

Article

Low-Latency Collaborative Predictive Maintenance: Over-the-Air Federated Learning in Noisy Industrial Environments

Ali Bemani ^{*,†}  and Niclas Björsell [†] 

Department of Electrical Engineering, Mathematics and Science, University of Gävle, 801 76 Gävle, Sweden; niclas.bjorsell@hig.se

* Correspondence: ali.bemani@hig.se

† These authors contributed equally to this work.

Abstract: The emergence of Industry 4.0 has revolutionized the industrial sector, enabling the development of compact, precise, and interconnected assets. This transformation has not only generated vast amounts of data but also facilitated the migration of learning and optimization processes to edge devices. Consequently, modern industries can effectively leverage this paradigm through distributed learning to define product quality and implement predictive maintenance (PM) strategies. While computing speeds continue to advance rapidly, the latency in communication has emerged as a bottleneck for fast edge learning, particularly in time-sensitive applications such as PM. To address this issue, we explore Federated Learning (FL), a privacy-preserving framework. FL entails updating a global AI model on a parameter server (PS) through aggregation of locally trained models from edge devices. We propose an innovative approach: analog aggregation over-the-air of updates transmitted concurrently over wireless channels. This leverages the waveform-superposition property in multi-access channels, significantly reducing communication latency compared to conventional methods. However, it is vulnerable to performance degradation due to channel properties like noise and fading. In this study, we introduce a method to mitigate the impact of channel noise in FL over-the-air communication and computation (FLOACC). We integrate a novel tracking-based stochastic approximation scheme into a standard federated stochastic variance reduced gradient (FSVRG). This effectively averages out channel noise's influence, ensuring robust FLOACC performance without increasing transmission power gain. Numerical results confirm our approach's superior communication efficiency and scalability in various FL scenarios, especially when dealing with noisy channels. Simulation experiments also highlight significant enhancements in prediction accuracy and loss function reduction for analog aggregation in over-the-air FL scenarios.

Keywords: predictive maintenance; over-the-air federated learning; analog aggregation; low latency; channel noise



Citation: Bemani, A.; Björsell, N. Low-Latency Collaborative Predictive Maintenance: Over-the-Air Federated Learning in Noisy Industrial Environments. *Sensors* **2023**, *23*, 7840. <https://doi.org/10.3390/s23187840>

Academic Editors: Yichuang Sun, Haeyoung Lee and Oluyomi Simpson

Received: 20 August 2023

Revised: 7 September 2023

Accepted: 8 September 2023

Published: 12 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The pervasive connectivity of numerous devices and sensors, fueled by recent advancements in communication networks and Internet of Things (IoT) applications, has resulted in the generation of immense volumes of data. The copious volume of data at hand serves as the training datasets for machine learning (ML) algorithms that find utility across various applications in industry, such as process optimization, defining product quality, and PM, critical components within the purview of Industry 4.0 [1]. Conventionally, the training process for these ML algorithms has followed a centralized approach, wherein multiple devices transmit their raw and occasionally sensitive data to a PS possessing robust computational resources dedicated to training tasks. Nevertheless, the data required for training these ML algorithms are produced by numerous assets and devices in the industry, which often necessitate privacy-preserving measures to safeguard data integrity. Furthermore, the presence of bandwidth constraints poses a bottleneck when transmitting

a massive amount of data from the devices to the PS [2]. For this reason, there has been considerable focus on federated and distributed learning algorithms, primarily due to their ability to train ML models in a fully decentralized manner.

FL algorithms have been proposed as an alternative scheme for privacy-preserving distributed ML, where each device participates in the training process using exclusively locally available data facilitated by a PS [3]. In the context of FL, devices communicate with the PS by exchanging model parameters and their respective local updates, while ensuring that the raw data remain localized on the devices. This approach offers not only privacy advantages but also proves to be a compelling solution for wireless edge devices, particularly when dealing with substantial dataset sizes. In the domain of PM applications, the volume of data obtained from online sensors is significant. Consequently, it becomes imperative to account for this substantial data volume when implementing federated learning techniques for PM. Additionally, time sensitivity and temporal awareness are essential attributes for the effective execution of PM activities. Therefore, time latency should be considered for PM applications.

