transmission and Plant Disease Detection**

Derek Kwaku Pobi Asiedu, Kwadwo Boateng Ofori-Amanfo, Kwabena Ebo Bennin, Mustapha Benjillali, Kyoung-Jae Lee, Dennis Agyemanh Nana Gookyi, Samir Saoudi

## ► To cite this version:

Derek Kwaku Pobi Asiedu, Kwadwo Boateng Ofori-Amanfo, Kwabena Ebo Bennin, Mustapha Benjillali, Kyoung-Jae Lee, et al.. Precision Agriculture Deep Neural Network Driven Multi-hop Plant Image Noisy Data Transmission and Plant Disease Detection. ISIVC 2024: IEEE International Conference on Signal, Image, Video and Communications, May 2024, Marrakech, Morocco. <10.1109/ISIVC61350.2024.10577918>. <hal-04600029>

**HAL Id: hal-04600029**

**<https://imt-atlantique.hal.science/hal-04600029v1>**

Submitted on 4 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Precision Agriculture Deep Neural Network Driven Multi-hop Plant Image Noisy Data Transmission and Plant Disease Detection

Derek Kwaku Pobi Asiedu\*, Kwadwo Boateng Ofori-Amanfo<sup>†</sup>, Kwabena Ebo Bennin<sup>‡</sup>, Mustapha Benjillali<sup>§\*</sup>, Kyoung-Jae Lee<sup>†</sup>, Dennis Agyemanh Nana Gookyi<sup>¶</sup>, and Samir Saoudi\*

\*IMT-Atlantique, Lab-STICC, Brest, France. Emails: kwakupobi@ieee.org, and samir.saoudi@imt-atlantique.fr

<sup>†</sup> Electronic Engineering Department, HBNU, Daejeon, South Korea. Email: gigaofori11@gmail.com, kyoungjae@hanbat.ac.kr

<sup>‡</sup> Information Technology Group, WUR, 6706 KN, The Netherlands. Email: kwabena.bennin@wur.nl

<sup>§</sup> Communication Systems Department, INPT, Rabat, Morocco. Email: benjillali@ieee.org

<sup>¶</sup> Institute for Scientific and Technological Information, CSIR, Accra, Ghana. Email: dennisgookyi@gmail.com

**Abstract**—Under precision agriculture (PA), plant disease detection (PDD) is imperative regarding farm crops’ life quality and crop yield. However, the data captures for PDD is influenced by the noisy data captured by farm sensors due to wireless noisy transmission channels. Hence, this work considers the onsite or offsite (remote) farm PDD through onsite farm monitoring PA sensor networks (PAN). Here, effects on captured sensor image (plant leaf image) data transmitted through the PAN to an PDD application are studied. Where both traditional decode-and-forward (DF) data routing and channel-effect considering machine learning data autoencoder routing are used for image data transmission. In addition, a PDD deep learning algorithm is developed to predict whether or not a farm plant is diseased, based on the noisy image data captured by the PAN through data routing. From the PAN-PDD simulation, the proposed ML PAN-PDD algorithm showed fair performance over the DF PAN-PDD.

**Index Terms**—Deep learning (DL), multi-hop wireless sensor network (WSN), plant disease detection (PDD), convolutional neural network (CNN), autoencoder.

## I. INTRODUCTION

With the advent of the Internet-of-Things (IoT) and Machine Learning (ML), intelligent entities such as smart homes, industry, agriculture and applications have been developed and integrated into society [1]–[3]. The application of IoT networks and ML in “*smart*” agriculture is termed precision agriculture (PA), and the development of its technology falls under the coined term Farming 4.0 [1], [2], [4]. Farming 4.0 covers farm monitoring (e.g., soil nutrients, soil moisture, plant and livestock monitoring and pests monitoring), intelligent inference and protocol execution (e.g., plant and livestock disease detection, production planning and farm machinery control). Farming 4.0 seeks to reduce the impact of farming

on the climate, make farming more efficient, and to satisfy specific United Nations (UN) sustainable development goals (SDGs) set for or associated with modernized farming. The aim is to reduce farming cost and operation, human labor, increase availability of quality food stuffs and improve farming culture in modern farming. However, the development of the PA sensor networks (PANs) used for farm monitoring and mechanization, and the intelligent farm applications (e.g., software and mobile apps) still need to be improved for efficiency and satisfy the UN SDGs [2].

