Wireless Data Acquisition for Edge Learning: Data-Importance Aware Retransmission

Dongzhu Liu, Guangxu Zhu, Jun Zhang, and Kaibin Huang

Abstract

By deploying machine-learning algorithms at the network edge, edge learning can leverage the enormous real-time data generated by billions of mobile devices to train AI models, which enable intelligent mobile applications. In this emerging research area, one key direction is to efficiently utilize radio resources for wireless data acquisition to minimize the latency of executing a learning task at an edge server. Along this direction, we consider the specific problem of retransmission decision in each communication round to ensure both reliability and quantity of those training data for accelerating model convergence. To solve the problem, a new retransmission protocol called *data-importance aware automatic-repeat-request* (importance ARQ) is proposed. Unlike the classic ARQ focusing merely on reliability, importance ARQ selectively retransmits a data sample based on its *uncertainty* which helps learning and can be measured using the model under training. Underpinning the proposed protocol is a derived elegant communication-learning relation between two corresponding metrics, i.e., *signal-to-noise ratio* (SNR) and data uncertainty. This relation facilitates the design of a simple threshold based policy for importance ARQ. The policy is first derived based on the classic classifier model of *support vector machine* (SVM), where the uncertainty of a data sample is measured by its distance to the decision boundary. The policy is then extended to the more complex model of *convolutional neural networks* (CNN) where data uncertainty is measured by entropy. Extensive experiments have been conducted for both the SVM and CNN using real datasets with balanced and imbalanced distributions. Experimental results demonstrate that importance ARQ effectively copes with channel fading and noise in wireless data acquisition to achieve faster model convergence than the conventional channel-aware ARQ. The gain is more significant when the dataset is imbalanced.

I. Introduction

With the prevalence of smartphones and *Internet-of-Things* (IoT) sensors on the network edge, known as *edge devices*, people envision an incoming new world of ubiquitous computing and ambient intelligence. This vision motivates Internet companies and telecommunication operators to

D. Liu, G. Zhu, and K. Huang are with the Dept. of Electrical and Electronic Engineering at The University of Hong Kong, Hong Kong. J. Zhang is with the Dept. of Electronic and Information Engineering at the Hong Kong Polytechnic University, Hong Kong. Corresponding author: K. Huang (email: huangkb@eee.hku.hk).

develop technologies for deploying machine learning on the (network) edge to support intelligent mobile applications, named as *edge learning* [1]–[4]. This trend aims at leveraging enormous real-time data generated by billions of edge devices to train AI models. In return, downloading the learnt intelligence onto the devices will enable them to respond to real-time events with human-like capabilities. Edge learning crosses two disciplines, wireless communication and machine learning, which cannot be decoupled as their performances are interwound under a common goal of fast learning.

As data-processing speeds are increasing rapidly, wireless acquisition of high-dimensional training data from many edge devices has emerged to be a bottleneck for fast edge learning, which faces the challenges due to high mobility and unreliable devices (see e.g., [5]). This calls for designing highly efficient techniques for radio resource management targeting edge learning. For conventional techniques, data bits (or symbols) are assumed of equal importance, which simplifies the design criterion to be rate maximization but fails to exploit the features of learning. In contrast, for learning, *the importance distribution in a training dataset is non-uniform*, namely that some samples are more important than others. For instance, for training a classifier, the samples near decision boundaries are more critical than those far away [6]. This fact motivates the proposed design principle of *importance-aware resource allocation*. In this work, we apply this principle to redesign the classic technique of *automatic repeat-request* (ARQ) for efficient wireless data acquisition in edge learning.

A. Wireless Communications for Edge Learning

Conventional communication techniques are designed mostly for either reliable transmission or data-rate maximization without awareness of data utility for learning. Such a “communication-learning separation” principle does not yield efficient solutions for acquiring large-scale distributed data in edge learning. Its increasingly critical communication bottleneck calls for re-designing communication techniques with a new objective of low-latency execution of learning tasks. Research opportunities in this largely uncharted area can be roughly grouped under three topics: radio resource allocation, multiple access, and signal encoding. The new idea in radio resource allocation for edge learning, the topic of our interest, is to consider data usefulness for learning in allocating resources for data uploading from devices to a server [7]. In this paper, we consider retransmission which is a simple time-allocation method for ensuring reliable communication in the presence of channel hostility [8]. The widely used ARQ protocols repeat the transmission of a data packet until it is reliably received. Thereby, channel uses are allocated to packets under a reliability constraint. While existing ARQ designs purely target data reliability [9], [10], accelerating edge learning calls for new protocols incorporating the new feature of considering data importance in retransmission decision. This motivates our work.

The second key topic in the area is low-latency multi-access for distributed edge learning. Recent research focuses on federated learning, where edge devices transmit their local model updates to collaboratively update the global AI model by aggregation at the server [11]. One idea proposed recently is to perform "over-the-air" aggregation by exploiting the waveform superposition property of a multi-access channel [12]–[14]. Such a scheme allows simultaneous access and hence can dramatically reduce multi-access latency.

Last, signal encoding for communication efficient edge learning represents another research thrust. Relevant research aims at integrating feature extraction, source coding, and channel encoding to compress transmitted data without significantly compromising learning performance. Examples include analog encoding on Grassmannian for high mobility data classification [15] and quantized stochastic gradient descent [16].

B. Wireless Data Acquisition

Efficient data acquisition is a classic topic in designing *wireless sensor network* (WSN) with a rich literature [17]–[21]. The main challenge is how to overcome the energy constraints of sensors to allow fusion centers to collect distributed sensing data without interruptions. There exist diversified solutions such as wireless power transfer [17], multi-hop transmission [18], [19], and UAV-assisted data collection [20]. One approach that shares the same spirit as the current work is to schedule sensors based on their data quality evaluated using criteria including cost, sensing accuracy and timeliness [21]. On the other hand, the ARQ protocol proposed in the current work also involves data evaluation which, however, is based on a different criterion, namely importance for learning. Overall, data utilization (i.e., computing or learning) is considered out of scope in prior work and not accounted for in existing techniques for data acquisition, leaving some space for performance improvement.

In machine learning, one topic relevant to data acquisition is *active learning* [6]. Consider the scenario where unlabeled data are abundant but manually labeling is expensive. Active learning aims to selectively label informative data (by querying an oracle), such that a model can be trained using as few labelled data samples as possible, thus reducing the labelling cost. Roughly speaking, the informativeness of a sample is related to how uncertain the prediction of this sample is under the current model. To be specific, the more uncertain the prediction is, the more useful the sample can be for model learning. Several commonly used uncertainty measures are *entropy* [22], *expected model change* [23], and *expected error reduction* [24]. In active learning, wireless communication is irrelevant. However, the uncertainty measures developed therein are useful for this work and integrated with a retransmission protocol to enable intelligent data acquisition in an edge learning system.

C. Contributions and Organization

This work concerns wireless data acquisition in edge learning. In this work, we propose a new retransmission protocol called *data-importance aware ARQ*, or *importance ARQ* for short, which adapts retransmission decisions to both data importance and reliability (or equivalently the channel state). As a result, the allocation of channel uses is biased towards protecting important data samples against channel noise while ensuring the quantity of acquired data. Balancing the two aspects in the design results in the combined effects of accelerating model convergence and reducing the required budget of channel uses. To the authors' best knowledge, this work represents the first attempt on exploiting the non-uniform distribution of data informativeness to improve the communication efficiency of an edge learning system.

The main contributions of this work are summarized as follows.

- **Importance ARQ for SVM:** First, consider the classic classifier model of *support vector machine* (SVM). The importance ARQ is designed to improve the quality-vs-quantity tradeoff. The protocol selectively retransmits a data sample based on its underlying importance for training an SVM model which is estimated using the real-time model under training. For SVM, a suitable importance measure is proposed to be the shortest distance from a data sample to decision boundaries. The theoretical contribution of the design lies in a derived *elegant communication-learning relation* between two corresponding metrics, i.e., *signal-to-noise ratio* (SNR) and *data importance*, for targeted learning performance. This new relation facilitates the design of a simple threshold based policy for making retransmission decisions, where the SNR threshold is shown to be proportional to the importance measure.
- **Extension to general classifiers:** The derived importance-ARQ policy for SVM models is extended to general classifier models. Particularly, the SNR threshold is designed to be proportional to a monotonically increasing *reshaping function* of a general importance

measure. The design captures the heuristic that more important data should be better protected against noise by a higher target SNR. Moreover, general guidelines on how to select the reshaping function and the SNR-importance scaling factor are discussed. Subsequently, a case study on designing importance ARQ for the modern *convolutional neural networks* (CNN) classifier is presented.

- **Experiments:** We evaluate the performance of the proposed importance ARQ via extensive experiments using real datasets with balanced and imbalanced distribution. The results demonstrate that the proposed method avoids learning performance degradation caused by channel fading and noise while achieving faster convergence than the conventional channel-aware ARQ. Furthermore, the performance gain is found to be more significant for the imbalanced data distribution.

The remainder of the paper is organized as follows. Section II introduces the communication and learning models. Section III presents some initial experimental results and motivates the design of an intelligent retransmission protocol. The principle of importance ARQ is proposed for SVM in Section IV. It is extended to general classifiers in Section V. Section VI provides experimental results, followed by concluding remarks in Section VII.

II. Communication and Learning Models

In this section, we first introduce the communication system model and learning models. Then data uncertainty metrics are defined for different learning models.

