World Academy of Science, Engineering and Technology
International Journal of Information and Communication Engineering
Vol:14, No:9, 2020

Discussing Embedded versus Central Machine Learning in Wireless Sensor Networks

Anne-Lena Kampen, Øivind Kure

***Abstract*—**Machine learning (ML) can be implemented in Wireless Sensor Networks (WSNs) as a central solution or distributed solution where the ML is embedded in the nodes. Embedding improves privacy and may reduce prediction delay. In addition, the number of transmissions is reduced. However, quality factors such as prediction accuracy, fault detection efficiency and coordinated control of the overall system suffer. Here, we discuss and highlight the trade-offs that should be considered when choosing between embedding and centralized ML, especially for multihop networks. In addition, we present estimations that demonstrate the energy trade-offs between embedded and centralized ML. Although the total network energy consumption is lower with central prediction, it makes the network more prone for partitioning due to the high forwarding load on the one-hop nodes. Moreover, the continuous improvements in the number of operations per joule for embedded devices will move the energy balance toward embedded prediction.

***Keywords*—**Central ML, embedded machine learning, energy consumption, local ML, Wireless Sensor Networks, WSN.

I. Introduction

THE overriding focus in WSNs has been to preserve energy. Sending and receiving data have been the major energy consumer in such networks. One approach is therefore to process data locally, i.e. embedded processing, and only submit control actions or calculated data instead of transferring all underlying information. The result is smart environments, where decisions are moved from a central location to the embedded device. Delay can be reduced since no transmissions are needed before taking decisions and activating appropriate action. Preventing transmission and central storage of raw-data improve privacy and security. In addition, the consequences of network partitioning are reduced. These advantages are precious for a wide range of areas from autonomous vehicles, drone navigation and robotics to health technology. Another important advantage of reduced transmissions is that energy usage may be reduced, especially for the nodes in the vicinity of the central in multihop networks, which must forward all traffic sent by nodes located further away. Reducing energy increases network lifetime.

Smart environments are important parts of IoT and Industry 4.0 [1]. Embedded ML has a potential to significantly extend where such an approach is beneficial [2]. The deep learning algorithms that are the basis for ML were introduced in the late 20th century [3] and have had a rapid development in the past decade producing successful information retrieval, computer vision [4], speech recognition [5] and image analysis [6].

When designing a network, the benefits of embedded ML must be weighed against the benefits of central solutions. For embedded ML, the process decisions are made solely on local data, which may result in sub-optimal performance. On the contrary, central predictions take advantage of full knowledge of all the raw-data gathered in the network to improve the quality of the results, and to make more coordinated decisions.

Training of ML models requires excessive processing capacity and is time, energy and memory consuming. The training process is therefore generally not suitable for embedded systems. Prediction on the other hand, may be performed on embedded devices, for instance using TensorFlow Lite which is a lightweight version of TensorFLow [7]. TensorFlow Lite is designed to enable low-latency prediction of deep feed-forward neural networks on embedded and mobile devices.

In this paper, we investigate the trade-off between sending data to a central location for processing, versus local processing, i.e. embedded ML. Our motivation is to assess to what extent embedded ML is favorable in WSNs, especially in terms of network energy usage. WSNs are an interconnected collection of sensor nodes whose data are sent to a collection node, in this paper called the central. Some of the nodes are not in direct contact with the central such that nodes must forward data on behalf of each other. We highlight and discuss the various factors that affect the embedded-central trade-off, such as network size in terms of number of hops, amount of data sampled by the sensors and quality requirement set by the applications. Furthermore, to get a more insightful assessment of the trade-offs between central and embedded ML, we estimate energy usage in WSN. To this end, the nodes are running a ML model that predicts soil moisture. The soil moisture is decided by several factors [8]; increasing temperature of applied water reduces the time for wetting, irreversible drying process can induce water repellence and so forth. Thus, predicting of soil moisture is a hard problem [9]. However, rather than pursuing an accurate moisture prediction, the goal is to develop a reasonable model as a means to gain insight into the sought energy usage trade-off.

