Cooperative Learning Strategy for Human Behavior Prediction Using Multi-Modal Data

Doyun Lee[†] and Hoon Lee[*]
[‡]Department of Intelligent Robot Engineering, Pukyong National University, Busan, Korea
[*]Department of Electrical Engineering, UNIST, Ulsan, South Korea
Email: [†]ehdbs8964@pukyong.ac.kr, [*]hoonlee@unist.ac.kr

***Abstract*—Understanding human behaviors leads to fully-automated systems in the near future. This paper investigates a deep learning solution that forecasts human activity patterns based on sensing signals measured by internet-of-things devices. Practical limitations on these small-form-factor sensors request remote deep learning services at a distant edge computing server. Therefore, we need to involve impairments in sensor-server communication phases, such as random packet loss, resource constraint, and propagation noise, in the design of the remote learning architecture. To address these challenges, we propose a collaborative learning strategy among the sensors and server. Each sensor is equipped with its own encoding neural network that compresses high-dimensional sensing signals to communication messages. These are forwarded to the server through imperfect backhaul channels. Then, a classifier at the server infers desired labels. A joint training mechanism of the encoders and classifier is developed along with the channel impairment. By doing so, we can obtain a robust prediction model for arbitrary communication noises. Numerical results demonstrate the viability of the proposed methods.**

***Index Terms*—Lifelog dataset, multi-modal features, human behavior prediction.**

I. Introduction

As human everyday life is prevalent with a variety of activities and behaviors, recording and analyzing this information becomes important. It is necessary to understand the various characteristics of human behavior and physiological signs. The lifelog datasets [1], [2] provide deep insights into such metacognitive tasks. These datasets collect a wide range of modalities measured by sensors embedded in wearable devices, such as images, videos, vital signs, and location data, that provide an unprecedented opportunity to investigate the intricacies of daily experiences. As a result, they have played a crucial role in advancing the research across multiple disciplines to investigate a variety of challenges. Existing studies have explored activity recognition, predicting emotional states from facial expressions, and understanding patterns in daily routines. Such metacognitive tasks of heterogeneous human behaviors invoke the use of deep learning techniques to predict target labels accurately.

Practical constraints on small-form-factor sensors and coupled nature of multimodality features block the execution of deep neural network models at wearable devices straightforwardly. This requests the uplink communications from multiple sensors to a server holding deep prediction models. However, most of conventional works have not directly accounted for the unique challenges induced by the communication issues, including wireless resource constraints and channel impairments. In addition, the design of sensor-server coordination protocols is essential to facilitate decentralized sensor encoding processes and collaborative inference at the server.

These challenges would be addressed by the vertical federated learning (VFL) framework [3]–[5] where multiple client having different features of the identical dataset supports the remote inference at a server. The successes of existing VFL approaches have been confined to a single modal case in image classification applications. In addition, resource constraints in a client-server communication step were not involved in conventional studies. Such a special property requests a practical design of multiple access schemes for the VFL framework.

This paper proposes a decentralized cooperative learning strategy for the lifelog dataset where a server aims at estimating human behaviors by receiving signals measured by individual sensors. Multimodal signals at individual sensors exhibit heterogeneous properties with variable dimensions for each sampling duration. To overcome this difficulty, encoding neural networks of sensors are carefully designed such that they can address heterogeneous inputs with arbitrary lengths. The resulting encoded signals are then forwarded to the server through uplink backhaul channels. Imperfections in this uplink coordination step involve random packet loss in which some of the sensing signals would be dropped due to a severe propagation environment. To tackle this problem, we present an embedding dropout (ED) operation which includes arbitrary losses of encoded signals into the training step. This guarantees the robustness of trained neural networks at the sensors and the server to practical channel imperfections.