The migration learning from centralized clouds to the edge enables edge servers to quickly obtain real-time data generated by edge devices, facilitating rapid AI model training. As a result, distributing these models from servers to devices in close proximity enables the devices to respond effectively to real-time events, making them well-suited for PM applications. Despite the rapid advancement of computing speeds, wireless transmission of large amounts of data by any device faces limitations due to limited radio resources and the adverse conditions of wireless channels. This creates a communication bottleneck that hinders fast-edge learning [4,5]. Hence, communication efficiency has been at the forefront of FL, and the paramount objective is to achieve high model accuracy while minimizing the number of communication rounds of resource usage. In the realm of cutting-edge research, the fusion of digital twins and multi-access edge computing has gained significant attention [6]. This innovative approach represents a crucial technology in the context of 6G networks, primarily serving as a fundamental enabler for the Industrial IoT. The primary objective of this research is to minimize the total task completion delay for IoT devices by optimizing various parameters.

Communication schemes for FL can be categorized as digital or analog. Digital communication, though burdensome for wireless networks, assigns communication resources to each client's ML model parameters. Analog communication reduces overhead by allowing shared resources for transmitting FL models. Early research aimed to reduce communication rounds or payload size. However, most FL literature assumes a perfect communication channel shifting focus to ML design. Recent research addresses this gap by emphasizing system design, especially for wireless FL [7–9]. Although early studies have delved deeply into optimizing communication design for FL, the crucial and practical matter of a noisy channel with over-the-air communication and computation (OACC) has not been thoroughly investigated. Incorporating the impact of noisy channels in OACC complicates convergence analysis due to noise propagation during each communication round. Moreover, the collective impact of these noisy communications in OACC on the final learning performance necessitates a comprehensive design and analysis approach.

This work addresses the impact of communication-induced noise during FL training on the convergence and accuracy performance of the ML model, and then proposes robust algorithms within the OACC framework that mitigate these effects and optimize client resources concurrently. Our focus lies in analog communication for model updates [10–12] and the exploration of new distributed algorithms capable of withstanding high levels of channel noise and low signal-to-noise ratio (SNR) in industrial environments, in accordance with PM applications.

1.1. Related Work

In recent years, substantial endeavors have been undertaken to implement the FL framework over wireless networks, with its origins traced back to McMahan's seminal

works [13,14]. A comprehensive survey of FL has been conducted in [15], offering an overview of this paradigm where statistical models are trained on distributed networks at the edge. These studies explore the distinctive challenges and characteristics of FL compared to traditional approaches while identifying open problems that necessitate interdisciplinary research efforts. However, these works overlook the practical implications of channel effects and assume a seamless integration of FL algorithms into wireless networks. Recently, there has been significant attention given to exploring methods for mitigating the impact of these effects on FL algorithms [16–18].

This study primarily focuses on analog aggregation schemes for over-the-air transmission, which are driven by the inherent superposition property of signals in the wireless multiple-access channel. Analog over-the-air aggregation is a highly promising technique extensively used in Federated Averaging (FedAvg) due to the fact that the ps or clients require only the sum of their local gradients or model parameters. These schemes have been explored in various studies, including [2,19], and other relevant works.

In order to provide context and underscore the contributions of this article, we will now discuss previous studies that have investigated FL in the presence of imperfect or noisy communication. The main objective of [20] was to tackle the issue of noise in wireless communications for federated learning. The paper addressed this challenge by formulating the problem using an expectation-based model and a worst-case model. They introduced a sampling-based successive convex approximation algorithm for solving the problem. This approach successfully handles noise by incorporating it as a regularizer in the loss function during the training process. Simulation results showcased improved prediction accuracy and reduced loss function values, affirming the effectiveness of the proposed methods in mitigating the impact of noise. In [21], Amiri et al. examined the impact of a noisy channel on FedAvg. They determined that when dealing with noisy downlink transmission, the presence of noise cannot be neutralized through step-size design, resulting in an inability to ensure precise convergence. Consequently, addressing this fundamental problem has necessitated imposing strict demands on the estimation noise associated with the aggregated model weights.