A. Communication System Model

We consider an edge learning system as shown in Fig. 1 comprising an edge server and multiple edge devices, each equipped with a single antenna. A machine learning classifier is trained at the server using a labelled dataset distributed over devices. Denote the k-th data sample $(\mathbf{x}_k, c_k)$ with $\mathbf{x}_k \in \mathbb{R}^p$, p its dimensions, and $c_k \in \{1, 2, \cdots, C\}$ its label. The devices time share the channel and take turn to transmit local data to the server. The time sharing is coordinated by a channel-aware scheduler while importance-aware scheduling is noted to be an interesting direction for future investigation. Note that a label has a much smaller size than a data sample (e.g., a $0-9$ integer versus a vector of a million coefficients). Thus two separate channels are planned: a low-rate *label channel* and a high-rate *data channel*. The former is assumed to be noiseless for simplicity. Reliable uploading of data samples over the noisy and

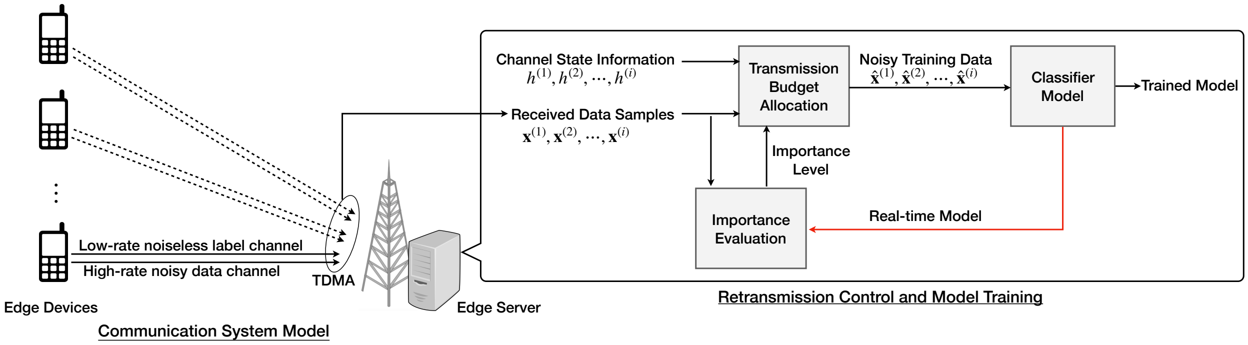

Edge Devices
Low-rate noiseless label channel
High-rate noisy data channel
TDMA
Communication System Model
Edge Server
Channel State Information
$h^{(1)}, h^{(2)}, \cdots, h^{(i)}$
Received Data Samples
$\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(i)}$
Transmission Budget Allocation
Noisy Training Data
$\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \cdots, \hat{\mathbf{x}}^{(i)}$
Classifier Model
Trained Model
Importance Level
Importance Evaluation
Real-time Model
Retransmission Control and Model Training

Figure 1. An edge learning system.

fading channel is the bottleneck of wireless data acquisition and the focus of this work. Time is slotted into symbol durations, called *slots*. Transmission of a data sample requires p slots, called a *symbol block*.

Upon receiving a data sample, the edge server makes a *binary decision* on whether to request a retransmission to improve the sample quality or a new sample from the scheduled device. The decision is communicated to the device by transmitting either a *positive ACK* or a *negative ACK*. The device is assumed to have backlogged data. Upon receiving a request from the server, the device transmits either the previous sample or a new sample randomly picked from its buffer.

The data channel is assumed to follow block-fading, where the channel coefficient remains static within a symbol block and is *independent and identically distributed* (i.i.d.) over different blocks. The transmit data sample $\mathbf{x} = [X_1, X_2, \cdots, X_p]^{\mathsf{T}}$ is a random vector. During the i-th symbol block, the active device sends the data $\mathbf{x}^{(i)}$ using linear analog modulation, yielding the received signal given by

$$\mathbf{y}^{(i)} = \sqrt{P} h^{(i)} \mathbf{x}^{(i)} + \mathbf{z}^{(i)}, \tag{1}$$

where P is the transmit power, the channel coefficient $h^{(i)}$ is a complex *random variable* (r.v.) with a unit variance, and $\mathbf{z}^{(i)}$ is the *additive white Gaussian noise* (AWGN) vector with the entries following i.i.d. $\mathcal{CN}(0, \sigma^2)$ distributions. Analog uncoded transmission is assumed not only for tractability but also to allow fast data transmission [25] and a higher energy efficiency (compared with the digital counterpart) [26]. We assume that perfect *channel state information* (CSI) on $h^{(i)}$ is available at the server. This allows the server to compute the instantaneous SNR of a received data sample and make the retransmission decision.

1) Retransmission Combining: To exploit the time-diversity gain provided by multiple independent noisy observations of the same data sample from retransmissions, the *maximal-ratio combining* (MRC) technique is used to coherently combine all observations for maximizing the receive SNR. To be specific, consider a data sample $\mathbf{x}$ retransmitted T times. All T received copies, say from symbol block $n+1$ to $n+T$, can be combined by MRC to acquire the received sample, denoted as $\hat{\mathbf{x}}(T)$, as follows:

$$\hat{\mathbf{x}}(T) = \frac{1}{\sqrt{P}} \Re \left(\sum_{i=n+1}^{n+T} \frac{(h^{(i)})^*}{\sum_{m=n+1}^{n+T} |h^{(m)}|^2} \mathbf{y}^{(i)} \right), \tag{2}$$

where $\mathbf{y}^{(i)}$ is given in (1). In (2), we extract the real part of the combined signal for further processing since the data for learning are real-valued in general (e.g., photos, voice clips or video clips). As a result, the effective receive SNR for $\hat{\mathbf{x}}(T)$ after combining is given as

$$\mathsf{SNR}(T) = \frac{2P}{\sigma^2} \sum_{i=n+1}^{n+T} |h^{(i)}|^2, \tag{3}$$

where the coefficient 2 at the right-hand side arises from the fact that only the noise in the real dimension with variance $\frac{\sigma^2}{2}$ affects the received data. The summation in (2) has a value growing as the number of retransmissions T increases. The SNR expression in (3) measures the reliability of a received data sample and serves as a criterion for making the retransmission decision as discussed in Section IV.

2) Latency Constrained Transmission: Either due to the application-specific latency requirement for the learning task or limited radio resources, the objective of designing the communication system is to minimize the duration of wireless data acquisition or equivalently maximize the speed of model convergence. Under this objective, the retransmission protocol is designed in the sequel to bias channel-use allocation towards providing better protection for more important data samples against channel noise.

B. Learning Models

For the learning task, we consider supervised training of a classifier. Prior to training, we assume that the edge server has a small set of clean observed samples, denoted as $\mathcal{L}_0$. This allows the construction of a coarse initial classifier, which is used for making retransmission decisions at the beginning. The classifier is refined progressively in the data acquisition (and training) process. In this paper, we consider two widely used classifier models, i.e., the classic SVM classifier and the modern CNN classifier as introduced below.

1) SVM Model: As shown in Fig. 2, the SVM algorithm is to seek an optimal hyperplane $\mathbf{w}^{\mathsf{T}}\mathbf{x} + b = 0$ as a decision boundary by maximizing its margin γ to data points, i.e., the minimum distance between the hyperplane to any data sample [27]. The points lie in the margin

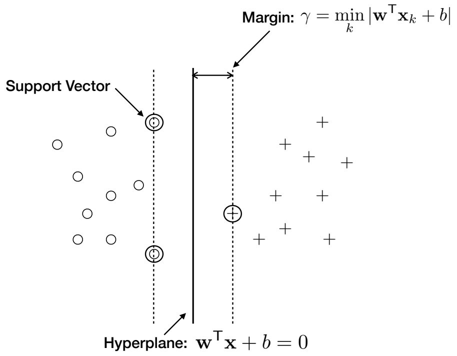

Margin: $\gamma = \min_k |\mathbf{w}^\mathsf{T}\mathbf{x}_k + b|$
Support Vector
Hyperplane: $\mathbf{w}^\mathsf{T}\mathbf{x} + b = 0$

Figure 2. A binary SVM-classifier model.

are referred to as *support vectors* which directly determine the decision boundary. Let $(\mathbf{x}_k, c_k)$ denote the k-th data-label pair in the training dataset. A convex optimization formulation for the SVM problem is given as

$$\min_{\mathbf{w},b} \quad \|\mathbf{w}\|^2 \tag{4}$$

$$\text{s.t.} \;\; c_k(\mathbf{w}^\mathsf{T}\mathbf{x}_k + b) \geq 1, \quad \forall k. \tag{5}$$

The original SVM works only for linearly separable datasets, which is hardly the case when the dataset is corrupted by channel noise in the current scenario. To enable the algorithm to cope with a potential outlier caused by noise, a variant of SVM called *soft margin SVM* is adopted. The technique is widely used in practice to classify a noisy dataset that is not linearly separable by allowing misclassification but with an additional penalty on the objective in (4) (see [27] for details). After training, the learnt SVM model can be used for predicting the label of a new sample, denoted by $\mathbf{x}$, by computing its output score. The binary-classification case is as follows:

$$(\textbf{Output Score}) \qquad s(\mathbf{x}) = (\mathbf{w}^\mathsf{T}\mathbf{x} + b)/\|\mathbf{w}\|, \tag{6}$$

where $\|\cdot\|$ represents the Euclidean norm and s(**x**) is a normalized score. Then the sign of the output score yields the prediction of the binary label.

2) CNN model: CNN is made up of neurons that have adjustable weights and biases to express a non-linear mapping from an input data sample to class scores as outputs [28]. Fig. 3 illustrates the implementation of CNN, which consists of an input and an output layers, as well as multiple hidden layers. The hidden layers of a CNN typically include convolutional layers, ReLu layers, pooling layers, fully connected layers and normalization layers. Without the explicitly defined decision boundaries as for SVM, CNN adjusts the parameters of hidden layers to minimize the prediction error, calculated using the outputs of the softmax layer and the true labels of training

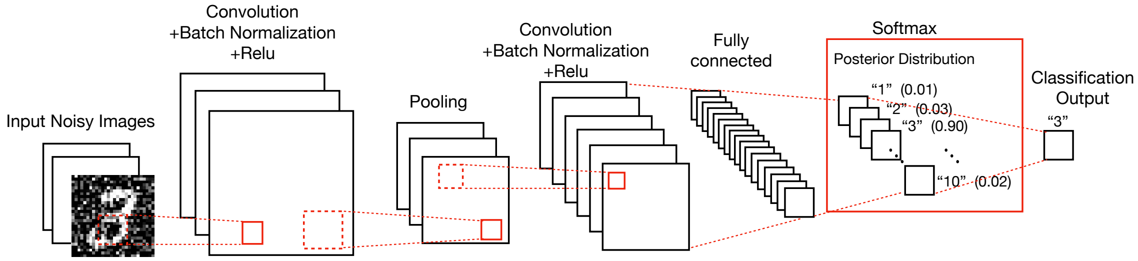


Figure 3. A CNN classifier model.

data. After training, the learnt CNN model can then be used for predicting the label of a new sample by choosing one with the highest posterior probability, which is obtained from the outputs of the softmax layer.

