A federated binarized neural network model for constrained devices in IoT healthcare services

Hyeontaek Oh
Institute for Information Technology Convergence, KAIST, Daejeon, Rep. of Korea, 34141
Email: hyeontaek@kaist.ac.kr

Jongmin Yu
Institute for Information Technology Convergence, KAIST, Daejeon, Rep. of Korea, 34141
Email: andrew.yu@kaist.ac.kr

Nakyoung Kim
Institute for Information Technology Convergence, KAIST, Daejeon, Rep. of Korea, 34141
Email: nkim71@kaist.ac.kr

Dongyeong Kim
Institute for Information Technology Convergence, KAIST, Daejeon, Rep. of Korea, 34141
Email: deathquin@kaist.ac.kr

Jangwon Lee
Institute for Information Technology Convergence, KAIST, Daejeon, Rep. of Korea, 34141
Email: walker0723@kaist.ac.kr

Jinhong Yang
Department of Healthcare Information Technology, Inje University, Gimhae, Rep. of Korea, 50834
Email: jinhong@inje.ac.kr

***Abstract*—In IoT healthcare environment, the devices are not sufficiently powerful for operating recent deep learning models, and data collected by the devices are usually decentralized. Moreover, data are unavailable to share between devices because of information security issues. Therefore, a concept of federated learning has emerged to overcome data sharing issues, and a concept of binarized neural network has emerged to generate lightweight deep learning models. This paper proposes a federated binarized neural network model to derive a reliable healthcare system in this circumstance. This paper shows an overview of considered system model with constrained IoT healthcare devices. In addition, this paper shows illustrations of implementing the proposed federated learning model with the proposed binarized MLP networks by utilizing an open-source library. The experiment results show that the binarized MLP network shows comparable performances compared to the full-precision MLP network while the binarized MLP requires about 10-times less model size for training.**

***Index Terms*—Federated learning, Binarized neural network, Internet of Things, Healthcare**

I. Introduction

Today, with a huge development of machine learning (ML) and artificial intelligence (AI) techniques, the application of AI/ML is inevitable to all IT based applications and services. Particularly, the emergence of deep learning has accelerated the innovations in all IT related areas [1], [2].

Deep learning based AI techniques require the entire dataset as centralized manner. However, these kinds of centralized approaches have raised several issues, such as data security and privacy. Particularly, to preserve privacy of users, the necessity of distributed approaches for AI/ML techniques has increased. Therefore, a concept of federated learning has been emerged [3], [4]. Unlike to centralized AI models, in federated learning environment, each device contains AI models for train and inference. The device sends AI models, including detailed information of parameter weights. Since data from the device are not shared to public, using federated learning becomes more popular in industrial sectors, particularly, that handles various personal data such as healthcare.

With the advantages of federated learning, various approaches have been proposed in IoT based healthcare services [5]–[7]. Particularly, Wu *et al.* [8] proposed a personalized federated learning model for in-home health monitoring by exploiting generative convolutional autoencoder in cloud-edge computing environment. Chen *et al.* [9] proposed a federated transfer learning framework for wearable healthcare devices. These kinds of approaches produced remarkable performances; however, deep neural network based federated learning models have a very high complexity model in general. Therefore, it may not suitable for constrained devices. In IoT healthcare environments, the complexity of AI/ML models becomes an issue because many applications have utilized constrained devices.

To overcome this issue, as one of possible approaches to reduce the complexity of AI model, a concept of binarized (or binary) neural network has emerged [10]. Since the concept of binarized neural network can significantly reduce size and complexity of deep neural networks, it has been mainly considered in image processing area [11] to reduce computational complexity of convolutional neural network. In IoT environments, few studies have leveraged BNN. Using the concept of binarized neural network, this paper proposes a binarized multi-layer perceptron model for constrained devices in IoT environment. Cerutti *et al.* [12] proposed a sound event detection model based on binary neural network for power-constrained IoT devices. Verca *et al.* [13] proposed a method for detecting network intrusion with binarized neural network that can be utilized in embedded IoT devices.

