AGE OF INFORMATION AND UNBIASED FEDERATED LEARNING IN ENERGY HARVESTING ERROR-PRONE CHANNELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ZEYNEP ÇAKIR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

AUGUST 2022

Approval of the thesis:

AGE OF INFORMATION AND UNBIASED FEDERATED LEARNING IN ENERGY HARVESTING ERROR-PRONE CHANNELS

submitted by **ZEYNEP ÇAKIR** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil KALIPÇILAR
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İlkay ULUSOY
Head of Department, **Electrical and Electronics Engineering**

Assist. Prof. Dr. Elif Tuğçe CERAN ARSLAN
Supervisor, **Electrical and Electronics Engineering, METU**

Examining Committee Members:

Prof. Dr. Elif UYSAL
Electrical and Electronics Engineering, METU

Assist. Prof. Dr. Elif Tuğçe CERAN ARSLAN
Electrical and Electronics Engineering, METU

Assoc. Prof. Dr. Hüseyin Uğur YILDIZ
Electrical and Electronics Engineering, TED University

Assoc. Prof. Dr. Ayşe Melda YÜKSEL TURGUT
Electrical and Electronics Engineering, METU

Assist. Prof. Dr. Serkan SARITAŞ
Electrical and Electronics Engineering, METU

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Zeynep Çakır

Signature :

ABSTRACT

AGE OF INFORMATION AND UNBIASED FEDERATED LEARNING IN ENERGY HARVESTING ERROR-PRONE CHANNELS

Çakır, Zeynep
M.S., Department of Electrical and Electronics Engineering
Supervisor: Assist. Prof. Dr. Elif Tuğçe CERAN ARSLAN

August 2022, 82 pages

Federated learning is a communication-efficient and privacy-preserving learning technique for collaborative training of machine learning models on vast amounts of data produced and stored locally on the distributed users. In this thesis, unbiased federated learning methods that achieve a similar convergence as state-of-the-art federated learning methods in scenarios with various constraints like error-prone channel or intermittent energy availability are investigated. In addition, a prevalent metric called the age of information (AoI), which quantifies the staleness of the information at the destination, is studied under energy constraints and exploited to increase the performance of federated learning algorithms.

Firstly, a constrained Markov decision problem that aims to minimize the average age of information over an imperfect channel and under energy constraints is investigated. An optimal threshold-based scheduling policy is obtained and the optimal time average AoI and age violation probabilities are derived. Secondly, a federated learning algorithm that jointly designs the unbiased user scheduling and gradient weighting according to the energy and channel profile of each user is presented. It is shown that the proposed algorithm provides a high test accuracy and a convergence guarantees, which is close to the algorithms that have no energy or channel constraints. Lastly, the effect of AoI on federated learning with heterogeneous users and different datasets is studied, and the performance is demonstrated by experiments.

Keywords: federated learning, energy harvesting, age of information, momentum, wireless communications

ÖZ

HATAYA AÇIK KANALLAR ÜZERİNDE ENERJİ HASADI İLE TARAFSIZ FEDERE ÖĞRENME VE BİLGİ YAŞI

Çakır, Zeynep
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Elif Tuğçe CERAN ARSLAN

Ağustos 2022 , 82 sayfa

Federe öğrenme, dağıtılmış kullanıcılar üzerinde yerel olarak üretilen ve depolanan büyük miktarda veri üzerinde makine öğrenimi modellerinin işbirlikçi eğitimi için iletişim açısından verimli ve gizliliği koruyan bir öğrenme tekniğidir. Bu tezde, hataya açık kanal veya kesintili enerji varışı gibi çeşitli kısıtlamalara sahip senaryolarda modern federe öğrenme yöntemlerine benzer bir yakınsamayı sağlayan tarafsız federe öğrenme yöntemleri araştırılmaktadır. Ek olarak, varış noktasındaki bilginin eskiliğini ölçen bilgi yaşı (AoI) adı verilen yaygın metrik, enerji kısıtlamaları altında incelenir ve federe öğrenme algoritmalarının performansını artırmak için kullanılır.

İlk olarak, kusurlu bir kanal üzerinde ve enerji kısıtlamaları altında ortalama bilgi yaşını en aza indirmeyi amaçlayan kısıtlı bir Markov karar problemi incelenmiştir. Optimal eşik tabanlı bir zamanlama politikası önerilmiş ve zamana göre ortalama AoI ve yaş ihlali olasılıkları elde edilmiştir. İkinci olarak, her kullanıcının enerji ve kanal profillerine göre kullanıcı çizelgelemesini ve gradyan ağırlıklandırmasını ortaklaşa tasarlayan, tarafsız bir federe öğrenme algoritması sunulmuştur. Önerilen algoritmanın, enerji veya kanal kısıtlaması olmayan algoritmalara yakın, yüksek bir test doğruluğu ve yakınsama garantisi sağladığı gösterilmiştir. Son olarak, bilgi yaşının heterojen kullanıcılar ve farklı veri kümeleri ile federe öğrenme üzerindeki etkisi incelenmiş ve performansı deneylerle gösterilmiştir.

Anahtar Kelimeler: federe öğrenme, enerji hasadı, bilgi yaşı, momentum, kablosuz haberleşme

to my beloved parents

ACKNOWLEDGMENTS

I am thankful to my advisor, Assist. Prof. Dr. Elif Tuğçe CERAN ARSLAN, for sharing her amazing knowledge with me and being very understanding about working and this thesis going in parallel. It was wonderful to take this journey with her, and I hope I made her proud, as me being one of her first graduate students. Also, I am thankful to Prof. Dr. Elif UYSAL for giving me a chance to be a part of her research group at the very beginning, sharing her amazing knowledge with me and guiding me in this journey.

I am thankful to my advisor in my undergraduate years and one of the distinguished members of the examining committee, Assoc. Prof. Dr. Hüseyin Uğur YILDIZ, for always being there for me whenever I need his guidance and support. I will be forever grateful to him for his golden touches in my life, and I hope I made him proud, as me being one of his first students.

I am thankful to all of the distinguished professors in the examining committee, for their valuable feedbacks and their contributions to this thesis.

I am thankful to my manager Gökhan BİRCAN and my employer Roketsan Inc. for making this master's degree possible.

I am thankful to Ümit ERONAT for his endless love, support and understanding. He is one of the biggest reasons I was able to write this thesis, and I will be forever grateful to him.

Most of all, I am thankful to my parents, Türkan and Alaaddin ÇAKIR, for all of their sacrifices and endeavours for me to live a successful, peaceful and happy life, and guiding me with their brightest light whenever I need them. They are the biggest treasures of my life and I hope I made them proud, and continue to do so.

And life, I am thankful to you.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AoI	Age of Information
IoT	Internet of Things
WET	Wireless Energy Transfer
FL	Federated Learning
FEEL	Federated Edge Learning
SGD	Stochastic Gradient Descent
CNN	Convolutional Neural Network
IID	Independent and Identically Distributed
FedAvg	*FederatedAveraging*

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

In today's world, a vast amount of data is produced by various types of devices. The need to store, process, and use this big data is one of the main focuses of up-to-date machine learning applications. Since an orchestral server conducts the model training, collecting, storing, and processing the data produced by the devices was getting harder, and it was a burden for the server to work with that substantial amount of data. In addition, the data produced by a device can be sensitive and private, and privacy violations may occur because of the need to upload data. Motivated by providing a solution to these problems, Google researchers introduced a concept named "federated learning" [1], and it became a commonly-used method for private and efficient machine learning/deep learning.

Federated learning is a communication-efficient and privacy-preserving learning technique for training machine learning models on vast amounts of data produced and stored locally on the participant users. It allows users to be part of a global machine learning model training without sharing their local data. Training is performed using distributed stochastic gradient descent (SGD) coordinated by a central server responsible for the global model. To train the global model, each user uses their local dataset, and the goal is to train a machine learning model on the combined dataset. While designing a method for a federated learning setup with several constraints, ensuring that there is no bias between users is an important guarantee. Federated learning found a place in many areas, such as the defense industry, the Internet of Things (IoT), medical applications, and many more. Recently, there has been substantial research on federated learning and its applications. Energy harvesting, which comprises energy gathering by electric, magnetic, or electromagnetic fields, has played an essential role in extending the current federated learning algorithms. Federated edge learning is a version of federated learning performed by wireless devices, with constrained energy and bandwidth, on their local datasets, supported by a remote parameter server. A concept of weighting model differences by a "cooldown multiplier," based on the time elapsed between two most recent energy arrivals, is introduced. Additionally, the method "momentum," an extension of the stochastic gradient descent method, is essential in accelerating the convergence or increasing the accuracy for non-homogeneous data distribution on participant users. Supported by numerous convergence and performance analyses, federated learning is a trustworthy method for conducting machine learning applications effectively and privately.

A prevalent metric called the Age of Information (AoI) quantifies the staleness of the information at the destination. It is defined as the time elapsed since the generation time of the most recent status update packet successfully received at the destination. Especially for status update applications, AoI is a critical performance indicator. Its implementation areas include machine-type communications, industrial applications, the Internet of Things, social networks, etc. In the federated learning area, AoI is commonly stated as the time elapsed between receiving the local updates from a participant user. It is an essential and unique metric for increasing the performance of federated learning algorithms and provides a new perspective to existing methods and applications.

This thesis explores federated learning strategies that achieve a similar convergence as state-of-the-art federated learning methods in contexts with diverse restrictions such as error-prone channels or intermittent energy availability.

1.1 Contributions and Novelties

To the best of our knowledge, this thesis is the first work on bringing together the two main concepts of the age of information and unbiased federated learning with energy harvesting with the error-prone channels. The contributions of this thesis are listed as follows:

- A constrained Markov decision problem, aiming to minimize the average age of information under energy constraints, is studied, and as a solution, an optimal threshold-based decision policy is proposed. Part of this work has been presented as a poster presentation at London Symposium on Information Theory (LSIT) in 2019, and an extension of this work is studied in [2] and presented at the 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt) in 2021.

- A channel and energy-aware federated learning method is proposed for an unbiased and heterogeneous federated learning network that is prone to intermittent energy arrivals and channel errors. It is shown that the proposed method achieves the same convergence guarantees as the federated learning algorithms with no energy or channel constraints. Part of this work has been presented at the 30th Signal Processing and Communications Applications Conference (SIU) in 2022.

- The effect of AoI on federated learning with channel and energy-aware scheduling is studied, combined with the dynamic weighting of the updates and the acceleration of AoI-aware momentum for independent and identically distributed (IID) and not independent and identically distributed (non-IID) datasets, is studied.

- Performances of proposed methods are verified by several experiments, and numerical results are provided.

1.2 The Outline of the Thesis

This thesis includes six chapters. In Chapter 2, background information and literature review on federated learning and the age of information are provided. In Chapter 3, achieving optimal age of information with wireless energy transfer with system model, proposed methods, and experimental results are provided. In Chapter 4, federated learning with channel and energy-aware scheduling with system model, proposed methods, and experimental results are provided. In Chapter 5, the effect of age on federated learning with channel and energy-aware scheduling combined with dynamic weighting and AoI-aware momentum with system model, proposed methods, and experimental results are provided. In Chapter 6, a summary of the work in this thesis with some important results and future research/work areas are provided.

CHAPTER 2

BACKGROUND INFORMATION

2.1 Federated Learning

Models trained using the data produced, processed, and used in mobile devices have the potential to pave the way for promising technologies in the future. In the case of the model training process being dependent on a single center, data privacy violation may occur on the part of the participating users since all the data for which the model will be trained must be on the server. Additionally, it requires a lot of time and energy due to the high processing load. Since first introduced by Google researchers in 2016 [1] as a solution to the need to carry out a training process without needing to store the local data set of each user on a central server, federated learning and its applications have become a popular approach for such concerns.

To train a model in federated learning, a training scenario where K users work together is considered. Each user has their local dataset, and the goal is to train a machine learning model on the combined dataset. Dataset can be either IID (ex. shuffling the dataset and splitting it between the users) or non-IID (ex. sorting and dividing the dataset and assigning each part to a client). Training is performed using distributed stochastic gradient descent (SGD) coordinated by a central server responsible for the global model. Illustration of SGD is provided in Figure 2.1. Note that the learning rate determines how quickly the model adapts to the situation, which is why it is one of the most crucial hyperparameters [3]. A lower learning rate might cause the training process to be slower, whereas a greater learning rate might lead the model to converge too soon to an unreliable result.

The server sends users the current estimate of the model parameters after each training

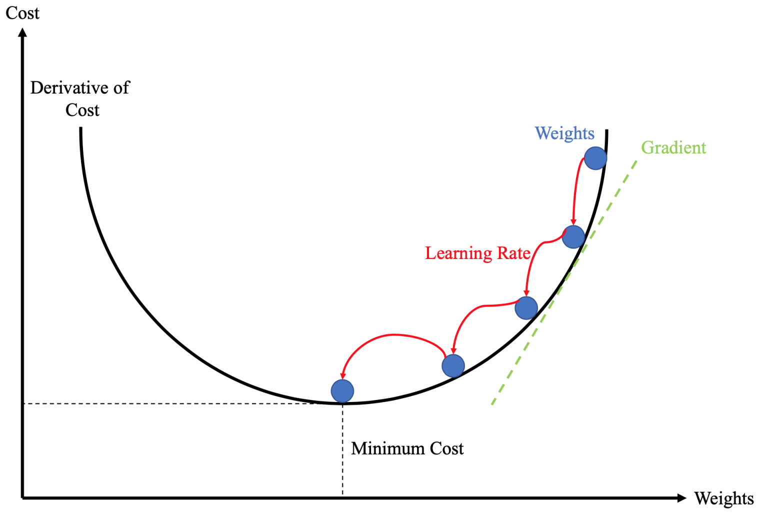

Figure 2.1: An illustration about the operation of SGD.

round. Users then update the global model by calculating a local gradient on local datasets and sending the results to the server. The server then collects users' local updates, updates the global model, and returns the updated model to the users. This method aims to ensure privacy and security since the data is not shared with the server, and the training process takes place on a user basis. An illustration about federated learning is provided in Figure 2.2 *(Retrieved from: https://ai.googleblog.com/ 2021/10/fedjax-federated-learning-simulation.html)*.

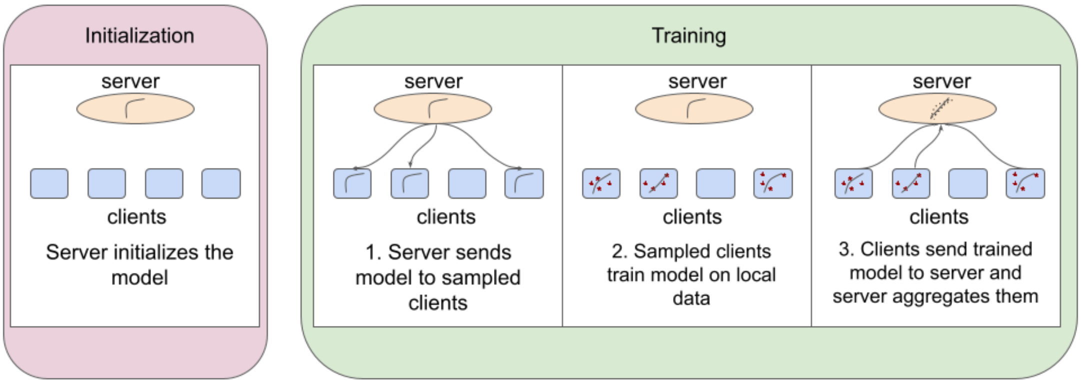
Initialization
server
clients
Server initializes the model
Training
server
clients
1. Server sends model to sampled clients
server
clients
2. Sampled clients train model on local data
server
clients
3. Clients send trained model to server and server aggregates them

Figure 2.2: An illustration about federated learning.

The term "heterogeneous users" for a federated learning setup is used for users with different network characteristics, such as energy arrival or channel availability. In such terms, the parameter server may give particular importance to users who can participate in the process more frequently and produce better results than the other users. This situation leads to a bias in federated learning. It is a situation not desired because the parameter server would prefer the users that are more advantageous than the other users in terms of participation, resulting in a performance loss. In sum, while designing a method for a federated learning setup with several constraints, ensuring that there is no bias between users is an important guarantee.

Along with the first introduction of federated learning by Konecny et al. [4], [1], another reference guide is presented by McMahan et al. [5] in 2017. In this study, the concept of federated learning is explained, and the *FederatedAveraging* algorithm, which forms the basis of many following studies, is introduced. In this algorithm, each participant user performs local training on the current global model using its local dataset, and the parameter server takes a weighted average of the locally trained model parameters. This method provides the advantage of multiple computations on each user. This algorithm is provided in Algorithm 1. The convergence analysis of this algorithm was carried out by Li et al. in [6], and it was carried out separately for the datasets that are equally distributed and not equally distributed to the users. In this context, it has shed light on many convergence analysis studies. While there is an assumption in traditional federated learning algorithms that users participate in the training process as soon as they are scheduled, Güler et al. [7] studied adding energy harvesting to the federated learning concept. In this study, the criterion of users' participation according to their energy level was added, and convergence analysis and experiments were carried out for the functionality of this criterion. This study motivated adding new measures to the federated learning concept. To introduce, energy harvesting, which comprises the gathering of electrical energy without wires using time-dependent electric, magnetic, or electromagnetic fields, has been indicated as a feasible preference for numerous communication systems [8, 9, 10]. Similarly, Güler et al. [11] examined the energy harvest criterion and the federated learning concept concerned with the sustainability of future smart ecosystems. Gündüz et al. [12] studied communicative constraints, inspected the divergence of existing coding and communication schemes and learning algorithms, and suggested new approaches to combine these concepts. Özfatura et al. [13] studied the demon-

Algorithm 1 FederatedAveraging

```
Require: Total number of global rounds $T$, total number of local iterations $L$, number of users $K$, the fraction of users $C$, local minibatch size $B$, set of indexes of data points $P_k$, initialized model parameters $w^{(0)}$
Ensure: Trained model parameters $w^{(T)}$
  Initialize $w^{(0)}$
  for Global round $t = 1, 2, ..., T$ do
    Determine $m = \max(C * K, 1)$
    Determine $S_t$ = Random set of m clients
    for User $i$ in $S_t$ do
      Split $P_k$ into batches of size $B$
      for Local round $l$ in $L$ do
        Local training with $w^{(t)}$ according to the corresponding minibatch
      end for
      Return $w_i^{(t+1)}$ to the server
    end for
    Update the global model as $w^{(t+1)} = \sum_{k=1}^{K} \frac{n_k}{n} w_i^{(t+1)}$, where $n_k$ is the local dataset size and $n$ is the total dataset size
    Send model parameters $w^{(t+1)}$ to the users
  end for
```

stration of how taking wireless channel characteristics, such as resource allocation, scheduling, and so on, into consideration may considerably enhance the speed and overall performance of distributed learning approaches. Generic communication reduction approaches, including sparsification, quantization, and local iterations, are inspected, and an overview of device scheduling and resource allocation methodologies for wireless distributed learning is provided.

In addition to the valuable works above, many works focus on industrial applications. Since the main idea of federated learning is focused on the privacy and security of local data of the participating users, it can easily find a place in military applications. One of its popular areas of interest in applications is unmanned aerial vehicle (UAV) networks. Zhang et al. investigated the image classification problems in the context of UAV-assisted exploration, where the coordination of UAVs is handled by a center located in a strategic but inaccessible area, where available energy is finite [14]. In this work, federated learning is used to reduce the communication cost between the UAVs and the center, as well as the computational complexity [14]. Brik et al. reviewed both the advantages and the main objections, and possible research areas of UAV-based networks with federated deep learning [15]. Pham et al. studied improving the UAV transmit power efficiency by optimizing transmission time, bandwidth allocation, power management, and UAV positioning [16]. Lim et al. studied the trade-off between age and service latency in the federated learning for contract-theoretic incentive system to fairly rewarding users based on the expense of updating data [17].