In [22], researchers conducted an investigation into the influence of a noisy channel in uplink and downlink analog transmission on the training process. The findings of this analysis led the authors to conclude that for the FedAvg algorithm to converge, it is necessary for the noise variance to decrease at a rate of $\mathcal{O}(1/k^2)$, where k represents the communication round. Consequently, in order to effectively mitigate the impact of channel noise on the training procedure, the authors recommend two approaches. Firstly, it is advised to employ an increased transmission power gain of $\mathcal{O}(k)$ in both uplink and downlink transmissions. Alternatively, if the power gain remains fixed, extending the transmission time to $\mathcal{O}(k)$ is suggested. The implementation of either of these strategies is crucial for preserving the integrity of the training process when faced with channel noise.

Another critical issue is analog transmission, which is widely used in over-the-air aggregation within the wireless channel. It is extensively utilized and plays a crucial role in enhancing spectral efficiency and reducing multi-access latency. Gau et al. [23] developed a hardware transceiver and application software to train a real-world FL task using over-the-air analog aggregation. They focused on developing an over-the-air aggregation solution for wireless FL based on orthogonal frequency-division multiplexing (OFDM). The main challenge they faced was achieving perfect waveform superposition, which was complicated by the presence of frame timing offset and carrier frequency offset. To tackle these challenges, they proposed a two-stage waveform pre-equalization technique.

The focus of this study pertains to the exploration of algorithms aimed at resolving optimization problems through over-the-air analog aggregation. While numerous papers have discussed communication-efficient solutions for distributed learning problems, it is of paramount importance to thoroughly investigate the optimization of analog FL prob-

lems over noisy communication channels. This particular problem presents significant importance and complexity, necessitating meticulous attention and consideration.

1.2. Our Contributions

The major contributions of this work are summarized as follows.

- We propose a hierarchical approach to PM, building upon our prior work [1]. The key distinction lies in the utilization of OACC for the FL algorithm at the factory level. This choice is motivated by the benefits of low latency, making it suitable for PM applications while also improving spectral efficiency. At higher levels, such as fog and cloud servers, occasional requests are made to the factory level for the aggregate model, enabling averaging over multiple factory parameters, FedAvg. Our primary emphasis is on the factory level, specifically investigating FLOACC as the focal point of our research.
- We propose FSVRG-OACC as a distributed approach to solve the optimization problem for PM at the factory level based on OACC. FSVRG-OACC leverages analog over-the-air aggregation, which enables it to effectively handle highly noisy communication channels and allows for improved convergence in minimizing the cost function associated with the ML algorithm.
- FSVRG-OACC facilitates the transmission of local gradient updates by individual agents, capitalizing on the advantages of computation over the air. This algorithm effectively mitigates the impact of channel perturbations on convergence by incorporating the effects of the communication channel into the algorithm update process. The utilization of FSVRG-OACC ensures that convergence is not compromised, enabling efficient and robust optimization in the presence of varying channel conditions.
- The simulation results demonstrate the substantial reduction in convergence sensitivity to noise achieved by our proposed algorithm. This finding holds significant implications for the implementation of ML algorithms on analog over-the-air aggregation in highly noisy industrial environments.

The remainder of this paper is structured as follows. In Section 2, we introduce the system model, starting with the description of FL over-the-air analog aggregation and extending it to FLOACC. Section 3 presents our proposed algorithm, FSVRG-OACC, along with a comparison to other stochastic gradient descent algorithms. Following that, in Section 4, we evaluate the performance of FSVRG-OACC through a PM application and present the corresponding experimental results. Finally, in Section 5, we provide concluding remarks and discuss future research directions.