Based on these backgrounds, this paper proposes a framework with federated binarized neural network model in IoT

This research was financially supported by the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D program. (Project No. 0011879). Corresponding author: Jongmin Yu (andrew.yu@kaist.ac.kr).

healthcare environments with various constrained devices. The contributions of this paper are summarized as follows:

- This paper proposes a federated learning based binarized neural network model for considering the characteristics of constrained devices in IoT healthcare environments.
- This paper proposes a binarized multi-layer perceptron (MLP) network model for handling IoT data. It also illustrates a implementation procedures of the proposed federated binarized MLP using an open-source [14].
- Using the real-world dataset [15] captured by a radar-based contactless biometric monitoring testbed, this paper shows the performance of the proposed binarized neural network model in federated learning environment. The results show that the proposed binarized neural network model performs comparable performance, compared with the full-precision model, with 10-times less model size for training.

II. System model

This paper considers a healthcare monitoring system consisting of a global server and N distributed gateways deployed at users' house. Each gateway manages various health/biometric monitoring devices in the house. Each monitoring device collects various healthcare data from users. The collected data are transferred to the local gateway device. Figure 1 shows an illustration of the proposed federated learning framework.

A. Federated learning for IoT devices

For applying federated learning based AI applications, both local clients and global server share the same neural network structures. To update the model parameters, for each communication round, a federated learning is operated as follows.

1) Local clients utilize the data collected from monitoring devices to train local neural network model.
2) When a local model training is done, the trained model is sent to the global server. At this time, some information such as general statistics of dataset is also transferred for applying a federated learning algorithm in the global server; however, no raw data are directly transferred to the global server.
3) When the global server aggregates all local models from the gateways, the server combines the local models into one single global model and validates the new global model.
4) After the validation, the newly updated global model is sent back to all local gateways so that local clients can also update their local model.

A federated learning iteratively performs to improve the performance of both local and global model.

In the proposed model, both clients and server have deep neural network based model. The primary objective of federated learning with deep neural network is to minimize risk, which means how accurately achieve a global objective by combining the results from local objective functions. Let F^k and D_k denote the local objectives and the set of indexes of

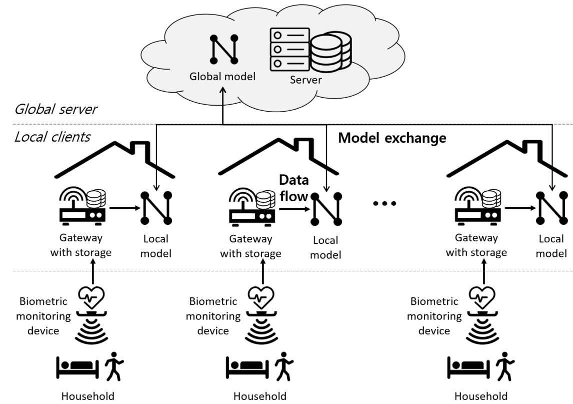

Fig. 1. An overview of considered IoT healthcare environment

local data on device k. Then, the risk minimization problem or federated learning can be shown as follows [16].

$$\min_{\omega \in \mathcal{R}_d} f(\omega) = \sum_{k=1}^{K} \frac{n_k}{n} F^k(\omega), \text{where } F^k(\omega) = \frac{1}{n_k} \sum_{i=D_k} F_i^k(\omega). \tag{1}$$

In equation (1), $n = \sum_{k=1}^{K} n_k$ means the total number of samples, where K is the number of active devices participating in federated learning. The global objective $f(\omega)$ in federated learning is able to be represented as a linear sum of the local objectives $F_k(\omega)$, where ω denotes the parameters of the model (e.g., the weights and bias in a deep neural network). Each local objective is defined by averaging the outputs of the local objectives with respect to each local dataset: $F^k(\omega) = \frac{1}{n_k} \sum_{i=D_k} F_i^k(\omega)$.

FederatedAverage (FedAvg) algorithm, which is one of the most well-known approaches in federated learning, is utilized as the fundamental framework in federated learning setting [16]. FedAvg utilizes an iterative model averaging scheme with collected local stochastic gradient descent (SGD) from local devices. In each iteration t, each device k calculates the average gradient $g_k = \nabla F_k(\omega_t)$ of the local model at the current model parameters ω_t. Then, the server aggregates all of gradients from devices and updates the global as follows,

$$\omega_{t+1} \leftarrow \omega_t - \eta \nabla f(\omega_t), \text{ where } \nabla f(\omega_t) = \sum_{k=1}^{K} \frac{n_k}{n} g_k. \tag{2}$$

This paper also adopts the concept of FedAvg algorithm for performing a federated learning with a binarized neural network. The global model in server can be derived by using FedAvg algorithm.