Additionally, there are recent studies about analyzing and improving the performance of federated edge learning (FEEL). Supported by a remote parameter server, federated edge learning is performed by wireless devices, with constrained energy and bandwidth, on their local datasets. Aygün et al. studied a FEEL scenario among users that harvest energy with over-the-air (OTA) aggregation [18]. Users are assumed to participate in the process only when their energy level is enough, and they send local updates concurrently over the same channel bandwidth. Applicable for different energy arrival processes, it is proposed that the model differences can be weighted by a "cooldown multiplier", based on the time elapsed between the two most recent energy arrivals. Supported by the convergence analysis and experiments, it is shown that the proposed algorithm does not violate the convergence guarantees. Again, Aygün et al. studied a federated learning setup where mediating servers create clusters near users to deal with the challenge of being away from the parameter server and related channel effects [19]. Supported by the numerical analysis and experiments, proposed system design performs better than traditional federated learning methods in the manner of convergence rate and accuracy. Amiri et al. investigated the effect of the bandwidth-shared wireless communication on the performance of FL from both uplink and downlink transmissions' perspectives, mainly downlink transmission [20]. In the downlink transmission, the server sends the global model to the users; in the uplink transmission, it is the reverse. This study focuses on the effect of noisy downlink transmission on FEEL. Supported by the experiments, it is pointed out that analog downlink transmission is much more efficient in a non-IID scenario, with fewer local SGD iterations.

In addition, there is a method called "momentum" for increasing the efficiency of stochastic gradient descent applied in federated learning. Momentum is a variant of gradient descent optimization aiming to speed up the optimization process. Momentum brings a new parameter (mostly called momentum attenuation factor) to the equation that regulates the quantity of previous data to incorporate in the update equation. The momentum attenuation factor is in the range of 0 to 1, and 0 corresponds to gradient descent without momentum. A large value of the momentum attenuation factor means that the current update is strongly affected by the previous update, whereas a lower value means the reverse [21]. An illustration about momentum is provided in Figure 2.3.

There are many applications of momentum in federated learning. Xu et al. studied expanding the *FederatedAveraging* algorithm introduced in [5], by adding a momentum factor to it, supported by the convergence analysis and experiments [22]. This method is named as *FedCM*, aiming to solve the problem of client heterogeneity and partial participation. It is stated that this method introduces a correction term to the local gradient direction as the momentum attenuation factor, and the smaller this term is, the more global gradient information is included in the update. Kim et al. proposed another method named *FedAGM*, to deal with the challenge of low convergence rate [23]. *FedAGM* uses momentum to accelerate the model training process and aims to

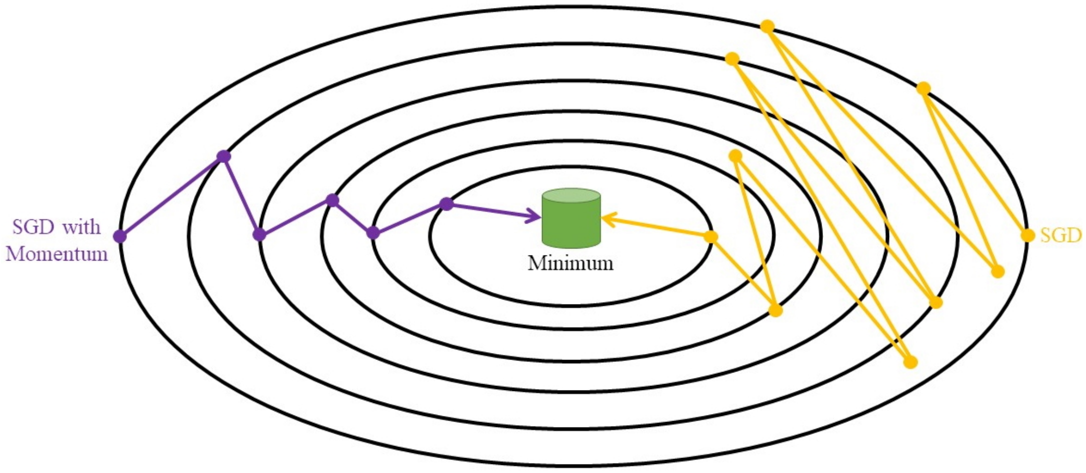

Figure 2.3: Comparison of SGD with and without momentum.

improve the convergence and accuracy of the model. Supported by the experiments, it is shown that the proposed algorithm performs better than the other various federated learning algorithms, including *FedCM*. Liu et al. proposed a method named *Momentum Federated Learning (MFL)* and studied defining the proposed method's global convergence characteristics and estimating an upper bound on the convergence rate [24]. Supported by the numerical analysis and experiments, it is shown that the proposed method has a positive effect on the convergence rate. The circumstances under which the proposed method accelerates convergence are also examined. Huo et al. studied on *FederatedAveraging* and their proposed method named *FedMom* from a perspective of non-convex problems and showed that their proposed method does not violate convergence guarantees for non-convex problems [25].

2.2 Age of Information

The age of information was introduced by Kaul et al. in [26] and [27], to adjust the freshness of information in status-update systems. The AoI quantifies the staleness of the information at the destination and is defined as the time elapsed since the generation time of the most recent status update packet successfully received at the destination. The effectiveness of AoI requires low-latency packets to be received promptly.

AoI is a substantial performance criterion, especially for status update applications, which are becoming increasingly crucial in machine-type communications, industrial applications, the Internet of Things, social networks, etc. Mathematically, AoI is defined as the following:

$$\Delta_t = t - u(t), \tag{2.1}$$

where Δ_t is the AoI and $u(t)$ is the time of the last update for any time slot t. Time-average age for the second update is visualized in Figure 2.4 and analyzed by the yellow area. This can be adapted to all updates. A general formula for calculating the time-average age is provided as follows:

$$\begin{aligned} Area &= \frac{1}{2}(X_n + Y_n)^2 - \frac{1}{2}(Y_n)^2, \\ \Delta_t &= \frac{E[Area]}{E[Y_n]} = \frac{E[Y_n T_n] + E[Y_n^2]/2}{E[Y_n]}, \end{aligned} \tag{2.2}$$

where $E[.]$ denotes the expectation, X_n denotes the time elapsed between the n-th and $(n+1)$-th system updates, and Y_n denotes the time between the n-th system update and its receivement.

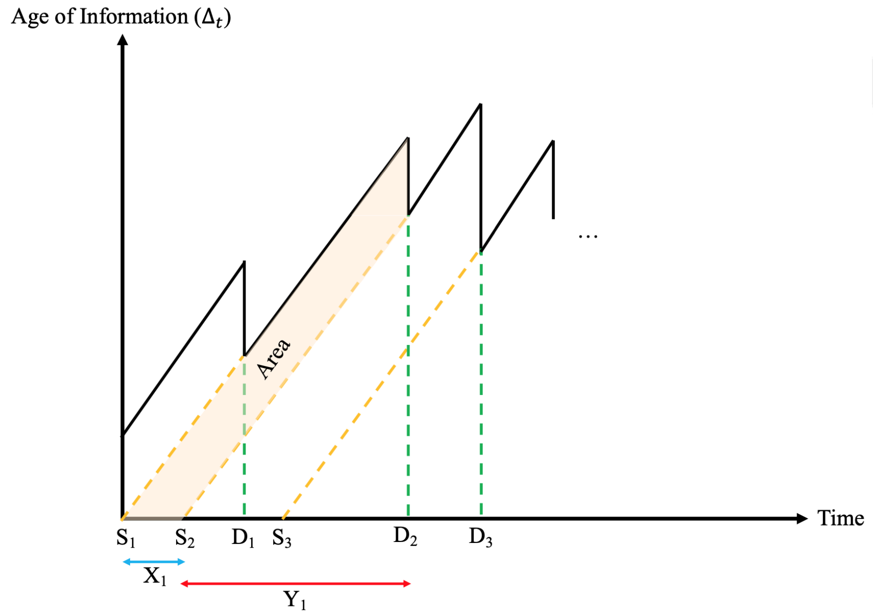

Figure 2.4: Sample change of age of information for a first-come-first-serve network. X_1 denotes the time elapsed between the first two system updates, and Y_1 denotes the time between the first system update and its receivement.

Optimal transmission scheduling, first formulated in [28] and [29], is the concern of modifying rate and power in time to energy and data arrivals and channel diversity to transfer data as precisely as possible. Today, there has been a growing interest in minimizing the AoI in communication systems. Bacınoğlu et al. studied solving the continuous-time problem of optimizing status updates for reducing the average age of information [30]. Kadota et al. studied the formulation of a discrete-time decision problem of a scheduling policy that minimizes the AoI of the clients in the network [31]. Sun et al. studied managing the AoI of status updates sent from a source to a remote monitor via a network server and formulated a constrained semi-Markov decision process (SMDP) problem [32]. Their work was an important reference because of the detailed optimization problem solutions. Ceran et al. examined both standard automatic repeat request (ARQ) and hybrid ARQ (HARQ) protocols and with a contribution of reinforcement learning, minimizing the long-term average AoI under a constraint on the average number of transmissions at the source node [33]. Scheduling problems conditioned on energy efficiency also take part in communication systems. Yates et al. mainly examine the issue of status updates by an energy harvesting source [34]. Their work is important in the manner of combining and analyzing the freshness and energy harvesting concepts together. Bacınoğlu et al. studied formulating two offline transmission scheduling problems for the transmitter-centric and the receiver-centric wireless energy transmission (WET) [35]. Bacınoğlu et al. also studied non-linear age penalty optimization under the restriction that the number of energy units possible to be stored at one time is restricted by the battery capacity [36].

In the federated learning area, age is defined as the time elapsed between receiving the local updates from a participant user. Yang et al. studied a metric called “age of update” and a scheduling policy is proposed, which takes channel parameters and age into account [37]. The aim is to find the minimum age of update, with the constraints of maximum transmit power, avoiding interference and rate exceeding a threshold. To define the age-optimal number of total and earliest participant users, Büyükateş et al. studied the metric of the average age of information of each client, and numerical results show that the suggested communication strategy not only ensures timeliness but also reduces average iteration durations without negatively affecting the convergence [38]. Liu et al. focused on an age-aware communication method for federated learning over wireless networks that takes both the staleness of parameters and capabilities of end devices into account to achieve precise and efficient model training over non-IID data [39]. Numerical results support the performance of the proposed method. Aygün et al. defined a metric called "cooldown multiplier", the time elapsed between two most recent energy arrivals, to weight the model differences [18].

CHAPTER 3

ACHIEVING OPTIMAL AGE OF INFORMATION WITH WIRELESS ENERGY TRANSFER

3.1 Introduction

In this chapter, an optimal threshold-based decision policy that aims to provide the lowest long-term average AoI is studied. A receiver pulls data from a transmitter on a binary channel by providing the transmitter with enough energy by wireless energy transfer (WET). It is assumed that the receiver has infinite energy to perform this energy transfer, and the goal of the receiver is to control the time-average AoI by taking the previous action and the channel state into account. The channel state as ON/OFF is known instantaneously by the receiver. The optimization problem is modeled as a constrained Markov decision problem, and both the optimal decision policy and the threshold are obtained. The optimal time average AoI and age violation probabilities are also provided. The performance of the proposed decision policy is evaluated by comparing it to a benchmark uniform transmission policy. In Section 3.2, system model and problem definition are presented. In Section 3.3.1, steady-state analysis is presented. In Section 3.3.2, optimality of the decision policy is demonstrated. In Section 3.3.3, optimal AoI threshold is derived. In Section 3.3.4, age violation probability is derived. In Section 3.4, experimental results and evaluation of the proposed decision policy are provided.

Parts of this work has been presented as a poster presentation at London Symposium on Information Theory (LSIT) in 2019, and an extension of this work has been studied in [2] and presented at the 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt) in 2021.

3.2 System Model and Problem Definition

As illustrated in Figure 3.1, a point-to-point channel including a transmitter-receiver pair is considered. The receiver gets data from the sensor based on the requests it obtains. It pulls data from the transmitter by sending energy to be harvested, named receiver-centric scheduling. Data requests arrive intermittently to the receiver, and it is expected from the receiver to determine whether the energy transmission should be performed or not by taking the previous action and the channel state into account, thus, controlling the long-term average age of information (AoI). It is important to point out that the transmitter is responsible for transmitting data to the receiver only by using the energy it received, so the harvested energy is not stored. The system model is simplified in the sense that each transmission requires one unit of energy, and the long-term energy usage is limited by the long-term average energy constraint, denoted by λ, per time slot. It is also assumed that the receiver has an infinite energy source. The channel state changes as ON and OFF from one time slot to the other as in independent and identically distributed (IID) form with their corresponding probability values. It is assumed that when the channel state is ON, any transmitted packet is correctly decoded, and when the channel state is OFF, there is no successful transmission. AoI at time slot t, denoted by Δ_t, is provided to the receiver by the network, and it is assumed that unless the age is greater than a specific threshold value, the transmission does not occur. Throughout this chapter, it is assumed that the AoI increases by one when a transmission fails, whereas it decreases to one when a successful transmission occurs. The system model is illustrated in Figure 3.1.

3.3 Proposed Method

A non-decreasing time-average age penalty function, $g(\Delta)$, is a function of AoI and is defined to evaluate the staleness of data packets under diverse scenarios. Suppose the age penalty function is an identity function. In that case, the expected age penalty becomes the time-average AoI. If $g(\Delta) = \mathbb{1}_{\Delta>\gamma}$, where γ is the age violation threshold, the expected age penalty becomes the age violation probability. Note that these are the two extensively used timeliness metrics in the literature.

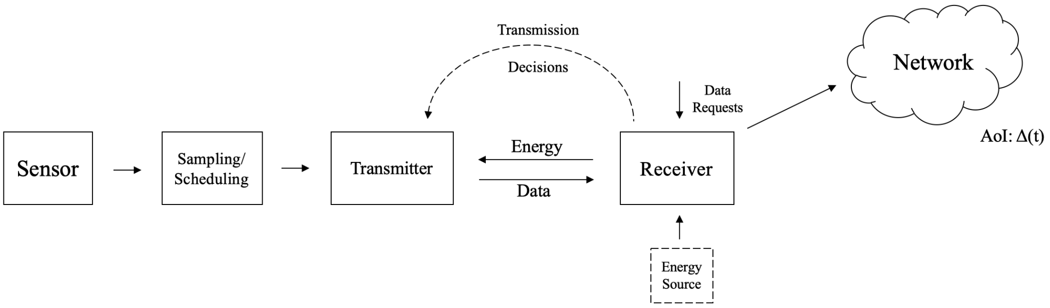

Figure 3.1: Receiver-centric system model. Data requests arrive intermittently to the receiver and the receiver determines whether the energy transmission should be performed or not. The transmitter is responsible for transmitting data to the receiver only by using the energy it receives.

The problem can be modeled as an infinite-state, constrained Markov decision process (CMDP). The constrained Markov decision process formulation is defined by the 5-tuple: (S, A, P, c, d): the countable set of states $\mathcal{S} = \mathbb{Z}^+ \times \{\mathrm{ON}, \mathrm{OFF}\}$ and the finite action set $A = \{0, 1\}$ are defined. 1 denotes that the transmission will be performed, and 0 denotes that no transmission occurs. The state s_t consists of the AoI Δ_t at time t and the channel state C_t at time t. P refers to the transition function, where P$(s'|s,a) = Pr\{s_{t+1} = s'|s_t = s, a_t = a\}$ is the probability that action a in state s at time t will lead to state s' at time $t+1$. The cost function c is the AoI at the destination, and is defined as $c(s,a) = g(\Delta_t)$. It is a non-decreasing function of state and AoI, which are independent of each other. The transmission cost d is identical for each transmission, $d = 1$ if $a = 1$ and $d = 0$ otherwise. A stationary policy is a decision rule denoted by π, which maps the states s into actions a with some probability $\pi(a|s)$.

Given the initial state s_0 = (1, ON) and under an energy constraint, the goal is to minimize AoI under the optimal policy π with the help of age penalty function $g(\Delta_t)$. The AoI, Δ_t, either increases by 1 or drops to 1, depending on the success of transmissions:

$$\Delta_{t+1} = \begin{cases} 1, & \text{if } C_t = ON \text{ and } a_t = 1 \\ \Delta_t + 1, & \text{otherwise} \end{cases} \tag{3.1}$$

Channel states are ON and OFF in each time slot in an IID fashion with probabilities

Table 3.1: Chapter 3: Parameter Symbols and Definitions

Parameter	Definition
Δ, Δ_t	Age of information, age of information at time slot t
A, a_t	Action set, action at time slot t
S, s_t	State set, state at time slot t
C_t	Channel state at time slot t
P	Transition function
$c(s, a)$	Cost function
λ	Long-time average energy constraint
d	Transmission cost
P_{ON}	Probability that channel state is ON
P_{OFF}	Probability that channel state is OFF
π^*	Optimal decision policy
θ	Optimal age of information
p_θ	Randomization coefficient
γ	Age violation threshold
$g(\Delta)$	Age penalty function

P_{ON} and P_{OFF} (P_{ON} > 0), and are not affected by the actions:

$$Pr\{C_{t+1} = c\} = \begin{cases} P_{ON}, & \text{if } c = ON \\ P_{OFF}, & \text{if } c = OFF \end{cases} \tag{3.2}$$

The CMDP optimization problem is defined below, where $E[.]$ represents the expectation concerning the distribution of AoI produced by policy π and channel states C_t:

$$\begin{aligned} \min_{\pi} \Delta^{\pi}(s_0) &= \lim_{T\to\infty} \frac{1}{T} E\left[\sum_{t=1}^{T} g(\Delta_t)|s_0\right], \\ \text{subject to } &\lim_{T\to\infty} \frac{1}{T} E\left[\sum_{t=1}^{T} a_t^{\pi}|s_0\right] \leq \lambda \end{aligned} \tag{3.3}$$

The main focus of this work is to obtain an optimal decision policy π^* that solves the time-average expected AoI minimization problem defined in (3.3). In the following sections, it will be shown that an optimal stationary policy exists, and the structure of the optimal policy will be defined. In general, CMDPs with countably infinite

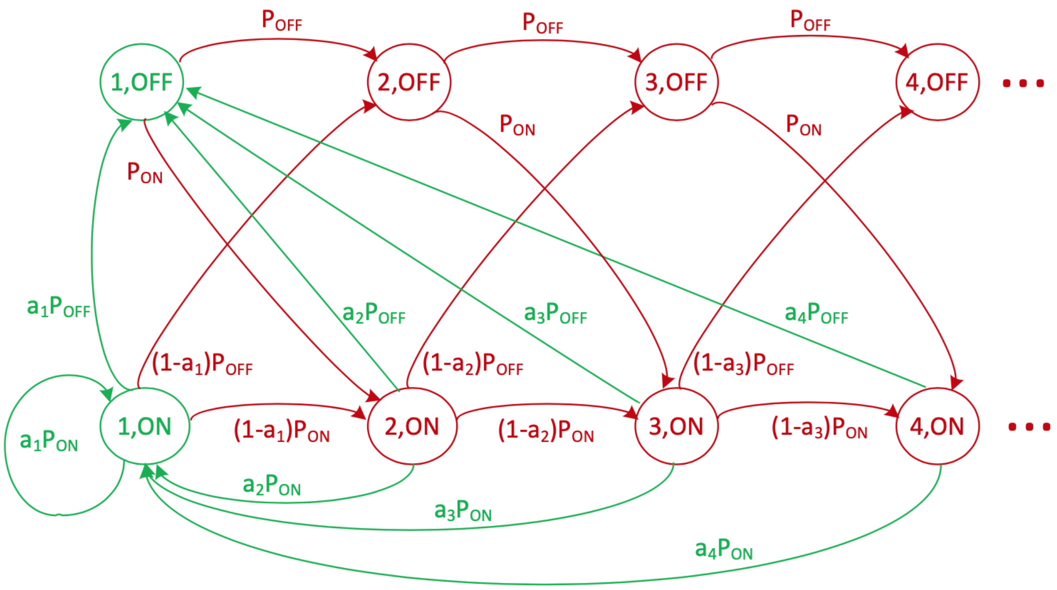

Figure 3.2: Markov chain representation of the system model. a_k represents the transmission probability when age is equal to k and the channel is ON. Green arrows indicate successful transmissions, and red arrows otherwise.

state spaces are difficult to solve because a stationary optimal policy, or an optimal policy in general, is not guaranteed to exist [40]. In the following theorem, it will be demonstrated that an optimal stationary policy exists, and the structure of such policy will be outlined.