In addition, mutual interference among the sensors would result in severe degradation of the inference performance at the server. Inspired by resource sharing schemes, we provide orthogonal multiple access (OMA) and non-orthogonal multiple

This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the MSIT (Intelligent 6G Wireless Access System) under Grant 2021-0-00467, in part by Korea Research Institute for defense Technology planning and advancement (KRIT) grant funded by the Korea government (DAPA(Defense Acquisition Program Administration)) (21-106-A00-007, Space-Layer Intelligent Communication Network Laboratory, 2022), and in part by National Research Foundation(NRF), South Korea, under project BK21 FOUR (Smart Robot Convergence and Application Education Research Center).

TABLE I
Lifelog dataset [1]

(a) Features

Feature	Description	Dimension	Max. length
e4Acc	Accelerometer samples measured by wristband	3	1,920
e4Bvp	Blood volume pressure samples measured by wristband	1	3,840
e4Eda	Electrodermal activity samples measured by wristband	1	240
e4Hr	Heart rate samples measured by wristband	1	60
e4Temp	Skin temperature samples measured by wristband	1	240
mAcc	Accelerometer samples measured by smartphone	3	1,800
mGps	GPS samples measured by smartphone	3	25
mGyr	Gyroscope samples measured by smartphone	6	1,800
mMag	Magnetometer samples measured by smartphone	3	1,800

(b) Labels

Label	Num. samples
IN VEHICLE	12,603
ON BICYCLE	365
ON FOOT	15,615
STILL	207,833
UNKNOWN	36,470
WALKING	135

access (NOMA) protocols. Although the NOMA approach has recently been studied [3], they cannot accommodate arbitrary given resource constraints since the output dimension of the encoder neural networks should be identical to that of the label, thereby posing a rigid architecture. We tackle this difficulty by allowing the encoders to yield output signals with arbitrary dimensions. Impacts of these channel impairments are included in the inference design from the sensors to the server. We propose a joint training algorithm that optimizes an end-to-end prediction model of the lifelog dataset over arbitrary channel imperfections. Numerical results validate the superiority of the proposed learning strategy.

II. Lifelog Dataset

A. Description

The lifelog dataset [1] aims at investigating human behaviors and their underlying semantic contexts such as emotion, location, and physical/social activities. Features and labels of the considered dataset are summarized in Table I. A variety of multi-modal sensory data samples are obtained by using portable and wearable devices including smartphones and smart wristbands. As shown in Table I(a), the lifelog dataset consists of 9 features of 72 participants collected over at least 12 hours per day. In this study, we consider 22 participants. Each participant was involved in the data collection experiment within 12-30 days. Therefore, the considered dataset can be viewed as time series data samples whose sequence lengths are different for participants. Each data sample is partitioned into 60-second-long segments. Due to the heterogeneous sampling frequency of sensors, the sequence length is different for participants and features. The maximum sequence length of each feature is shown in Table I(a). It is observed that the lifelog dataset requires for processing high-dimensional feature vectors having up to $6 \times 1,800 = 10,800$ elements for the mGyr feature.

Based on these time series features, it is desired to predict user activity labels. Activities of each participant are labeled among 6 different categories as presented in Table I(b). This can be formalized as a classification problem that categorizes an input sample having 9 features into 6 classes. For accurate human behavior prediction, we leverage a deep neural network (DNN) model trained over the lifelog dataset. The number of the samples for each class is provided in Table I(b), resulting in total 140,114 samples. This is split into the train, validation, and test datasets each consisting of 60 %, 20 %, and 20 % samples of the entire dataset.

B. Challenges

Practical wearable devices have no sufficient power to implement real-time computations of very deep architectures. Therefore, it is necessary to adopt the notion of the mobile edge computing network where wearable devices send their sensing signals to a cloud computing server equipped with a classifier DNN. By doing so, we can facilitate remote AI services by offloading AI computations of battery-powered wearable devices to the server having intensive computing units. However, sharing raw sensory samples over wireless/wired media with limited bandwidth invokes excessive communication latency to convey high-dimensional features. To reduce communication overheads, each sensor needs to encode collected samples into low-dimensional representation. This requests a novel DNN architecture that suits device-server networks.