After the global model is learned, the model can be applied to the local devices. However, if the global model is directly overwritten to local models, each local model has lost its locality that contains the characteristics of individual. Therefore, in this paper, the local model is updated as follows

$$\omega_{k,t+1} = \omega_{G,t+1} + \lambda \omega_{k,t}. \tag{3}$$

In equation (3), $\omega_{G,t+1}$ means the results of FedAvg in global server at time $t+1$, and $\omega_{k,t}$ and $\omega_{k,t+1}$ indicate model

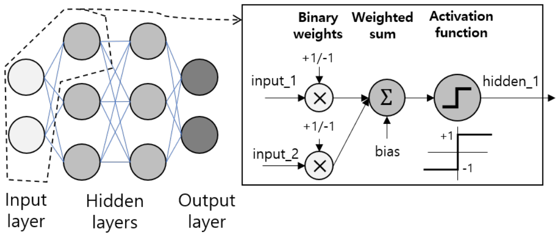

Fig. 2. A concept of binarized neural network

parameters of a local client k at time t and $t+1$, respectively. Here, λ is a balance weight for the local model parameters at the previous time frame. If $\lambda = 0$, it ignores the local model parameters.

B. Binarized neural network model

Binarized neural network (BNN) is a deep neural network that has weight parameters only consists of either -1 or 1 [17]. Figure 2 shows a concept of BNN. When an input It quantizes a weight parameter into either -1 or 1 as follows.

$$q(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ -1 & \text{if } x < 0, \end{cases} \tag{4}$$

where x is a weight parameter of a original model and $q(x)$ is an activation function to make binarized neural network.

In general, each weight parameter of a neural network model is represented as floating number. Therefore, it requires at least 32-bit for storing one weight parameter of the model. However, in a BNN, it requires only 1-bit for storing the weight parameter because the weight is either -1 or 1. Therefore, theoretically, BNN can consume 32-times lower storage and communication costs. Since the communication costs are expensive in federated learning, these kinds of cost reduction is very helpful to operating IoT systems, particularly, consists of many constrained devices. For example, the throughput of LoRaWAN varies from 300 bps to 37.5 Kbps [18].

Various neural network models can be applied to IoT environment. In this paper, a binarized multi-layer perceptron (MLP) network is utilized to train and predict healthcare datasets from constrained devices. The reason to choose MLP is that the dimensionality of input data from constrained devices is relatively lower than other environments.

By using the proposed federated learning framework and binarized MLP network, the next section shows how the proposed federated binarized neural network model is implemented for experiments with the real-world dataset from IoT healthcare environment.

III. Implementation

For implementing a binarized MLP and a federated learning process, this paper utilizes Python, *Tensorflow*, and an open-source *Larq* [14] that provides BNN library.

```python
class BNN:
  @staticmethod
  def build(attributes, classes):
    kwargs = dict(input_quantizer="ste_sign",
                  kernel_quantizer="ste_sign",
                  kernel_constraint="weight_clip",
                  use_bias=False)
    model = tf.keras.models.Sequential()
    model.add(larq.layers.QuantDense(64, kernel_quantizer="ste_sign",
                                     kernel_constraint="weight_clip",
                                     use_bias=False, input_shape=(attributes,),
                                     ))
    model.add(tf.keras.layers.BatchNormalization(momentum=0.9, scale=False))
    model.add(larq.layers.QuantDense(32, **kwargs))
    model.add(tf.keras.layers.BatchNormalization(momentum=0.9, scale=False))
    model.add(larq.layers.QuantDense(16, **kwargs))
    model.add(tf.keras.layers.BatchNormalization(momentum=0.9, scale=False))
    model.add(larq.layers.QuantDense(classes, **kwargs))
    model.add(tf.keras.layers.BatchNormalization(momentum=0.9, scale=False))
    model.add(tf.keras.layers.Activation("softmax"))
    return model
```