Theorem 1 *There exists an optimal stationary policy for the CMDP in Problem 3.3, and it is randomized in at most a single point in the state-space S.*

Proof. The proof can be sketched out as follows: First, it is demonstrated that Assumptions 1-4 of [41] hold for Theorem 2.5, Proposition 3.2, and Lemma 3.9 of [41]. Then, according to Theorem 2.5 of [41], there exists an optimal stationary policy that is a combination of two deterministic policies that differ in at most one state, and there exists a randomization coefficient such that the corresponding decision policy satisfies the condition with equality [2]. The detailed proof can be achieved by performing the method described in [33].

3.3.1 Steady-State Analysis

For each $j \geq 1$, if the channel state is OFF between t and $t + j$, then age rises by j with probability 1, so that all states in the corresponding Markov chain presented in Figure 3.2 are accessible from the $\Delta = 1$ state. In this analysis, the predicted time between successive transmissions is considered to demonstrate that the state $\Delta = 1$ is positively recurring. The average energy cost would be zero if the predicted time were infinite, and such a policy would be lesser than the policies that meet the constraint in Problem 3.3. If the predicted time between transmissions is finite, then the Markov chain is positive recurrent, and the expected return time to the $\Delta = 1$ state is also finite. As a result, a policy that results in a steady-state distribution can be obtained [2].

3.3.2 Optimality of the Decision Policy

Optimality will be examined for the cases of $\lambda \geq P_{ON}$ and $\lambda < P_{ON}$. For the first case, notably, if $\lambda \geq P_{ON}$, the transmitter might not have to idle at a transmission opportunity because the infinite battery assumption makes such an unconstrained policy possible. Any policy that skips a transmission opportunity will only perform worse since any possible age plot will surpass the age plot of any policy that fully utilizes all ON slots in any sample route of the channel state process, which includes random occurrences of ON and OFF slots. The age will be lower than or equal to any other potential age graph reachable on the same sample route since the zero wait policy reduces the age to 1 at all ON slots. To conclude, for $\lambda \geq P_{ON}$, zero-wait policy is optimal [2]. For the second case, the following theorem can be stated:

Theorem 2 *There exists a stationary policy π^* that includes an integer Θ with the following probabilities:*

1. $Pr\{a = 1 \mid \Delta < \Theta, C = ON\} = 0$

2. $Pr\{a = 1 \mid \Delta > \Theta, C = ON\} = 1$

3. $Pr\{a = 1 \mid C = OFF\} = 0$

4. Pr{a = 1} = λ

The detailed version and the proof of this theorem are provided in [2]. In this theorem, the first two probabilities represent the structure of the threshold policy, and the last two represent energy utilization. The optimality proof starts by showing that the probability mass function of age at steady-state is monotonic. To show that, at any time t, the equation below can be stated:

$$Pr\{\Delta = j+1\} = Pr\{\Delta = j\}(1 - P_{ON}(Pr\{a_t = 1|s_t = (j, ON)\})) \tag{3.4}$$

This implies that $Pr\{\Delta = j+1\} \leq Pr\{\Delta = j\}$, so the monotonicity holds. Next, the age violation probability will be inspected to show the optimality, which is demonstrated in Lemma 1 from [2]. The lower bound defined in [2] is valid for a successful transmission occurs with probability λ at steady-state. The energy constraint must be completely exploited, and any available energy must not be used on transmitting while the channel is OFF. Also, there must be no successful transmission when the age is below λ. In general, the lower bound holds if transmission occurs when the channel state is ON and the AoI is greater than or equal to j. To maintain larger violation thresholds, Lemma 2 is defined in [2]. The optimal decision policy can be defined as follows:

$$\pi^*(a_t = 1|s_t = (\Delta_t, C_t)) = \begin{cases} 1, & \Delta_t > \Theta \text{ and } C_t = ON \\ p_\Theta, & \Delta_t = \Theta \text{ and } C_t = ON \\ 0, & \Delta_t < \Theta \text{ or } C_t = OFF \end{cases} \tag{3.5}$$

The derivation of Θ and p_Θ will be performed in the following section.

3.3.3 Derivation of the Optimal AoI Threshold

Let q_k denote the steady state probability of $\Delta_t = k$, $Pr\{\Delta_t = k\}$. The state transition probabilities and total probability equation are as follows:

$$\begin{aligned} &q_k = q_1 \text{ if } k \leq \Theta \\ &q_{\Theta+1} = q_\Theta(1 - p_\Theta P_{ON}) \\ &q_k = q_{k-1}P_{OFF} \text{ if } k \geq \Theta + 2 \\ &\sum_{k=1}^{\infty} q_k = 1 \end{aligned} \tag{3.6}$$

A closed-form solution of q_k can be obtained by solving these equations together, with the first element of the series being equal to:

$$q_1 = \frac{1}{\Theta - p_\Theta + \frac{1}{P_{ON}}} \tag{3.7}$$

Since q_1 is the expected average energy consumption, the following can be stated:

$$\lambda = \frac{1}{\Theta - p_\Theta + \frac{1}{P_{ON}}} \tag{3.8}$$

Finally, since Θ is an integer and $p_\Theta \in (0, 1]$, the parameters of the optimal policy can be derived as follows:

$$\Theta = \left\lfloor 1 + \frac{1}{\lambda} - \frac{1}{P_{ON}} \right\rfloor \tag{3.9}$$

$$p_\Theta = \Theta - \left(\frac{1}{\lambda} - \frac{1}{P_{ON}} \right) \tag{3.10}$$

Note that the optimal threshold value depends on the value of λ; the smaller the value, the greater the optimal threshold. For a greater value of the optimal threshold, noting that the receiver waits to pull data until that threshold is achieved, the probability that the transmission occurs decreases significantly.

3.3.4 Derivation of the Age Violation Probability

Age violation probability is defined as the probability that age is above a constant integer value, denoted by γ. If transmission occurs at $s = (j, ON)$ state and if the channel is OFF and age is j at time t, the above equation can be defined for any integer m:

$$P_{OFF}^m Pr\{\Delta = j\} \leq Pr\{\Delta = j + m\} \tag{3.11}$$

With the help of (3.11), the following can be derived:

$$\begin{aligned} Pr\{\Delta \geq j+1\} &= \sum_{m=1}^{\infty} Pr\{\Delta = j + m\} \\ \geq \sum_{m=1}^{\infty} P_{OFF}^m Pr\{\Delta = j\} &= \frac{P_{OFF}}{P_{ON}} Pr\{\Delta = j\} \end{aligned} \tag{3.12}$$

It is known that $Pr\{\Delta \geq j\} - Pr\{\Delta \geq j+1\} = Pr\{\Delta = j\}$. This can be used for transforming (3.12) into:

$$P_{OFF} Pr\{\Delta \geq j\} \leq Pr\{\Delta \geq j+1\} \tag{3.13}$$

To generalize:

$$P^{m}_{OFF}Pr\{\Delta \geq j\} \leq Pr\{\Delta \geq j+m\} \tag{3.14}$$

From Lemma 1 from [2], below can be defined:

$$Pr\{\Delta \geq \gamma+1\} \geq P^{\gamma-r}_{OFF}Pr\{\Delta \geq r+1\} \geq P^{\gamma-r}_{OFF}(1-\gamma r) \tag{3.15}$$

The age violation probability under the optimal policy is given in the following:

$$\Pr\{\Delta_t > \gamma\} = \sum_{k=\gamma+1}^{\infty} q_k = \begin{cases} 1-\lambda\gamma, & \gamma \leq \Theta \\ P^{\gamma-\Theta}_{OFF}\left(1-\lambda\Theta\right), & \gamma \geq \Theta \end{cases} \tag{3.16}$$

3.4 Performance Evaluation

To verify the proposed policy's efficiency, simulations are performed by comparing the performance of the proposed decision policy with a uniform transmission policy. In uniform transmission, transmission occurs only when there is an energy arrival and the channel is available at the same time, and age is not taken into account. The value of λ alters in a range of values starting by 0.05, increasing by 0.005, and reaching 1. For each value of λ, simulations are performed for $3{*}10^4$ time slots with 300 iterations and for two different P_{ON} values as 0.2 and 0.5. Channel is either ON or OFF uniformly randomly. Comparison of uniform transmission and optimal threshold policy in terms of time-average AoI and average energy consumption, for different values of P_{ON}, is provided in Figures 3.3 and 3.4. The same comparison in terms of age violation probability and average energy consumption is provided in Figures 3.5 and 3.6. Numerical results show that the proposed threshold policy is simple and computationally cost-effective, yet improves the performance considerably. Time-average age of uniform transmission is markedly higher compared to the proposed threshold policy for both of the probabilities that the channel is ON. As expected, if the value of P_{ON} increases, the time-average age decreases. Similarly, age violation probability is decreased with the proposed threshold policy, and the value of P_{ON} affects the age violation probability in the same way as the time-average age. Note that the average energy consumption does not exceed P_{ON}, which is in parallel with the assumption in Section 3.3.2. For the threshold policy, average energy consumption starts from the minimum value of the λ, satisfying the constraint defined in Problem 3.3. The optimal threshold directly depends on the value of λ and P_{ON}. Note that a higher value of λ and P_{ON} denotes that the value of AoI is decreasing, because if the channel is available with a higher probability, energy used for transmissions in a full round (in this case, $3 * 10^4$ time slots with 300 iterations) is going to be increased. The same logic is valid from the AoI perspective. To conclude, even in this simple setting with an ON/OFF channel, uniformly pulling data is inefficient. Getting fresh data through wireless energy transmission is possible by using an age threshold that depends on the average power available, and the channel capacity.

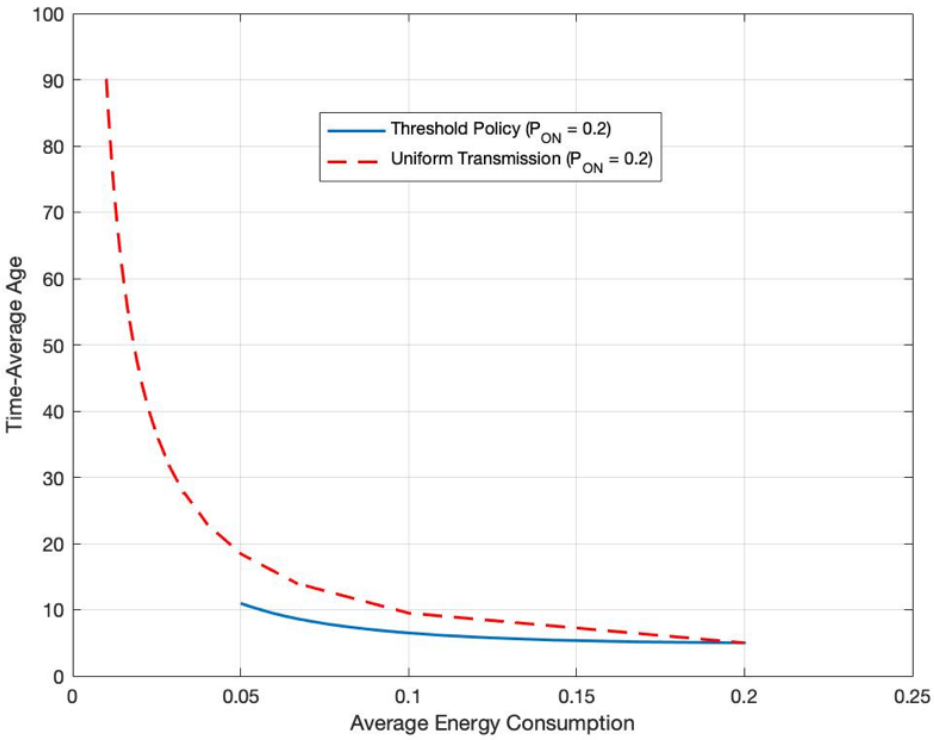

Figure 3.3: Comparison of uniform transmission and optimal threshold policy in terms of time-average AoI and average energy consumption.

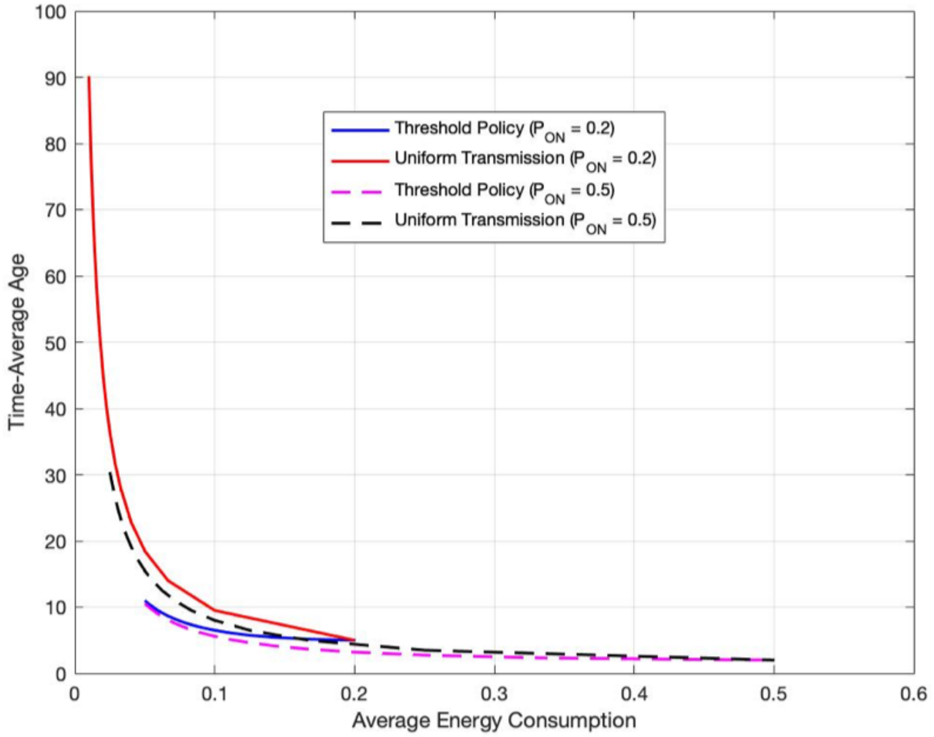

Figure 3.4: The effect of the probability that the channel is ON on the uniform transmission and optimal threshold policy in terms of time-average AoI and average energy consumption.

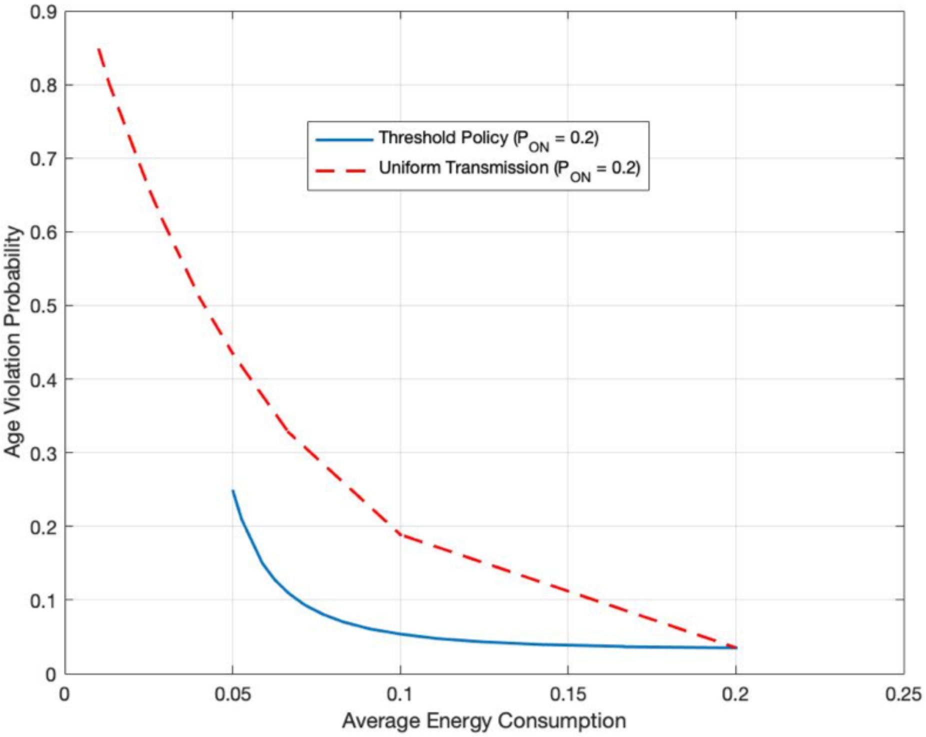

Figure 3.5: Comparison of uniform transmission and optimal threshold policy in terms of age violation probability and average energy consumption.

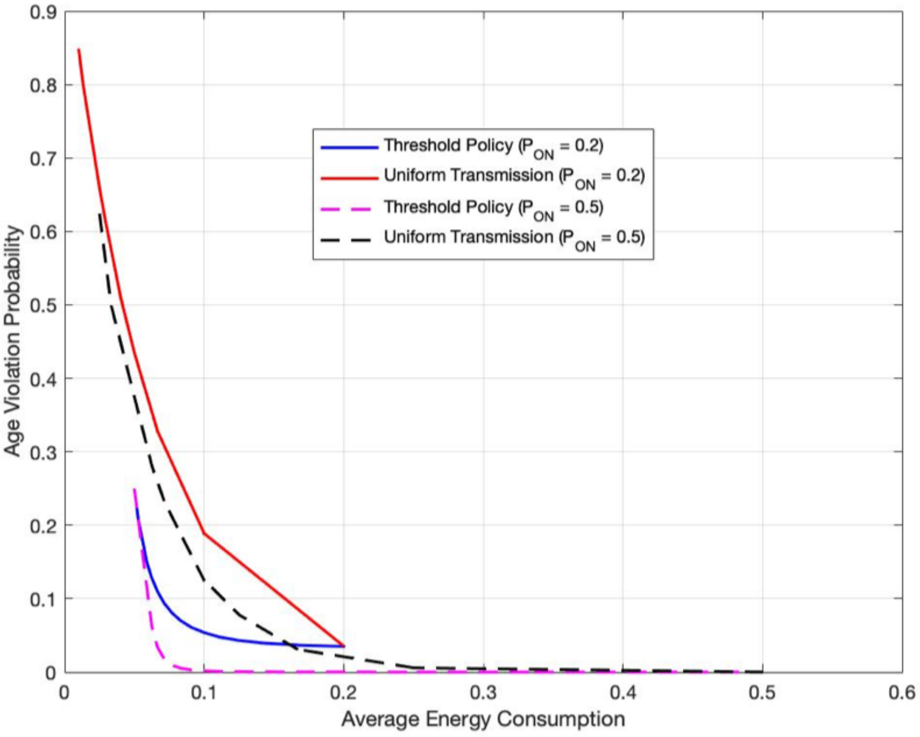

Figure 3.6: The effect of the probability that the channel is ON on the uniform transmission and optimal threshold policy in terms of age violation probability and average energy consumption.

CHAPTER 4

FEDERATED LEARNING WITH CHANNEL AND ENERGY AWARE SCHEDULING

4.1 Introduction

In this chapter, a federated learning algorithm that schedules users and weighting their local gradients according to the energy and channel profiles of each user is presented. A federated learning setup in which several users that harvest energy from the environment and collaboratively train a machine learning model under the constraints of intermittent energy arrivals and channel availability is studied. The main focus is to develop an algorithm that achieves a similar convergence as modern federated learning methods in a scenario with an error-prone channel and intermittent energy availability. Supported by the experiments, it has been seen that the proposed scheduling method provides higher test accuracy and lower train loss compared to the other methods. In Section 4.2, the system model and problem definition, including channel and energy models, are provided. In Section 4.3, scheduling methods for deterministic and stochastic energy arrivals and known and unknown channel states are presented. In Section 4.4, convergence analysis is presented. In Section 4.5, experimental results and evaluation of the proposed scheduling methods are provided.