C. Data Uncertainty Metrics

The importance of a data sample for learning is usually measured by its *uncertainty*, as viewed by the model under training [6]. Two uncertainty measures targeting SVM and CNN respectively are introduced as follows.

1) Uncertainty Measure for SVM: For SVM, the uncertainty measure of a data sample is synonymous with its distance to the decision boundary [29]. The definition is motivated by the fact that a classifier makes less confident inference on a data sample which is located near the decision boundary. Based on this fact, we measure the uncertainty of a data sample by the inverse of its distance to the boundary. Given a data sample $\mathbf{x}$ and a binary classifier, the said distance can be readily computed by the absolute value of the output score [see (6)] as follows

$$d(\mathbf{x}) = |s(\mathbf{x})| = |\mathbf{w}^{\mathsf{T}}\mathbf{x} + b|/\|\mathbf{w}\|. \tag{7}$$

Then the distance based uncertainty measure is defined as

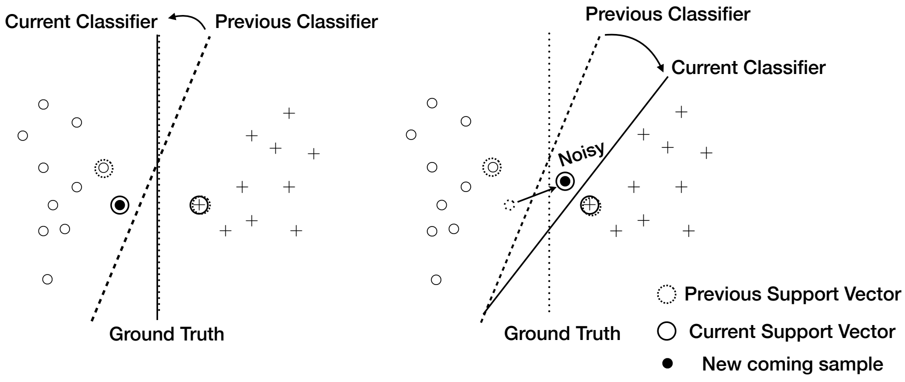
$$\mathcal{U}_{\mathsf{d}}\left(\mathbf{x}\right) = \frac{1}{d^2(\mathbf{x})} = \|\mathbf{w}\|^2/|\mathbf{w}^{\mathsf{T}}\mathbf{x} + b|^2. \tag{8}$$

One can observe that the measure diverges as a data sample approaches the decision boundary, and it reduces as the sample moves away from the boundary.

2) Uncertainty Measure for CNN: For CNN, a suitable measure is *entropy*, an information theoretic notion, defined as follows [22]:

$$\mathcal{U}_{\mathsf{e}}\left(\mathbf{x}\right) = -\sum_{c} P_{\theta}\left(c|\mathbf{x}\right) \log P_{\theta}\left(c|\mathbf{x}\right), \tag{9}$$

where c denotes a class label and θ the set of model parameters to be learnt. To be precise, the model parameters are the weights and biases of the neurons in CNN.



(a) Noiseless data.

(b) Noisy data.

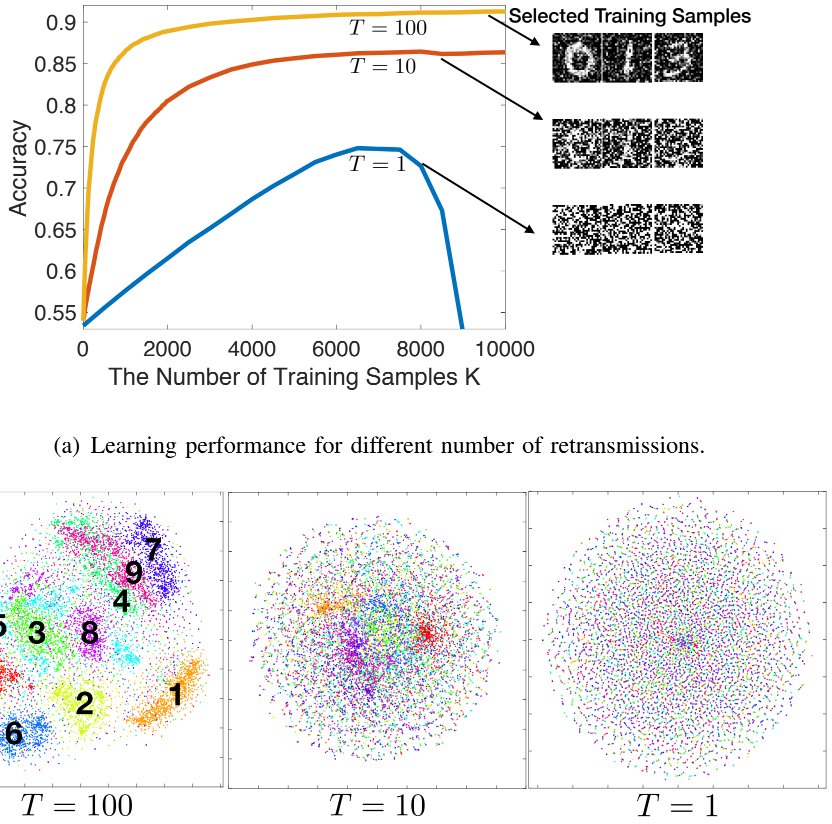
Figure 4. Illustration of the data-label mismatch issue for SVM.

III. Wireless Data Acquisition by Retransmission

A. Why Retransmission is Needed?

Given a noisy data channel and a reliable label channel, the classifier model at the edge server is trained using noisy data samples with correct labels. The channel noise and fading can cause a data sample to cross the ground-truth decision boundary, thereby resulting a mismatch between the sample and its label, referred to as a *data-label mismatch*. The issue can cause incorrect learning as illustrated in Fig. 4. Specifically, for the noiseless transmission case in Fig. 4(a), the new data sample helps refine the current decision boundary to approach the ground-truth one. However, for the case of noisy transmission in Fig. 4(b), noise corrupts the new sample and causes it to be an outlier falling into the opposite (wrong) side of the decision boundary. The situation will be exacerbated when the SVM classifier is used since the outlier may be selected as the supporter vector (or indirectly affect the decision boundary by increasing the penalty in soft-margin SVM).

Retransmission can exploit time diversity to suppress channel noise and fading so as to improve data reliability and hence the learning performance. To visualize the benefit of retransmission, we compare in Fig. 5 the performance of classifiers which are trained using the noise corrupted dataset with a varying number of retransmissions. Specifically, we consider the learning task of handwritten digit recognition using the well-known MNIST dataset that consists of 10 categories ranging from digit “0” to “9” [30]. The level of channel-noise is controlled by the average transmit SNR which is set as $\bar{\rho} = 4\text{dB}$. We train three SVM classifiers with different fixed numbers of retransmissions: $T = 1, 10, 100$. The curves of their test accuracy are shown in



(a) Learning performance for different number of retransmissions.

(b) Visualization of received 10000 noisy training samples.

Figure 5. The impact of retransmission on the accuracy of the learnt model.

Fig. 5(a), with the corresponding received dataset distribution visualized in Fig. 5(b) using the classic *t-distributed stochastic neighbor embedding* (t-SNE) algorithm for projecting the images onto the horizontal plane. It is observed from the case without retransmission ($T = 1$), after receiving a certain number (i.e., 8000) of noisy data samples, the training of the classifier fails as reflected in the abrupt drop in test accuracy. The reason is that the strong noise effect [see Fig. 5(a)] accumulates to cause the divergence of the model [see Fig. 5(b)]. As the number of retransmission increases, the noise effect is subdued to a sufficiently low level ensuring that the class structure of the ideal dataset can be resolved, leading to a converged model and a high test accuracy. The experiment demonstrates the effectiveness of retransmission in edge learning. To further improve learning performance and more efficiently utilize the transmission budget, retransmission should be adapted to the importance levels of individual data samples, which is the focus of the remainder of the paper.

B. Problem Statement

The objective of designing importance ARQ is to adapt retransmission to both the data importance and the channel state so as to efficiently utilize the finite transmission budget for optimizing the learning accuracy. The challenges faced by the design are reflected in two issues described as follows.

- *Quality-vs-Quantity Tradeoff*: The learning performance can be improved by either increasing the reliability (quality) of the wirelessly transmitted training dataset by more retransmissions, or increasing its size (quantity) by acquiring more data samples at the cost of their quality. Given a limited transmission budget, a tradeoff exists between the two aspects, called the *quality-vs-quantity tradeoff*. An efficient retransmission design must exploit the tradeoff to optimize the learning performance.
- *Non-uniform Data Importance*: In conventional data communication, bits are implicitly assumed to have equal importance. This is not the case for training a classifier where data samples with higher uncertainty are more informative and thus more important than those with lower uncertainty. Considering the non-uniform importance in training data provides a new dimension for improving the communication efficiency, which should be also leveraged in the design.

IV. Data-Importance Aware Retransmission

In this section, we consider the task of training an SVM classifier at the edge. First, the concept of noisy data alignment is introduced to relate wireless transmission and learning performance. By applying a relevant constraint, the importance ARQ protocol is derived to intelligently allocate channel uses to the acquisition of individual data samples so as to accelerate model convergence. The protocol is first designed for binary classification and then extended to multi-class classification.

A. Probability of Noisy Data Alignment

The direct design of importance ARQ for optimizing the learning performance is difficult as there lacks a tractable mapping from data quality to learning accuracy. In this section, the difficulty is overcome by deriving a condition for retaining the usefulness of received data for learning in the presence of channel noise, which can differentiate data importance levels. The condition is derived based on the following fact: *a noisy received data sample can mislead* *the model training if its label as predicted by the model differs from the ground truth received without noise*. To avoid this problem in the context of SVM, a pair of transmitted and received data samples should be forced to lie at the same side (ground-truth) of the decision hyperplane of the classifier model so that they have the same predicted labels. This event is referred to as *noisy data alignment* and denoted as $\mathcal{A}$. Its probability is called the *data-alignment probability*. From the distance based uncertainty defined in (8) for SVM, one can see that data samples with higher uncertainty are more vulnerable to noise corruption. To be specific, a small noise perturbation can push a highly uncertain data sample across the decision boundary to result in the aforementioned data-label mismatch (see Fig. 4). The high vulnerability of important data is the reason that importance ARQ allocates more resources to ensure their reliability, giving the protocol its name. The objective of designing importance ARQ is to satisfy a constraint on the data-alignment probability.