Fig. 3. An illustration of binarized MLP using *Larq*

```python
with larq.context.quantized_scope(True):
  global_model = BNN().build(attributes, classes)

  for comm_round in range(EPOCHS):
    global_weights = global_model.get_weights()
    scaled_local_weight_list = []

    for device_id in devices:
      bnn_local = BNN()
      local_model = bnn_local.build(attributes, classes)
      local_model.compile(optimizer='adam',
                          loss='categorical_crossentropy', metrics=['accuracy'])

      weights = []
      lw = scale_model_weights(local_models[device_id].get_weights(), 1)
      weights.append(lw)
      gw = scale_model_weights(global_model.get_weights(), 0.2)  # lambda
      weights.append(gw)
      local_weights = sum_scaled_weights_bin(weights)
      local_model.set_weights(local_weights)

      dataset = devices_dataset[device_id]
      local_model.fit(dataset['X_train'],
                      tf.keras.utils.to_categorical(dataset['y_train'],
                                                    num_classes=classes),
                      batch_size=32, epochs=1, verbose=0)

      local_models[device_id] = local_model

      scaling_factor = weight_scalling_factor(device_id)
      scaled_weights = scale_model_weights(local_model.get_weights(),
                                           scaling_factor)
      scaled_local_weight_list.append(scaled_weights)

      y_pred = np.argmax(local_model.predict(dataset['X_test']), axis=1)

      K.clear_session()

    average_weights = sum_scaled_weights_bin(scaled_local_weight_list)

    global_model.set_weights(average_weights)
    global_model.compile(optimizer='adam',
                         loss='categorical_crossentropy', metrics=['accuracy'])

    y_pred = np.argmax(global_model.predict(global_dataset['X_test']), axis=1)
```

Fig. 4. An illustration of federated learning using *Tensorflow* with *Larq*

A. Binarized Multi-Layer Perceptron

Figure 3 shows an implementation of the proposed binarized MLP network using *Larq* library. Since the dimension of data is relatively low, a MLP network is adopted on both the server and the devices at users' house for model training and prediction. The dimension of input is 14 (from a dataset [15]). The MLP network is composed of 4 fully-connected layers with 64, 32, 16, and 6 (i.e., the number of classes) units for extracting features from input data. A binary quantizer (which can be used as activation function in BNN), called SteSign quantizer with a standard weight clip constraint is used. Using this quantizer, the gradient is estimated using the straight-through estimator. *Softmax* is adopted in the final fully-connected layer of the MLP network to calculate the probability of the output results. According to the guideline from [14] inspired by [19], batch normalization layers with momentum of 0.9 are inserted between network layers.

B. Federated learning with binarized MLP

Figure 4 shows an implementation of the proposed federated learning framework for the simulation. The implementation of

TABLE I
ACCURACY AND F1-SCORE OF VARIOUS METHODS

Model	Accuracy	F1-score	Model Size
Fed-Bin-MLP (global)	0.783	0.783	1.36 KB
Fed-Full-MLP (global)	0.802	0.802	14.80 KB
Fed-Bin-MLP (local avg.)	0.843	0.843	1.36 KB
Fed-Full-MLP (local avg.)	0.833	0.833	14.80 KB

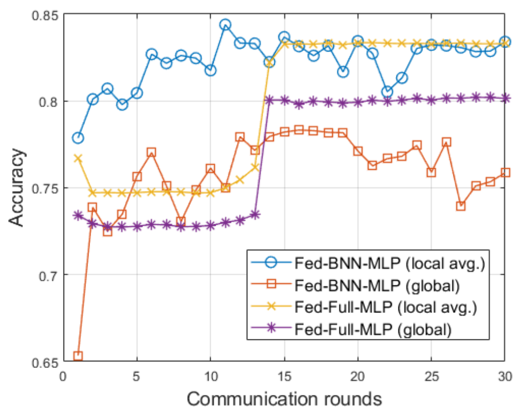

Fig. 5. A trend of accuracy for federated MLP models

federated learning is inspired by [20]. The detailed implementation is follows.

Line 1 means the model utilizes the binary weights (i.e., either -1 or 1). First, global model is initialized in line 2. A variable in line 6 uses to calculate scales weights for applying equations (1) and (2).

For each local device, local model learning is performed from line 8. From line 9 to 11, it shows a local model initialization. The model is trained by Adam optimizer with a loss function of categorical cross-entropy.