2. System Model

2.1. Federated Edge Learning System

We consider a distributed learning system specifically designed for a PM application at the factory level. This system comprises a single parameter server and K edge nodes, as depicted in Figure 1. Collaboratively, the edge nodes at each factory train a shared learning process involving the global model \mathbf{w} . Each machine collects a fraction of labeled training data via the interaction with a local dataset, denoted as $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_K$. The local loss function of the model vector \mathbf{w} on \mathcal{D}_k is given by

$$F_k(\mathbf{w}) = \frac{1}{\phi_k} \sum_{(\mathbf{x}_j, y_j) \in \mathcal{D}_k} f(\mathbf{w}, \mathbf{x}_j, y_j), \quad (1)$$

where ϕ_k is the number of data points stored on data partition \mathcal{D}_k and $f(\mathbf{w}, \mathbf{x}_j, y_j)$ is the loss function quantifying the prediction error of the model \mathbf{w} for each data sample j , which consists of the training sample \mathbf{x}_j and its ground true label y_j . For convenience, we write $f(\mathbf{w}, \mathbf{x}_j, y_j)$ as $f_j(\mathbf{w})$ and assume uniform sizes for local datasets ϕ_k for all k . Table 1 presents the typical loss function used in FL for various applications. In the specific case of anomaly detection over edge nodes at the factory, the squared-SVM loss function has been employed

in this study as a leading ML method that offers flexibility in modeling complex nonlinear boundaries between normal and abnormal data points. With these definitions, the global loss function on all the distributed datasets can be defined as

$$F(\mathbf{w}) = \frac{\sum_{j \in \cup_k \mathcal{D}_k} f_j(\mathbf{w})}{\cup_k \phi_k} = \frac{1}{K} \sum_{k=1}^K F_k(\mathbf{w}), \tag{2}$$

where $|\cdot|$ denotes the size of the datasets and each dataset satisfies $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$ when $i \neq j$. The training target is to minimize the global loss function $F(\mathbf{w})$ according to the distributed process to find \mathbf{w}^* .

$$\mathbf{w}^* = \arg \min F(\mathbf{w}) \tag{3}$$

In this section, we begin by presenting the problem formulation of FL in the context of a hierarchical PM scenario. Subsequently, we delve into the specific case of FL-OACC, where all agents at the factory levels broadcast their updated models over the air, leveraging the advantages of computation through analog communication.

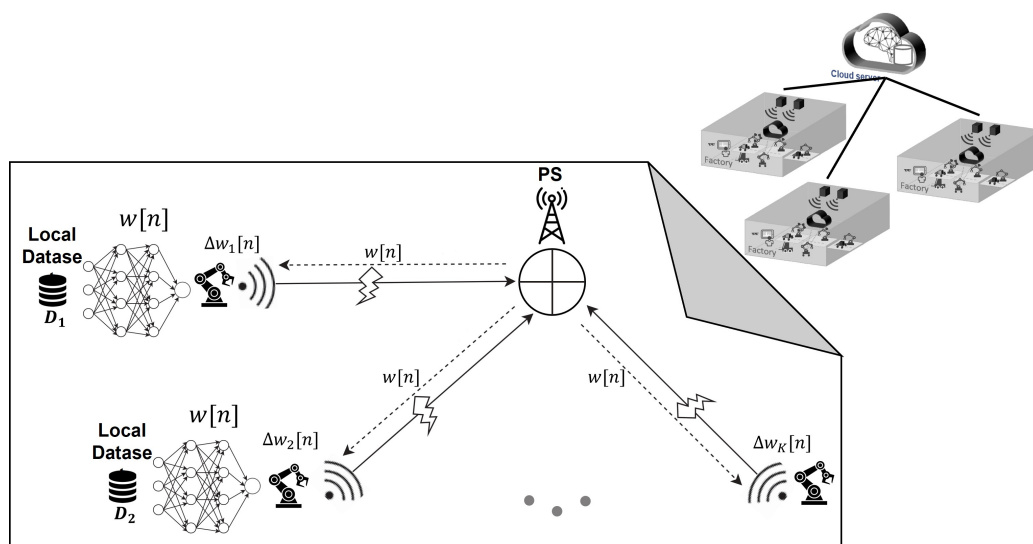


Figure 1. FL edge system model for a hierarchical PM scenario at the factory level.

Table 1. Several examples of loss functions.