Next, the data-alignment probability is defined mathematically for a binary classifier. Since the ground-truth model is unknown, the occurrence of the event $\mathcal{A}$ is evaluated using the current model under training as a *surrogate*. As a result, the output scores defined in (6) must yield the same signs for a pair of transmitted and received data samples if they are aligned. Consider an arbitrary transmitted data sample $\mathbf{x}$ and its received version $\hat{\mathbf{x}}(T)$ after T retransmissions as defined in (2). The event $\mathcal{A}$ is specified as

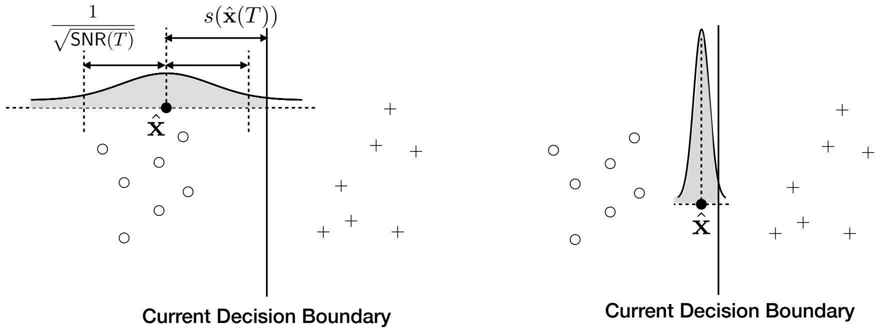
$$\{\mathcal{A} \mid s(\mathbf{x})s(\hat{\mathbf{x}}(T)) > 0\}. \tag{10}$$

Then data alignment probability can be mathematically defined as follows.

Definition 1 (Data-alignment probability)**.** Conditioned on the received data sample, the *data-alignment probability* is defined as:

$$\mathcal{P}\left(\hat{\mathbf{x}}(T)\right) = \mathsf{Pr}\left(\mathcal{A} \mid \hat{\mathbf{x}}(T)\right). \tag{11}$$

The remainder of the sub-section is focused on analyzing the probability. To begin with, the distribution of the transmitted sample score $s(\mathbf{x})$ conditioned on the received data sample $\hat{\mathbf{x}}(T)$ can be obtained from the conditional distribution of the transmitted sample, i.e., $p(\mathbf{x}|\hat{\mathbf{x}}(T))$, as derived below.


$\frac{1}{\sqrt{\mathsf{SNR}(T)}}$
$s(\hat{\mathbf{x}}(T))$
$\hat{\mathbf{x}}$
Current Decision Boundary
$\hat{\mathbf{x}}$
Current Decision Boundary

(a) Small uncertainty.

(b) Lagre uncertainty.

Figure 6. Illustration of the probability of noisy data alignment.

Lemma 1. Conditioned on the received sample $\hat{\mathbf{x}}(T)$, the distribution of the transmitted sample $\mathbf{x}$ follows a Gaussian distribution:

$$\mathbf{x}|\hat{\mathbf{x}}(T) \sim \mathcal{N}\left(\hat{\mathbf{x}}(T), \frac{1}{\mathsf{SNR}(T)}\mathbf{I}\right), \tag{12}$$

where $\mathsf{SNR}(T)$ is the effective SNR given in (3).

Proof: See Appendix A.

With the result, the useful distribution $p(s(\mathbf{x})|\hat{\mathbf{x}}(T))$ can be readily derived using the linear relationship in (6). The derivation simply involves projecting the high-dimensional Gaussian distribution onto a particular direction specified by $\mathbf{w}$, which yields a univariate Gaussian distribution of dimension one as elaborated below.

Lemma 2. Conditioned on the estimated sample $\hat{\mathbf{x}}(T)$, the distribution of the transmitted sample score $s(\mathbf{x})$ follows a unit-variate Gaussian distribution, given by

$$s(\mathbf{x})|\hat{\mathbf{x}}(T) \sim \mathcal{N}\left(s(\hat{\mathbf{x}}(T)), \frac{1}{\mathsf{SNR}(T)}\right). \tag{13}$$

Based on Lemma 2, the data-alignment probability is presented in the following proposition.

Proposition 1. Consider the training of a binary SVM classifier at the edge. Conditioned on the received sample $\hat{\mathbf{x}}(T)$, the data-alignment probability is given as

$$\mathcal{P}\left(\hat{\mathbf{x}}(T)\right) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\sqrt{\mathsf{SNR}(T)} \times \frac{|s(\hat{\mathbf{x}}(T))|}{\sqrt{2}}\right)\right], \tag{14}$$

where $\operatorname{erf}(\cdot)$ is the well known error function defined as $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$.

Proof: As shown in Fig. 6, the conditional distribution for the transmitted data score $s(\mathbf{x})$ is a Gaussian and the probability of data alignment is equal to the area shaded in grey. Mathematically, the probability can be derived using Lemma 2 as follows:

$$\mathcal{P}\left(\hat{\mathbf{x}}(T)\right) = 0.5 + \sqrt{\frac{\mathsf{SNR}(T)}{2\pi}} \int_0^{|s(\hat{\mathbf{x}}(T))|} e^{-\mathsf{SNR}(T)\frac{t^2}{2}} dt. \tag{15}$$

The integral therein can be further expressed using the error function $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$. □

Remark 1. (How does retransmission affects noisy data alignment?) Retransmission contributes to increasing the data-alignment probability. Specifically, retransmission affects both the mean and variance of the conditional distribution $p(s(\mathbf{x})|\hat{\mathbf{x}}(T))$ in (13). From the mean perspective, retransmission helps align the average of retransmitted samples with its ground truth. To be specific, the received estimate approaches the ground-truth value as the number of retransmissions grows:

$$\lim_{T\to\infty} s(\hat{\mathbf{x}}(T)) \to s(\mathbf{x}). \tag{16}$$

From the variance perspective, retransmission continuously reduces the variance by increasing the receive SNR or equivalently the number of retransmissions T. Particularly, it follows from the definition of SNR [see (3)] that

$$\frac{1}{\mathsf{SNR}(T)} = O(1/T) \quad \text{and} \quad \lim_{T\to\infty} \frac{1}{\mathsf{SNR}(T)} \to 0. \tag{17}$$

Combining the two aspects, one can further apply the Chernoff bound to (15) and obtain:

$$\mathcal{P}\left(\hat{\mathbf{x}}(T)\right) = 1 - O(e^{-aT}), \tag{18}$$

where $a > 0$ is a positive constant. As a result, the probability of noisy data alignment approaches one at an exponential rate as T grows.

Last, given the data alignment probability in (14), it is ready to specify the aforementioned condition for ensuring the usefulness of wirelessly acquired data for learning as the following constraint on a received sample $\hat{\mathbf{x}}(T)$ with T retransmissions:

$$(\textbf{Data Alignment Constraint}) \quad \mathcal{P}\left(\hat{\mathbf{x}}(T)\right) > p_c, \tag{19}$$

where $p_c \in (0.5, 1)$ is a given constant.

B. Importance ARQ for Binary Classification

In this section, the importance ARQ protocol is designed for binary SVM classification under the data alignment constraint in (19) and the optimal control policy is proved to have a threshold based structure.

First, it is shown that the constraint in (19) leads to a varying receive-SNR constraint on a data sample that depends on its importance level. The result is given below, which follows directly from the monotonicity of the error function.

Proposition 2. Consider the training of a binary SVM classifier at the edge. For a received data sample $\hat{\mathrm{x}}(T)$, the data alignment constraint in (19) is satisfied if and only if the receive SNR exceeds an importance based threshold:

$$\mathsf{SNR}(T) > \theta_0\, \mathcal{U}_{\mathsf{d}}\left(\hat{\mathrm{x}}(T)\right), \tag{20}$$

where $\mathcal{U}_{\mathsf{d}}\left(\cdot\right)$ is the uncertainty measure given in (8) and $\theta_0 = \left[\sqrt{2}\mathrm{erf}^{-1}\left(2p_c - 1\right)\right]^2$.

It is remarked that the scaling factor θ_0 in (20) can be interpreted as a conversion ratio specifying the rate at which the uncertainty measure is translated into the SNR requirement. The factor grows as the data-alignment constraint, p_c, becomes more stringent, and vice versa.

Next, using the result in Proposition 2, the importance ARQ protocol is designed as follows. Since the effective receive SNR after combining is a monotone increasing function of the number of retransmission, the constraint in (19) can be translated into a threshold based retransmission policy. On the other hand, the SNR threshold in (20) can diverge for an extremely uncertain data sample. Hence, it is necessary to limit the threshold value to avoid resource-wasteful excessive retransmission. The resultant simple protocol is described as follows.

Protocol 1 (Importance ARQ for binary SVM classification)**.** Consider the acquisition of a data sample $\mathbf{x}$ from a scheduled edge device. The edge server repeatedly requests the device to retransmit $\mathbf{x}$ until the effective receive SNR satisfies

$$\mathsf{SNR}(T) > \min(\theta_0\, \mathcal{U}_{\mathsf{d}}\left(\hat{\mathbf{x}}(T)\right), \theta_{\mathsf{SNR}}), \tag{21}$$

where θ_{SNR} is a given maximum SNR.

Remark 2 (Importance-aware SNR control)**.** The importance ARQ protocol is a threshold based control policy with a SNR threshold adapted to data importance. From (21), the SNR threshold is proportional to the distance-based uncertainty of the data sample, $\mathcal{U}_{\mathsf{d}}(\mathbf{x})$. It is aligned with the intuition that a data sample of higher uncertainty should be more reliably received. To better understand this result, a graphical illustration is provided in Fig. 6. For a pre-specified p_c, a highly uncertain sample near the decision hyperplane requires a slim distribution with small variance (corresponding to a higher receive SNR and hence more retransmissions) to meet the requirement on the data-alignment probability (the area shaded in grey) to be larger than p_c [see Fig. 6(b)]. On the other hand, for a less uncertain data sample, the requirement of p_c can be easily satisfied with a relatively flat distribution with a large variance and low receive SNR [see Fig. 6(a)].