Our contribution is to discuss and compare the trade-off between central versus embedded ML. According to energy consumption, the discussion is supported by measurements and calculations. It is revealed that, although transmission is currently cheaper than prediction, embedding is likely to

Dr. Anne-Lena Kampen is with the Department of Computer science, Electrical engineering and Mathematical sciences, HVL Western Norway University of Applied Sciences, Bergen, Norway (e-mail: alk@hvl.no)

Prof. Øivind Kure is with the Department of Technology Systems, University of Oslo, Oslo, Norway (e-mail: oivind.kure@its.uio.no).

lengthen the network lifetime in large multihop networks.

Section II presents related work, the central versus embedded ML discussion is presented in Section III. Section IV presents moisture prediction followed by energy evaluation in Section V. Section VI presents the conclusion.

II. Related Work

Our aim is to investigate the energy trade-off between central and embedded ML. There are few examples of full implementation of embedded deep learning found in the literature [10]. The training processes are almost always performed off-device, and even running the pre-trained model can be too resource-demanding for small embedded systems [11]. Still, the offloading part of ML processing is possible with the right combination of ML algorithms and embedded device. In [12], prediction is shared between the low power MSP423 device, which performs feature extraction and logistic regression, and the gateway which performs classification using Gradient Boosting Tree. They found that energy consumed using their approach reduces the energy usage of the IoT device to 1/16 of transmitting the raw data. In [13], the energy and time consumed for prediction on mobile and IoT class hardware is presented. The prediction is based on deep learning algorithms, Deep Neural Network (DNN) and Convolutional NN (CNN), to process audio and image data. The processing time reported varies from less than 1 ms for models that are simple in terms of number of model parameters, to 4.7 minutes for more complex models. Some part of the deep learning computation can be performed on mobile devices, while some are offloaded to the cloud for maximum performance and energy efficiency [14] or improving leaning performance and reducing network traffic [15]. An example of complete implementation of both training and classification on embedded devices is presented in [16], where Gaussian mixture model (GMM) is used as a kind of simple embedded ML. Furthermore, [17] presents a solution where training is distributed to small sensor nodes by making the nodes cooperate to conduct the training.

An applicable method to reduce energy usage is to optimize the sensing process by reducing the sampling frequency as suggested in [18], where the energy impact of distributing increasing parts of the ML process to embedded platforms is evaluated. The ML process is divided into optimization of sampling frequency, embedded featuring and embedded classification. The result shows that optimization of sampling has by far the highest impact to reduce energy usage.

Studies of embedded ML in WSN as well as control system issues related to WSN are presented in literature. A survey of wireless network design for control systems is presented in [19], taking interactive variables such as sampling period, message delay, message dropout, and network energy into consideration. A discussion of central versus embedded use of deep learning used in the WSN is presented in [20]. The conclusion is that embedding can offload the processing burden of the central and reduce traffic. However, centralized approaches are still the dominating solution applied. Alsheikh et al. [21] study ML for routing, localization, clustering and so forth, and emphasize that the limited resources of the network must be taken into consideration. Applying WSN as part of a control system is discussed in the survey [22], where the focus is industrial requirements and candidate protocols. In [23], a wireless sensor-actuator system for industrial control system is studied, focusing on real-time performance. Our focus is directed toward issues specifically related to use of ML in multihop network topologies.

Various solutions to predict soil moisture are proposed in the literature. Neural Network (NN) is used to predict soil moisture one hour ahead in [24] in order to optimize irrigation. The horizontal and vertical forces acting on a chisel represent the input dataset to an auto-regressive error function combined with NN models to estimate soil moisture in [25]. SVM is used to predict soil moisture using weather data along with soil temperature from 11 meteorological stations as input are presented in [26].