Model	Loss Function
Linear regression	$\ 1 - y_j \mathbf{w}^T x_j\ ^2$
Logistic regression	$-\log(1 + \exp(-y_j \mathbf{w}^T x_j))$
K-means	$\min_l \ x_j - \mathbf{w}_l\ ^2$
Cross-Entropy	$-\sum y_c p(y = c x, \mathbf{w})$
Squared-SVM	$\lambda \ \mathbf{w}\ ^2 + \max 0; 1 - y_j \mathbf{w}^T x_j$

To compute $F(\mathbf{w})$, one method involves directly uploading all local data, which raises privacy concerns. To address this issue, the FL framework is employed to solve the problem outlined in Equation (3) through a distributed approach. There are two approaches based on FedAvg for solving this distributed optimization problem:

1. Model averaging: In this approach, each agent minimizes its local loss function and transmits the model parameters to the PS for aggregation. In the second round of iteration, the agent receives the updated model from the PS.
2. Gradient averaging: In this approach, each agent calculates the gradient of its loss function and transmits the gradient to the PS for aggregation. In the second round of

iteration, the agent can update its model based on the gradient averaging received from the PS.

The agents generally employ a gradient descent (GD) algorithm or stochastic gradient descent (SGD) algorithm to minimize the loss function described in Equation (1). A single-step SGD for updating the model parameter for device K can be defined following the model and gradient averaging, respectively.

$$\mathbf{w}_k[n+1] = \mathbf{w}[n] - \alpha \nabla F_k(\mathbf{w}[n]), \quad (4)$$

$$\mathbf{w}_k[n+1] = \mathbf{w}_k[n] - \alpha \nabla F(\mathbf{w}[n]), \quad (5)$$

where α is the step size and ∇ is the gradient operator. Hence, the only operation performed by the PS is to compute the average of the model parameters or the gradients received from the agents.

$$\mathbf{w}[n+1] = \frac{1}{K} \sum_{k=1}^K \mathbf{w}_k[n+1], \quad (6)$$

$$\nabla F(\mathbf{w}[n]) = \frac{1}{K} \sum_{k=1}^K \nabla F_k(\mathbf{w}_k[n]). \quad (7)$$

The learning process entails iterating between Equations (4) and (6) or Equations (5) and (7) until the model converges. The averaging process on the PS serves as a motivation for the low-latency FL scheme, utilizing FLOACC. Further details about FLOACC are provided below.

2.2. FL Over-the-Air Communication and Computation

In this section, we discuss the details of Federated Learning over-the-Air Communication and Computation, abbreviated as FLOACC. This method provides an efficient multi-access scheme in a low-latency scenario, which is crucial for applications like PM that require very fast and real-time task decision making. The idea of the over-the-air computation model for FL has been examined in several previous works, such as [8,16]. Their approach was inspired by the PS's lack of interest in individual model weight vectors. Instead, the server solely needs the average of the model weights, which is automatically provided by the wireless multiple-access channel in the form of their sum.

As depicted in Figure 2, the FLOACC enables simultaneous transmission of result vectors from all devices and assets at the factory level to the PS in an analog manner. Let $\mathbf{w}_k = [w_{k,1}, \dots, w_{k,q}]^T$ denote $q \times 1$ local model parameter vector and $\nabla F_k(\mathbf{w}) = [\nabla F_{k,1}, \dots, \nabla F_{k,q}]^T$ be the local gradient vector of the loss function from the k -th device. In FLOACC, it is assumed that the local gradient of each model is transferred over an analog medium. In this scenario, the transmitted symbols are denoted by $\{\tilde{\nabla} F_{k,i}\}$ and are normalized to have zero mean and unit variance $E(\tilde{\nabla} F_{k,i} \tilde{\nabla} F_{k,i}^*) = 1$. By employing OFDM, it becomes feasible to allocate each element of the gradient vector to a distinct sub-carrier OFDM channel. This approach enables a significant reduction in the learning process latency for FLOACC.