Parts of this work are presented at 30th Signal Processing and Communications Applications Conference (SIU) in 2022.

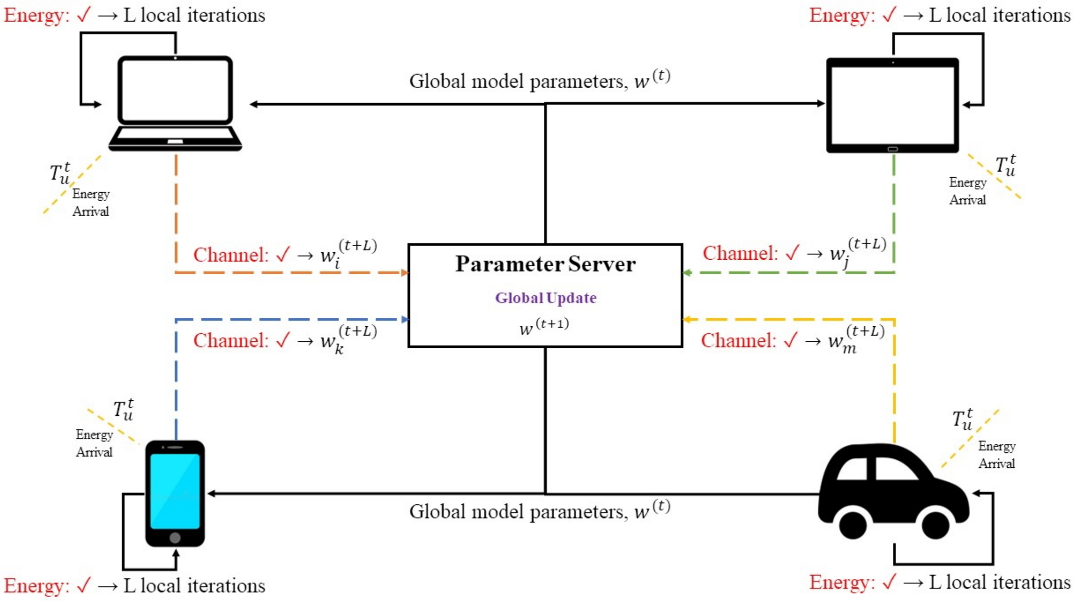
Energy: ✓ → L local iterations
Energy: ✓ → L local iterations
Global model parameters, $w^{(t)}$
T_u^t
Energy Arrival
T_u^t
Energy Arrival
Channel: ✓ → $w_i^{(t+L)}$
Parameter Server
Global Update
$w^{(t+1)}$
Channel: ✓ → $w_j^{(t+L)}$
Channel: ✓ → $w_k^{(t+L)}$
Channel: ✓ → $w_m^{(t+L)}$
T_u^t
Energy Arrival
T_u^t
Energy Arrival
Global model parameters, $w^{(t)}$
Energy: ✓ → L local iterations
Energy: ✓ → L local iterations

Figure 4.1: System model. Users are connected to a central parameter server and receive energy through an energy harvesting process. A user can join the global model update only if there is enough energy and the channel is available.

4.2 System Model and Problem Definition

As illustrated in Figure 4.1, a federated learning system with K users on the network is considered. These users are connected to a central parameter server and receive energy through energy harvesting. The arrival of energy can be deterministic or stochastic. In addition to the energy constraint, channel availability is also a criterion for the user to participate in the training. A user can join the training only if there is enough energy and the channel is available. The energy model and the channel model are the two determining factors in how scheduling will be done. The aim is to minimize the global loss function under the awareness of energy and channel state. The illustration of the system model is in Figure 4.1.

Assuming that a user $i \in \{1, \dots, K\}$ has D_i data points in its local dataset, the total number of data points for all users can be defined as D. With these definitions, the global loss function can be defined as follows:

$$F(w) = \sum_{i=1}^{K} p_i F_i(w), \tag{4.1}$$

In this equation, K is the number of users, p_i is the ratio of the user i's local dataset size to the entire dataset size ($p_i = \frac{D_i}{D}, \sum_{i=1}^{K} p_i = 1$), and the function $F_i(w)$ represents the local loss function. The local loss function of user i is defined as follows:

$$F_i(w) = \frac{1}{D_i} \sum_{j=1}^{D_i} l(w, x_{ij}), \tag{4.2}$$

The value $l(w, x_{ij})$ in this equation indicates the loss of the point x_{ij} in user i in the local dataset.

Training is performed by using the distributed SGD method. In this method, the model parameters are constantly updated in the negative direction of the gradient. Estimation of the model parameters for the global round $t \in 0, 1, 2, ..$ is represented by $w^{(t)}$. In the distributed SGD method, the parameter server sends the value $w^{(t)}$ to participating users. The number of local training iterations (local rounds) performed by the participant user is defined by L. Users $i \in 1, 2, ..., K$ calculate a local stochastic gradient with L local iterations:

$$g_i(w^{(t)}, \xi_i^t) = \nabla F_i(w^{(t)}, \xi_i^{(t)}), \tag{4.3}$$

The value $\xi_i^{(t)}$ specifies a uniformly random sample from the local dataset. This ensures that the stochastic gradient is not biased. Under this assumption, the actual gradient value of user i can be defined as:

$$E_{\xi_i^{(t)}}[\nabla F_i(w^{(t)}, \xi_i^{(t)})] = \nabla F_i(w^{(t)}), \tag{4.4}$$

In this equation, the value $\nabla F_i(w^{(t)})$ specifies the gradient of the local loss function. The gradient of the global loss function is defined as follows:

$$\nabla F(w^{(t)}) = \sum_{i=1}^{K} p_i \nabla F_i(w^{(t)}) \tag{4.5}$$

After users complete their local calculations, local gradient values are sent to the parameter server. The parameter server updates the model as follows:

$$w^{(t+1)} = w^{(t)} - \eta \sum_{i=1}^{K} p_i g_i(w^{(t)}, \xi_i^{(t)}) \tag{4.6}$$

In this equation, the value η indicates the learning rate. After the update, the model is sent back to participating users, and the cycle continues until the global training process is complete.

4.2.1 Energy Model

In this study, it is assumed that the users get energy through the energy harvesting process. Energy can be provided by the environment in various ways. In this study, it is assumed that a step in the SGD method, including calculating the local gradient and sending it to the parameter server, costs each user a unit amount of energy. It is also assumed that each user has a unit battery that stores enough energy for one SGD step.

The energy arrival process of users is indicated by E_i^t. It is assumed that if there is an energy arrival, $E_i^t = 1$, otherwise $E_i^t = 0$. The distribution of energy arrivals varies depending on whether the harvesting process is deterministic or stochastic.

4.2.1.1 Deterministic Energy Arrivals

In the case of deterministic energy arrival, users know when the energy will arrive. It is assumed that there is just one energy arrival in the same global round, and it is unit energy. In this energy arrival state, the T_i^t parameter specifies the time elapsed between the two energy arrivals of the user i at time t. This parameter will be used as the "energy arrival parameter" in the following sections.

4.2.1.2 Stochastic Energy Arrivals

In the case of stochastic energy arrival, users do not know exactly when the energy will arrive, but they know the probabilistic model of the energy arrival process. In this study, the particular focus is on the "binary energy arrival" setup for the stochastic energy state. Binary energy arrival is defined as a Bernoulli process with energy arrival probabilities as β_i for each user i. The user i receives a unit amount of energy with β_i per global round. The value of β_i is between 0 and 1, which may vary from user to user.

$$E_i^t = \begin{cases} 1, & \text{with the probability } \beta_i \\ 0, & \text{with the probability } 1 - \beta_i \end{cases} \tag{4.7}$$

4.2.2 Channel Model

The channel state, shown as Q_i^t, indicates whether the channel of user i is available at global round t, and is defined as a Bernoulli process with channel error probabilities as q_i for each user i. The channel of user i is available with probability $1 - q_i$. The value of q_i is between 0 and 1, which may vary from user to user.

$$Q_i^t = \begin{cases} 0, & \text{with the probability } q_i \\ 1, & \text{with the probability } 1 - q_i \end{cases} \tag{4.8}$$

This chapter examines the cases where the channel status is known and not known by the users separately.

4.3 Proposed Methods

In this section, an algorithm that schedules users and weighting their local gradients according to each user's energy and channel profiles is proposed and adapted into combinations of different schemes: deterministic and stochastic energy arrivals, and channel status is known and not known.

4.3.1 Federated Learning with Deterministic Energy Arrivals

In this section, two scheduling methods are going to be explained for a setup with deterministic energy arrivals and known and unknown channel status, respectively.

4.3.1.1 Case 1: Channel Status is Known

When the channel state is known, the user determines an integer J with a certain probability in the range of 0 to $T_i^t - 1$, if it has enough energy. The value of this

Table 4.1: Chapter 4: Parameter Symbols and Definitions

Parameter	Definition
K	Total number of users
D_i	Number of data points in the local dataset of user i
p_i	The ratio of user i's local dataset size to the entire dataset size
L	Total number of local training rounds
$w^{(t)}$	Estimation of the model parameters for round t
$\xi_i^{(t)}$	Uniformly random sample from the local dataset of user i for round t
η	Learning rate
E_i^t	Energy arrival process for the user i for round t
T_i^t	Energy arrival parameter for the user i for round t
β_i^t	Energy arrival probability for the user i for round t
Q_i^t	Channel availability process for the user i for round t
q_i^t	Channel error probability for the user i for round t
χ_i	The scaling coefficient for $P(J = 0)$ for the user i
φ_i	The scaling coefficient for $P(0 < J \leq T_i^t - 1)$ for the user i
T	Total number of global rounds
S_t	The set of users who have successfully participated

integer depends on the energy arrival parameter of the participating user and the error probability of the channel. It is determined with the help of the following two different probability values:

$$P(J = 0) = \frac{1}{T_i^t - T_i^t q_i + q_i} \tag{4.9}$$

$$P(0 < J \leq T_i^t - 1) = \frac{1 - q_i}{T_i^t - T_i^t q_i + q_i} \tag{4.10}$$

The derivation of these probabilities is provided in Appendix A.

After the integer J is determined, the user is scheduled by adding J to the current global round value $(t + J)$. Once the scheduling is complete, the learning process can begin. In each global round, it is checked whether users can participate in the learning process in accordance with the scheduling, taking the channel availability into account. The user can participate in the learning process if the channel is available at the current global round. If not, the user is scheduled to participate in the next global round. Since the participation check of the users is performed in each global round, it is aimed that the user can participate in the learning process as soon as possible by finding the closest global round to which it can participate.

The parameter server sends the model parameters to the participating users, and the users perform the learning process by making local gradient calculations. The user-specific local gradient values are multiplied by a coefficient based on the energy arrival parameter and the channel error probability for each participant user. The scaling coefficient for $P(J = 0)$ for the user i is defined as:

$$\chi_i = \frac{1}{P(J = 0)} = T_i^t - T_i^t q_i + q_i$$

The scaling coefficient for $P(0 < J \leq T_i^t - 1)$ for the user i is defined as:

$$\varphi_i = \frac{1}{P(0 < J \leq T_i^t - 1)} = \frac{T_i^t - T_i^t q_i + q_i}{1 - q_i}$$

The scaling, dependent on the value of J, is performed as in the following:

$$g_i^{(t+L)} = [\chi_i \text{ or } \varphi_i](g_i(w^{(t)}, \xi_i^t)) \tag{4.11}$$

With the scaled local gradients, locally trained model parameters are obtained.

After the local learning is finished, locally trained model parameters, denoted as $w_i^{(t+L)}$, are sent to the server. The parameter server updates the global model as follows:

$$w^{(t+1)} = p_i * \sum_{i \in S_t} w_i^{(t+L)} \tag{4.12}$$

In this equation, S_t represents the set of users who have successfully participated in the learning process. Scheduling and training steps can be seen in Algorithm 2. This method allows users to participate more in the learning process as the channel status is known. In a worst case scenario where all of the users participate in the training and channels for all users are always available, the time complexity of Algorithm 2 is $\mathcal{O}(T * (KL + K)) = \mathcal{O}(TKL)$, where T is the total number of global rounds, K is the total number of users and L is the total number of local training rounds (local iterations).

4.3.1.2 Case 2: Channel Status is Unknown

In case the channel status is not known, if there is enough energy for the user, J is determined uniformly random between 0 and $T_i^t - 1$, and the user is scheduled by adding J to the current global round. Once the scheduling is complete, the learning process can begin. After users complete the local learning process with L local iterations, locally trained model parameters are sent to the server. The user-specific local gradients are obtained by multiplying the gradients with a different coefficient than before:

$$g_i^{(t+L)} = \frac{T_i^t}{1 - q_i}(g_i(w^{(t)}, \xi_i^t)) \tag{4.13}$$

If the channel is available at that global round, the parameter server updates the global model after obtaining the participants' locally trained model parameters.

$$w^{(t+1)} = p_i * \sum_{i \in S_t} w_i^{(t+L)} \tag{4.14}$$

On the other hand, if the channel is not available, the user will not be able to participate in the parameter server global model update. Scheduling and training steps can be seen in Algorithm 3. In this method, since the channel status is unknown, the closest global round that users can participate in cannot be obtained. For this reason, fewer users participate in the learning process than in Case 1. In a worst case scenario where all of the users participate in the training, the time complexity of Algorithm 3

Algorithm 2 Federated Learning with Deterministic Energy Arrivals When Channel Status Is Known

```
Require: Total number of global rounds $T$, number of users $K$, channel status for
    user $i$ $Q_i$, channel error probability of user $i$ $q_i$, initialized model parameters $w^{(0)}$
Ensure: Trained model parameters $w^{(T)}$
  Initialize $K_i^t = 0$ for t ∈ [T]
  for Global round $t = 0, ..., T - 1$ do
    for User $i$ in $K$ do
      if $E_i^t$ =1 then
        Determine $J$ using (4.9) and (4.10)
        Schedule $K_i^{t+J}$ =1
      end if
      if $K_i^t$ =1 then
        if $Q_i^t$ =1 then
          for Local iteration $m$ in $L$ do
            Calculate and scale the local gradient $g_i(w^{(t)}, \xi_i^{(t)})$
          end for
          Send the locally trained model parameters to the parameter server
        else if $Q_i^t = 0$ then
          Schedule $K_i^{t+J} = 0$ and $K_i^{t+J+1} = 1$
        end if
      end if
    end for
    Parameter Server:
    Update the global model using (4.12)
    Send model parameters $w^{(t+1)}$ to the users
  end for
```

is $\mathcal{O}(T * (KL + K)) = \mathcal{O}(TKL)$, where T is the total number of global rounds, K is the total number of users and L is the total number of local training rounds.

Algorithm 3 Federated Learning with Deterministic Energy Arrivals When Channel Status Is Unknown

```
Require: Total number of global rounds $T$, number of users $K$, channel error probability of user $i$ $p_i$, initialized model parameters $w^{(0)}$
Ensure: Trained model parameters $w^{(T)}$
  Initialize $K_i^t = 0$ for t ∈ [T]
  for Global round $t = 0, ..., T-1$ do
    for User $i$ in $K$ do
      if $E_i^t$ =1 then
        Determine $J$ as uniformly random between $0, ..., T_i^t - 1$
        Schedule $K_i^{t+J}$ =1
      end if
      if $K_i^t$ =1 then
        for Local iteration $m$ in $L$ do
          Calculate and scale the local gradients $g_i(w^{(t)}, \xi_i^t)$
        end for
        Send the locally trained model parameters to the parameter server
      end if
    end for
    Parameter Server:
    Update the global model using (4.14)
    Send model parameters $w^{(t+1)}$ to the users
  end for
```

4.3.2 Federated Learning with Stochastic Energy Arrivals

The system architecture of the stochastic energy arrivals is the same as the deterministic energy arrivals. However, there is no need to determine J in scheduling, and the coefficient multiplied by the local gradient values differs as the energy arrival method changes. Within the scope of stochastic energy arrival, the case of binary energy arrival has been examined. Different from the scheduling method for deterministic energy arrival, the battery status will be necessary for scheduling. The scheduling method is explained as follows: If there is an energy arrival, the user is directly scheduled. If there is no energy arrival but energy available at the user's battery, the channel status is checked. If the channel is available, the user is scheduled. If the channel is not available, the user is scheduled for the next round. With this approach, it is aimed to avoid the waste of energy.

After the scheduling, in each global round, it is checked whether users can participate in the learning process by taking the channel and energy availability into account before starting the local training. The user can participate in the learning process if the channel is available in the current global round. The user is scheduled to participate in the next global round if the channel is unavailable. Likewise, since the participation check of the users is performed in each global round, it is aimed that the user can participate in the learning process as soon as possible. When scheduling is complete, and users participate in the learning process and complete the local training with L local iterations, the user-specific local gradient value for both channel cases is multiplied by the coefficient obtained by replacing T_i^t with $\frac{1}{\beta_i}$. Likewise, this coefficient is used when updating the global model on the parameter server side. Algorithm 4 and 5 defines the learning process for stochastic energy arrivals. For both Algorithm 4 and Algorithm 5, in a worst case scenario where all of the users participate in the training, the time complexity is $\mathcal{O}(T * (KL + K)) = \mathcal{O}(TKL)$, where T is the total number of global rounds, K is the total number of users and L is the total number of local training rounds.

4.4 Convergence Analysis

To show that the proposed algorithm for deterministic energy arrivals and IID data does not violate the convergence guarantees, a few assumptions must be revisited:

Assumption 1 *(Variance Bound) The variance of the stochastic gradients from* (4.3) *are bounded:*

$$E_{\xi_i^{(t)}}[||g_i(w^{(t)}, \xi_i^{(t)}) - \nabla F_i(w^{(t)})||^2] \leq \sigma^2, i \in [K] \tag{4.15}$$

Algorithm 4 Federated Learning with Stochastic Energy Arrivals When Channel Status Is Known

```
Require: Total number of global rounds $T$, number of users $K$, channel status for
    user $i$ $Q_i$, channel error probability of user $i$ $q_i$, initialized model parameters $w^{(0)}$
Ensure: Trained model parameters $w^{(T)}$
  Initialize $K_i^t = 0$ and $B_i^t = 0$ for t ∈ [T]
  for Global round $t = 0, ..., T - 1$ do
    for User $i$ in $K$ do
      if $E_i^t$ =1 then
        Schedule $K_i^t$ =1
        Battery level $B_i^t = 1$
      else
        if $B_i^t$ =1 then
          if $Q_i^t$ =1 then
            Schedule $K_i^t$ =1
          else
            Schedule $K_i^{t+1}$ =1
            Schedule $K_i^t$ =0
          end if
        end if
      end if
      if $K_i^t$ =1 then
        if $Q_i^t$ =1 then
          for Local iteration $m$ in $L$ do
            Calculate and scale the local gradients $g_i(w^{(t)}, \xi_i^{(t)})$
          end for
          Send the locally trained model parameters to the parameter server
        else if $Q_i^t = 0$ then
          Schedule $K_i^t = 0$ and $K_i^{t+1} = 1$
          Battery level $B_i^{t+1} = 1$
        end if
      end if
    end for
```

```
        **Parameter Server:**
        Update the global model
        Send model parameters $w^{(t+1)}$ to the users
    **end for**
```

Algorithm 5 Federated Learning with Stochastic Energy Arrivals When Channel Status Is Unknown

```
**Require:** Total number of global rounds $T$, number of users $K$, channel error probability of user $i$ $q_i$, initialized model parameters $w^{(0)}$
**Ensure:** Trained model parameters $w^{(T)}$
    Initialize $K_i^t = 0$ for t $\in$ [T]
    **for** Global round $t = 0, ..., T - 1$ **do**
        **for** User $i$ in $K$ **do**
            **if** $E_i^t$ =1 **then**
                Schedule $K_i^t$ =1
            **end if**
            **if** $K_i^t$ =1 **then**
                **for** Local iteration $m$ in $L$ **do**
                    Calculate and scale the local gradients $g_i(w^{(t)}, \xi_i^t)$
                **end for**
                Send the locally trained model parameters to the parameter server
            **end if**
        **end for**
        **Parameter Server:**
        Update the global model
        Send model parameters $w^{(t+1)}$ to the users
    **end for**
```

Assumption 2 *(Second Moment Bound) The expected square norm of the stochastic gradients from* (4.3) *are bounded:*

$$E_{\xi_i^{(t)}}[||g_i(w^{(t)}, \xi_i^{(t))}||^2] \leq G^2, i \in [K] \tag{4.16}$$

Assumption 3 *(μ-Strong Convexivity) The local loss functions of the participating users and the global loss function are μ-strongly convex: For all $\boldsymbol{v}$ and $\boldsymbol{w}$,*

$$F_i(\boldsymbol{v}) \geq F_i(\boldsymbol{w}) + (\boldsymbol{v} - \boldsymbol{w})^T \nabla F_i(\boldsymbol{w}) + \frac{\mu}{2}||\boldsymbol{v} - \boldsymbol{w}||_2^2 \tag{4.17}$$

Assumption 4 *(L-Smoothness) The local loss functions of the participating users and the global loss function are L-smooth: For all $\boldsymbol{v}$ and $\boldsymbol{w}$,*

$$F_i(\boldsymbol{v}) \leq F_i(\boldsymbol{w}) + (\boldsymbol{v} - \boldsymbol{w})^T \nabla F_i(\boldsymbol{w}) + \frac{L}{2}||\boldsymbol{v} - \boldsymbol{w}||_2^2 \tag{4.18}$$

Let the scaling coefficient of the local gradients for $J = 0$ be χ_i and for $0 < J \leq T_i^t - 1$ be φ_i. Using these parameters, the following lemma can be defined:

Lemma 1 *(Unbiasedness) For distributed SGD with deterministic energy arrivals,*

$$\mathbb{E}_{S_t}\left[\sum_{i \in S_t} p_i \chi_i g_i(w^{(t)}, \xi_i^{(t)})\right] = \sum_{i=1}^{N} p_i g_i(w^{(t)}, \xi_i^{(t)}) \tag{4.19}$$

$$\mathbb{E}_{S_t}\left[\sum_{i \in S_t} p_i \varphi_i g_i(w^{(t)}, \xi_i^{(t)})\right] = \sum_{i=1}^{N} p_i g_i(w^{(t)}, \xi_i^{(t)}) \tag{4.20}$$

for $J = 0$ and $0 < J \leq T_i^t - 1$, respectively.