Currently, the use of massive machine type communications (mMTC) in PAN for farm data acquisition and task execution is growing in academic and industrial research [1], [2], [5]. In addition, ML deep neural networks (DNNs) have seen increasing use in PA for intelligent farm monitoring, and process automation [3]. mMTC PANs and DNNs have been increasingly used for crop/plant disease detection (PDD) applications in Farming 4.0 research and development, with the goal of improving crop production and life quality [6], [7]. Limitingly, the PAN and DNN-PDD have been considered and studied separately in their research and development stages [4], [5], [7], [8]. This approach does not factor in the significant influence both systems (PAN and DNN PDD) have on each other. For example, the PAN acquires and transfers noisy data during data routing due to the farm environment (i.e., the wireless communication channel structure), and sensor hardware and operational imperfections. This PAN data transmission effect needs to be factored into the images used in training the DNN-PDD applications currently being developed. This is because, current DNN-PDD applications are trained using pixel perfect, clean and ideal images, which are not as practical as actual PAN noisy data. In addition, with mMTC devices, the amount of bits and processing power available at the transmitting and routing nodes are limited. Hence, processed image sizes must be reduced to allow transmission through farm mMTC devices, or are transmitted in concurrent packets through the PAN. This puts a toll on the PAN network, reducing its efficiency and

The EU’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 8995462 funded this work. D. Gookyi’s work was supported by the UNESCO-TWAS programme “Seed Grant for African Principal Investigators” financed by the German Federal Ministry of Education and Research (BMBF) (TWAS-SG-NAPI-4500474961).

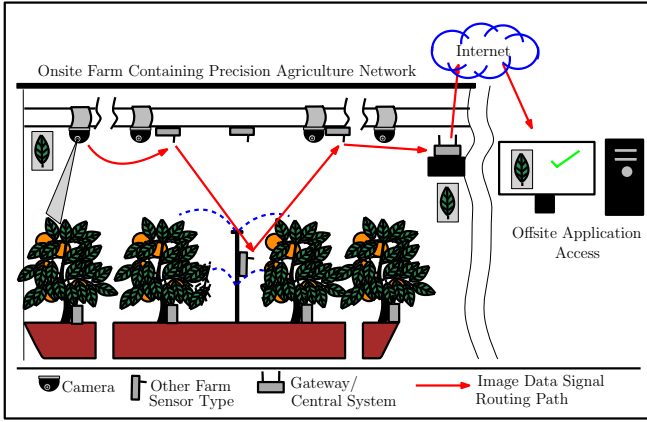


Fig. 1. On- and offsite farm deep learning PAN architecture.

network lifespan. As a result, there is a reduction in the quality of images processed and available for PDD. Impractical in real world implementations, the current research and development into DNN-PDD work hinges on the assumption capturing and acquiring perfect onsite farm data. Therefore, leading to farm deployed PDD applications having poor performance [9], [10].

*Motivation:* To highlight the influence of the PAN on PDD, this work proposes the deployment of an onsite (farm) DNN PAN using an autoencoder, which factors in noisy data (image data) transmission. The proposed DNN PAN is combined with an DNN PDD algorithm, which mimics practical PAN-PDD implementation and interaction. This proposed PAN-PDD approach or implementation can aid in further investigations and development of smart farms monitoring and mechanization systems. Note, from our reviews, this practical PAN-PDD scenario has not been explored in existing research.

*Contribution:* The contribution of this work are in two folds.

- i To counteract the negative effect of data size (crop image) and distortions, an autoencoder transceiver design are proposed to be implemented at each sensor node within the PAN. The autoencoder aids in data size reduction for transmission through the limited resourced routing sensor nodes within the PAN to the on/offsite DNN PDD<sup>1</sup>.
- ii The proposed DNN PDD training, testing, validation, and predictions are based on noisy data (crop images), unlike majority of current research which use perfect images for their DNN PDD training, testing, and validation.

Finally, through comparison of the proposed DNN PAN-PDD technique (MDF) to the transitional decode-and-forward (DF) routing PAN with PDD implementation (TDF) showed the superiority of the proposed MDF over the TDF benchmark.

## II. PRECISION FARMING SYSTEM MODEL

This work investigates an on/offsite farm DNN PAN-PDD algorithms development based on acquired noisy data (crop image) transmitted from an onsite farm PAN as shown in Fig. 1. Specifically, the system model consists of (i) onsite

<sup>1</sup>Here, Onsite DNN PDD refers to a centralized mission center build on the farm for smart farming. While the on/offsite DNN PDD refers to a mobile application accessed by the farmer for remote smart farming.

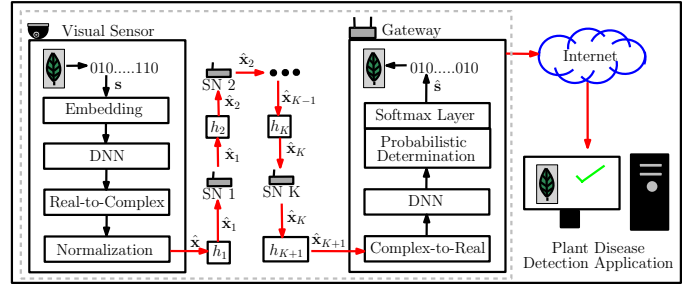


Fig. 2. Autoencoder multihop PAN communication model.