III. Embedded versus Central ML

This section discusses the trade-off between central and embedded ML for multihop WSNs. We use 'central' ML as a collective term for any kind of cloud or other centralized systems where the data must be transmitted from the nodes toward a more centralized location for management. The border between embedded and central approaches is not firm. The embedded device can take control of an increasing part for the ML process. In this discussion, the ML process is divided into prediction and training.

The scenarios discussed consist of battery charged sensor nodes, in addition, actuators can be located adjacent to the sensors to provide process control. The actuators can be controlled either from the central or from the adjacent sensors for the embedded solution. The latter requires no transmission over the network. Thus, if the sensors' raw-data are used to control the actuators, the actuators respond without the delay otherwise introduced by ML processing or transmission. Reducing delay is essential for critical processes in industry and automotive networks [27], for instance for motion control applications [28].

If prediction is used as part of a control action, a priori it is uncertain whether embedded prediction or central prediction is advantageous. Central prediction will normally be faster due to substantially larger computing resources compared to an embedded device. On the other hand, central prediction will incur transmission and queuing delay, and must handle packet loss.

For robustness against network failures, the advantages of embedded solutions are more pronounced. Wireless systems cannot always provide consistent and resilient connection with the central. However, sensor sampling is not affected by network failures thus the adjacent actuators continue to receive instructions. Such uninterrupted continuation of processes can be crucial for instance for industrial and healthcare processes that should continue even under denial-of-service attacks [29].

Another important advantage of embedded prediction is enhanced privacy and security since it avoids potential

exposure of data to a third party [30]. The raw-data do not traverse the wireless medium, which is prone to eavesdropping. In addition, the data are not centrally cached. Caching all data centrally makes the central an inspiring target for hackers [29]. Enhancing privacy is especially beneficial for sensitive data subject to regulations.

Using embedded prediction, the energy-usage between the nodes is balanced, and the nodes are more likely to deplete simultaneously, which may be an advantage for network management. In contrast, transmission gives an unbalanced energy sage between the nodes. Specifically, nodes in the proximity of the central must forward data from nodes located further away such that the forwarding load increases with decreasing hop-count. Thus, the one-hop are the fasted depleting nodes since they undergo the heaviest forwarding load [31]. In addition, they are the most critical to keep the network connected. However, for embedded solutions the predictions are based solely on local sensor readings, which raise issues. Data from several neighboring sensors cannot be compared to detect noisy or false sensor readings. False readings may lead to harmful controller actions and input data with noise are likely to give inaccurate predicted values. In addition, the control performance may be sub-optimal since the individual controller lacks information about the state or output of other controllers [32]. In contrast, centralizing prediction takes advantage of full knowledge of the raw-data to improve the quality of the results. All nodes transmit all sampled data to the central. In addition, the improved memory resources of the central mean that data for a longer period can be cached to search for unexpected temporal changes, and better investigation of the reasons for failing readings. Improved overview, understanding and knowledge of the surveyed scenario are gained for further improvements of the processes, and coordinated control of the overall system. In addition, troubleshooting regarding both raw-data and the prediction processes are easier.

The discussion above shows that there is an important quantity – quality trade-off between embedded and centralized predictions. Although, embedded predictions are appealing regarding reduced network traffic, quality aspects like prediction accuracy and efficient troubleshooting need to be considered when deciding the most appropriate scheme. Centralized approaches may be needed for critical processes requiring coordinated control and efficient troubleshooting. The improvement in network lifetime that embedded prediction may provide must be balanced against these advantages. On the other hand, control may be sacrificed for more continuity in the control process. Embedded processes work even when the network is disconnected.

So far, the focus has been on central versus embedded prediction and control, training, on the other hand, is generally performed centrally. The most apparent advantage relates to the limited processor, memory and energy resources of the nodes [33]. The limitations result in infeasible long processing time, inability to cache the amount of data point that may be needed for training and early depletion of sensors nodes. Battery replacement of depleted nodes increases maintenance cost and should therefore be avoided [18]. In addition, administration of the training process may require individual management of each node, which is more challenging than managing a common central.