Theorem 3 *For training a machine learning model* (4.1) *using Algorithm 2 with deterministic energy arrivals and a learning rate $\eta \leq min\{\frac{1}{2\mu}, \frac{1}{L}\}$, the global loss function can be upper bounded as follows:*

$$\begin{aligned}\mathbb{E}_{S_t,\xi_t}[||w^{(t+1)} - w^*||^2 \leq (1 - \eta\mu)\mathbb{E}_{S_t,\xi_t}[||w^{(t)} - w^*||^2 + \eta^2(\sum_{i=1}^{K} p_i^2(\alpha_{i,max} - 1) \\ + \sum_{i=1}^{K}\sum_{j=1}^{K} p_i p_j)G^2\end{aligned} \tag{4.21}$$

in T iterations, where $\boldsymbol{w}^*$ denotes the optimal parameters that minimize the global loss function.

Proof. By letting $g_i^t \triangleq g_i(w^{(t)}, \xi_t)$, $\boldsymbol{w}^* \triangleq argmin_{\boldsymbol{w}} F(\boldsymbol{w})$, $\xi_t = (\xi_1^{(t)}, \xi_2^{(t)}, ..., \xi_K^{(t)})$, from (4.12), and $\alpha_i^t = \chi_i$ for J = 0 and $\alpha_i^t = \varphi_i$ otherwise, we find that:

$$\begin{aligned}\mathbb{E}_{S_t,\xi_t}[||w^{(t+1)} - w^*||^2] &= \mathbb{E}_{S_t,\xi_t}[||w^{(t)} - \eta \sum_{i \in S_t} p_i(\alpha_i^t g_i(w^{(t)}, \xi_i^{(t)})) - w^*||^2] \\ &= \mathbb{E}_{S_t,\xi_t}[||w^{(t)} - w^*||^2 - 2\eta \mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i \in S_t} p_i(\alpha_i^t g_i^t)\rangle] \\ &+ \eta^2 \mathbb{E}_{S_t,\xi_t}[||\sum_{i \in S_t} p_i(\alpha_i^t g_i^t)||^2]]\end{aligned} \tag{4.22}$$

The second term in (4.22) can be expanded as in the following:

$$\begin{aligned}\mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i \in S_t} p_i(\alpha_i^t g_i^t)\rangle] = \mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i \in S_t} p_i(\alpha_i^t g_i^t) \\ - \sum_{i=1}^{K} p_i \nabla F_i(w^{(t)}) + \sum_{i=1}^{K} p_i \nabla F_i(w^{(t)})\rangle]\end{aligned} \tag{4.23}$$

$$\begin{aligned}= \mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i \in S_t} p_i(\alpha_i^t g_i^t) - \sum_{i=1}^{K} p_i \nabla F_i(w^{(t)})\rangle] \\ + \mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i=1}^{K} p_i \nabla F_i(w^{(t)})\rangle]\end{aligned} \tag{4.24}$$

Because of Lemma 1, the first term in (4.24) vanishes, and by using Assumption 3 and (4.5), we get that:

$$\mathbb{E}_{S_t,\xi_t}[\langle w^{(t)} - w^*, \sum_{i=1}^{K} p_i \nabla F_i(w^{(t)})\rangle] = \langle w^{(t)} - w^*, \nabla F(w^{(t)})\rangle \tag{4.25}$$

$$\geq F(w^{(t)}) - F(w^*) + \frac{\mu}{2}||w^* - w^{(t)}||^2 \tag{4.26}$$

The third term in (4.22) can be expanded as in the following:

$$\mathbb{E}_{S_t,\xi_t}[||\sum_{i \in S_t} p_i \alpha_i^t g_i^t||^2]] = \mathbb{E}_{S_t,\xi_t}[||\sum_{i \in S_t} p_i \alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t + \sum_{i=1}^{K} p_i g_i^t||^2]] \tag{4.27}$$

$$\begin{aligned}= \mathbb{E}_{S_t,\xi_t}[||\sum_{i \in S_t} p_i \alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2]] \\ - 2\mathbb{E}_{S_t,\xi_t}[\langle \sum_{i \in S_t} p_i \alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t, \sum_{i=1}^{K} p_i g_i^t\rangle] + \sum_{i=1}^{K} ||p_i g_i^t||^2\end{aligned} \tag{4.28}$$

The second term in (4.28) vanishes, and the equation becomes:

$$\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t||^2]] = \mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2]] + \mathbb{E}_{S_t,\xi_t}[\sum_{i=1}^{K} ||p_i g_i^t||^2] \tag{4.29}$$

By combining (4.22), (4.26) and (4.29) and using Assumption 3, we get that:

$$\begin{aligned}\mathbb{E}_{S_t,\xi_t}[||w^{(t+1)} - w^*||^2] &= \mathbb{E}_{S_t,\xi_t}[||w^{(t)} - w^*||^2 - 2\eta\langle w^{(t)} - w^*, \nabla F(w^{(t)})\rangle \\ &+ \eta^2(\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2]] + \mathbb{E}_{S_t,\xi_t}[\sum_{i=1}^{K} ||p_i g_i^t||^2])\end{aligned} \tag{4.30}$$

$$\begin{aligned}&\leq (1-\eta\mu)\mathbb{E}_{S_t,\xi_t}[||w^{(t)} - w^*||^2] - 2\eta(F(w^{(t)}) - F(w^*)) \\ &+ \eta^2\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2]] + \eta^2\mathbb{E}_{S_t,\xi_t}[\sum_{i=1}^{K} ||p_i g_i^t||^2]\end{aligned} \tag{4.31}$$

Let $U_i^t = \begin{cases} 1, & \text{if the user participates at time } t \\ 0, & \text{otherwise} \end{cases}$ and $P(U_i^t = 1) = \alpha_i$. Under this definition, the third term in (4.31) can be written as in the following:

$$\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2] = \mathbb{E}_{U_t,\xi_t}[||\sum_{i=1}^{K} p_i(\alpha_i^t g_i^t - g_i^t)||^2] \tag{4.32}$$

$$\begin{aligned}&= \sum_{i=1}^{K} p_i^2\mathbb{E}_{U_t,\xi_t}[||\alpha_i^t g_i^t - g_i^t||^2] \\ &+ \sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K} \mathbb{E}_{U_t,\xi_t}[\langle p_i(\alpha_i^t g_i^t - g_i^t), p_j(\alpha_j^t g_j^t - g_j^t)\rangle]\end{aligned} \tag{4.33}$$

Because of independence, the second term in (4.33) vanishes:

$$\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2] = \sum_{i=1}^{K} p_i^2\mathbb{E}_{U_t,\xi_t}[||\alpha_i^t g_i^t - g_i^t||^2] \tag{4.34}$$

$$= \sum_{i=1}^{K} p_i^2(U_i^2)\mathbb{E}_{\xi_t}[\mathbb{E}_{U_t|\xi_t}[(U_i^t - \frac{1}{\alpha_i^t})^2 - ||g_i^t||^2|\xi_t] \tag{4.35}$$

By using Assumption 2, it can be stated that:

$$\mathbb{E}_{S_t,\xi_t}[||\sum_{i\in S_t} p_i\alpha_i^t g_i^t - \sum_{i=1}^{K} p_i g_i^t||^2] \le \sum_{i=1}^{K} p_i^2(\alpha_{i,max} - 1)G^2 \tag{4.36}$$

From Cauchy-Schwarz inequality, the last term in (4.31) can be expressed as in the following:

$$\eta^2\mathbb{E}_{S_t,\xi_t}[\sum_{i=1}^{K} ||p_i g_i^t||^2] \le \sum_{i=1}^{K} p_i^2\mathbb{E}_{\xi_t}[||g_i^t||^2] + \sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K} p_i p_j\mathbb{E}_{\xi_t}[||g_i^t||||g_j^t||] \tag{4.37}$$

$$\le \sum_{i=1}^{K} p_i^2\mathbb{E}_{\xi_t}[||g_i^t||^2] + \sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K} \frac{p_i p_j}{2}\mathbb{E}_{\xi_t}[||g_i^t||^2 + ||g_j^t||^2] \tag{4.38}$$

$$\le \sum_{i=1}^{K}\sum_{j=1}^{K} p_i p_j G^2 \tag{4.39}$$

Equation (4.38) holds by using AM-GM Inequality, and Equation (4.39) is stated by using Assumption 2. Finally, by combining (4.36) and (4.39) and noting that $-2\eta(F(w^{(t)}) - F(w^*)) \le 0$, it can be stated that:

$$\mathbb{E}_{S_t,\xi_t}[||w^{(t+1)} - w^*||^2 \le (1-\eta\mu)\mathbb{E}_{S_t,\xi_t}[||w^{(t)} - w^*||^2 + \eta^2(\sum_{i=1}^{K} p_i^2(\alpha_{i,max} - 1) + \sum_{i=1}^{K}\sum_{j=1}^{K} p_i p_j)G^2 \tag{4.40}$$

This completes the proof.

4.5 Performance Evaluation

Experiments were performed as an image classification task with 10 classes of 40 users, for 1000 global rounds and 5 local training rounds, using the CIFAR-10 dataset [42]. Sample images of CIFAR-10 dataset are provided in Figure 4.3 *(Retrieved from: https://www.cs.toronto.edu/ kriz/cifar.html)*. Dataset was distributed as 50,000 training samples and 10,000 test samples, with a batch size of 64. Images were preprocessed before the training to train the model more accurately, including horizontal

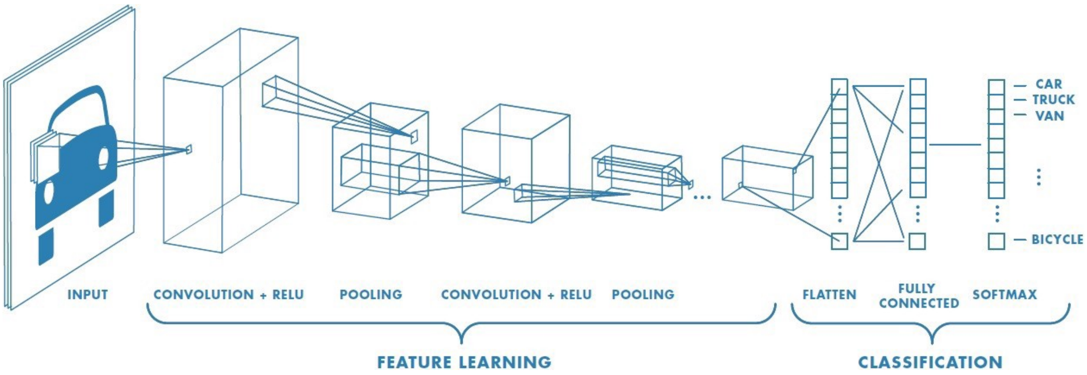

Figure 4.2: Architecture of the convolutional neural network (CNN).

and vertical flip, color jittering, resizing, and normalization. As the optimizer, SGD, which implements stochastic gradient descent, is used. The learning rate is set to 0.01. As the architecture, the convolutional neural network (CNN) is used, which includes three 3x3 convolutional layers (with 32, 64, and 64 channels, respectively, the first two with 2x2 pooling layers), a 0.25 dropout layer, a 64-unit fully connected layer, and an output layer for this specific scenario. The architecture of CNN is provided in Figure 4.2 *(Retrieved from: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53)*.

In the experiments, four different methods were simulated for the same scenario. In the first and second methods, the federated learning process in cases where the channel state is known or not known for deterministic energy arrivals (Algorithm 2 and Algorithm 3), respectively, is performed. In the third method, Conservative Algorithm is simulated, in which the learning process takes place only when all users have enough energy for deterministic energy arrivals, and the gradients are not multiplied by a coefficient. In the fourth method, as a baseline and reference for many federated learning algorithms in the literature, *FederatedAveraging* [5] algorithm is simulated. Note that this algorithm represents the performance in perfect conditions, i.e., no energy or channel constraint is included.

To show the effect of non-homogeneous energy arrivals, Güler et. al. [7] used a method of dividing users into four equal groups and assigning different energy arrival parameters to each group. This method is also adapted to the proposed algorithms,

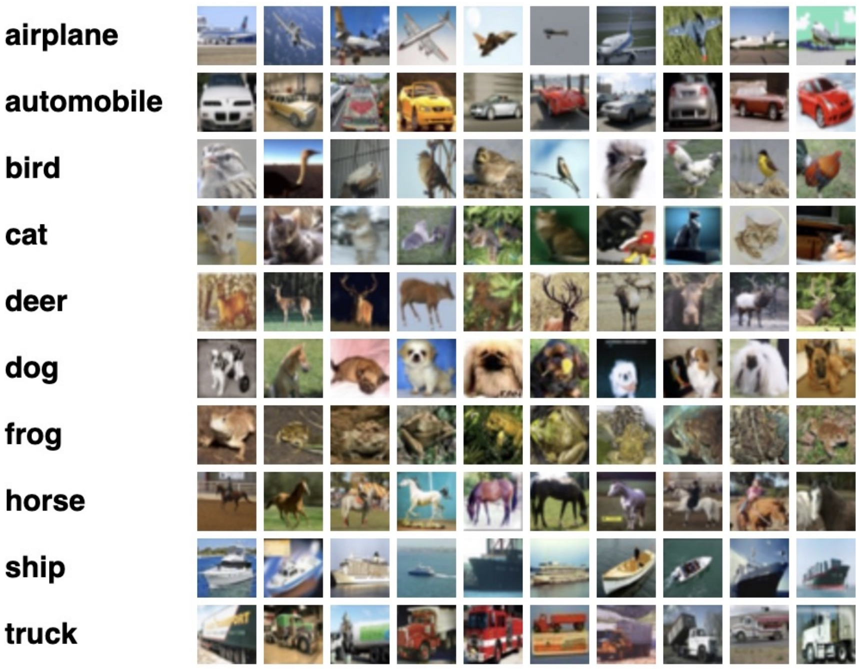

Figure 4.3: CIFAR-10 Dataset: Image classes and samples.

with the addition that different channel error probabilities were also assigned for each group. In addition, the channel models were generated in a way that the channel error probabilities were randomly assigned, independent of the energy harvesting. With this method, it is aimed to show that the proposed algorithm balances the participation of each user in the training process. The performance evaluation is conducted by calculating the test accuracy for each image class and taking the average.

In the first case of deterministic energy arrivals, the training dataset is IID to the users. This is performed by shuffling the dataset and splitting it between the users. Figure 4.9 shows the test accuracy versus the number of rounds for these methods and their comparison with the *FederatedAveraging* algorithm. It can be seen that the method with deterministic energy arrival, in which the channel state is known, provides high test accuracy compared to other methods. It is important to point out that the reference algorithm, *FederatedAveraging*, does not have any channel, processing power, or time constraints, and still, it provided 75% test accuracy, which is close to the accuracy of channel-aware scheduling (Algorithm 2). Note that the model used in these simulations is basic, and the proposed scheduling methods can be applied to more complicated and high-efficient models. The main goal of these experiments is

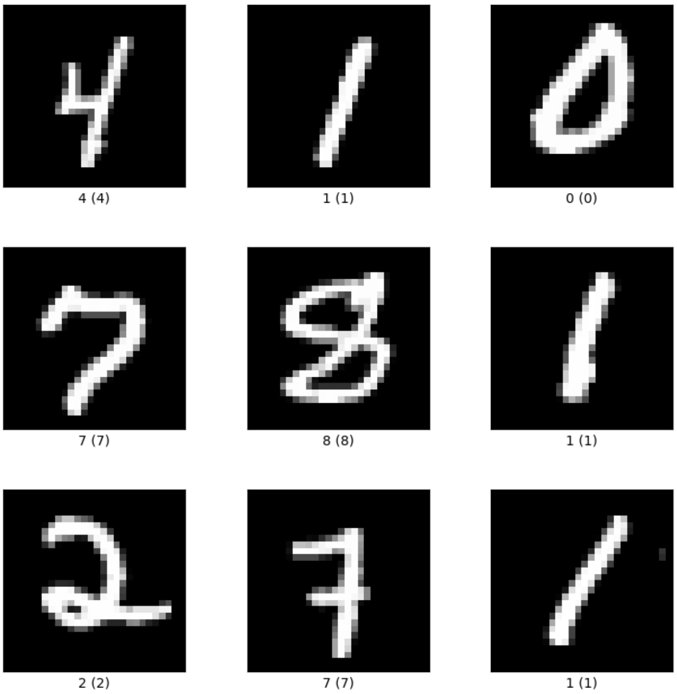

Figure 4.4: Samples from MNIST dataset.

to show that the proposed algorithms do not violate convergence guarantees and not to achieve higher test accuracy.