Since the devices and sensors have heterogeneous sampling rates, these time series features are of variable lengths. Features of different participants are measured over a different time duration, thereby leading to variable-length data samples even for the same modality. Handling such user- and modal-wide heterogeneity is one of the major challenges in addressing the lifelog dataset. To address this issue, we need to develop a proper encoding strategy at wearable devices that can handle variable-length sensory signals.

In practice, users might leverage lightweight wearable devices without expensive sensors. Furthermore, several features would be dropped in the communication channel from the devices to the server. This poses random imputations in data samples where some parts or the entire set of each feature is not available both at the devices and the server. Therefore, it is essential to build a valid DNN model robust to random defects in input features.

III. Proposed Cooperative Inference

For the remote prediction of human behaviors, as illustrated in Fig. 1, we consider a client-server learning architecture where each client corresponds to each embedded sensor, e.g., accelerometer, gyroscope, and heart rate sensors. There are

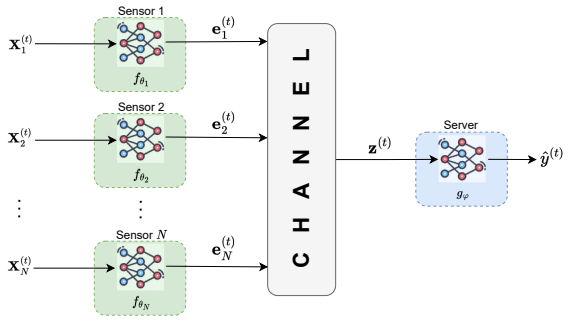

Fig. 1. Proposed cooperative inference system

$N = 9$ different sensors collecting multi-modal features in Table I(a). These sensing signals are then forwarded to the server through uplink backhaul channels for the estimation of labels shown in Table I(b). Total M wireless resource blocks (RBs) are assigned to the uplink coordination where each RB is assumed to convey one real-valued number reliably.

A. Inference Structure

We propose a cooperative inference structure among sensors and server. At a certain sampling time t ($t = 1, \cdots, T$), an input feature of sensor i ($i = 1, \cdots, N$) is represented by a matrix $\mathbf{X}_i^{(t)} \in \mathbb{R}^{s_i^{(t)} \times d_i}$ with time-varying sequence length $s_i^{(t)}$ and feature dimension d_i. To accommodate resource-constrained backhaul channels, each sensor i encodes the sensing data matrix $\mathbf{X}_i^{(t)}$ using the dedicated encoder neural network $f_{\theta_i}(\cdot)$, where θ_i indicates a set of trainable parameters. The resulting output, denoted by $\mathbf{e}_i^{(t)} \in \mathbb{R}^E$ of length E, becomes a low-dimensional embedding vector of raw sensing data $\mathbf{X}_i^{(t)}$. As will be discussed, the embedding dimension E is set according to the number of the RBs M. The encoding step at sensor i is expressed as

$$\mathbf{e}_i^{(t)} = f_{\theta_i}(\mathbf{X}_i^{(t)}). \tag{1}$$

The sensors transmit a set of embedding vectors $\mathbf{e}^{(t)} \triangleq \{\mathbf{e}_i^{(t)} : \forall i\}$ to the server. The time-varying channel transfer function of the backhaul links is denoted by $h^{(t)}(\cdot)$, which captures the impacts of channel impairments and multiple accessing protocols adopted at the sensors. Then, the received signal vector at the server, denoted by $\mathbf{z}^{(t)}$, can be written as

$$\mathbf{z}^{(t)} = h^{(t)}(\mathbf{e}^{(t)}). \tag{2}$$

Based on a classifier $g_\varphi(\cdot)$ with learnable parameter φ, the server obtains an estimate $\hat{y}^{(t)}$ for the ground truth label $y^{(t)}$ of the sensing data matrices $\{\mathbf{X}_i^{(t)} : \forall i\}$ as

$$\hat{y}^{(t)} = g_\varphi(\mathbf{z}^{(t)}) = g_\varphi\left(h^{(t)}\left(\{f_{\theta_i}(\mathbf{X}_i^{(t)}) : \forall i\}\right)\right). \tag{3}$$