Last, the importance ARQ protocol is compared with the conventional channel-aware counterpart. For the latter, the retransmission policy is merely channel-aware, and a fixed SNR threshold is set for all data samples without differentiating their importance, as described below.

Protocol 2 (Channel-aware ARQ)**.** Consider the acquisition of a data sample $\mathbf{x}$ from a scheduled edge device. The edge server repeatedly requests the device to retransmit $\mathbf{x}$ until the required effective SNR, θ_{SNR}, is attained:

$$\mathsf{SNR}(T) > \theta_{\mathsf{SNR}}, \tag{22}$$

where $\mathsf{SNR}(T)$ is defined in (3).

Remark 3 (Uniform vs. heterogenous reliability)**.** As the SNR requirement in (22) is independent of data uncertainty, the channel-aware protocol achieves *uniform reliability* for data samples. If deployed in an edge learning system, it can lead to inefficient utilization of radio resource due to unnecessary retransmissions for unimportant data, resulting in sub-optimal learning performance. In contrast, the proposed importance ARQ protocol achieves *heterogeneous reliability* for data samples according to their importance levels. This allows more efficient resource utilization via improving the quality-vs-quantity tradeoff, thereby accelerating learning.

C. Implementation of Multi-Class Classification

In this subsection, the principle of importance ARQ developed in the preceding sub-section for binary classification is generalized to multi-class classification. Note that a C-class SVM classifier can be trained using the so-called *one-versus-one* implementation [31]. The implementation decomposes the classifier into $L = C(C-1)/2$ *binary component classifiers* each trained using the samples from the two concerned classes only. As a result, for each input data sample $\mathbf{x}$, a C-class SVM outputs a L-dimension vector, denoted as $\mathrm{s} = [s_1(\mathbf{x}), s_2(\mathbf{x}), \cdots, s_L(\mathbf{x})]$, which records the L output scores as defined in (6), from the component classifiers. To map the output s to one of the class indexes, a so-called *reference coding matrix* of size $C \times L$ is built and denoted by $\mathbf{M}$, where each row gives the "reference output pattern" corresponding to the associated class. An example of the reference coding matrix with C = 4 and hence 6 binary component classifiers is provided as follows:

$$\mathbf{M}= \begin{array}{c} \\ \text{class1} \\ \text{class2} \\ \text{class3} \\ \text{class4} \end{array} \begin{array}{c} \begin{array}{cccccc} \text{binary1} & \text{binary2} & \text{binary3} & \text{binary4} & \text{binary5} & \text{binary6} \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix} \end{array}.$$

Given $\mathbf{M}$, the prediction of the class index of s involves simply comparing the Hamming distances between s and different rows in $\mathbf{M}$, and choosing the row index with the smallest distance as the predicted class index. Particularly, the Hamming distance between s and the c-th row of $\mathbf{M}$ is defined by

$$d(\mathbf{s}, \mathbf{m}_c) = \sum_{\ell=1}^{L} |m_{c\ell}|[1 - \mathrm{sgn}(m_{c\ell} s_\ell(\mathbf{x}))]/2, \tag{23}$$

where $m_{c\ell}$ denotes the ℓ-th element in vector $\mathbf{m}_c$, and $\mathrm{sgn}(\mathrm{x})$ denotes the sign function taking a value from $\{1, 0, -1\}$ corresponding to the cases $x > 0$, $x = 0$ and $x < 0$, respectively. One can observe from the distance definition that not all the component classifiers' output scores have an effect on predicting a particular class. For example, the scores from binary classifiers $2, 3$ and 6 have no effect on determining class 2 as they are assigned a zero weight in computing the Hamming distance between s and $\mathbf{m}_2$. In other words, only binary classifiers $1, 4$ and 5 are active when class 2 is predicted.

Having obtained the predicted label $\hat{c} = \arg\min_c d(\mathbf{s}, \mathbf{m}_c)$, all the active component classifiers determining the current predicted label should satisfy the requirement of data alignment probability predefined in (19). Consequently, the *single-threshold policy* for importance ARQ defined in (21) can be then extended to a *multi-threshold policy* as defined below:

$$\mathsf{SNR}(T) > \frac{\theta_0}{|s_\ell(\hat{\mathbf{x}}(T))|^2}, \quad \forall \ell \in \{\ell \mid m_{\hat{c}\ell} \neq 0\}. \tag{24}$$

V. Extension to General Classifiers

In this section, we extend the proposed importance ARQ protocol designed in the preceding section for the SVM classifier model to a general model, and present a case study using the modern CNN model.

A. Importance ARQ for a Generic Model

The derivation of Protocol 1 targets for SVM and may not be directly extended to a generic classifier model (e.g., CNN), due to the lack of explicitly defined decision boundaries, and thus an explicit distance based uncertain measure. Nevertheless, the following insight derived for the SVM model is applicable to a generic model: *the receive-SNR threshold in wireless data acquisition with retransmission should be adapted to data uncertainty.* This motivates the generalization of the importance ARQ protocol by modifying Protocol 1 as follows.

Protocol 3 (Importance ARQ for generic classifier)**.** Consider the acquisition of a data sample $\mathbf{x}$ from a scheduled edge device. The edge server repeatedly requests the device to retransmit $\mathbf{x}$ until

$$\mathsf{SNR}(T) > \min\left(\theta_0\ \mathcal{L}(\mathcal{U}_{\mathsf{x}}\left(\hat{\mathbf{x}}(T)\right)), \theta_{\mathsf{SNR}}\right), \tag{25}$$

where $\mathcal{U}_{\mathsf{x}}$ is an uncertainty measure, θ_0 is a given conversion ratio between the uncertainty measure and the target SNR, and $\mathcal{L}(\cdot)$ is a monotonically increasing function.

The main difference of the generic protocol from Protocol 1 for SVM is that the distance-based uncertainty measure in the latter is replaced by a general *monotonically increasing* function of a general uncertainty measure. The function is called (uncertainty) *reshaping function*. The main motivation for introducing the function is to accommodate various forms of uncertainty measures. In particular, this function provides the flexibility to reshape a selected uncertainty measure to allow it to have certain desired properties as discussed in the sequel. Furthermore, the monotonicity of the function enforces the intuition that more uncertain data should be more reliably received.

To apply the general Protocol 3 to training a specific classifier model, the uncertainty measure, the reshaping function, and the conversion ratio should be carefully designed for efficient radio-resource utilization to achieve the desired learning performance. Several **design guidelines** are provided as follows.

- *Selection of Uncertainty Measure:* In general, the uncertainty measure should be selected for ease of computation according to the output of the learning model. For example, for SVM, the output score evaluated by *linear decision boundaries* allows easy evaluation of the distance-based uncertainty in (7). In contrast, for CNN, the *softmax* output, which gives the posterior probability for each predicted class, makes the *entropy* in (9) a more natural choice for measuring uncertainty.
- *Design of Reshaping Function and Conversion Ratio:* The reshaping function and the conversion ratio should be jointly designed to address the following two practical issues.
  - *Unregulated SNR for Data with Zero Uncertainty:* The minimum value of some uncertainty measures, e.g. entropy, can be zero. Its direct use in (25) without proper modification may lead to a corrupted training dataset. To be more specific, since the corresponding SNR thresholds have zero values, data samples with zero uncertainty fail to trigger retransmission and thus may be received with unacceptably low reliability in the case of strong noise. The use of such corrupted data in model training can cause model divergence. This issue can be addressed by a proper design of the reshaping function.
  - *Low Differentiability in SNR Threshold:* An issue can arise in practice due to a narrow dynamic range of a selected uncertainty measure. For example, if the uncertainty is measured by entropy, the corresponding dynamic range is given by $\mathcal{U}_{\mathsf{e}}(\mathbf{x}) \in [0, \log C]$, where C denotes the number of classes. For 10-class classification, we have $\mathcal{U}_{\mathsf{e}}(\mathbf{x}) \in [0, 2.3]$, which can be too narrow in retransmission implementation. In particular, without any reshaping function or a suitable conversion ratio, the SNR thresholds set as in (25) for the most and least important data would be about the same, making importance ARQ insensitive to uncertainty and barely "importance aware".

B. Implementing Importance ARQ for CNN

In this subsection, we use CNN as an example to illustrate how the generic importance ARQ in Protocol 3 can be particularized to a mode of choice based on the guidelines in the preceding subsection. To begin with, as discussed, *entropy* is chosen as a suitable measure of data uncertainty for CNN. Then, we design the reshaping function to have the following form: $\mathcal{L}(x) = 1 + \gamma x$, where γ is a scaling factor to be determined in the sequel.[1] Note that the bias term 1 in $\mathcal{L}(x)$

[1] An alternative such as the nonlinear increasing functions $\mathcal{L}(x) = (1 + x)^{\gamma}$ is also a suitable choice as verified by experiments.

is added to address the issue of zero SNR threshold. Particularly, we set the bias term to be 1 rather than other positive values as it allows the conversion ratio θ_0 to be also interpreted as the minimum quality requirement for the least uncertain data with the entropy being zero. This allows θ_0 to be set easily following the typical settings in a wireless communication system (e.g., θ_0 = 10 dB). Note from (25) that θ_{SNR} denotes the maximum quality requirement for the data with the largest uncertainty. Thus the scaling factor γ can be determined by solving the equality $\theta_0\left[1+\gamma \mathcal{U}_{\max}\right] = \theta_{\mathsf{SNR}}$ where the maximum entropy $\mathcal{U}_{\max} = \log C$. The above designs lead to the importance ARQ for the CNN classifier as shown below.

Protocol 4 (Importance ARQ for CNN)**.** Consider the acquisition of a data sample $\mathbf{x}$ from a scheduled edge device for training a CNN classifier model. The edge server repeatedly requests the device to retransmit $\mathbf{x}$ until

$$\mathsf{SNR}(T) > \min\left(\theta_0\left[1+\gamma\, \mathcal{U}_{\mathsf{e}}\left(\hat{\mathbf{x}}(T)\right)\right], \theta_{\mathsf{SNR}}\right), \tag{26}$$

where γ is a scaling factor given as $\gamma = \frac{1}{\mathcal{U}_{\max}}\left(\frac{\theta_{\mathsf{SNR}}}{\theta_0} - 1\right)$.