Aggregation can be used to exploit spatial and temporal correlations in the data to reduce the number of transmissions needed for central training. The energy gains of aggregation increase with aggregation period. However, the longer the aggregation interval, the poorer is the granularity of the input to the training process which may give less accurate ML models. Generally, fast changing processes should not be aggregated; at least great care must be taken to ensure that all variations of input parameters are included in the training dataset for the trained model to be correct. Thus, the accuracy of the trained model must be balanced against reduced network traffic [34].

The main disadvantage of centralized training applies to reduced privacy and increased network traffic. In addition, training individual models for each node or a common model with data from all the nodes as input, places significant workload on the central.

IV. ML

This section presents ML predictions of soil moisture. The data form a weather station along with the data from moisture sensors have been collected over a period of about four months. The data are transmitted over wire from the gauging instrument to a server. From the server the data are downloaded to a computer and raspberry Pi for processing. ML has been implemented using scikit-learn library [35] with Python programming language.

The parameters that are measured are rainfall, rain-rate, humidity, dewpoint, outside temperature, radiation and windspeed. For ML, DNN is implemented with multi-layer perception (MLP) for supervised regression problem. The stochastic gradient-based optimizer Adam [36] is used for training. R^2 is used find the optimal DNN topology. The value of R^2 ranges from -∞ to 1. A value less than zero indicates that the observed mean is better than the predicted value, and a value of 1 indicates a perfect agreement between the predicted value and the measured soil moisture. Fig. 1 displays the moisture prediction. The black curves represent the reference moisture values, provided by the moisture sensor. The dimmed curves represent the predicted values.

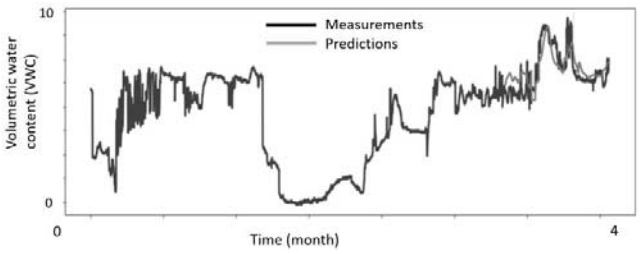

Fig. 1 Measured and predicted soil moisture

V. Energy Usage

This section investigates the energy balance between embedded and central ML based on the energy usage of the embedded devices. Raspberry Pi, which according to [37] is one of the best-known hardware IoT platforms, is used for estimation. The raspberry Pi cannot enter a real sleep state like other low power nodes; thus, the idle mode consumption will reduce the profit of offloading resource-demanding processes from the embedded device to the central. In addition, our measurements are based on complex models that detect soil moisture; other prediction models would give other values. However, the measurements show a valid trend for the energy usage associated with the various ML processes, and our results are therefore useful when designing energy efficient WSNs.

We use the power-models presented in [38] and [39] to estimate the power consumed by the raspberry Pi when processing the data. See the references for the exact equations. We report average values for 20 measurements.

The CPU consumes energy for processing of sensor data. In contrast, the energy consumed during communication is generally dominated by the radio, especially in WSNs using small sensor nodes such as [40]. Thus, the energy consumed for communication is focused toward the time the radio is active to transmit and receive.

To investigate transmission, we chose a data packet size of 128 Byte. The size of the packet is selected by counting the bytes needed to carry all required information, such as readings from seven sensors represented as integer numbers in addition to general management fields.

For energy usage, we assume a completely fair workload between the nodes, which means that the number of predecessor nodes is equally balanced between all the nodes at a given hop-distance from the central. Predecessors of a node are all the nodes whose data are forwarded through this node to reach the central. The immediate predecessors are called child nodes; accordingly, their immediate successor is called parent node. Each parent node has three child nodes in our scenarios.