data acquisition and processing using autoencoder transceiver sensor nodes, (ii) multi-hop data through the PAN to the gateway/central system (GW) for further processing and transfer via the internet to the offsite PDD application, and (iii) predicting whether or not a plant is diseased using the offsite PDD application. The PAN, data transfer and data processing considered in this work are shown in Fig. 1. From Fig. 1, within the proposed onsite farm PAN, the crop (maize) data (leaf image) is captured using an onsite visual sensor (camera) (VS). The acquired data is pre-processed using an autoencoder (i.e., the encoder: transmission data size reduction and multi-hop transmission effects compensation<sup>2</sup>) and transmitted through the farm monitoring sensor nodes (SNs) within the PAN to the PAN GW. The GW contains the other portion of the autoencoder (i.e., the decoder portion, which estimates the captured original plant leaf image), which processes the received plant image data. The GW then transmits the recovered data through an external network (internet+cloud) to the on/offsite PDD application to predict whether or not the considered plant is diseased. The PAN consists of sensor nodes (SNs) with limited computation and communication resources. Therefore, reducing the data size with an autoencoder minimizes the strain on the PAN multi-hop SNs. Next, the multi-hop PAN autoencoder design is discussed. This work focused on PAN as a specific application of the proposed cascaded autoencoder nodes. However, the proposed DNN PAN is applicable to other wireless sensor network for noisy data reduction.

## III. ONSITE FARM MULTI-HOP PAN TRANSMISSION

This section discusses the physical layer of the onsite farm PAN data acquisition, processing, and transmission to the offsite PDD application. As shown in Fig. 1, the VS captures the farm plant (leaf) image, processes the data, and transmits the data through the  $K \geq 0^3$  PAN multihop SNs to the gateway. Each PAN SN is equipped with a single antenna. Also, multihop is done by using either an DNN (autoencoder [11], [12] as done for TinyML implementations)

<sup>2</sup>The compensation here refers to consideration of the multi-hop cascading channel gains (noise) within the autoencoder training phase.

<sup>3</sup>Depending on the selected optimal multihop path and VS location;  $K = 0$  or  $K \geq 1$  for VS-to-GW direct or multi-hop transmission, respectively.

or the transitional decode-and-forward technique (DF) [13]<sup>4</sup>. In this work, it is assumed that the multihop SNs and path from the VS to the GW are already known.

#### A. Signal Flow for Autoencoder PAN

The data processing using the autoencoder ML approach from the transmitter through the PAN multi-hop routing nodes to the onsite gateway receiver are detailed as follows.

**Transmitter (Encoder Portion):** At the VS, the captured image<sup>5</sup> goes through the following autoencoder (encoder portion) steps before transmitting through the SNs<sup>6</sup>.

- Image data processing (Embedding stage): the image is converted to binary format ( $\mathbf{s}$ ) data stream (i.e., the embedding phase: conversion from 3D pixel matrix ( $\mathbf{S}$ ) to 2D decimal matrix ( $\tilde{\mathbf{S}}$ ) and then 1D bits vector ( $\mathbf{s}$ )). Hence, the image data transformation is  $f : \mathbf{S} \rightarrow \mathbf{s}$ .
- DNN data processing: the  $\mathbf{s}$  data vector is fed to the DNN for data stream dimension reduction. Here, the DNN performs the dimension reduction using a symbol mapping (i.e., number of bits per symbol ( $k$ )) to match a chosen modulation scheme (e.g. QPSK ( $2^k, k = 2$ ), 16QAM ( $2^k, k = 4$ )). In detail, the image bits ( $\mathbf{s}$ ) is converted to a set of symbols  $\mathcal{M}$ , where each symbol consists of  $2^k$  bits based on the modulation scheme and each symbol  $\hat{S}$  is represented by a value in the set  $\{1, 2, \dots, M\}$ . Now, for each symbol to be transmitted, the symbol message is transformed to an  $\hat{\mathbf{s}} \in \mathbb{R}^n$  of size  $n$ , where  $n$  equals the size of transmit signal and number of data stream  $\hat{\mathbf{s}}$  accessed channels. In summary, the encoder portion does the dimension reduction  $f : \mathbf{S} \rightarrow \tilde{\mathbf{S}} \rightarrow \mathbf{s} \rightarrow \hat{\mathbf{S}} \in \{1, 2, \dots, M\} \rightarrow \hat{\mathbf{s}} \in \mathbb{R}^n$ .
- Transmit data normalization: the resulting data stream is converted to a complex-valued data set (i.e., equal portions of real and imaginary components). Finally, the transmit data is normalized to satisfy the transmit power constraint, that is,  $\hat{\mathbf{x}}_0 = \|\hat{\mathbf{s}}\|^2 \leq P_{VS}$ .