As a result, the joint training task of a group of encoders $\{f_{\theta_i}(\cdot) : \forall i\}$ and the classifier $g_\varphi(\cdot)$ is formulated as

$$\min_{\Theta} \mathbb{E}\left[l(y^{(t)}, \hat{y}^{(t)})\right], \tag{4}$$

where $\Theta \triangleq \{\theta_i : \forall i\} \bigcup \varphi$ stands for the set of all training parameters, $l(y^{(t)}, \hat{y}^{(t)})$ indicates a loss function between the label $y^{(t)}$ and its estimate $\hat{y}^{(t)}$, and the expectation operator $\mathbb{E}[\cdot]$ is taken over the training dataset $\mathbf{X} = \{\mathbf{X}_i^{(t)} : \forall i, t\}$ as well as random backhaul channel $h^{(t)}(\cdot)$. The training problem (4) can be readily addressed by the stochastic gradient descent (SGD) method and its variants, e.g., the Adam algorithm.

B. Channel Model

In practice, the backhaul coordination step invokes random imputation in the received signal $\mathbf{z}^{(t)}$. More precisely, data packets conveying a particular embedding signal $\mathbf{e}_i^{(t)}$ would be dropped due to the deep fading case with a severe propagation environment. Such a scenario prevails in the considered human behavior prediction task where some sensors or wearable devices are not available for particular users. We introduce a binary number $b_i^{(t)} \in \{0, 1\}$ to indicate the loss of sensor i at time t. If sensor i is deactivated or its encoded signal $\mathbf{e}_i^{(t)}$ is dropped in the backhaul communications, its status is denoted by $b_i^{(t)} = 0$. Otherwise, i.e., when $\mathbf{e}_i^{(t)}$ is reliably transmitted to the server, we set $b_i^{(t)} = 1$. The channel transfer function $h^{(t)}(\cdot)$ at time t is then determined by a set of binary indicators $\{b_i^{(t)} : \forall i\}$ as well as multiple access schemes of encoded signals $\mathbf{e}_i^{(t)}$, $\forall i$, which will be described in the following.

C. Multiple Access Schemes

According to the RB sharing policy, we consider two multiple access schemes: orthogonal multiple access (OMA) and non-orthogonal multiple access (NOMA). Detailed processes of each scheme are given in the following.

1) Orthogonal Multiple Access: In the OMA protocol, each sensor utilizes exclusive RBs to transmit its embedding signal. To this end, total $M = NE$ RBs are evenly assigned to each sensor. As a consequence, each sensor i can send its embedding signal $\mathbf{e}_i^{(t)} \in \mathbb{R}^E$ to the server without incurring any interference. The received signal vector $\mathbf{z}_{OMA}^{(t)}$ in (2) for the OMA case can be written as

$$\mathbf{z}_{OMA}^{(t)} = \bigoplus_{i=1}^{N} (b_i^{(t)} \mathbf{e}_i^{(t)}), \tag{5}$$

where $\bigoplus_{i=1}^{N} \mathbf{a}_i$ indicates the concatenation operation of vectors $\mathbf{a}_1, \cdots, \mathbf{a}_N$. In (5), the scalar multiplication $b_i^{(t)} \mathbf{e}_i^{(t)}$ stands for the ED layer which drops the embedding signal $\mathbf{e}_i^{(t)}$ according to the Bernoulli random number (7). The dimension M of the received signal vector $\mathbf{z}_{OMA}^{(t)}$ becomes $M = NE$.