For centralized approaches, the sampled data from all the nodes are transmitted to the central for prediction. That is, data form nodes further away must be forwarded reach the central as explained in Section III. In addition, all nodes overhear traffic from nodes in their vicinity, i.e. their neighboring nodes. The energy usage for overhearing is fairly similar to the energy used during transmission. Within our rough approximation, they can be considered equal. Both the overall energy consumption and the energy use by the nodes one-hop away from the central will then scale linearly with the number of overhearing nodes. In actual deployment, the nodes will have different number of nodes they overhear. For our illustration we believe that overhearing four nodes is a reasonable assumption. Typically, sleep algorithms and MAC protocols will incorporate some robustness or non-optimality.

Comparing central versus embedded ML means that the energy consumed when all data are transmitted to the central for prediction is compared against the energy consumed when the prediction is performed at the embedded device, thus no transmissions. We do not consider the energy impact of any action that results from the prediction, whether they initiate a control action, or they just are logged centrally. We focus solely on the impact on energy usage of processing raw data locally or centrally.

The total energy consumed by the network is the sum of the consumption of each individual node.

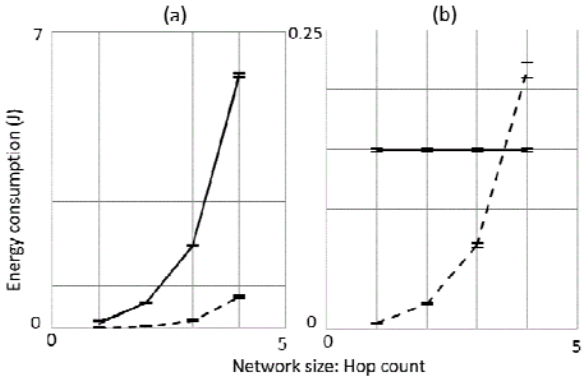

Fig. 2 The continuous graphs represent prediction and the dotted lines represent transmission. (a) compares the total energy consumed for embedded versus central prediction. (b) compares the energy usage for the one-hop nodes during embedded versus central prediction

In Fig. 2 (a), the total energy consumed for embedded prediction is compared against the total energy consumed for transmission to the central, i.e. embedded versus central ML. The curves show that transmitting to the central saves energy. The reason is that energy scales with process time, and transmission is a much faster process than prediction, see Table I. However, the total energy consumption does not convey a full picture. In a wireless network, the nodes one-hop away from the central will have the highest energy consumption during transmission, since they forward the highest number of packets. These nodes will therefore deplete first, by which the network is partitioned. Fig. 2 (b) compares the one-hop node's consumption during embedded prediction (horizontal line) and central ML, i.e. when all nodes transmit their data to the central (dotted curve). When the network size increases beyond three hops, transmission consumes more energy than prediction. Thus, to maintain a connected network and increase network lifetime, embedded prediction should be considered, especially for large networks.

TABLE I
Consumption for Prediction and Training

	Power consumption [W]	Runtime [s]	Energy consumed [J]
Prediction	1.4893	0.1001	0.149
Tx 128 bytes	2.1939	0.0004	0.0009
Training 1.5-moth	1.6435	117557	193200
Training 4-month	1.6441	208755	343209

Training is, as expected, several orders of magnitude more expensive in both time and energy compared to prediction, see Table I. The results support the argument presented in [11]

that training must be performed off-device. The argument is strengthened as the size of the dataset increases. The reason is that the energy usage is related to the size of the dataset, which is also supported by our measurements. The number of rows in the 4-month dataset is 1.83 times the length of the 1.5-month dataset, while the related energy factor is 1.78.

VI. Conclusion

The numbers presented here are just snapshots in a fast-changing technology. There is a rapid improvement in the number of operations per joule. Predictions will therefore consume less energy as technology improves. This improvement is more rapid than the development of more energy efficiency for the radio. The energy-reduction for raspberry Pi is likely to follow the exponential graph presented in Fig. 5 in [39], where the number of computations achieved per kWh is doubled every 19 months. Thus, in three years' time the energy usage for the prediction process is about a quarter of today. That is, the horizontal line in Fig. 2 (b) moves down to 0,038 J, such that embedded prediction becomes the most energy efficient approach when the network size is larger than two hops. Other radio technologies and network topologies would give other results; however, the trend is similar. In addition, there is an effect of moving prediction for general purpose CPU to specialized FPGA or chips. Overall, our examples illustrate that one must expect that balance point between embedded prediction versus transmission to a central point will shift towards embedding, also in networks with fewer nodes than used in our illustration. Thus, embedding the prediction process saves energy in future networks, however the price paid is reduced quality and robustness as discussed in Section III.