**PAN multihop:** The VS data stream is transmitted through  $G$  SNs to the GW for final decoding and transmission to the offsite unit. The data received at a multihop SN is deduced as

$$\hat{\mathbf{x}}_g = h_g \hat{\mathbf{x}}_{g-1} + \phi_g, g = 1, 2, \dots, G, \quad (1)$$

where  $h_g$  represents the inter-node channel (modelled as Additive White Gaussian Noise (AWGN) channels) and  $\phi_g$  represents the antenna noise at SN  $g$ . Each SN uses either the MDF (autoencoder) to decode and re-transmit information. The received signal at the GW is given as

$$\hat{\mathbf{x}}_{G+1} = h_{G+1} \hat{\mathbf{x}}_G + \phi_{G+1}. \quad (2)$$

**Receiver (GW):** At the GW, the decoder portion of the autoencoder is implemented as follows.

<sup>4</sup>The DF technique involve each relaying node decoding the its received data signal before transmitting the decoded data to the next relaying node. TinyML is a type of ML implementation that allows models to run on smaller and low power devices such as farm sensors.

<sup>5</sup>In this proof of concept work, leaf images from PlantVillage was use.

<sup>6</sup>The autoencoder in this work was based on the modification of the wireless communication autoencoder model proposed and used in [11], [12], [14].

- The received complex data is concatenated (real and imaginary components,  $\hat{\mathbf{x}}_{G+1} \rightarrow \tilde{\mathbf{x}}$ ), and the real value components are passed to the DNN.
- DNN data processing: the reverse of the DNN in the encoder occurs here, where the  $n$  data stream vector is transformed to an estimated symbol of size  $2^k$  bits using softmax function (softmax layer) ( $\tilde{\mathbf{x}} \rightarrow \tilde{\mathbf{s}}$ ).
- Image recovery: the estimated 1D image bit vector  $\tilde{\mathbf{s}}$  is reshaped to the estimated 2D decimal matrix ( $\tilde{\tilde{\mathbf{S}}}$ ), then to the estimated 3D pixel matrix ( $\tilde{\mathbf{S}}$ ). The total decoding process is presented as  $f : \tilde{\mathbf{x}} \in \mathbb{R}^n \rightarrow \tilde{\mathbf{S}} \in \{1, 2, \dots, M\} \rightarrow \tilde{\mathbf{s}} \rightarrow \tilde{\tilde{\mathbf{S}}} \rightarrow \tilde{\mathbf{S}}$ .

The recovered image data is then transmitted over the internet to the the DNN PDD offsite system for disease detection predictions. Note, the training of the autoencoder involves the use of random generated bits as input and outputs for the training and validation sets of the autoencoder [11], [12]. In addition, the autoencoder training and validation incorporated the channel characteristics into the learning process.

#### B. Signal Flow for Decode-and-Forward PAN (Benchmark)

A summary of the signal flow for traditional DF routing PAN used in this work is presented as follows<sup>7</sup>.

**Transmitter:** The image data processing portion for the DF technique is same as that of the encoder approach. Similarly the 1D bits data ( $\mathbf{s}$ ) is then modulated using the chosen modulation scheme ( $\hat{\mathbf{s}}$ ), and transmitted over several/single packet depending on the device coherence time allowable packetsize. Note,  $\hat{\mathbf{x}}_0 = \|\hat{\mathbf{s}}\|^2 \leq P_{VS}$  must be satisfied.

**PAN multihop:** Transmission through the  $G$  SNs is achieved through the traditional DF (TDF) approach, where the received data (equation (1)) as SN  $g$  is decode and then decoded information is re-transmitted (equation (2)) to SN $_g$  or GW.

**Receiver (GW):** At the GW, the received data is demodulated, and the image data recovered using the same encoder image recovery process. Next, the DNN PDD is presented.

## IV. PLANT DISEASE DETECTION USING CNN

To test and affirm our noisy data theory and proposed cascaded encoder design for PAN-PDD, the acquired data must be used in PDD training. Hence, we briefly discuss the PDD algorithm used in this confirmation.

The high precision and widely used VGG16 Convolutional Neural Network (CNN) is adopted in this work for the PDD. The architecture of the VGG16 is presented in Fig. 3. The VGG16 consists of the typical CNN model which has three layers, namely, convolutional layer, a pooling layer, and a fully connected layer [16]. The VGG16 is adopted because of its high precision in PDD as shown in [17].

<sup>7</sup>An argument can be made for the VS (Visual sensor) having the capability to run the PDD algorithm (via TinyML) and rather forward the prediction data. However, this may require additional hardware and programming inclusion to the VS devices. This will reduced in increased purchase cost of devices an operational cost (VS node battery consumption, replacement, and lifespan) which is proportional to the number of VS devices needed for the farm. Also, this defeats the purpose of using mMTC devices for smart precision farming equipped with 5G and beyond technologies.