During each round n , all the local devices at a factory simultaneously transmit their local gradient based on the distributed loss function as $h_k[n] p_k[n] \nabla F_k(\mathbf{w}[n])$, where $h_k[n] \sim \mathcal{CN}(0, 1)$ is the small-scale fading of the channel between the k -th device and the PS, and $p_k[n]$ is the allocated transmission power for each device k . In particular, the aggregated gradient in the n -th communication round, denoted by $y[n]$, is expressed as follows.

$$\hat{\nabla} F(\mathbf{w}[n]) = \sum_{k=1}^K h_k[n] p_k[n] \nabla F_k(\mathbf{w}_k[n]) + Z[n], \quad (8)$$

where $Z[n] \sim \mathcal{CN}(0, \sigma_z^2 \mathbf{I})$ is additive white Gaussian noise. Note that in over-the-air analog aggregation, it is possible to transfer the model parameters as well. However, in FLOACC, we only consider the local gradient during aggregation. This is because the aggregated gradient is less sensitive in the optimization algorithms compared to the model parameters. We will discuss this issue further in the next section.

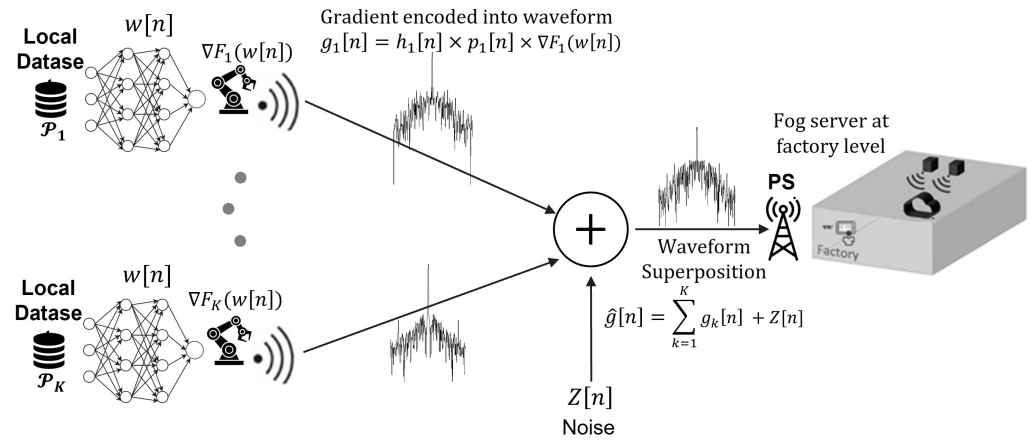


Figure 2. FLOACC scenario at the factory level.

We assumed that, similar to the LTE system, there are common downlink reference symbols in the radio resource block for the devices at the factory level to estimate the channel fading coefficient. Then, the local devices can transmit the signal with the specific power they calculated. The received signal at the PS has been depicted with gain blocks in Figure 3, so the PS can estimate the aggregated gradient as follows:

$$y[n] = \sum_{k=1}^K \frac{h_k[n] p_k[n] \nabla F_k(\mathbf{w}_k[n])}{\sqrt{\eta}} + \frac{Z[n]}{\sqrt{\eta}}, \quad (9)$$

where η is a receiver scaling factor. Owing to the imposed physical constraints, the transmission of each device is subject to a long-term transmission power constraint.

$$\mathbb{E} \left[\left| \sum_{n=1}^N |p[n](h[n])|^2 \right| \right] \leq P_0. \quad (10)$$

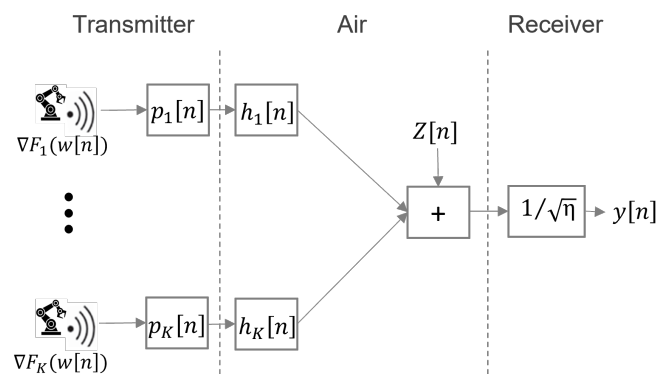


Figure 3. Illustration of the system design for FLOACC using analog transmission over-the-air aggregation.