VI. Experimental Results

A. Experiment Setup

1) Channel Model: We assume the classic Rayleigh fading channel with channel coefficients following i.i.d. complex Gaussian distribution $\mathcal{CN}(0,1)$. The average transmit SNR defined as $\bar{\rho} = P/\sigma^2$ is by default set as 4 dB.

2) Learning Performance Metrics: The performance metrics are defined separately for the cases of *balanced* and *imbalanced* training datasets, depending on whether the dataset has more instances of certain classes than others. A balanced dataset is an ideal setting while the imbalanced setting is more likely to happen in real-world applications, e.g., fraud detection, medical diagnosis and network intrusion detection [32]. Given a balanced dataset, the learning performance is measured by the test accuracy. However, the overall accuracy is unable to reflect the performance using a highly skewed dataset. For example, a naive classifier that predicts all test samples as the majority class could achieve a high accuracy. However, it is unable to detect the minority but critical class. To tackle the issue, two performance metrics, i.e., G-mean and F-measure, widely used for imbalanced classification are adopted [32]. Both are based on the following confusion matrix defined for binary classification for imbalanced data, where positive and negative classes correspond to minority and majority classes, respectively:

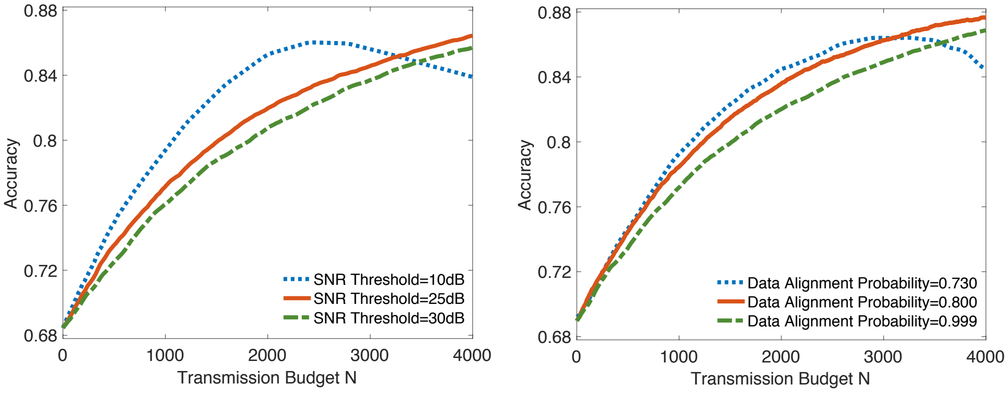
$$\textbf{Confusion Matrix}= \begin{array}{c} \\ \text{real positive} \\ \text{real negative} \end{array} \begin{array}{cc} \text{predicted positive} & \text{predicted negative} \\ \left(\begin{array}{c} \text{true positive (TP)} \\ \text{false positive (FP)} \end{array}\right. & \left.\begin{array}{c} \text{false negative (FN)} \\ \text{true negative (TN)} \end{array}\right) \end{array}.$$

Based on the confusion matrix, several useful metrics can be defined, followed by the definitions of G-mean and F-measure:

$$\text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \quad \text{specificity} = \frac{\text{TN}}{\text{TN}+\text{FP}}, \quad \text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}},$$
$$\text{G}-\text{mean} = \sqrt{\text{recall}\times\text{specificity}}, \quad \text{F}-\text{measure} = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}.$$

Recall and specificity measure the relevance between the predicted and ground-true results for the positive class and negative class, respectively. On the other hand, precision is the prediction accuracy for the positive class. As seen, G-mean is the geometric mean of recall and specificity, representing the average detection rate of positive and negative classes. However, one may be only interested in the highly effective detection for the rare case in some applications, e.g., cancer detection. In this case, F-measure is adopted which concerns only the positive class, integrating the detection and prediction rates as a single metric.

3) Experimental Dataset: We consider the learning task of training classifiers using the well-known MNIST dataset of handwritten digits as described in Section III-A. The training and test sets consist of $60,000$ and $10,000$ samples, respectively. Each sample is a grey-valued image of 28×28 pixels that gives the sample dimensions $p = 784$. For binary classification, we consider both balanced and imbalanced datasets. For a balanced dataset, we choose the relatively less differentiable class pair of “3” and “5” (according to t-SNE visualization). For an imbalanced data set, the relatively compact class “1” is chosen as the minority class and the majority class is made up of the remaining classes. The training set used in experiments is partitioned as follows. At the edge server, the priorly available collection of clean observations $\mathcal{L}_0$ are constructed by randomly sampling the global training dataset based on fixed ratios over classes: a) 2 samples for each class for the case of balanced data; b) 1 sample for minority and 8 samples for majority for the case of imbalanced data. The remaining training data are evenly and randomly distributed over edge devices. The maximum transmission budget N is set to be 4000 and $20,000$ (channel uses) for binary and multi-class classification, respectively. All results are averaged over 200 and 20 experiments for binary and multi-class cases, respectively.



(a) Channel-aware ARQ

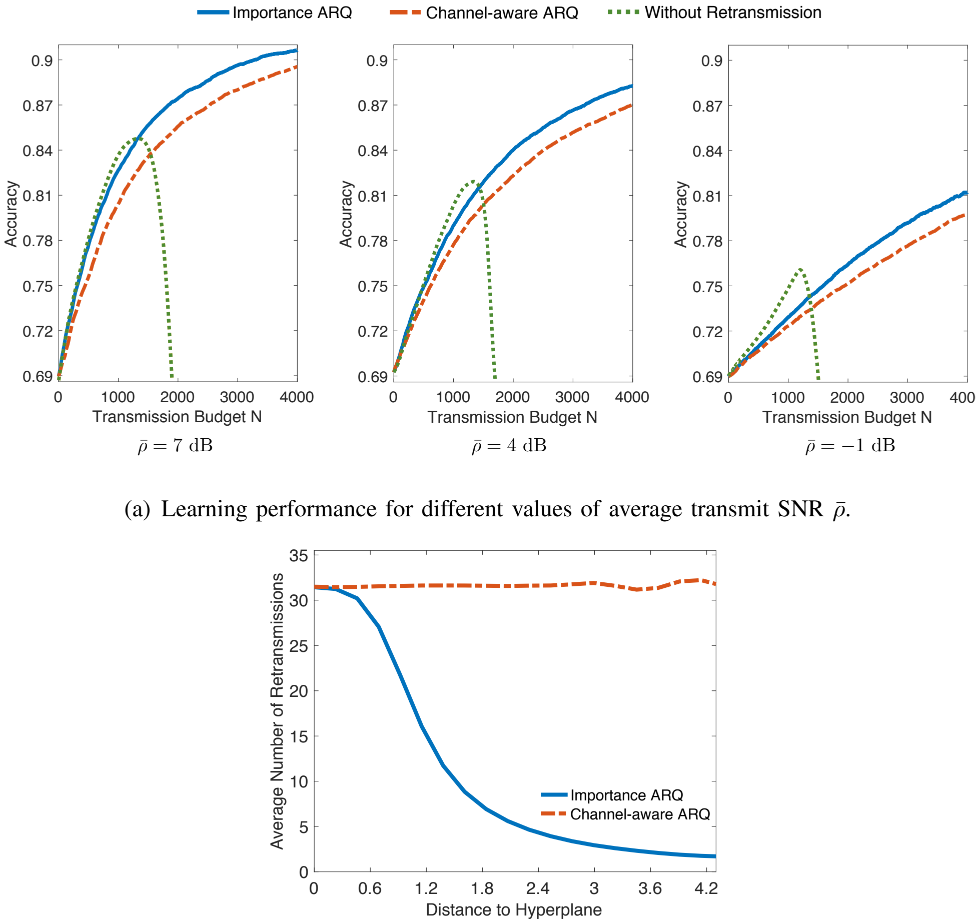
(b) Importance ARQ

Figure 7. Quality-vs-quantity tradeoff in wireless data acquisition.

4) Learning Model Implementation: The considered classifier models include the following: binary SVM, multi-class SVM, and CNN. For binary SVM, the soft-margin SVM is implemented with slack variable set as 1. *Iterative Single Data Algorithm* (ISDA) [33] is used for solving the SVM problem with maximum 10^6 iterations. The multi-class SVM is built on 45 binary SVMs as described in Section IV-C. For the implementation of CNN, we use a 6-layer CNN as illustrated in Fig. 3, including two 3×3 convolution layers with batch normalization before ReLu activation (the first with 16 channels, the second with 32), the first one followed with a 2×2 max pooling layer and the second one followed with a fully connected layer, a softmax layer, and a final classification layer. The model is trained using stochastic gradient descent with momentum [34]. The mini-batch size is 2048, and the number of epochs is 120. To accelerate training, the CNN is updated in a batch mode with the incremental sample size as 10.

B. Quality-vs-Quantity Tradeoff

To demonstrate the quality-vs-quantity tradeoff in wireless data acquisition, Fig. 7 displays the curves of learning accuracy versus transmission budget for both channel-aware ARQ and importance ARQ. In Fig. 7(a), we test the performance of channel-aware ARQ with three SNR thresholds, i.e., $\theta_{\mathsf{SNR}} = 10,\ 25$ and 30 dB, from low to high data-reliability requirements. Similar cases for importance ARQ are considered in Fig. 7(b) with the reliability requirements specified by the data-alignment probability: $p_c = 0.730,\ 0.800$ and 0.999. It is observed from both Fig. 7(a) and 7(b) that setting the thresholds too low (e.g., $\theta_{\mathsf{SNR}} = 10$ and $p_c = 0.730$) leads to a fast convergence rate but at a cost of performance degradation as the errors accumulate. In contrast, a too high threshold (e.g., $\theta_{\mathsf{SNR}} = 30$ and $p_c = 0.999$) also leads to poor learning performance



(a) Learning performance for different values of average transmit SNR $\bar{\rho}$.

(b) Retransmission Distribution

Figure 8. Learning performance for a binary SVM classifier trained using with wirelessly acquired data.

due to insufficient acquired samples. This suggests that the retransmission threshold should be carefully picked for optimizing the quality-vs-quantity tradeoff and thereby improving the learning performance. In the following experiments, we select thresholds based on observations in this sub-section to optimize performance.