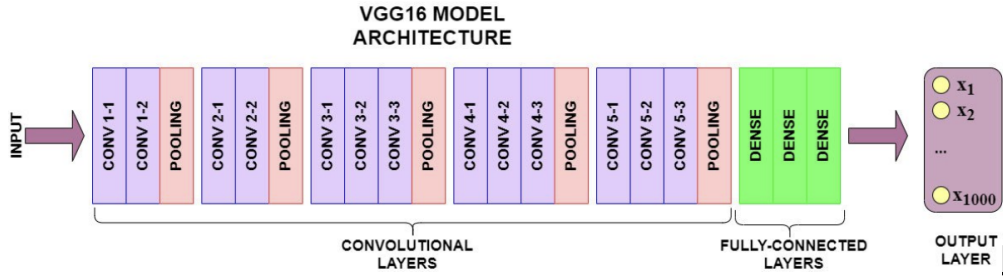


Fig. 3. Architecture of the VGG16 model [15]

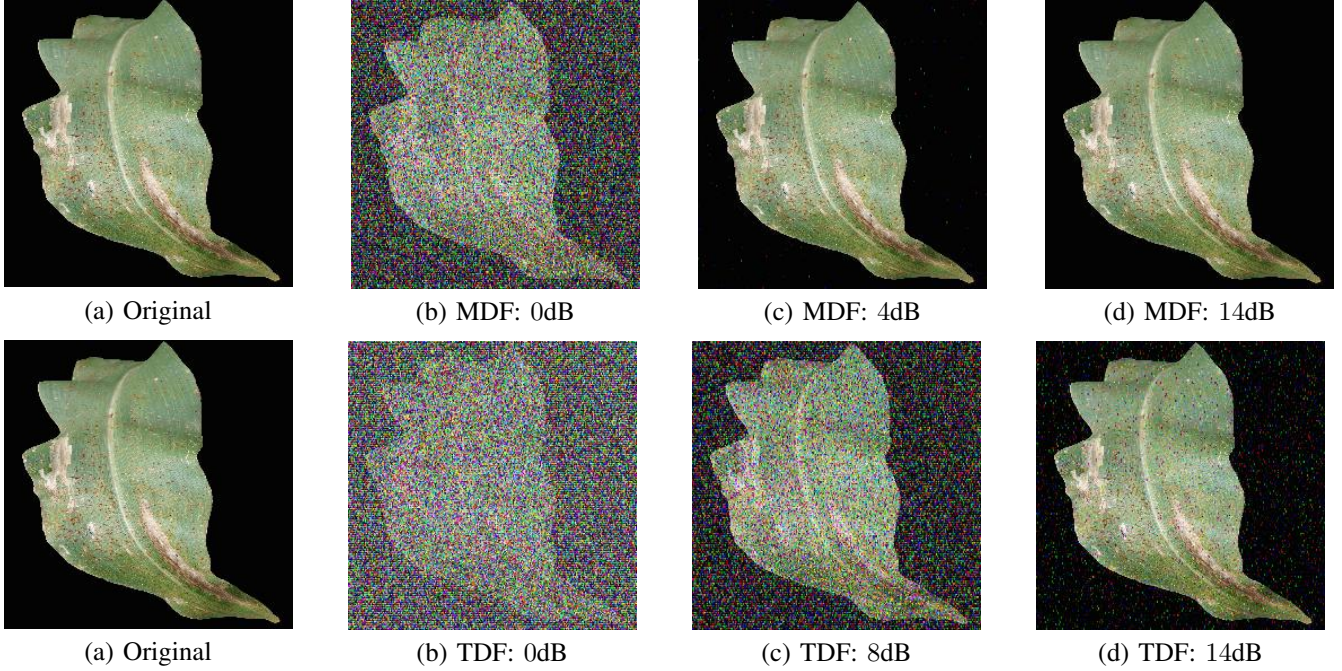


Fig. 4. Example of the original and recovered images for increasing signal-to-noise transmit power values.

TABLE I  
DNN PAN SIMULATION SET-UP.

Parameter	Value
Inter-node SNR $E_0$	0 dB to 14 dB
Channel model	AWGN
{relays, bits, channels} number	{ $G = 2, k = 2, n = 2$ }
Number of plant images	100
Plant disease	maize common rust

## V. SYSTEM EVALUATION

### A. Onsite Farm Multihop PAN Simulation

1) *Simulation Setup*: PAN simulation set-up is summarized in Table I. Details of the autoencoder implementation adopted for the data routing follows a similar structure as those found in [11], [12], [14]. The plant (corn) images used for the PAN and DNN PDD simulations were acquired from “PlantVillage” [18]. From the PAN images transmission simulations, we acquired 1600 unhealthy and 1600 healthy noisy plant images. Next the results (augmented images) from the PAN simulation are presented. To discuss data quality, the Block Error Rate (BLER) is chosen as the quality-of-service. In addition to the TDF benchmarks, two upper bound benchmarks, namely, DF coded multi-hop simulation (Coded DF) and its analytical

version (DF QPSK) are presented in the BLER plots. This is to evaluate the performance of the MDF approach, as well as, affirm the simulation is right. Each BLER plot was acquired using  $10^2$  iterations (experiments).