Figure 4.5 shows the effect of the learning rate on the proposed Algorithm 2. Note that the learning rate determines how quickly the model adapts to the situation, which is why it is one of the most crucial hyperparameters [3]. A lower learning rate might cause the training process to be slower, whereas a greater learning rate might lead the model to converge too soon to an unreliable result. It can be observed that Figure 4.5 reflects that statement: when the learning rate increases, the model converges more quickly, but the test accuracy becomes less consistent and reliable. Similarly, the training loss decreases quickly but changes more often, so it is also unreliable. Experimental results were obtained for IID data and deterministic energy arrival. In the first column, the learning rate was set as 0.01, and it can be seen that the model reaches an average test accuracy of 70% and a minimum of train loss of 0.37, which

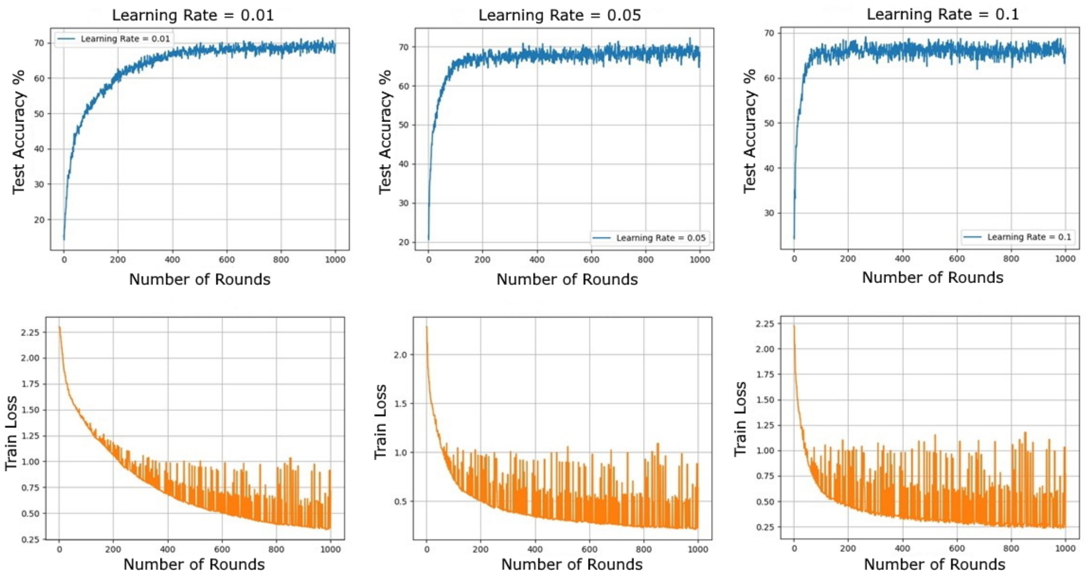

Figure 4.5: Test accuracy and train loss of channel aware scheduling (Algorithm 2) for different learning rates, for IID data and deterministic energy arrival.

demonstrates good performance. In the second column, the learning rate was set as 0.05, and it can be seen that the model reaches 70% test accuracy, but with a bit of a variance. Also, the minimum train loss was obtained approximately as 0.3, with a bit of a variance. Note that the convergence time decreased due to the increase in the learning rate. Lastly, in the third column, the learning rate was set as 0.1, and it can be seen that the model cannot reach 70% test accuracy, and variates more than the other cases. Also, the minimum train loss was obtained at approximately 0.25, with increased variance. Convergence is faster than in the other cases. These results support the conception explained in the beginning.

Figure 4.6 shows the effect of the number of local iterations on the proposed Algorithm 2. Note that the number of local iterations defines how often the participant user will go through its dataset, perform its training and converge to adequate model parameters. A smaller number of local iterations might cause the training process to be faster but less accurate, whereas a greater number of local iterations might lead the model to converge more slowly but to a reliable result. Supporting this conception, it can be seen from Figure 4.6 that when the number of local iterations increases, the model converges more slowly, whereas the test accuracy becomes better.

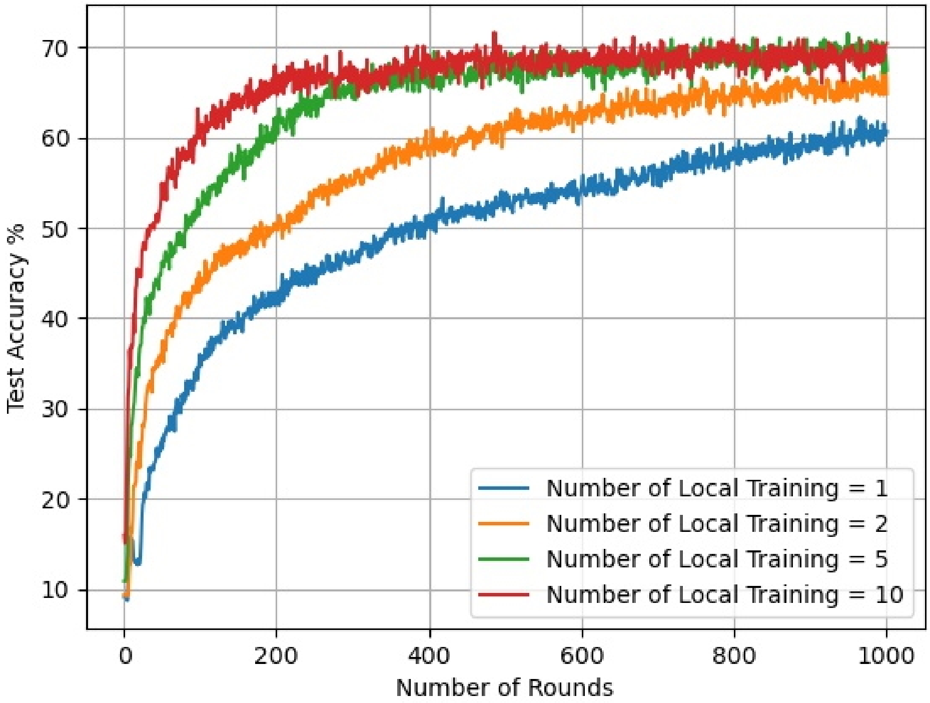

Figure 4.6: Test accuracy and train loss of channel aware scheduling (Algorithm 2) for different numbers of local training rounds, for IID data and deterministic energy arrival.

Figure 4.7 and Figure 4.8 shows the performance of proposed algorithms, Conservative Algorithm and *FederatedAveraging* algorithm for MNIST dataset [43], instead of CIFAR-10 dataset, for both IID and non-IID data. MNIST is an introductory yet useful dataset, which includes handwritten digits from 0 to 9 and has 60,000 samples for training and 10,000 samples for testing. Sample images of MNIST dataset are provided in Figure 4.4 *(Retrieved from: https://www.tensorflow.org/datasets/catalog/ mnist)*. The digits are in normal size and centered in a fixed-size image. It can be observed that the performance significantly improved for all of the algorithms compared to the CIFAR-10 dataset. This is because images in the MNIST dataset are relatively clean and easy to recognize and learn.

In the second case of deterministic energy arrivals, the training dataset is non-IID

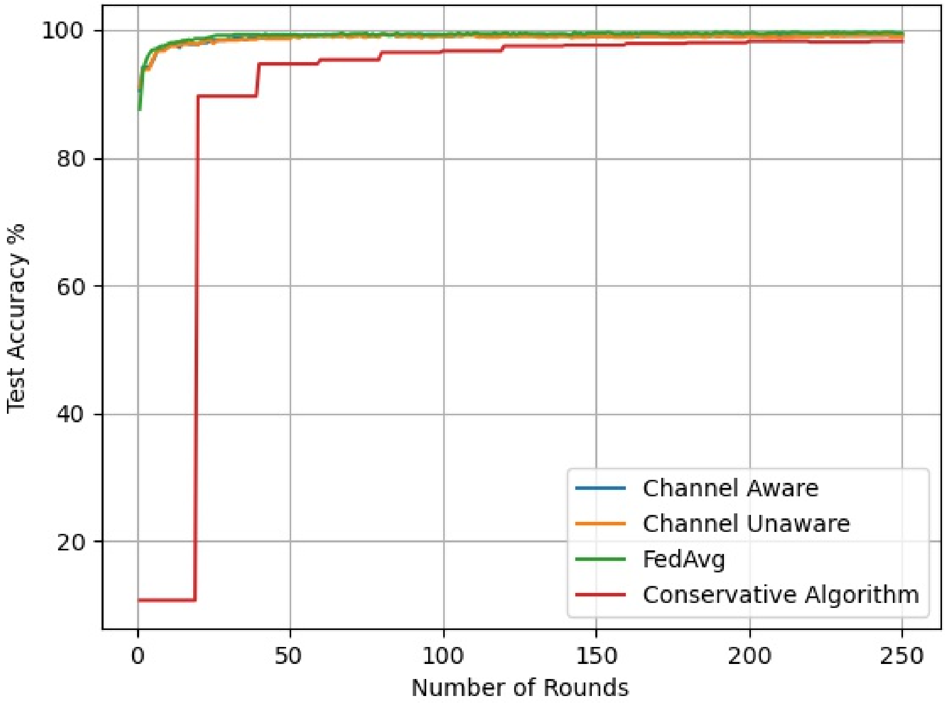

Figure 4.7: Test accuracy of channel aware scheduling (Algorithm 2) for MNIST for IID data and deterministic energy arrival according to the global rounds and comparison with channel unaware scheduling (Algorithm 3), Conservative Algorithm and *FederatedAveraging*.

to the users. This is performed by sorting the dataset by the digit label, dividing the data into 200 shards of size 250, and assigning two shards to each user. Figure 4.10 shows the test accuracy versus the number of rounds for these methods and their comparison with the *FederatedAveraging* algorithm. As expected, the accuracy goes back and forth because of the unbalanced data distribution among the participant users. Still, it can be observed that channel-aware scheduling (Algorithm 2) provides higher test accuracy compared to other methods. Another observation can be stated as the oscillation of the test accuracy, and the number and variation of participating users can be related. As an example, the test accuracy of the Conservative Algorithm is much more consistent compared to the others, or the test accuracy of Algorithm 3 does not oscillate as much as Algorithm 2. Note that in the Conservative Algorithm,

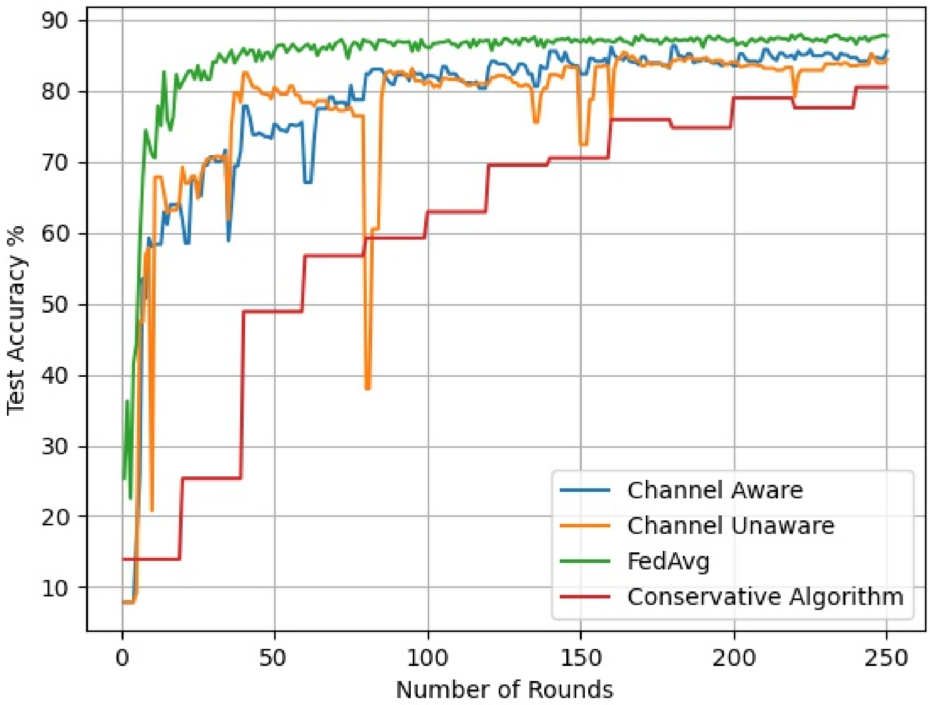

Figure 4.8: Test accuracy of channel aware scheduling (Algorithm 2) for MNIST for non-IID data according to the global rounds and comparison with channel unaware scheduling (Algorithm 3), Conservative Algorithm and *FederatedAveraging*.

all users participate every 20 rounds, so the scheduling is very certain and predictable, resulting in consistent, settled test accuracy. In the scheduling process of Algorithm 3, the integer J is determined as uniformly random, and if the channel is not available at that round, users are not re-scheduled. This reduces the deviation from the initial scheduling so that the oscillation is not as much as Algorithm 2. Algorithm 2 and *FederatedAveraging* algorithms have more oscillation because the scheduling in these algorithms is more random. Note that in *FederatedAveraging* algorithm, in every global round, users are randomly selected; and in Algorithm 2, users are scheduled according to the energy arrivals and the value of integer J, which is determined by a probability value that is dependent to a randomly available channel.

Additionally, as another method, a non-IID dataset can be synthetically produced.

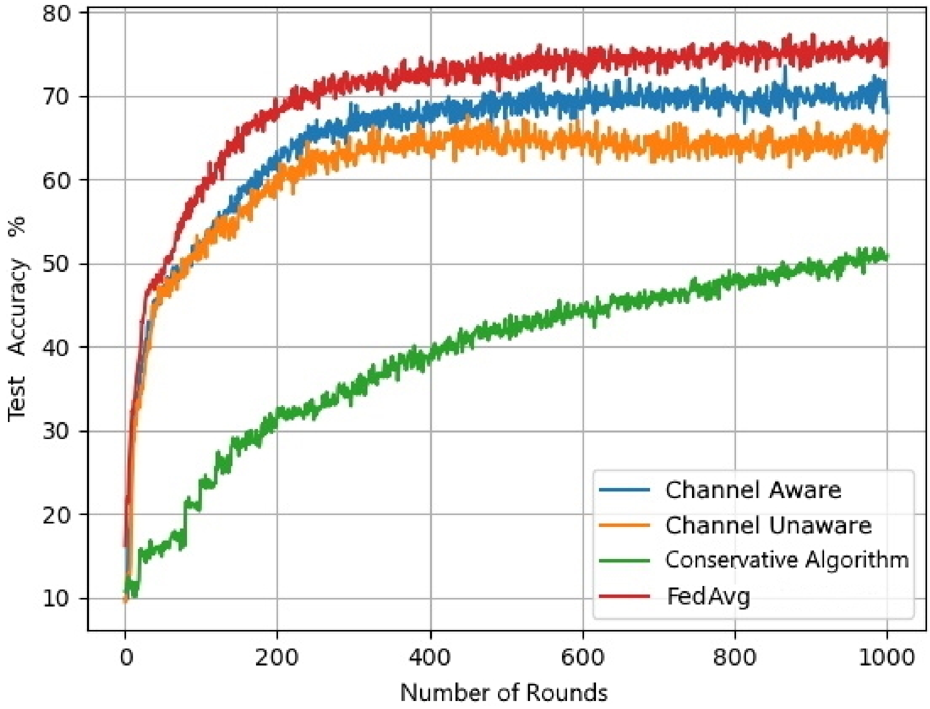

Figure 4.9: Test accuracy of channel aware scheduling (Algorithm 2) for CIFAR-10 for IID data and deterministic energy arrival according to the global rounds and comparison with channel unaware scheduling (Algorithm 3), Conservative Algorithm and *FederatedAveraging*.

To introduce, a synthetic dataset can be defined as an alternative dataset that has been created intentionally, rather than actual occurrences. Synthetic data is generated by modeling the original data statistically and utilizing those models to make new data values that replicate the statistical characteristics of the original. It is used to validate mathematical models and train machine learning models. Compared to the numerical results of the experiments with non-IID data, it can be observed that the numerical results with synthetic datasets are more settled because of the difference in the dataset characteristics. Note that CIFAR-10 is a dataset produced by real-life events, and even the *FederatedAveraging* algorithm without any communicative constraints does not have a settled test accuracy. Similar oscillation can be observed in the numerical results in [44] and [45]. To reduce the oscillation, a possible approach

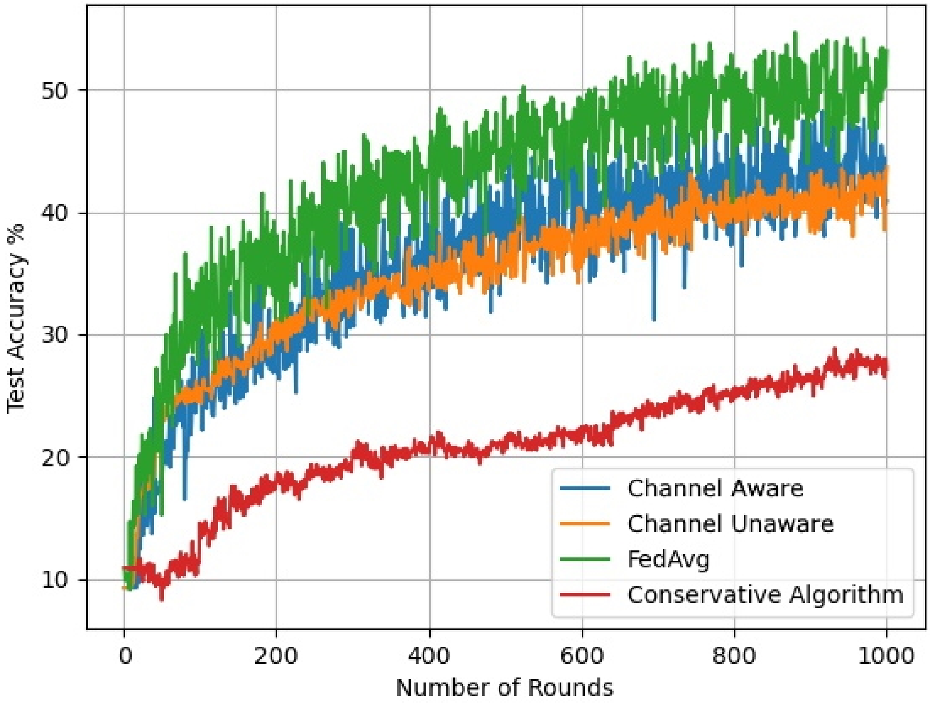

Figure 4.10: Test accuracy of channel aware scheduling (Algorithm 2) for CIFAR-10 for non-IID data according to the global rounds and comparison with channel unaware scheduling (Algorithm 3), Conservative Algorithm and *FederatedAveraging*.

can be eliminating several image classes in the training process.

For the stochastic energy arrivals, the experimental setup is the same as the previous experiments, only with the difference in the energy arrival process. In the first case, similar to deterministic energy arrivals, the training dataset is IID to the users, in the same way. Energy arrival processes are modeled as binary energy arrival processes defined in (4.7). In addition to the channel error probabilities, different energy arrival probabilities were assigned to each user group. Because of the change in energy arrival processes, the coefficient T_i^t is replaced by $\frac{1}{\beta_i}$ in all of the calculations. Note that there is no need to determine the integer J for scheduling because of the stochastic energy arrival. Figure 4.11 shows the test accuracy versus the number of rounds for these methods and their comparison with the *FederatedAveraging* algorithm. It can be seen from the figures that the performance of both proposed algorithms increased compared to the deterministic arrival case. For Algorithm 5, aiming not to waste substantial energy, the scheduling process includes checking both the channel and battery status when there is no energy arrival. Note that in Algorithm 2, when there is no energy arrival, there is no scheduling. Additionally, there is no J included in the stochastic energy arrival case, because both the scheduling process and scaling parameter provide unbiasedness among users. These factors lead to a better convergence result, because the participation of the user is much more guaranteed. On the other hand, in Algorithm 6, users are directly scheduled if there is an energy arrival, and no channel or battery status is included. Because of that, the performance of Algorithm 6 is a bit worse than Algorithm 5. In the second case, the training dataset is non-IID to the users, the same as the distribution in the experiments for deterministic energy arrival. Figure 4.12 shows the test accuracy versus the number of rounds for these methods and their comparison with the *FederatedAveraging* algorithm. Like the deterministic energy arrival case, channel aware scheduling outperforms channel unaware scheduling.