2) Non-Orthogonal Multiple Access: The NOMA scheme allows all sensors to share the entire RBs. Thus, the dimension of the embedding signal $\mathbf{e}_i^{(t)}$ is directly set to the number of the total RBs, i.e., $E = M$. The received signal $\mathbf{z}_{NOMA}^{(t)} \in \mathbb{R}^E$ of the NOMA scheme is given as the superposition of all embedding signals as

$$\mathbf{z}_{NOMA}^{(t)} = \sum_{i=1}^{N} b_i^{(t)} \mathbf{e}_i^{(t)}. \tag{6}$$

Compared to the OMA case (5) which needs $M = NE$ RBs, in the NOMA, the total number of the RBs $M = E$

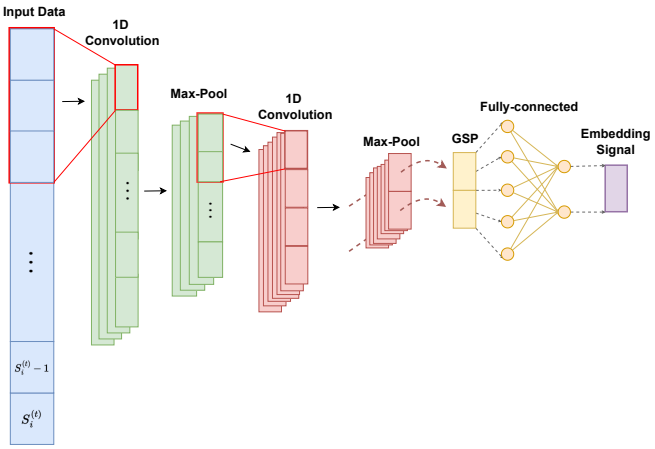

Fig. 2. Encoder architecture.

is no longer proportional to the number of the sensors. Consequently, we can save the backhaul coordination resources and improve resource utilization efficiency. However, at the same time, the multi-sensor interference in (6) might reduce the final classification accuracy. Such a tradeoff relationship will be discussed in the simulation result section.

IV. Proposed Learning Strategy

This section provides a structure of the encoder $f_{\theta_i}(\cdot)$ that can handle variable-length samples $\mathbf{X}_i^{(t)}$. It is then followed by the joint training policy of the encoders and classifier.

A. Model Design

Fig. 2 illustrates the encoder architecture which consists of two convolutional layers and one fully-connected layer. The convolutional layers perform one-dimensional (1D) convolutional operations for the time series input $\mathbf{X}_i^{(t)} \in \mathbb{R}^{s_i^{(t)} \times d_i}$ in the time domain of sequence length $s_i^{(t)}$. The l-th convolutional layer of encoder $f_{\theta_i}(\cdot)$ adopts $C[l]$ filters with stride 1, and it is followed by the max-pooling layer of filter size $(2,1)$. No padding is employed. For the first convolutional layer, we leverage $C[1] = 16$ filters of shape $(5, d_i)$. After the max-pooling, the output shape of the first convolutional layer becomes $(16, S_{i,1}^{(t)}, 1)$, where the first axis indicates the channel of the convolutional layer, $S_{i,1}^{(t)} \triangleq \lceil s_i^{(t)} - 4 \rceil / 2$ is the sequence length of the output of the first convolutional layer, and $\lceil \cdot \rceil$ stands for the ceiling operator.

The second convolutional layer employs $C[2] = 32$ filters of shape $(5,1)$. Combining with the max-pooling layer, the corresponding output shape becomes $(32, S_{i,2}^{(t)}, 1)$ with $S_{i,2}^{(t)} \triangleq \lceil S_{i,1}^{(t)} - 4 \rceil / 2$ being the sequence length of the output of the second convolutional layer. Notice that the output shape depends on sample-wise sequence length $s_i^{(t)}$. To handle such variable-length data, we apply the global sum-pooling (GSP) to the second axis which reduces a sequence for each convolutional channel to a scalar signal. Thus, the output shape of the global sum-pooling becomes $(32, 1, 1)$. We then employ a fully-connected layer to get the encoded signal $\mathbf{s}_i^{(t)}$ of length E.