From line 14 to 20, it represents the proposed local weight update procedure in equation (3). The balance weight λ is set to 0.2 tuned using cross-validation procedure.

From line 22 to 28, it loads a local datasets of each device and trains a local binarized MLP network model. Batch size for training local models is set to 32. In line 28, the trained local model is saved for performing the proposed local weight update procedure described above.

From line 30 to 33, is performs weight calculation for federated learning according to equation (1). After local model training is done, from line 39, the global binarized MLP is updated according to equation (2).

IV. EXPERIMENTS

This section shows various simulation results of the proposed federated binarized MLP model compared with various methods. For the simulation, this paper refers a dataset from [15] that consists of various biometric information (e.g., heart rate, breath rate, the intensity of movement etc.) collected from radar-based contactless biometric monitoring testbed.

The total number of participants considered in this experiment is 20. The dataset contains 14 attributes (e.g., heart

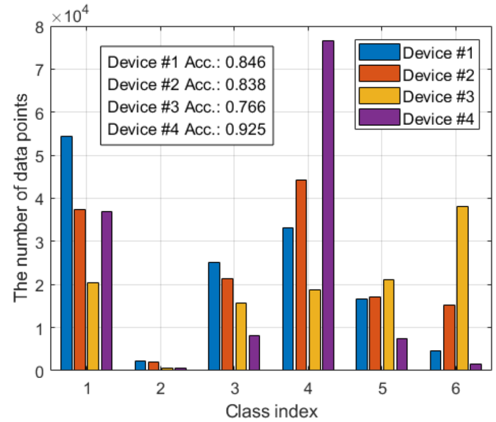

Fig. 6. Data distribution of selected devices and their prediction accuracies for federated binarized MLP

rate, breath rate, etc.) collected from radar-based contactless biometric monitoring system [15]. The total number of classes is 6 that represents the status of users. Data with *Null* element are removed for experiments. Consequently, each user has the average 100K data points. Each dataset from user has non-IID characteristics. A min-max scalar is applied for the entire dataset. For binarized network, the data additionally normalized for setting the value in [-1, 1]. Since it is a multi-class classification problem, accuracy and f1-score (which calculates metrics globally by counting the total true positives, false negatives and false positives) are chosen as performance metrics. RTX 2070 SUPER is used for these experiments.

Table I shows the performance of the proposed federated binarized MLP model (i.e., Fed-Bin-MLP) compared with federated MLP with full precision networks (i.e., Fed-Full-MLP). In the table, "global" means the performance of the global model and "local avg." means the averaged performance of all local models. As shown in Table I, full precision MLP shows a better performance than that of binarized MLP. Fed-Full-MLP shows accuracy and f1-score of 0.802 for the global model and those of 0.833 as average for local models, respectively. On the other hand, Fed-Bin-MLP shows accuracy and f1-score of 0.783 for the global model and those of 0.843 as average for local models. The values represent that MLP networks with either binary precision or full precision show similar performance. However, as shown in the table, a model size of Fed-Bin-MLP is about 1.4 KB while that of Fed-Full-MLP is about 15 KB.

Since constrained IoT devices produce relatively lower complex data (i.e., data with a lower dimensionality) compared to other devices, the binarized MLP network also can capture enough discriminative information compared with the MLP network with full precision weights. In addition, since local models is updated using raw datasets, the overall performance of local models is higher than that of the global model for both MLPs with binary and full precision weights.

Meanwhile, Table I shows the best case for each model, so it is necessary to check some trends of accuracy changed during the communication rounds. Figure 5 shows a trend of accuracy for federated MLP models with respect to communications rounds (i.e., as time goes). Basically, when time goes (i.e., the number of communication rounds is increased), overall performance of the federated MLP models increases. However, the trends of binarized MLP and full-precision MLP networks are a little different. The full-precision MLP networks show stable results. Both local and global models show significant performance improvement at a certain points. On the other hand, the binarized MLP networks show relatively unstable results. Since the binarized MLP contains less information in model parameters, it is more sensitive to variations of input data. Therefore, the accuracy values of both local and global models are fluctuated and sometimes overfitted.