2) *Simulation Discussion*: Fig. 4 shows the original “transmitted” image as well as the images received and “recovered” at the GW for both the MDF and TDF approaches. It can be seen from the recovered images that the MDF approach gives better image results compared to the TDF approach. This is because the MDF approach is based on a trained ML considering the channel effect. However, this does not apply to the TDF approach. Hence, the influence of the channel is not considered in the TDF approach. In addition, the TDF yields distorted (noise) recovered images, compared to the MDF approach which yields near perfect images (marginal noisy images compared to the original). However, both schemes (TDF and MDF) show very noisy and similar recovered images at the very low SNR value of 0dB. However, at higher SNR values from 8dB and 4dB less noisy images are recovered for the TDF and MDF approaches, respectively. This influence is further evident in the BLER against increasing transmit SNR simulation presented in Fig. 5. Here ((2, 2)) represents

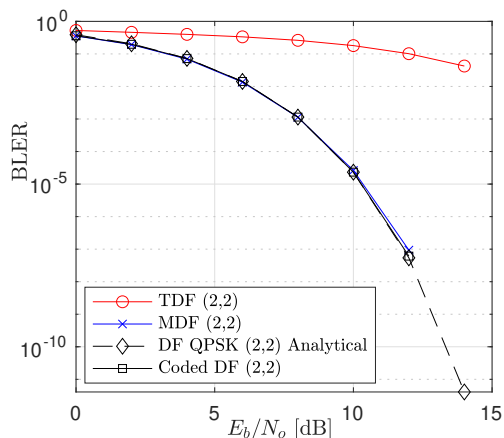


Fig. 5. Multi-hop PAN BLER vs SNR with  $K = 2$ .

TABLE II  
OVERVIEW OF THE TRAINING AND TESTING DATASETS.

Dataset	Healthy	Diseased
Corn (Default, TDF, MDF)	(800,800,800)	(800,800,800)
Total	2400	2400

(( $n = 2, k = 2$ )). From Fig. 5, it is observed that there is a large BLER performance gap between the MDF (better) and the TDF. However, the MDF has similar (close) performance to coded DF and its analytical results. As mentioned earlier due to the channel influence factored into the MDL training.

Next, the influence of increasing the number of routing hops (Multi-hops) has on the BLER in Fig. . It is observed that the BLER increases with increasing hops for all curves. This observation is due to the fact that as the number of routing hops increases the amount of channel effect (data noisy effect) increases, reducing the BLER. As Expected, the MDF outperforms the TDF approach, and marginally similar to the analytical benchmarks (DF QPSK and the Coded DF). This influence of noisy data is observed in the sample image data set for increasing routing hops (multi-hop ( $K$ )) in Fig. . Here, the MDF retains good image quality with marginal distortions while the TDF has images with increased distortions.

From our simulation results, it is evident that the quality of data acquired for processing and learning in precision farming needs to be improved and machine learning in communication is a viable option to mitigate this issue. Hence, by implementing an SN autoencoder within PAN yields better performance.

## B. Plant Disease Detection

1) *Dataset*: To assess the impact of PAN on the performance of PDD models, a sample dataset from PlantVillage [18], [19] is extracted and used as discussed above. The Corn dataset contains plant images which are essential for training an effective PDD model. The PlantVillage dataset is chosen because it is the most used open source plant disease dataset. The sizes and distribution of the datasets is found in Table II.

2) *Simulation Setup*: The DNN PDD experiments were run on a Apple MacBook computer with an M1 Max processor and

TABLE III  
OVERVIEW OF VGG16 HYPERPARAMETERS.

VGG16	Hyperparameters
Input size	(224,224,3)
(Batch size, Learning rate)	(32, 0.0002)
(Number of epochs, Optimizer)	(15, Adam)

TABLE IV  
VGG16 RESULTS ON PAN GENERATED DATASETS.

Dataset	Accuracy	Precision	Recall	$F1$ -score
Default	1.000	1.000	1.000	1.000
TDF	0.9313	0.8791	0.100	0.9357
MDF	0.9462	0.8939	0.100	0.944

32GB RAM. Microsoft Visual Studio [20] operating Python 3.9 was used to run the model. For each dataset, an 80/20 split is made where the 20% is used for testing and the 80% is further split into 80% training and 20% validation sets. We used the transfer learning model (VGG16 model) from Keras [21] for the DNN PDD development. We utilized the VGG16 model to classify the images. We used transfer learning where we fine tune the VGG16 model for plant disease detection using the Plant Village dataset. The fully connected layers of the model were removed and two layers were frozen during the training, switching from trainable to non-trainable. Table III shows the hyperparameters used for the experiments.