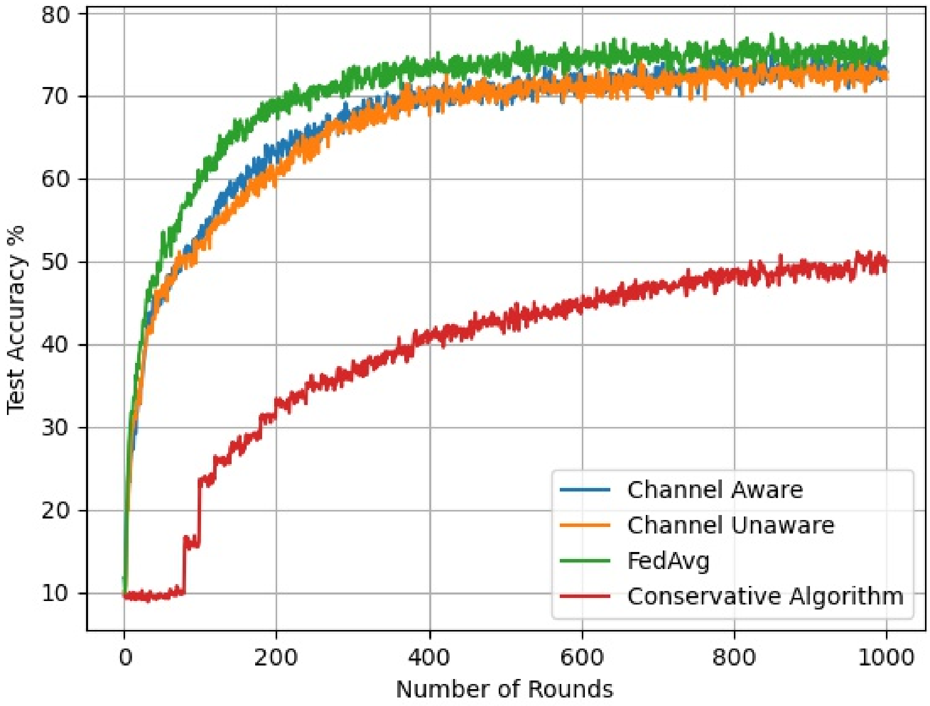

Figure 4.11: Test accuracy of channel aware scheduling (Algorithm 2) for CIFAR-10 for IID data and stochastic energy arrival according to the global rounds and comparison with channel unaware scheduling (Algorithm 3), Conservative Algorithm and *FederatedAveraging*.

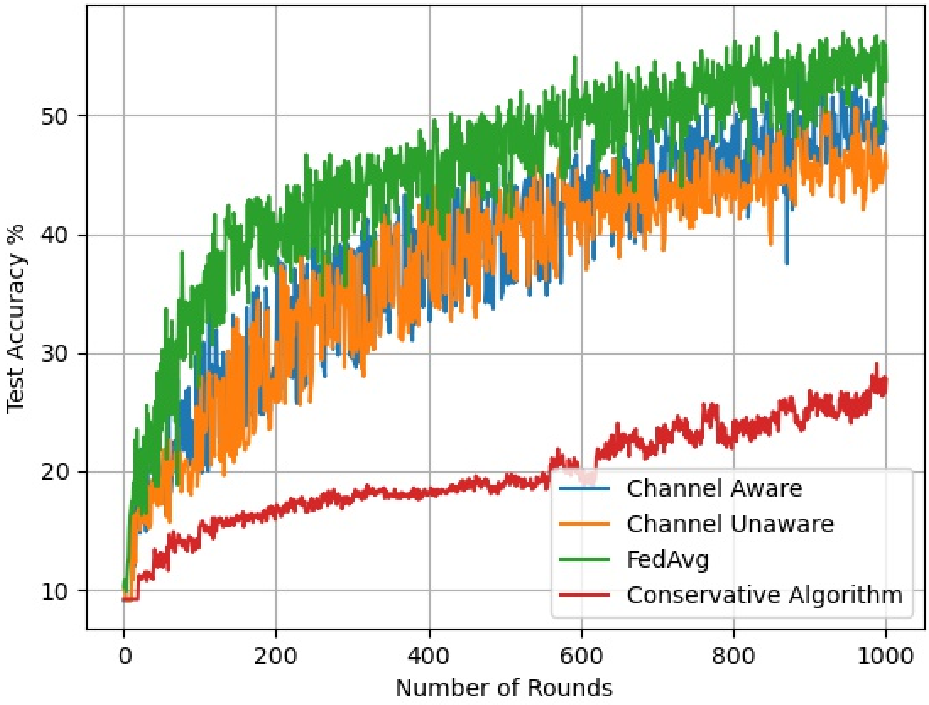

Figure 4.12: Test accuracy of channel aware scheduling (Algorithm 2) for CIFAR-10 for non-IID data and stochastic energy arrival according to the global rounds and comparison with channel unaware scheduling (Algorithm 3), Conservative Algorithm and *FederatedAveraging*.

CHAPTER 5

EFFECT OF AGE ON FEDERATED LEARNING WITH CHANNEL AND ENERGY AWARE SCHEDULING

5.1 Introduction

In this chapter, a federated learning algorithm that schedules users according to the energy and channel profiles and weighting their local gradients according to the same profiles with an extension of age (AoI) of each user is studied. Age is defined as the time elapsed between the two most recent participation in the training process. The setup is the same with Chapter 4. The main focus is to develop an age-involved algorithm that achieves a significant convergence rate in a scenario with an error-prone channel and intermittent energy availability. The experiments evaluate the performance of the proposed algorithms, and it has been seen that the proposed scheduling method provides similar test accuracy as the other methods. In Section 5.2, system model and problem definition are provided. In Section 5.3, age and momentum included scheduling methods for deterministic and stochastic energy arrivals for known channel states are presented. In Section 5.4, experimental results and evaluation of the proposed scheduling methods are provided.

5.2 System Model and Problem Definition

The system architecture is the same as in Chapter 4. Age (AoI), denoted as Δ_i^t, is defined as the time elapsed between the two most recent participation in the training process for user i. As an extension to the previous methods, age is added as a gradient update and momentum attenuation factor metric to train the model in a more

Table 5.1: Chapter 5: Parameter Symbols and Definitions

Parameter	Definition
K	Total number of users
p_i	The ratio of user i's local dataset size to the entire dataset size
Δ_i^t	The time elapsed between the two most recent participation in the training process
L	Total number of local training rounds
$w^{(t)}$	Estimation of the model parameters for round t
$\xi_i^{(t)}$	Uniformly random sample from the local dataset of user i for round t
δ_i^t	Momentum attenuation factor of participant user i
η	Learning rate
T	Total number of global rounds
S_t	The set of users who have successfully participated

balanced and accurate way. As it is defined in Chapter 2, momentum is a variant of gradient descent designed for performance improvement and optimization. Momentum attenuation factor of participant user i is denoted as δ_i^t. It is important to note that momentum improves the model's accuracy for non-IID data, and decreases the convergence time for IID data. This difference originated from the diversity of data of a participant user. To show the effect of age on model update and momentum more precisely, the work in this chapter is evaluated by both IID and non-IID datasets.

The global loss function is defined in (4.1), and the local loss function is defined in (4.2). Assuming that a user $i \in \{1, \ldots, K\}$ has D_i data points in its local dataset, the total number of data points for all users is denoted as D. p_i is defined as the ratio of the user i's local dataset size to the entire dataset size ($p_i = \frac{D_i}{D}, \sum_{i=1}^{K} p_i = 1$). The value $t \in 0, 1, 2, ..$ represents each global round, and L defines the number of local training iterations. K is the total number of users. Training is performed by using the distributed SGD method. Estimation of the model parameters for the round t is represented by $w^{(t)}$. In the distributed SGD method, the parameter server sends the value $w^{(t)}$ to participating users. Users $i \in 1, 2, ..., K$ calculate a local stochastic gradient with L local iterations as defined in (4.3), and the actual gradient value of user i is defined in (4.4). The gradient of the global loss function is defined in (4.5). After users complete their local calculations, local gradient values are sent to the parameter server. The parameter server updates the model as in (4.6). After the update, the model is sent back to participating users, and the cycle continues until the process is complete.

5.3 Proposed Methods

As an extension of the previous chapter, the main motivation for this chapter is that if the user cannot participate in the training process because of its energy arrival or channel availability processes, that user's previous data must be taken into account in a grander scale. In order to achieve that, the momentum attenuation factor must be set at a higher value. On the other hand, if the user participates in the training process more frequently, the momentum attenuation factor must be determined as a smaller value. The same motivation holds for the gradient scaling: if the time elapsed between the two most recent participation of the corresponding user is long, the scaling coefficient must be greater than the other users; since that user cannot participate in the training process frequently, so that the information obtained from that user must be important. To ensure this assumption, while scaling the local gradients, the age of the user is taken into account, which is going to be explained in the following.

When the channel state is known and if the user has enough energy, an integer J with a certain probability in the range of 0 to $T_i^t - 1$ is determined with the help of the following two different probability values, defined as in (4.9) and (4.10). After the integer J is determined, the user is scheduled by adding J to the current global round value $(t + J)$. The momentum attenuation factor is also determined in the scheduling process according to the age of the corresponding user:

$$\delta_i^t = \begin{cases} 0.1, & \text{if } \Delta_i^t = 1 \\ 0.5, & \text{if } 1 < \Delta_i^t \leq T_i^t \\ 0.9, & \text{if } \Delta_i^t > T_i^t \end{cases} \tag{5.1}$$

where T_i^t is the energy arrival parameter. Once the scheduling is complete, the learning process can begin. In each global round, it is checked whether users can participate in the learning process by the channel status. The user can participate in the training process if the channel is available at the current global round. If not, the user is scheduled to participate in the next global round. The parameter server sends the model parameters to the participating users, and the users perform the local training.

The information gathered from the user that participates in the training process less than the other users is more important and must be included in the process on a greater scale. With this information, the scaling coefficient is defined as the ratio of the age of the corresponding user to the total age of the users, $\frac{\Delta_i^t}{\Delta^t}$. During the local training, local gradients are scaled as follows:

$$g_i^{(t+L)} = \frac{\Delta_i^t}{\Delta^t}(g_i(w^{(t)}, \xi_i^t)) \tag{5.2}$$

If the time elapsed between the two most recent participation of the corresponding user is greater than the energy arrival parameter, the momentum attenuation factor must be greater; because the information gathered from that user becomes more important due to the lack of participation in the process. The momentum term can be obtained as in the following:

$$m_i(t+1) = \delta_i^t m(t) - \eta * g_i^{(t+L)} \tag{5.3}$$

The participant user obtains the locally trained model parameters as follows:

$$w^{(t+L)} = w^{(t)} + m_i(t+1) \tag{5.4}$$

After the local training, participant users send their locally trained model parameters to the server. The parameter server updates the global model and sends the global model back to the users until the global training is complete. Scheduling and training steps for deterministic energy arrivals can be seen in Algorithm 6, and for stochastic energy arrivals can be seen in Algorithm 7. For both Algorithm 6 and Algorithm 7, in a worst case scenario where all of the users participate in the training, the time complexity is $\mathcal{O}(T * (KL + K)) = \mathcal{O}(TKL)$, where T is the total number of global rounds, K is the total number of users and L is the total number of local training rounds.

5.4 Performance Evaluation

Similar to the experiments in Chapter 4, experiments were performed as an image classification task with 10 classes of 40 users, for 1000 global rounds and 5 local training rounds, using both the CIFAR-10 dataset [42], distributed as 50,000 training samples and 10,000 test samples; and MNIST dataset [43], distributed as 60,000

Algorithm 6 Age-Involved Federated Learning with Momentum for Deterministic Energy Arrivals and Known Channel Status

```
Require: Total number of global rounds $T$, number of users $K$, channel status for
    user $i$ $Q_i$, channel error probability of user $i$ $q_i$, initialized model parameters $w^{(0)}$
Ensure: Trained model parameters $w^{(T)}$
  Initialize $K_i^t = 0$ for t $\in$ [T]
  for Global round $t = 0, ..., T-1$ do
    for User $i$ in $K$ do
      if $E_i^t$ =1 then
        Determine $J$ using (4.9) and (4.10)
        Schedule $K_i^{t+J}$ =1
        Determine the momentum attenuation factor using (5.1)
      end if
      if $K_i^t$ =1 then
        if $Q_i^t$ =1 then
          for Local iteration $m$ in $L$ do
            Calculate and scale the local gradients $g_i(w^{(t)}, \xi_i^{(t)})$
          end for
          Send the locally trained model parameters to the parameter server
          $\Delta_i^{t+1} = 1$
        else if $Q_i^t = 0$ then
          Schedule $K_i^{t+J} = 0$ and $K_i^{t+J+1} = 1$
          $\Delta_i^{t+1} = \Delta_i^t + 1$
        end if
      end if
    end for
    Parameter Server:
    Update the global model
    Send model parameters $w^{(t+1)}$ to the users
  end for
```

Algorithm 7 Age-Involved Federated Learning with Momentum for Stochastic Energy Arrivals and Known Channel Status

```
Require: Total number of global rounds $T$, number of users $K$, channel status for
    user $i$ $Q_i$, channel error probability of user $i$ $q_i$, initialized model parameters $w^{(0)}$
Ensure: Trained model parameters $w^{(T)}$
  Initialize $K_i^t = 0$ and $B_i^t = 0$ for t ∈ [T]
  for Global round $t = 0, ..., T-1$ do
    for User $i$ in $K$ do
      if $E_i^t$ =1 then
        Schedule $K_i^t$ =1
        Determine the momentum attenuation factor using (5.1)
        Battery level $B_i^t = 1$
      else
        if $B_i^t$ =1 then
          if $Q_i^t$ =1 then
            Schedule $K_i^t$ =1
            Determine the momentum attenuation factor using (5.1)
          else
            Schedule $K_i^{t+1}$ =1 and $K_i^t$ =0
          end if
        end if
      end if
      if $K_i^t$ =1 then
        if $Q_i^t$ =1 then
          for Local iteration $m$ in $L$ do
            Calculate and scale the local gradients $g_i(w^{(t)}, \xi_i^{(t)})$
          end for
          Send the locally trained model parameters to the parameter server
          $\Delta_i^{t+1} = 1$
        else if $Q_i^t = 0$ then
          Schedule $K_i^t = 0$ and $K_i^{t+1} = 1$
          $\Delta_i^{t+1} = \Delta_i^t + 1$
          Battery level $B_i^{t+1} = 1$
        end if
      end if
    end for
    Parameter Server:
    Update the global model
    Send model parameters $w^{(t+1)}$ to the users
  end for
```

samples for training and 10,000 samples for testing, both of the datasets with the batch size as 64. In the experiments, Algorithm 6 and Algorithm 7 are simulated for both CIFAR-10 and MNIST datasets for the case of deterministic and stochastic energy arrivals. For the CIFAR-10 dataset, images were preprocessed before the training. SGD optimizer is used for both of the datasets, and the learning rate was set as 0.01. As the architecture, CNN is used with the same layer structure. The performance evaluation is conducted by calculating the test accuracy for each image class and taking the average. Lastly, to show the effect of momentum more clearly, both IID and non-IID training datasets are used in the simulations. Like in the previous simulation setup, IID is applied by shuffling the dataset and splitting it between the clients. Non-IID is applied by sorting the dataset by the digit label, dividing the data into 200 shards of size 250, and assigning two shards to each client. To evaluate the performance of the proposed algorithm, both *FedAvg* and Conservative Algorithm are also simulated for both cases.

Figure 5.1 shows the convergence of Algorithm 6 for deterministic energy arrivals, comparing with *FedAvg*, for non-IID MNIST and CIFAR-10 datasets. For the MNIST dataset, Algorithm 6 reaches approximately 88% of accuracy, which is slightly better than the accuracy of the channel aware algorithm, Algorithm 2, without age. Additionally, for CIFAR-10, accuracy reaches approximately 50%, which is greater than the maximum test accuracy of Algorithm 2, without age. Figure 5.2 shows the convergence of Algorithm 6 for deterministic energy arrivals for IID MNIST and CIFAR datasets. Algorithm 6 reaches 100% of accuracy, which is the same as the accuracy with Algorithm 2, without age. Similarly, for CIFAR-10, accuracy reaches approximately 70%, which is similar to the test accuracy of Algorithm 2, without age.

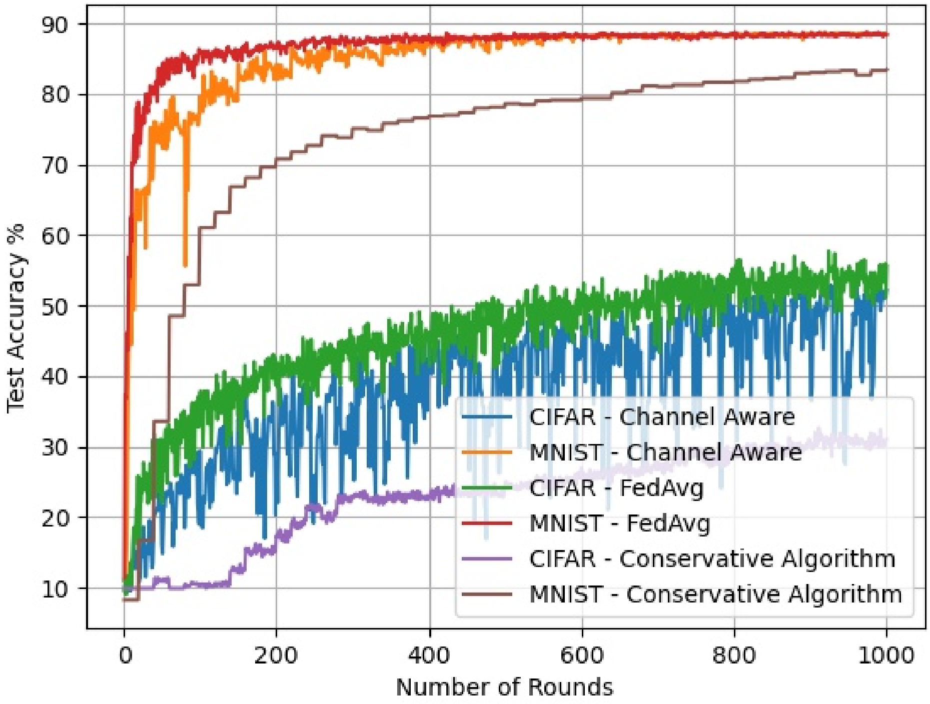

Figure 5.1: Test accuracy of Algorithm 6 for MNIST and CIFAR-10 datasets, for non-IID data and deterministic energy arrival. Note that channel status is known in this scenario.

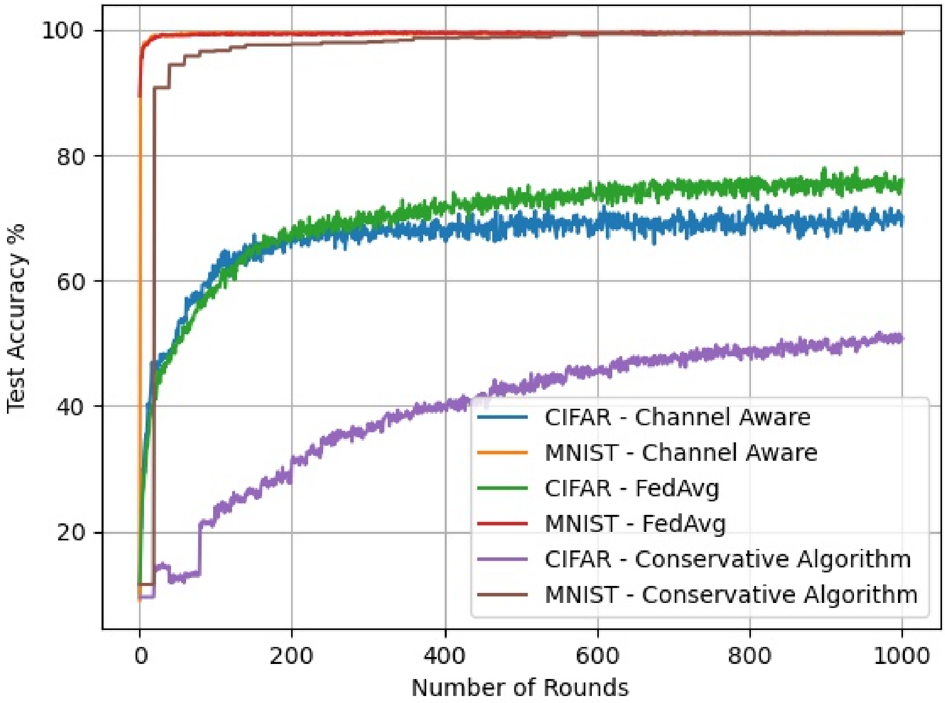

Figure 5.2: Test accuracy of Algorithm 6 for MNIST and CIFAR-10 datasets, for IID data and deterministic energy arrival. Note that channel status is known in this scenario.