Next, the classifier $g_\varphi(\cdot)$ at the server consists of three fully-connected layers whose output dimensions are set to 128, 100, and 6, respectively. We employ the rectified linear unit (ReLU) and softmax functions as activation functions at hidden and output layers, respectively.

B. Proposed Training Strategy

We propose a joint training strategy that optimizes encoders $f_{\theta_i}(\cdot)$, $\forall i$, and classifier $g_\varphi(\cdot)$, simultaneously. The impact of the stochastic data imputation $b_i^{(t)}$ in the backhaul coordination (5) and (6) should be injected into the training process for the robust optimization of the classifier $g_\varphi(\cdot)$ at the server. To this end, we propose an embedding dropout (ED) technique that randomly drops the embedding vector, i.e., the encoded signal $\mathbf{s}_i^{(t)}$, during the training. We generate the Bernoulli random variable $b_i^{(t)}$ as

$$b_i^{(t)} = \begin{cases} 0, & \text{with probability } p \\ 1, & \text{with probability } 1-p \end{cases} \tag{7}$$

where $p \in [0,1]$ is a hyperparameter contriving the probability of sensor i being dropped. Based on $b_i^{(t)}$, each embedding signal $\mathbf{e}_i^{(t)}$ is removed in the forward propagation (3) as well as in the gradient calculation. Thus, the ED technique (7) is regarded as a stochastic and non-trainable layer injected between the sensors and sever. Unlike the vanilla dropout layer which deactivates each element of latent vectors independently, the proposed ED layer performs group-wise dropout operations where the entire embedding vector $\mathbf{e}_i^{(t)}$ becomes active or inactive. By doing so, a number of artificial information loss scenarios can be easily injected both in the training and test steps. This brings the generalization ability for the classifier.

Let $\mathbf{X}^{(t)} \triangleq \{\mathbf{X}_i^{(t)} : \forall i\}$ be the set of all sensing signals sampled at time t. Then, the training dataset is defined as $\mathcal{X} = \{(\mathbf{X}^{(t)}, y^{(t)}) : \forall t\}$. The mini-batch stochastic gradient descent (SGD) algorithm is adopted where a mini-batch set $\mathcal{B}$ is uniformly sampled from the training set $\mathcal{X}$, i.e., $\mathcal{B}$ contains a number of tuples $(\mathbf{X}^{(\tau)}, y^{(\tau)})$ of arbitrary time instants $\tau \in \{1, \cdots, T\}$. Thus, the mini-batch set $\mathcal{B}$ can be simply denoted by the subset of all time instants $\mathcal{B} \in \{1, \cdots, T\}$. The mini-batch average loss function is then given as

$$L(\Theta) = \frac{1}{|\mathcal{B}|} \sum_{\tau \in \mathcal{B}} l(y^{(\tau)}, \hat{y}^{(\tau)}), \tag{8}$$

where $|\mathcal{B}|$ is the cardinality of $\mathcal{B}$. We can readily employ the standard SGD algorithm to minimize (8). The update rule of each training epoch is written by

$$\Theta \leftarrow \Theta - \eta \nabla_\Theta L(\Theta) \tag{9}$$

with $\nabla_\Theta L(\Theta)$ being the gradient of the loss function $L(\Theta)$ with respect to Θ.

Algorithm 1 summarizes the proposed training strategy which optimizes the encoder parameters $\theta_1, \cdots, \theta_N$ and the classifier parameter φ iteratively. For each mini-batch sample $\tau \in \mathcal{B}$, each sensor i encodes its sensing signal $\mathbf{X}_i^{(\tau)}$ into $e_i^{(\tau)}$ individually using their encoder $f_{\theta_i}(\cdot)$. We then employ the ED layer for each $\mathbf{e}_i^{(\tau)}$ to mimic the stochastic loss of the

TABLE II
Test accuracy performance.