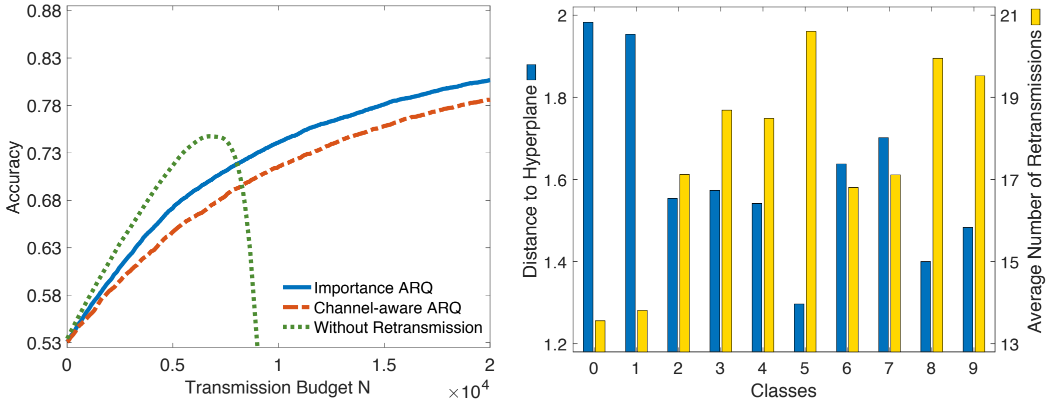
C. Learning Performance for Balanced Data

1) Binary SVM Classification: In Fig. 8, the learning performance of the proposed importance ARQ is compared with two baseline protocols, namely the channel-aware ARQ and the protocol without retransmission (maximum data quantity). It is observed that the performance of edge learning without retransmission dramatically degrades after acquiring a sufficiently large number of noisy samples. This is aligned with our previous observations from Fig. 5(a) and justifies the need for retransmission. Next, one can observe that importance ARQ outperforms the conventional channel-aware ARQ throughout the entire training duration. This confirms the performance gain from the intelligent resource utilization in data acquisition. Furthermore, the performance gain of importance ARQ is almost the same in varying SNR scenarios. This demonstrates the robustness of the proposed protocol against the hostile channel condition.

In Fig. 8(b), we further investigate the underlying reason for the performance improvement of importance ARQ by plotting the distribution of average numbers of retransmissions over a range of sample uncertainty (inversely proportional to sample distance to the decision hyperplane). One can observe close-to-uniform distribution for conventional channel-aware ARQ corresponding to uncertainty independence. In contrast, for importance ARQ, retransmission is concentrated in the high uncertainty region. This is aligned with the design principle and shows its effectiveness in adapting retransmission to data importance.

2) Multi-class SVM Classification: In Fig. 9(a), the learning performance of the proposed importance ARQ is compared with two baseline protocols in the scenario of multi-class classification. Similar trends as in the binary-class scenario are observed, and the importance ARQ is found to consistently outperform the benchmarking protocols in this more challenging scenario. The relation between importance ARQ and the multi-cluster structure of the training dataset is illustrated in Fig. 9(b). The blue bar indicates that samples in different classes in general have distinct average distances to their corresponding decision hyperplanes and thus with different uncertainty (inverse of distance). From the yellow bar, one can observe that importance ARQ can effectively adapt the average retransmission budget for different classes to their uncertainty levels. For example, class 5 has the shortest average distance to the hyperplane thus is allocated the largest transmission budget to protect its receive quality. In contrast, class 0 has the longest distance, and thus consumes less budget as desired.

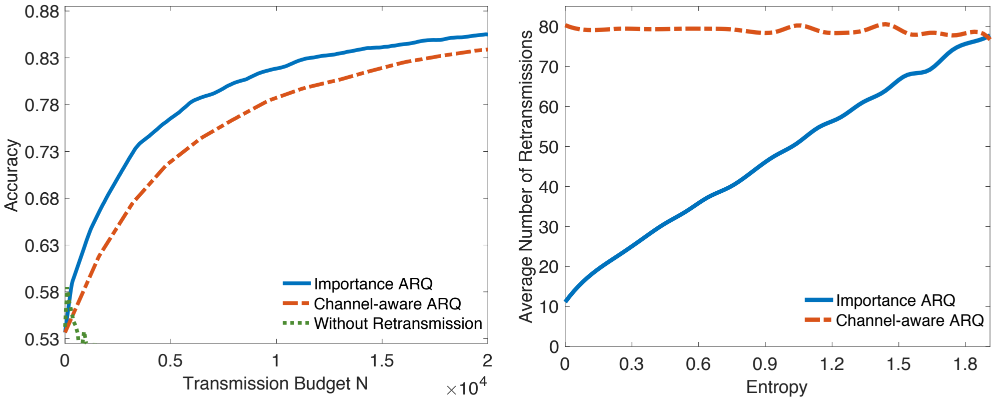
3) Multi-class Classification of CNN: Our heuristic design for CNN is tested in the scenario of multi-class classification and the related results are provided in Fig. 10. Fig. 10(a) displays the learning performance adopting entropy based uncertainty, which consistently outperforms two baseline protocols. One can notice that without retransmission the performance of CNN quickly degrades especially compared with the previous result for SVM, which implicates that CNN is more sensitive to noisy environment. This is due to the fact that, in CNN classifier, all samples contribute to define the decision hyperplane. As a result, the noisy effect accumulates faster in CNN than SVM where only a few support vectors determine the decision hyperplane. Therefore, CNN in general requires more retransmission to attain a higher receive SNR to guarantee the learning performance as shown in Fig. 10(b). Besides, Fig. 10(b) shows the linear relationship



(a) Performance evaluation.

(b) Uncertainty v.s. Retransmission.

Figure 9. Learning performance evaluation for multi-class SVM classification.



(a) Performance evaluation.

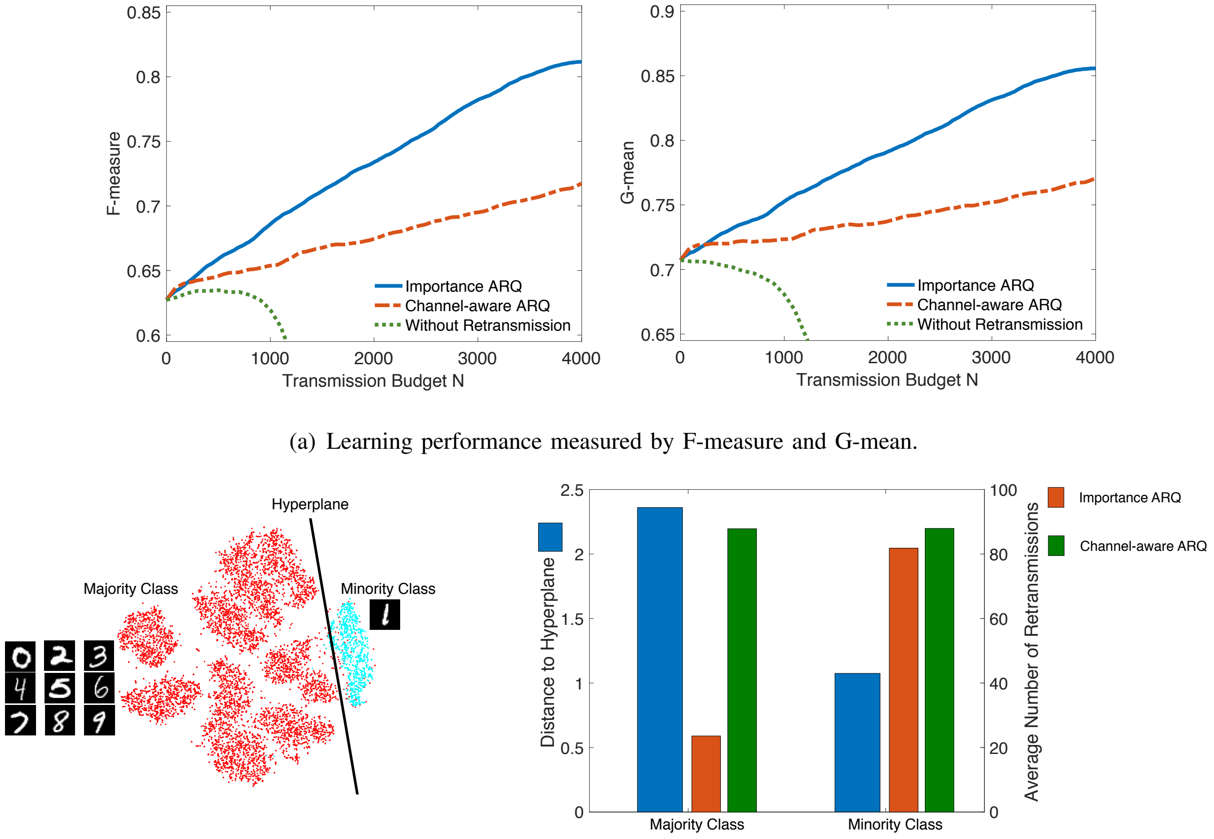
(b) Retransmission Distribution.

Figure 10. Learning performance evaluation for multi-class CNN classification.

between entropy and the number of retransmissions, which is consistent with our design.

D. Learning Performance for Imbalanced Data

1) Imbalanced Classification of SVM: In Fig. 11(a), both F-measure and G-mean of the proposed importance ARQ are compared with two baseline protocols in the scenario of imbalanced classification by using SVM. Compared with balanced classification (Fig. 8(a)), the performance curves in the imbalanced setting degrade faster if no retransmission is made, which implicates that imbalanced classification is more vulnerable to the hostile channel environment. This fact calls for an intelligent retransmission protocol to regulate the quality of each training sample. One can notice that importance ARQ could achieve a larger gain in imbalanced classification (nearly 10% performance improvement is observed compared with the conventional channel-aware ARQ). To further investigate the underpinning reason, we visualize the imbalanced dataset by using t-SNE



(a) Learning performance measured by F-measure and G-mean.

(b) Uncertainty v.s. Retransmission.