In Table 5.2 and Table 5.3, the maximum age values for deterministic energy arrivals and CIFAR-10 and MNIST datasets respectively are provided. Similar to the experimental setup in the previous chapter, users were divided into four equal groups and different energy arrival parameters and channel availability probabilities were assigned to each group. How the age changes is dependent to both the channel and energy status of the user, because it is directly obtained by the scheduling process.

Table 5.2: Chapter 5: Maximum Age Statistics for CIFAR Dataset and Deterministic Energy Arrivals

User Groups	IID Dataset	Non-IID Dataset
First User Group	6	7
Second User Group	14	18
Third User Group	177	205
Fourth User Group	160	120

Table 5.3: Chapter 5: Maximum Age Statistics for MNIST Dataset and Deterministic Energy Arrivals

User Groups	IID Dataset	Non-IID Dataset
First User Group	6	7
Second User Group	13	18
Third User Group	178	210
Fourth User Group	159	120

Figure 5.4 shows the convergence of Algorithm 6 for stochastic energy arrivals, comparing with *FedAvg*, for non-IID MNIST and CIFAR-10 datasets. For the MNIST dataset, Algorithm 6 reaches approximately 88% of accuracy, which is slightly better than the accuracy of the channel aware algorithm, Algorithm 2, without age. Additionally, for CIFAR-10, accuracy reaches approximately 53%, which is greater than the maximum test accuracy of Algorithm 2 for the stochastic energy arrivals, without age. Figure 5.3 shows the convergence of Algorithm 6 for stochastic energy arrivals for IID MNIST and CIFAR datasets. Algorithm 6 reaches 100% of accuracy, which is the same as the accuracy with Algorithm 2, without age. For CIFAR-10, accuracy reaches approximately 73%, which is very close to the accuracy of *FedAvg*.

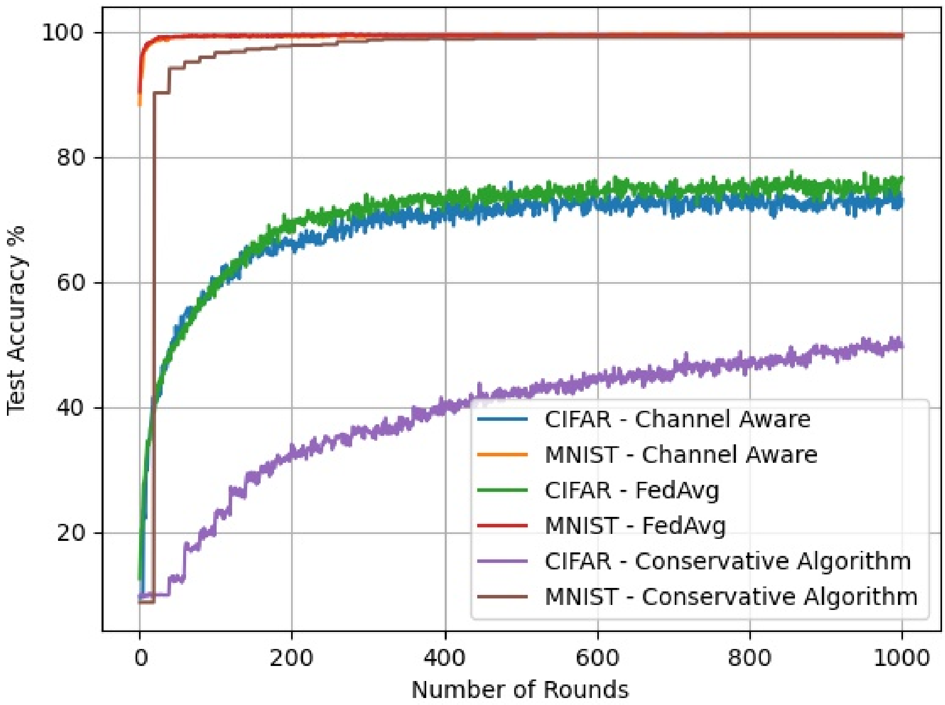

Figure 5.3: Test accuracy of Algorithm 6 for MNIST and CIFAR-10 datasets, for IID data and stochastic energy arrival. Note that channel status is known in this scenario.

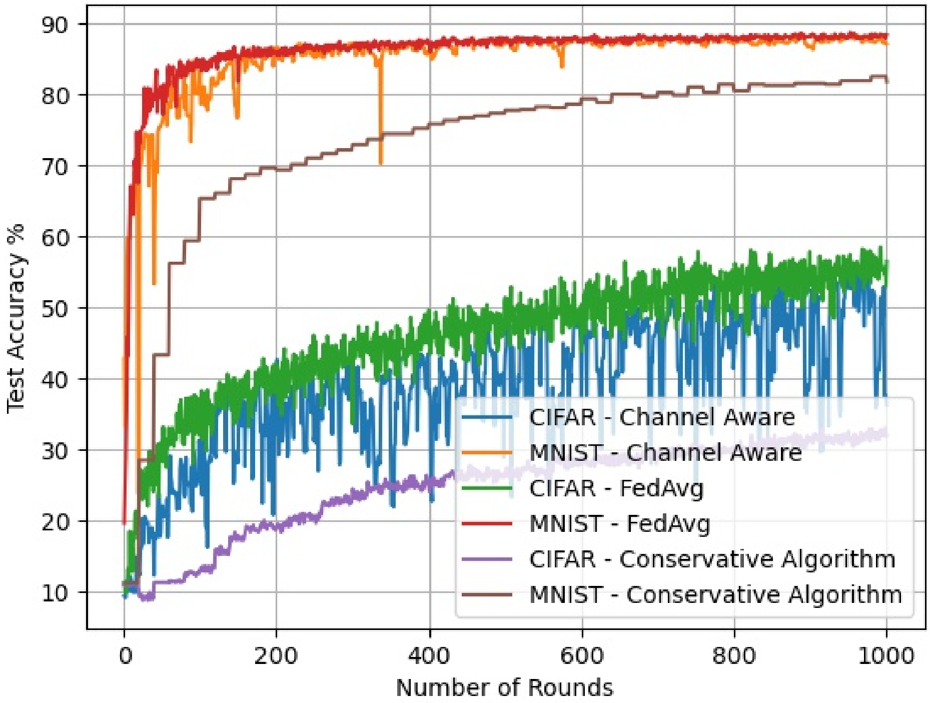

Figure 5.4: Test accuracy of Algorithm 6 for MNIST and CIFAR-10 datasets, for non-IID data and stochastic energy arrival. Note that channel status is known in this scenario.

It can be observed from the experimental results that adding age as a gradient scaling factor and a metric for momentum attenuation factor has a positive effect on both the test accuracy and convergence rate for non-IID datasets. Additionally, for IID datasets, the model converged significantly faster. Experimental results for Algorithm 2 in the previous chapter showed that the model started to converge around 500th global round. In contrast, in this chapter, with the help of age-based momentum, the model started to converge around 200th global round. As a result, these experimental results verify the claim that AoI-aware momentum improves the model's accuracy for non-IID data and decreases the convergence time for IID data.

CHAPTER 6

CONCLUSIONS

This thesis focused on developing federated learning algorithms that are extended by constraints such as channel availability, energy harvesting, and data freshness, and provide the same guarantee of convergence with the algorithms that have no constraints. In Chapter 3, an optimal threshold-based decision policy that aims to provide the lowest long-term average AoI is studied. A point-to-point channel with a receiver-centric scheduling policy, with channel states changing as ON and OFF in an IID fashion, is considered. It is assumed that unless the AoI is greater than a specific threshold value, the transmission does not occur. The performance of the proposed policy is compared with the uniform transmission policy, and it is observed that the proposed policy is significantly more efficient than the uniform transmission policy. In Chapter 4, the study on optimizing the federated learning process according to the intermittent energy harvesting and channel state without violating the theoretical convergence guarantees by proposing a scheduling algorithm sensitive to channel and energy state for a federated learning system prone to energy harvesting and channel errors is presented. As a result of this work, it has been seen that the proposed scheduling method provides higher test accuracy and lower loss than other methods. In Chapter 5, an extension of the work in Chapter 4 by studying the effect of AoI with momentum for the proposed channel and energy-aware scheduling methods is presented. As a result, it is shown that with AoI-aware momentum, the accuracy of the model for non-IID data increases, and the convergence time for IID data decreases.

For future work, there are many research areas focused on improving the efficiency of federated learning.

- **Federated Learning with Finite Battery:** In the proposed methods for deterministic energy arrival, for the sake of simplicity, the battery status of both the users and the parameter server were not included. It was only included in the stochastic energy arrival case because of the randomness of the energy arrivals, and it was aimed that the existing energy will not be wasted. The same logic can be adapted into the finite battery case: in the scheduling process, the battery status of each user can be another constraint, just as energy and channel. On the other hand, the parameter server may have finite battery and may not prefer to perform the model updating process, but this will lead to a slow convergence of the model. This tradeoff might be an interesting research problem to work on in the future.

- **Adaptive Model/Network Pruning:** After the model is initially trained using SGD for a fixed number of iterations, a specified proportion (referred to as the pruning rate) of weights with the least absolute values can be eliminated, which is called network pruning. This cycle can be repeated until the required model size is achieved. The advantage of this strategy is that training and pruning occur concurrently, resulting in a trained model with the desired size. However, current pruning strategies need the availability of training data in a central location, which is against the FL's principles [46]. Adapting it into the work in this thesis, the least absolute values may be produced from the users with less participation because of the several constraints referred to in this thesis. Model pruning can be adapted to these constraints to achieve more accurate model parameters. The research area would be determining the pruning rate for a network that includes users with different channel, energy, and age profiles.

- **Federated Edge Learning (FEEL):** Supported by a remote parameter server, federated edge learning is performed by wireless devices, with constrained energy and bandwidth, on their local datasets. Adapting it into the work in this thesis, FEEL can be extended by adding an AoI parameter as a constraint.

- **Federated Learning with Finite Blocklength:** When users participate in the process with the same channel, there may be interference among users. To avoid interference and lessen the burden on the channel, the packet including the locally trained model parameters can be compressed at a rate. There is a tradeoff for determining the compression rate: If less error in the transmission is desired, the packets will be compressed on a smaller scale, so that the packet size will be bigger and it will take time to transmit it, which will cause higher AoI. On the other hand, if the amount of error in the transmission is not prioritized, the packet will be compressed on a grander scale, so that it will take less time to transmit, which will lead to a smaller AoI value. Considering this problem, an interesting research problem would be determining a compression and sparsification method for federated learning.

- **Federated Learning with Incentives:** In the context of this thesis, users with less energy arrival or unavailable channel would not prefer to participate in the process not to waste their resources. Such users can be encouraged to participate in the process by offering an incentive by the parameter server. As an example, this incentive can be providing energy to the user. As a result of this application, more participation of users and more accurate models can be achieved more quickly, especially for non-IID datasets.

REFERENCES

[1] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[2] O. T. Yavascan, E. T. Ceran, Z. Cakir, E. Uysal, and O. Kaya, "When to pull data for minimum age penalty," in *2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, pp. 1–8, 2021.

[3] J. Brownlee, "Understand the Impact of Learning Rate on Neural Network Performance," 01 2019.

[4] J. Konečnỳ, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575*, 2015.

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

[6] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[7] B. Güler and A. Yener, "Energy-harvesting distributed machine learning," in *2021 IEEE Int. Symp. on Information Theory (ISIT)*, pp. 320–325, IEEE, 2021.

[8] N. Shinohara, "Development of rectenna with wireless communication system," in *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*, pp. 3970–3973, April 2011.

[9] Z. Popovic, "Cut the cord: Low-power far-field wireless powering," in *IEEE Microwave Magazine*, pp. 55–62, March/April 2013.

[10] R. J. Vyas, B. B. Cook, Y. Kawahara, and M. M. Tentzeris, "E-wehp: A batteryless embedded sensor platform wirelessly powered from ambient digital-tv

signal," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, pp. 2491–2505, June 2013.

[11] B. Guler and A. Yener, "Sustainable federated learning," *arXiv preprint arXiv:2102.11274*, 2021.

[12] D. Gündüz, D. B. Kurka, M. Jankowski, M. M. Amiri, E. Ozfatura, and S. Sreekumar, "Communicate to learn at the edge," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 14–19, 2020.

[13] E. Ozfatura, D. Gunduz, and H. V. Poor, "Collaborative learning over wireless networks: An introductory overview," *arXiv preprint arXiv:2112.05559*, 2021.

[14] H. Zhang and L. Hanzo, "Federated learning assisted multi-uav networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 14104–14109, 2020.

[15] B. Brik, A. Ksentini, and M. Bouaziz, "Federated learning for uavs-enabled wireless networks: Use cases, challenges, and open problems," *IEEE Access*, vol. 8, pp. 53841–53849, 2020.

[16] Q.-V. Pham, M. Zeng, R. Ruby, T. Huynh-The, and W.-J. Hwang, "Uav communications for sustainable federated learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3944–3948, 2021.

[17] W. Y. B. Lim, Z. Xiong, J. Kang, D. Niyato, C. Leung, C. Miao, and X. Shen, "When information freshness meets service latency in federated learning: A task-aware incentive scheme for smart industries," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 457–466, 2020.

[18] O. Aygün, M. Kazemi, D. Gündüz, and T. M. Duman, "Over-the-air federated learning with energy harvesting devices," *arXiv preprint arXiv:2205.12869*, 2022.

[19] O. Aygün, M. Kazemi, D. Gündüz, and T. M. Duman, "Hierarchical over-the-air federated edge learning," *arXiv preprint arXiv:2112.11167*, 2021.

[20] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of federated learning over a noisy downlink," *IEEE Transactions on Wireless Communications*, 2021.

[21] J. Brownlee, “Gradient Descent With Momentum from Scratch,” 10 2021.

[22] J. Xu, S. Wang, L. Wang, and A. C.-C. Yao, “Fedcm: Federated learning with client-level momentum,” *arXiv preprint arXiv:2106.10874*, 2021.

[23] G. Kim, J. Kim, and B. Han, “Communication-efficient federated learning with acceleration of global momentum,” *arXiv preprint arXiv:2201.03172*, 2022.

[24] W. Liu, L. Chen, Y. Chen, and W. Zhang, “Accelerating federated learning via momentum gradient descent,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 8, pp. 1754–1766, 2020.

[25] Z. Huo, Q. Yang, B. Gu, L. C. Huang, *et al.*, “Faster on-device training using new federated momentum algorithm,” *arXiv preprint arXiv:2002.02090*, 2020.

[26] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, “Minimizing age of information in vehicular networks,” in *Sensor, Mesh and Ad-Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference*, pp. 350–358, June 2011.

[27] S. Kaul, R. D. Yates, and M. Gruteser, “Real-time status: How often should one update?,” in *INFOCOM 2012*, p. 2731–2735, 2012.

[28] E. Uysal-Biyikoglu, B. Prabhakar, and A. E. Gamal, “Energy-efficient packet transmission over a wireless link,” in *IEEE Trans. on Networking*, vol. 10, pp. 487 – 499, August 2002.

[29] E. Uysal-Biyikoglu and A. E. Gamal, “Energy-efficient packet transmission over a multiaccess channel,” in *Proceedings IEEE International Symposium on Information Theory*, p. 153, 2002.

[30] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, “Age of information under energy replenishment constraints,” in *2015 Information Theory and Applications Workshop (ITA)*, pp. 25–31, Feb 2015.

[31] R. S. I. Kadota, E. Uysal-Biyikoglu and E. Modiano, “Minimizing the age of information in broadcast wireless networks,” in *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 844 – 851, 2016.

[32] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," in *The 35th Annual IEEE International Conference on Computer Communications*, pp. 1 – 9, 2016.

[33] E. T. Ceran, D. Gündüz, and A. György, "Average age of information with hybrid ARQ under a resource constraint," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2018.

[34] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 3008–3012, 2015.

[35] B. T. Bacinoglu, O. Kaya, and E. Uysal-Biyikoglu, "Energy efficient transmission scheduling for channel-adaptive wireless energy transfer," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, April 2018.

[36] B. T. Bacinoglu, Y. Sun, E. Uysal, and V. Mutlu, "Optimal status updating with a finite-battery energy harvesting source," *Journal of Communications and Networks*, vol. 21, no. 3, pp. 280–294, 2019.

[37] H. H. Yang, A. Arafa, T. Q. Quek, and H. V. Poor, "Age-based scheduling policy for federated learning in mobile edge networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8743–8747, IEEE, 2020.

[38] B. Buyukates and S. Ulukus, "Timely communication in federated learning," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, IEEE, 2021.

[39] X. Liu, X. Qin, H. Chen, Y. Liu, B. Liu, and P. Zhang, "Age-aware communication strategy in federated learning with energy harvesting devices," in *2021 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 358–363, IEEE, 2021.

[40] E. Altman, *Constrained Markov decision processes: stochastic modeling*. Routledge, 1999.

[41] L. I. Sennott, “Constrained average cost markov decision chains,” *Probability in the Engineering and Informational Sciences*, vol. 7, no. 1, pp. 69–83, 1993.

[42] A. Krizhevsky, G. Hinton, *et al.*, “Learning multiple layers of features from tiny images,” 2009.

[43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[44] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated learning with non-iid data,” *arXiv preprint arXiv:1806.00582*, 2018.

[45] H. Wang, Z. Kaplan, D. Niu, and B. Li, “Optimizing federated learning on non-iid data with reinforcement learning,” in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 1698–1707, IEEE, 2020.

[46] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, “Model pruning enables efficient federated learning on edge devices,” *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

APPENDIX A

DERIVATION OF THE PROBABILITY OF THE SCHEDULING PARAMETER

Let $\alpha_t = P_J(j)$ and the channel error probability of user i is q_i. To ensure fairness among all participant users, it is assumed that $\alpha_0 = \alpha_1 = \alpha_2 = ... = \alpha_{T_i-1}$. The probabilities of different J values can be defined as in the following:

$$
\begin{aligned}
\alpha_t(0) &= (1-q_i)P_J(0) \\
\alpha_t(1) &= (1-q_i)P_J(1) + q_i(1-q_i)P_J(0) \\
\alpha_t(2) &= (1-q_i)P_J(2) + q_i(1-q_i)P_J(1) + q_i^2(1-q_i)P_J(0) \\
&\;\;... \\
\alpha_t(T_i-1) &= (1-q_i)P_J(T_i-2) + q_i(1-q_i)P_J(T_i-3) \\
&\quad + ... + q_i^{(T_i-2)}(1-q_i)P_J(0)
\end{aligned}
$$

Because of the assumption:

$$
\begin{aligned}
(1-q_i)P_J(0) &= (1-q_i)P_J(1) + q_i(1-q_i)P_J(0) \\
&\hookrightarrow (1-q_i)P_J(0) = P_J(1)
\end{aligned}
$$

$$
\begin{aligned}
(1-q_i)P_J(1) + q_i(1-q_i)P_J(0) &= (1-q_i)P_J(2) + q_i(1-q_i)P_J(1) + q_i^2(1-q_i)P_J(0) \\
(1-q_i)P_J(1) + q_i(1-q_i)P_J(0) &= P_J(2) \\
&\hookrightarrow P_J(2) = P_J(1)
\end{aligned}
$$

$$
\begin{aligned}
P_J(3) &= (1-q_i)P_J(2) + q_i(1-q_i)P_J(1) + q_i^2(1-q_i)P_J(0) \\
&= P_J(2) = P_J(1) \\
&\hookrightarrow P_J(T_i-1) = ... = P_J(3) = P_J(2) = P_J(1)
\end{aligned}
$$

It is known that $\sum_{0}^{T_i-1} P_J(j) = 1$. This leads to:

$$\begin{aligned}
&P_J(0) + (T_i - 1)P_J(1) = 1 \\
&P_J(0)(1 + (T_i - 1)(1 - q_i)) = 1 \\
&P_J(0) = \frac{1}{T_i - T_i q_i + q_i} \\
&P_J(T_i - 1) = ... = P_J(3) = P_J(2) = P_J(1) = \frac{1 - q_i}{T_i - T_i q_i + q_i}
\end{aligned}$$

This completes the derivation.