3) *Simulation Discussion*: The results show that for all three datasets the transfer learning VGG16 model is able to distinguish the diseased images from the healthy images. Also, the DNN PDD performance for the MDL images is marginally better compared to the TDF images. The default represents the DNN PDD training on the original non-augmented (no multihop PAN) images, which shows superior performance. Note, the default had perfect results because the ideal (original) images were used, which are not practical images.

## VI. CONCLUSION

This paper presented work on onsite farm multihop PAN using traditional (TDF) and ML (MDF) decode-and-forward approaches and farm offsite deep neural network (DNN) plant disease detection (PDD) implementation. It was shown that the MDF approach is better compared to the TDF approach, and better image acquisition for DNN PPD offsite. However, for the DNN PDD execution, the MDF and TDF had similar prediction values. The next research stage involves designing and incorporating both a spectrum- and energy-efficient routing protocol based on traditional and ML approaches. In addition, other modulation schemes will be considered for the system model. Finally, physical implementation will be done.

## REFERENCES

- [1] Sofia Polymeni, Stefanos Plastras, Dimitrios N Skoutas, Georgios Kormentzas, and Charalabos Skianis, "The impact of 6G-IoT technologies on the development of agriculture 5.0: A review," *Electronics*, vol. 12, no. 12, pp. 2651, Jun. 2023.
- [2] Yu Tang, Sathian Dananjayan, Chaojun Hou, Qiwei Guo, Shaoming Luo, and Yong He, "A survey on the 5G network and its impact on agriculture: Challenges and opportunities," *Computers and Electronics in Agriculture*, vol. 180, pp. 105895, Jan. 2021.

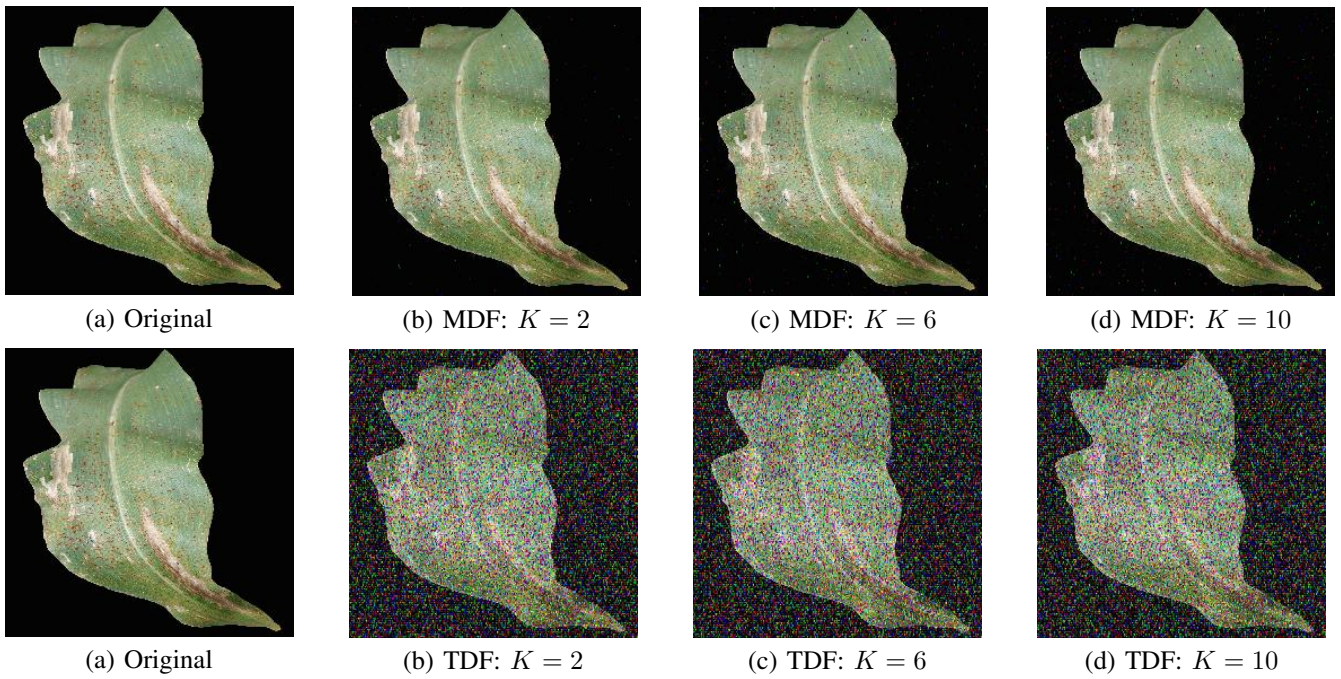


Fig. 6. Example of the original and recovered images for increasing number of routing multi-hops values.