Figure 11. Learning performance for SVM classifier training using imbalanced data.

and plot the relationship between retransmission and uncertainty. The left subfigure in Fig. 11(b) shows that, the minority class has a higher uncertainty value since the average distance to the hyperplane is shorter than the majority one. This is aligned with the blue bar in the right subfigure. It is also observed that a highly uncertain minority class consumes more retransmission budget in importance ARQ, as shown by the red bar. However, the green bar shows that channel-aware ARQ allocates equal transmission budgets to both majority and minority classes. The superiority of importance ARQ in the balanced setting further substantiates the theoretical gain brought by the intelligent adaptation of the radio resource allocation according to the data importance.

2) Imbalanced Classification of CNN: In Fig. 12, F-measure and G-mean are examined in the imbalanced classification by deploying CNN. Similar trends as the SVM classifier are observed, and the importance ARQ is found to consistently outperform the benchmarking protocols, which confirm the effectiveness of our extension in Section V.

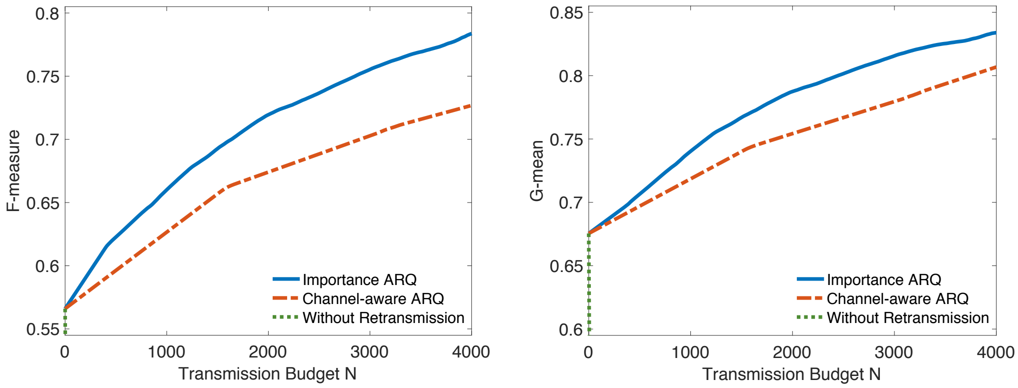


Figure 12. Learning performance for CNN classifier training using imbalanced data.

VII. Concluding Remarks

In this paper, we have proposed a novel retransmission protocol, namely importance ARQ, for wireless data acquisition in edge learning systems. It intelligently adapts retransmission to data-sample importance so as to enhance the learning performance given a transmission latency constraint. Comprehensive experiments using real datasets substantiate the performance gain of the proposed design.

At a higher level, the work contributes the new principle of exploiting the non-uniform distribution of data importance to improve the efficiency of wireless data acquisition for edge learning. Importance aware retransmission is just one way for materializing this principle. It can be applied to many other aspects of wireless data acquisition such as scheduling, power control, spectrum allocation, and energy efficient transmission. Thereby many promising research opportunities are presented.

Appendix A
Proof of Lemma 1

The received sample $\hat{\mathbf{x}}(T)$ in (2) can be rewritten as

$$\hat{\mathbf{x}}(T) = \mathbf{x} + \Re\left(\widetilde{\mathbf{z}}(T)\right),$$

where $\widetilde{\mathbf{z}}(T) = \frac{1}{\sqrt{P}}\left(\frac{\sum_{i=n+1}^{n+T}\left(h^{(i)}\right)^{*}\mathbf{z}^{(i)}}{\sum_{m=n+1}^{n+T}|h^{(m)}|^2}\right)$. Consequently, the transmitted sample is

$$\mathbf{x} = \hat{\mathbf{x}}(T) - \Re\left(\widetilde{\mathbf{z}}(T)\right), \tag{27}$$

where $\widetilde{\mathbf{z}}(T) = [\widetilde{z}_1(T), \cdots, \widetilde{z}_p(T)]^{\mathsf{T}}$ is the equivalent noise after combining with the entries being

$$\widetilde{z}_j(T) = \frac{1}{\sqrt{P}} \times \frac{\sum_{i=n+1}^{n+T}\left(h^{(i)}\right)^{*} z_j^{(i)}}{\sum_{m=n+1}^{n+T}|h^{(m)}|^2}, \;\; j = 1, 2, \cdots, p.$$

Since $z_j^{(i)}$ follows i.i.d $\mathcal{CN}\left(0, \sigma^2\right)$, each entries in $\widetilde{\mathbf{z}}(T)$ are i.i.d and the distributions are:

$$\widetilde{z}_j(T) \sim \mathcal{CN}\left(0, \frac{\sigma^2}{\sum_{i=n+1}^{n+T} |h^{(i)}|^2 P}\right), \; j = 1, 2, \cdots, p.$$

With effective SNR defined in (3), taking the real part of $\widetilde{\mathbf{z}}$ yields to the following distribution:

$$\Re\left(\widetilde{\mathbf{z}}(T)\right) \sim \mathcal{N}\left(0, \frac{\mathbf{I}}{\mathsf{SNR}(T)}\right),$$

which leads to the desired result in (12).

References

[1] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Towards an intelligent edge: Wireless communication meets machine learning," *[Online]. Available: https://arxiv.org/pdf/1809.00343.pdf*, 2018.

[2] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. of IEEE Int. Conf. Comput. Commun. (INFOCOM)*, (Honolulu, USA), April 2018.

[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys and Tutorials*, vol. 19, pp. 2322–2358, Aug. 2017.

[4] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *[Online]. Available: https://arxiv.org/pdf/1812.02858.pdf*, 2018.

[5] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan, *et al.*, "Towards federated learning at scale: System design," *[Online]. Available: https://arxiv.org/pdf/1902.01046.pdf*, 2019.

[6] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

[7] M. Chen, Y. Hao, K. Lin, Z. Yuan, and L. Hu, "Label-less learning for traffic control in an edge network," *IEEE Network*, vol. 32, pp. 8–14, Nov. 2018.

[8] C. She, C. Yang, and T. Q. Quek, "Radio resource management for ultra-reliable and low-latency communications," *IEEE Commun. Magazine*, vol. 55, pp. 72–78, June 2017.

[9] E. N. Onggosanusi, A. G. Dabak, Y. Hui, and G. Jeong, "Hybrid ARQ transmission and combining for MIMO systems," in *Proc. of Intl. Conf. on Commun. (ICC)*, (Anchorage, USA), May 2003.

[10] Q. Zhang and S. A. Kassam, "Hybrid ARQ with selective combining for fading channels," *IEEE J. Sel. Areas Commun.*, vol. 17, pp. 867–880, May 1999.

[11] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *[Online]. Available: https://arxiv.org/pdf/1610.05492.pdf*, 2016.

[12] G. Zhu, Y. Wang, and K. Huang, "Low-latency broadband analog aggregation for federated edge learning," *[Online]. Available: https://arxiv.org/pdf/1812.11494.pdf*, 2018.

[13] M. M. Amiri and D. Gunduz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *[Online]. Available: https://arxiv.org/pdf/1901.00844.pdf*, 2019.

[14] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *[Online]. Available: https://arxiv.org/pdf/1812.11750.pdf*, 2018.

[15] Y. Du and K. Huang, "Fast analog transmission for high-mobility wireless data acquisition in edge learning," *to appear in IEEE Wireless Commun. Lett.*, March 2018.

[16] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient sgd via gradient quantization and encoding," in *Proc. of Conf. on Neural Info. Process. Sys. (NIPS)*, (Long Beach, USA), Dec. 2017.

[17] K. Li, W. Ni, L. Duan, M. Abolhasan, and J. Niu, "Wireless power transfer and data collection in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 67, pp. 2686–2697, March 2018.

[18] J. Guo, S. Durrani, X. Zhou, and H. Yanikomeroglu, "Massive machine type communication with data aggregation and resource scheduling," *IEEE Trans. Commun.*, vol. 65, pp. 4012–4026, Sept. 2017.

[19] D. Malak, H. S. Dhillon, and J. G. Andrews, "Optimizing data aggregation for uplink machine-to-machine communication networks," *IEEE Trans. Commun.*, vol. 64, pp. 1274–1290, March 2016.

[20] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in uav enabled wireless sensor network," *IEEE Wireless Commun. Lett.*, vol. 7, pp. 328–331, June 2018.

[21] W. Nie, V. C. Lee, D. Niyato, Y. Duan, K. Liu, and S. Nutanong, "A quality-oriented data collection scheme in vehicular sensor networks," *IEEE Trans. Veh. Technol.*, vol. 67, pp. 5570–5584, July 2018.

[22] A. Holub, P. Perona, and M. C. Burl, "Entropy-based active learning for object recognition," in *Proc. of Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, (Anchorage, USA), June 2008.

[23] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Proc. of Conf. on Neural Info. Process. Sys. (NIPS)*, (Vancouver and Whistler, Canada), Dec. 2008.

[24] N. Roy and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction," in *Proc. of Intl. Conf. on Machine Learning (ICML)*, (Williamstown, USA), June 2001.

[25] T. L. Marzetta and B. M. Hochwald, "Fast transfer of channel state information in wireless systems," *IEEE Trans. Signal Process.*, vol. 54, pp. 1268–1278, April 2006.

[26] S. Cui, J.-J. Xiao, A. J. Goldsmith, Z.-Q. Luo, and H. V. Poor, "Energy-efficient joint estimation in sensor networks: Analog vs. digital," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, (Philadelphia, USA), March 2005.

[27] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. Springer, 2001.

[28] S. Haykin, *Neural Networks: A Comprehensive Foundation*, vol. 2. New York: Prentice hall, 1994.

[29] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *J. of Machine Learning Research*, vol. 2, pp. 45–66, Nov. 2001.

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, pp. 2278–2324, Nov. 1998.

[31] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin DAGs for multiclass classification," in *Proc. of Conf. on Neural Info. Process. Sys. (NIPS)*, (Denver, USA), Dec. 2000.

[32] Y. Sun, A. K. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *Intl. J. of Pattern Recognition and Artificial Intelligence*, vol. 23, pp. 687–719, June 2009.

[33] V. Kecman, T.-M. Huang, and M. Vogt, "Iterative single data algorithm for training kernel machines from huge data sets: Theory and performance," in *Support Vector Machines: Theory and App.*, pp. 255–274, Springer, 2005.

[34] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. of Intl. Conf. on Machine Learning (ICML)*, (Atlanta, USA), June 2013.