	$M = 9$		$M = 27$		$M = 45$	
	$p_{test} = 0$	$p_{test} = 0.2$	$p_{test} = 0$	$p_{test} = 0.2$	$p_{test} = 0$	$p_{test} = 0.2$
NOMA w/ ED	78.330	**77.750**	**79.051**	**78.374**	79.144	**78.552**
OMA w/ ED	77.344	76.637	77.808	77.214	78.394	77.817
NOMA w/o ED	**78.620**	76.117	79.034	75.542	**79.399**	76.484
OMA w/o ED	77.417	75.391	78.005	57.275	78.669	62.125

Algorithm 1 Proposed Training Strategy

Initialize the parameters $\theta_1, \cdots \theta_N$, and φ.
for each training epoch **do**
  Sample the mini-batch set $\mathcal{B} \subset \mathcal{X}$.
  for each mini-batch sample $\tau \in \mathcal{B}$ in parallel **do**
    for each sensor $i = 1, 2, \cdots, N$ in parallel **do**
      Calculate $\mathbf{e}_i^{(\tau)}$ from (1).
      Employ the ED layer (7) to $\mathbf{e}_i^{(\tau)}$.
    end for
    Obtain $\mathbf{z}^{(\tau)}$ from (2).
    Predict $\hat{y}^{(\tau)}$ from (3).
  end for
  Calculate the loss function $L(\Theta)$ from (8).
  Update the parameter Θ from (9).
end for

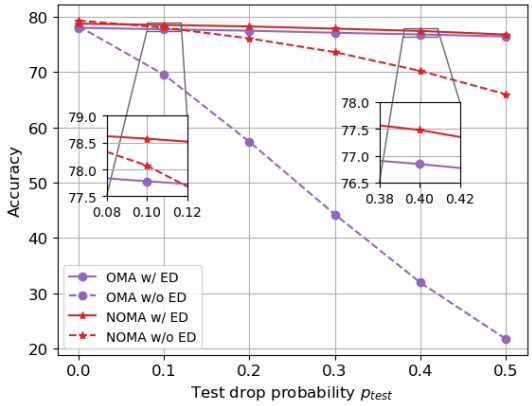

Fig. 3. Accuracy performance with respect to test ED probability p_{test} for $M = 27$ and $p_{train} = 0.2$.

sensor. This randomly drops some of the encoded signals in the training. The signal received by the server $\mathbf{z}^{(\tau)}$ is created based on the OMA (5) or NOMA (6) strategies. Then, we can calculate the estimate $\hat{y}^{(\tau)}$ and the loss function $L(\Theta)$ in (8). This is followed by the SGD update (9) where the gradient $\nabla_\Theta L(\Theta)$ can be computed using the standard backpropagation algorithm. After the training, optimized encoders and classifiers are exploited for the decentralized prediction of sensing data samples.

V. Numerical Results

This section assesses the proposed cooperative learning policy for the lifelog dataset problem. Among the total of 273,921 samples, 164,352 and 54,784 samples are used for the training and validation, respectively, and the remaining 54,785 samples are exploited for the final test. The drop probability p of the ED layer (7) in the training and test phases are denoted by p_{train} and p_{test}, respectively. Unless otherwise stated, we set $p_{train} = 0.2$. We employ the Adam algorithm to optimize encoders and a classifier, with a learning rate $\eta = 8 \times 10^{-6}$ and batch size $4,096$. In the simulations, the number of the RBs M is set to $M \in [9, 27, 45]$.

Table II compares the test accuracy of various schemes with the different numbers of the RBs M and the test packet loss probability p_{test}. Notice that the methods without the ED technique correspond to the case with $p_{train} = 0$, whereas those with the ED are trained with the drop probability $p_{train} = 0.2$. The best results for each M and p_{test} are highlighted by boldface letters. Regardless of M and p_{test}, the NOMA protocol along with the ED technique generally performs better than the OMA counterpart. This validates the effectiveness of the resource sharing policy for the proposed cooperative learning method. With random packet loss, i.e., $p_{test} = 0.2$, the test accuracy slightly decreases compared to the ideal case with $p_{tets} = 0$. Increasing M leads to the improvement of the prediction performance. These results confirm the fact that the trained neural encoders can exploit the communication resources actively to enhance the inference performance at the server.