2.3. Effective Noise and Definition of SNR in FLOACC

In order to keep the problem general, we adopt the effective noise model in analog aggregation. We assume accurate channel state information at the transmitters. To meet the aggregation requirement of FL, the local devices implement the channel inversion rule,

which yields the instantaneous transmit power of user k during the communication round n for gradient aggregation.

$$p_k[n] = \frac{h_k[n]^H}{|h_k[n]|} \cdot \frac{\sqrt{\eta}}{|h_k[n]|}, \quad (11)$$

where H is Hermitian transpose. Based on this definition, the received signal can be formulated as follows.

$$y[n] = \sum_{k=1}^K \nabla F_k(\mathbf{w}_k[n]) + \frac{Z[n]}{\sqrt{\eta}}. \quad (12)$$

η represents a scalar that signifies the average transmission power, based on which the received SNR of the global gradient can be expressed as follows.

$$\text{SNR}[n] = \mathbb{E} \left\| \frac{1}{K} \sum_{i=1}^q \frac{\sqrt{\eta} \sum_{k \in \mathcal{K}} \nabla F_k(\mathbf{w}_{k,i}[n])}{Z_i[n]} \right\|^2 = \frac{\eta \mathbb{E} \|\sum_{k \in \mathcal{K}} \nabla F_k(\mathbf{w}_k[n])\|^2}{\sigma_z^2 K^2}. \quad (13)$$

It is evident that the received SNR is consistent across all users due to devices with weaker channels compensating by transmitting at higher powers. In various studies, devices with significantly weak channels are often excluded from training due to their inability to pre-equalize their channels. Many research has been undertaken to optimize η to enhance model convergence over the wireless communication network. In [24,25], the optimal selection of p_k and η is determined by solving the corresponding optimization problem.

$$\text{MSE} = \mathbb{E} \left\| y[n] - \sum_{k=1}^K \nabla F_k(\mathbf{w}_k[n]) \right\|^2 = \frac{1}{K^2} \sum_{k=1}^K \left(\frac{|h_k| p_k}{\sqrt{\eta}} - 1 \right)^2 + \frac{\sigma_z^2}{\eta K^2}, \quad (14)$$

$$\begin{aligned} \min_{p_1, p_2, \dots, p_K, \eta} & \frac{1}{K^2} \sum_{k=1}^K \left(\frac{|h_k| \sqrt{p_k}}{\sqrt{\eta}} - 1 \right)^2 + \frac{\sigma_z^2}{\eta K^2} \\ \text{s.t. } & p_k \leq P_{\max}, \forall k \in \{1, 2, \dots, K\}. \end{aligned} \quad (15)$$

Extensive research has been conducted to optimize transmit power for the purpose of achieving model convergence and mitigating the effects of existing noise in the wireless transmission medium. However, to the best of our knowledge, no existing studies have investigated the comparison of convergence performance among different algorithms without any transmission power control. In the upcoming section, we delve into various gradient descent algorithms for ML optimization problems, considering the over-the-air scheme, and provide an analysis of which algorithm demonstrates superior convergence properties.

3. Proposed Adaptive FSVRG-OACC Algorithm

In this section, our specific focus lies on algorithms suitable for solving problem (3) within the context of over-the-air and analog aggregation. Firstly, in Section 3.1, we examine baseline algorithms that are compatible with distributed problems and analog gradient transmission (GT), while also highlighting the distinctions between model transmission (MT) and GT. Next, in Section 3.2, we delve into randomized methods that, in an initial approximation, integrate the advantages of cost-effective iterations from SGD with the rapid convergence of GD. Many of these methods can be categorized into one of two classes: dual methods of the randomized coordinate descent type and primal methods of the stochastic gradient descent with variance reduction type. Our emphasis lies on stochastic variance reduced gradient (SVRG), and we optimize this method for FL within the framework of analog aggregation while considering the presence of white Gaussian channel noise.