Lastly, since this paper handles non-iid datasets, 4 devices, which show the different range of accuracy for local models, are chosen for microscopic analysis. Figure 6 shows a data distribution of the selected devices (i.e., the number of data points of each class) and the accuracy performance of the devices with federated binarized MLP. As shown in the figure, each device collects data with different characteristics. For example, the largest amount of data from device 1 is class 1, but that from device 4 is class 4. The device 3, which has the largest amount data of class 6, shows the lowest performance among 4 devices. These kinds of non-iid characteristics significantly impacts on the performance of local and global models.

Since the global model can contain more generalized information from all local clients, therefore, it is possible to provide better performances for general circumstances. Meanwhile, from a local client points of view, generalization of the model means that it loses some local specific features for customization. Therefore, it is necessary that some kinds of adaptation techniques for both local and global models to improve the overall performance of the entire systems. Not only the characteristics of non-iid data environments but also the input sensitive characteristics of binarized neural networks should be considered. In future, these kinds of issues should be considered in future to improve overall performance of a federated binarized neural networks.

V. Conclusion

This paper has proposed a federated binarized neural network model that can be utilized in IoT healthcare environments with various constrained devices. For the proposed federated binarized neural network model, the detailed methods for model exchange process and a binarized multi-layer perceptron (MLP) network model are proposed. With implementation of the proposed federated learning model with the proposed binarized MLP networks, the experiment results show that the binarized MLP network shows comparable performances compared to the full-precision MLP network while the binarized MLP requires much less model size for training. To improve the overall performance of a federated binarized neural network for IoT healthcare environments, various issues should be more considered in future such as local-global model adaptation, handling non-iid datasets, more input stable binarized neural network models.

References

[1] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 2, pp. 869–904, 2020.

[2] W. Chen, X. Qiu, T. Cai, H.-N. Dai, Z. Zheng, and Y. Zhang, "Deep reinforcement learning for internet of things: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 3, pp. 1659–1692, 2021.

[3] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. Vincent Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 3, pp. 1622–1658, 2021.

[4] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained iot devices," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[5] N. Rieke, J. Hancox, W. Li, F. Milletarì, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, S. Ourselin, M. Sheller, R. M. Summers, A. Trask, D. Xu, M. Baust, and M. J. Cardoso, "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, no. 1, p. 119, Sep 2020.

[6] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, no. 1, pp. 1–19, Mar 2021.

[7] L. Yang, K. Yu, S. X. Yang, C. Chakraborty, Y. Lu, and T. Guo, "An intelligent trust cloud management method for secure clustering in 5g enabled internet of medical things," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[8] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[9] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[10] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, "Binary neural networks: A survey," *Pattern Recognition*, vol. 105, p. 107281, 2020.

[11] T. Simons and D.-J. Lee, "A review of binarized neural networks," *Electronics*, vol. 8, no. 6, 2019.

[12] G. Cerutti, R. Andri, L. Cavigelli, E. Farella, M. Magno, and L. Benini, "Sound event detection with binary neural networks on tightly power-constrained iot devices," in *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '20. New York, NY, USA: ACM, 2020, p. 19–24.

[13] J. Vreča, I. Ivanov, G. Papa, and A. Biasizzo, "Detecting network intrusion using binarized neural networks," in *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, 2021, pp. 622–627.

[14] L. Geiger and P. Team, "Larq: An Open-Source Library for Training Binarized Neural Networks," *Journal of Open Source Software*, vol. 5, no. 45, p. 1746, Jan. 2020. [Online]. Available: https://doi.org/10.21105/joss.01746

[15] H. Oh, J. Yu, C. J. Jung, and J. K. Choi, "Data Analysis and Consideration of Radar-Based Contactless Biometrics Monitoring Testbed for Single Elderly Households," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 46, pp. 1056–1064, 2021.

[16] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[17] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[18] A. Lavric and V. Popa, "Performance Evaluation of LoRaWAN Communication Scalability in Large-Scale Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2018, p. 6730719, Jun 2018.

[19] M. Courbariaux and Y. Bengio, "Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1," *CoRR*, vol. abs/1602.02830, 2016. [Online]. Available: http://arxiv.org/abs/1602.02830

[20] "Federated Learning: A Step by Step Implementation in Tensorflow," accessed Nov. 25, 2021. [Online]. Available: https://towardsdatascience.com/federated-learning-a-step-by-step-implementation-in-tensorflow-aac568283399