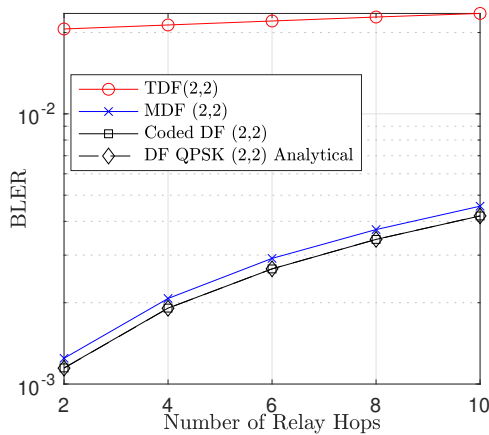


Fig. 7. Multi-hop PAN BLER vs  $K$  with  $E_b/N_o = 8\text{dB}$ .

[3] Muhammad K Shehzad, Luca Rose, M Majid Butt, István Z Kovács, Mohamad Assaad, and Mohsen Guizani, "Artificial intelligence for 6G networks: Technology advancement and standardization," *IEEE Veh. Technol. Mag.*, vol. 17, no. 3, pp. 16–25, May 2022.

[4] Sharada P Mohanty, David P Hughes, and Marcel Salathe, "Using deep learning for image-based plant disease detection," *Frontiers in plant science*, vol. 7, pp. 1419, Sep. 2016.

[5] Muhammad Anas Khan, Adeeb Khan, Mohammed Abuibaid, and Jun Steed Huang, "Harnessing 5G networks for enhanced precision agriculture: Challenges and potential solutions," in *Proc. Int. Conf. Smart Applications, Commun. Network.*, Jul. 2023, pp. 1–6.

[6] Michael James Roberts, *The value of plant disease early-warning systems: A case study of USDA's soybean rust coordinated framework*, Number 18. USDA Economic Research Service, 2006.

[7] Amreen Abbas, Sweta Jain, Mahesh Gour, and Swetha Vankudothu, "Tomato plant disease detection using transfer learning with C-GAN synthetic images," *Computers and Electronics in Agriculture*, vol. 187, pp. 106279, Aug. 2021.

[8] Byeongjun Min, Taehyun Kim, Dongil Shin, and Dongkyoo Shin, "Data augmentation method for plant leaf disease recognition," *Applied Sciences*, vol. 13, no. 3, pp. 1465, Jan. 2023.

[9] Mobeen Ahmad, Muhammad Abdullah, Hyeonjoon Moon, and Dongil Han, "Plant disease detection in imbalanced datasets using efficient convolutional neural networks with stepwise transfer learning," *IEEE Access*, vol. 9, pp. 140565–140580, Oct. 2021.

[10] Artzai Picon, Aitor Alvarez-Gila, Maximilian Seitz, Amaia Ortiz-Barredo, Jone Echazarra, and Alexander Johannes, "Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild," *Computers and Electronics in Agriculture*, vol. 161, pp. 280–290, Jun. 2019.

[11] Sebastian Dörner, Sebastian Cammerer, Jakob Hoydis, and Stephan Ten Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Dec. 2017.

[12] Timothy O'shea and Jakob Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cog. Commun. Network.*, vol. 3, no. 4, pp. 563–575, Oct. 2017.

[13] Derek Kwaku Pobi Asiedu, Hoon Lee, and Kyoung-Jae Lee, "Simultaneous wireless information and power transfer for decode-and-forward multihop relay systems in energy-constrained IoT networks," *IEEE Internet of Things J.*, vol. 6, no. 6, pp. 9413–9426, Aug. 2019.

[14] MathWorks R2023a, "Matlab documentation: Autoencoders for wireless communications," <https://fr.mathworks.com/help/comm/ug/autoencoders-for-wireless-communications.html>, MathWorks, Date Accessed:, 18 Aug. 2023.

[15] J McDermott, "Hands-on transfer learning with keras and the VGG16 model," 2021.

[16] Jinzhu Lu, Lijuan Tan, and Huanyu Jiang, "Review on convolutional neural network (CNN) applied to plant leaf disease classification," *Agriculture*, vol. 11, no. 8, pp. 707, 2021.

[17] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[18] MathWorks R2023a, "Data for: Identification of plant leaf diseases using a 9-layer deep convolutional neural network," <https://data.mendeley.com/datasets/tywbtsjrjy/1>, PlantVillage, Date Accessed:, 18 Aug. 2023.

[19] David Hughes and Marcel Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv:1511.08060*, 2015.

[20] Microsoft Visio, "Visual studio code: Code editing redefined," <https://code.visualstudio.com/>, PlantVillage, Date Accessed:, 18 Aug. 2023.

[21] Microsoft Visio, "Keras: Simple flexible powerful," <https://keras.io/>, PlantVillage, Date Accessed:, 18 Aug. 2023.