References

[1] X. Li, D. Li, J. Wan, A. V. Vasilakos, C.-F. Lai, and S. Wang, "A review of industrial wireless networks in the context of Industry 4.0," Wireless networks, vol. 23, no. 1, pp. 23-41, 2017.

[2] D.-H. Kim et al., "Smart machining process using machine learning: A review and perspective on machining industry," International Journal of Precision Engineering and Manufacturing-Green Technology, vol. 5, no. 4, pp. 555-568, 2018.

[3] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," Neurocomputing, vol. 234, pp. 11-26, 2017.

[4] K. Vamsikrishna, D. P. Dogra, and M. S. Desarkar, "Computer-vision-assisted palm rehabilitation with supervised learning," IEEE Transactions on Biomedical Engineering, vol. 63, no. 5, pp. 991-1001, 2015.

[5] F. Tao and C. Busso, "Gating neural network for large vocabulary audiovisual speech recognition," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 26, no. 7, pp. 1290-1302, 2018.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580-587.

[7] M. Abadi et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," arXiv preprint arXiv:1603.04467, 2016.

[8] L. Dekker and C. Ritsema, "Wetting patterns and moisture variability in water repellent Dutch soils," Journal of Hydrology, vol. 231, pp. 148-164, 2000.

[9] L. Meng and S. M. Quiring, "A comparison of soil moisture models using soil climate analysis network observations," Journal of Hydrometeorology, vol. 9, no. 4, pp. 641-659, 2008.

[10] N. D. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi, and F. Kawsar, "Squeezing deep learning into mobile and embedded devices," IEEE Pervasive Computing, vol. 16, no. 3, pp. 82-88, 2017.

[11] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2923-2960, 2018.

[12] F. Samie, S. Paul, L. Bauer, and J. Henkel, "Highly efficient and accurate seizure prediction on constrained iot devices," in 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2018: IEEE, pp. 955-960.

[13] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, and F. Kawsar, "An early resource characterization of deep learning on wearables, smartphones and internet-of-things devices," in Proceedings of the 2015 international workshop on internet of things towards applications, 2015, pp. 7-12.

[14] A. E. Eshratifar and M. Pedram, "Energy and performance efficient computation offloading for deep neural networks in a mobile cloud computing environment," in Proceedings of the 2018 on Great Lakes Symposium on VLSI, 2018, pp. 111-116.

[15] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," IEEE network, vol. 32, no. 1, pp. 96-101, 2018.

[16] J. Lee, M. Stanley, A. Spanias, and C. Tepedelenlioglu, "Integrating machine learning in embedded sensor systems for Internet-of-Things applications," in 2016 IEEE international symposium on signal processing and information technology (ISSPIT), 2016: IEEE, pp. 290-294.

[17] Y. Fukushima, D. Miura, T. Hamatani, H. Yamaguchi, and T. Higashino, "MicroDeep: In-network deep learning by micro-sensor coordination for pervasive computing," in 2018 IEEE International Conference on Smart Computing (SMARTCOMP), 2018: IEEE, pp. 163-170.

[18] X. Fafoutis, L. Marchegiani, A. Elsts, J. Pope, R. Piechocki, and I. Craddock, "Extending the battery lifetime of wearable sensors with embedded machine learning," in 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), 2018: IEEE, pp. 269-274.

[19] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," IEEE Communications Surveys & Tutorials, vol. 20, no. 2, pp. 978-1013, 2017.