Fig 3 illustrates the classification accuracy by changing the ED probability p_{test} for $M = 27$. Both the NOMA and OMA schemes with the ED technique are trained at a fixed drop probability $p_{train} = 0.2$. but are applied to a wide range of the test probability $p_{test} \in \{0, 0.1, \cdots, 0.5\}$ unseen in the training phase. It is shown that both methods exhibit a good accuracy performance at any given p_{test}, proving the generalization ability to arbitrary packet loss probabilities. The NOMA with the ED method performs better than the OMA with the ED method. Without the ED technique, the performance of the NOMA and OMA approaches highly degrade as p_{test} gets larger. In particular, the OMA without the ED fails to carry out a valid classification procedure. This validates the effectiveness of the proposed ED technique to get robust encoder and classifier models.

So far, we evaluate the test accuracy for stochastic packet losses with given probability p_{test}. In Fig. 4, we consider deterministic packet loss events where a certain sensor out of a total of nine sensors is dropped. The x-axis stands for the missing sensor. The NOMA with the ED method exhibits almost constant classification accuracy for all simulated cases. On the contrary, the performance of the OMA without the ED method highly fluctuates for the elimination of each sensor,

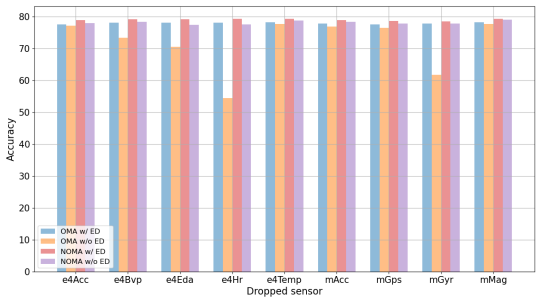

Fig. 4. Accuracy performance with deterministic packet loss for $M = 27$ and $p_{train} = 0.2$.

in particular, e4HR and mGyr features. Therefore, we can conclude that the NOMA protocol along with the ED layer is essential to learn effective sensor-server cooperation policies for arbitrary packet loss scenarios.

VI. Conclusions

In this paper, we have proposed cooperative learning strategies for the lifelog dataset in which multiple sensors and a server collaboratively estimate human activities based on multi-modal sensing signals. To this end, each sensor exploits neural encoder that encapsulates real-time measurements to low-dimensional messages conveyed to the server. Variable-length sensing signals can be handled by the proposed encoder structure having convolutional layers followed by the GSP operation. In addition, by leveraging the ED technique, our proposed system becomes robust to random drops of sensors and encoded signals in the sensor-server communication step. Furthermore, we have developed orthogonal and non-orthogonal types of resource sharing strategies for efficient backhaul coordination. Simulation results have confirmed that the proposed method with the ED technique can improve the robustness to the random or particular losses of sensing signals.

References

[1] S. Chung, C. Y. Jeong, J. M. Lim, J. Lim, K. J. Noh, G. Kim, and H. Jeong, "Real-world multimodal lifelog dataset for human behavior study," *ETRI J.*, vol. 43, no. 6, pp. 426–437, Jun. 2021.

[2] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The EPIC-KITCHENS dataset," *in Proc. Eur. Conf. Comput. Vis. (ECCV)*, p. 720–736, 2018.

[3] J. Zhang, S. Guo, Z. Qu, D. Zeng, H. Wang, Q. Liu, and A. Y. Zomaya, "Adaptive vertical federated learning on unbalanced features," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4006–4018, Dec. 2022.

[4] Y. Hu, D. Niu, J. Yang, and S. Zhou, "FDML:A collaborative machine learning framework for distributed features," *in Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, p. 2232–2240, 2019.

[5] T. Chen, X. Jin, Y. Sun, and W. Yin, " VAFL:A method of vertical asynchronous federated learning ," *arXiv:2007.06081*, 2020.