[20] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2224-2287, 2019.

[21] M. A. Alsheikh, S. Lin, D. Niyato, and H.-P. Tan, "Machine learning in wireless sensor networks: Algorithms, strategies, and applications," IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1996-2018, 2014.

[22] K. Ovsthus and L. M. Kristensen, "An industrial perspective on wireless sensor networks—A survey of requirements, protocols, and challenges," IEEE communications surveys & tutorials, vol. 16, no. 3, pp. 1391-1412, 2014.

[23] C. Lu et al., "Real-time wireless sensor-actuator networks for industrial cyber-physical systems," Proceedings of the IEEE, vol. 104, no. 5, pp. 1013-1024, 2015.

[24] B. Keswani et al., "Adapting weather conditions based IoT enabled smart irrigation technique in precision agriculture mechanisms," Neural Computing and Applications, vol. 31, no. 1, pp. 277-292, 2019.

[25] A. L. Johann, A. G. de Araújo, H. C. Delalibera, and A. R. Hirakawa, "Soil moisture modeling based on stochastic behavior of forces on a no-till chisel opener," Computers and Electronics in Agriculture, vol. 121, pp. 420-428, 2016.

[26] M. K. Gill, T. Asefa, M. W. Kemblowski, and M. McKee, "Soil moisture prediction using support vector machines 1," JAWRA Journal of the American Water Resources Association, vol. 42, no. 4, pp. 1033-1046, 2006.

[27] L. L. Bello and W. Steiner, "A Perspective on IEEE Time-Sensitive Networking for Industrial Communication and Automation Systems," Proceedings of the IEEE, vol. 107, no. 6, pp. 1094-1120, 2019.

[28] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," IEEE industrial electronics magazine, vol. 11, no. 1, pp. 17-27, 2017.

[29] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The

internet of things for health care: a comprehensive survey," IEEE Access, vol. 3, pp. 678-708, 2015.
[30] M. Verhelst and B. Moons, "Embedded deep neural network processing: Algorithmic and processor techniques bring deep learning to iot and edge devices," IEEE Solid-State Circuits Magazine, vol. 9, no. 4, pp. 55-65, 2017.
[31] Q. Wang et al., "Reducing delay and maximizing lifetime for wireless sensor networks with dynamic traffic patterns," IEEE Access, vol. 7, pp. 70212-70236, 2019.
[32] X. Ge, F. Yang, and Q.-L. Han, "Distributed networked control systems: A brief overview," Information Sciences, vol. 380, pp. 117-131, 2017.
[33] R. V. Kulkarni, A. Forster, and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: A survey," IEEE communications surveys & tutorials, vol. 13, no. 1, pp. 68-96, 2010.
[34] H. Xu, W. Yu, D. Griffith, and N. Golmie, "A survey on industrial Internet of Things: A cyber-physical systems perspective," IEEE Access, vol. 6, pp. 78238-78259, 2018.
[35] Scikit-learn. "Machine Learning in Python." https://scikit-learn.org/stable/ (accessed May 2020).
[36] D. P. K. a. J. L. Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION," arXiv:1412.6980v9, 2017. [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf.
[37] F. Javed, M. K. Afzal, M. Sharif, and B.-S. Kim, "Internet of things (IoT) operating Systems support, networking technologies, applications, and challenges: A comparative review," IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2062-2100, 2018.
[38] F. Kaup, P. Gottschling, and D. Hausheer, "PowerPi: Measuring and modeling the power consumption of the Raspberry Pi," in 39th Annual IEEE Conference on Local Computer Networks, 2014: IEEE, pp. 236-243.
[39] F. Kaup, S. Hacker, E. Mentzendorff, C. Meurisch, and D. Hausheer, "Energy models for NFV and service provisioning on fog nodes," in NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, 2018: IEEE, pp. 1-7.
[40] Advanticsys. "802.15.4 Mote Modules." https://www.advanticsys.com/shop/802154-mote-modules-c-7_3.html (accessed 2020).