3.1. Baseline Algorithms

A fundamental approach for solving a distributed optimization problem with the structure (4) involves employing the GD algorithm, particularly when the functions f possess smoothness and convexity characteristics. In a distributed setting, there are two possible approaches: the GT method, where gradients are transmitted over the air and an aggregated gradient signal is received, which is subject to noise for model updating using GD, or the MT method, where the model is updated through GD iterations and then transmitted for model aggregation. In the presence of noise, the convergence deteriorates due to the elevated sensitivity of the loss function to the model parameters. Therefore, in over-the-air distributed algorithms, the GT approach is preferred. This preference stems from the lower sensitivity of the cost function to the aggregated gradient compared to the individual model parameters. GD demonstrates a fast convergence rate. However, each iteration has the potential to be computationally intensive on each local device. In contrast, SGD selects a random data label and performs the update, which offers a more efficient alternative.

3.2. Fsvrg-Oacc Algorithm

An additional algorithm within the SGD category is SVRG [26]. The SVRG algorithm operates through two nested loops. The outer loop involves calculating the full gradient of the entire function, $\nabla F(\mathbf{w}^t[n])$, which is typically a computationally expensive operation to be avoided whenever possible. In the inner loop, the update step is iteratively computed as follows.

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \alpha \left[\nabla F^i(\mathbf{w}[n]) - \nabla F^i(\mathbf{w}^t[n]) + \nabla F(\mathbf{w}^t[n]) \right], \quad (16)$$

where $\nabla F^i(\mathbf{w}[n])$ represents the stochastic gradient computed based on a randomly selected data label, $\nabla F^i(\mathbf{w}^t[n])$ denotes the stochastic gradient computed over the entire dataset, and α is stepsize. This iteration is specific to a single device, and its fundamental concept lies in utilizing stochastic gradients to estimate the gradient change from point \mathbf{w}^t to \mathbf{w} , rather than directly estimating the gradient itself.

Indeed, this algorithm is naturally suited for centralized implementations since it necessitates computing the stochastic gradient over the complete dataset, thereby making it well-suited for centralized scenarios. But a notable contribution was made in [27], where they introduced FSVRG, which is particularly applicable in the context of distributed optimization. They demonstrated that existing SVRG algorithms are not suitable for distributed approaches and proposed the FSVRG algorithm, specifically designed for sparse distributed convex problems. The pseudocode for FSVRG is provided in Algorithm 1. This algorithm has been implemented and subjected to evaluation, and the results will be presented in the experimental section. The findings indicate that this algorithm does not demonstrate satisfactory convergence in the realm of FL for over-the-air analog aggregation, specifically in relation to the absence of power transfer control.

Let us now elucidate the motivation behind considering a different algorithm suitable for FL in the context of over-the-air analog aggregation. A crucial aspect that demands attention is the significant variation in the number of available data points among different devices, which may differ greatly from the average number of data points available to any single device. It looks like a similar issue to FSVR, but it should be noted that in our assumptions, analog communication is the sole communication type between local devices and the PS. As a result, the PS lacks information regarding the number of data points and the type of data distribution.

Additionally, this scenario frequently entails the local data being clustered around a specific pattern, which renders it unrepresentative of the overall distribution we aim to learn. Consequently, considering an aggregation on the entire gradient direction in each iteration could be a promising approach that could be undertaken in the concept of analog aggregation.

Algorithm 1: Federated SVRG

Parameters: ϕ = number of data points, ϕ_k = number of data points store on device k , α = stepsize, data partition $\{\mathcal{D}_k\}_{k=1}^K$.

for $n = 0, 1, \dots$ **do**

Compute $\nabla F(\mathbf{w}^t[n]) = \frac{1}{\phi} \sum_{i=1}^{\phi} \nabla F^i(\mathbf{w}^t[n])$ » Overall iterations

for $k = 1$ to K **do in parallel over device** k **do**

Initialize: $\mathbf{w}_k = \mathbf{w}^t$ and $\alpha_k = \alpha / \phi_k$ » Distributed loop

Let $\{i_t\}_{t=1}^{\phi_k}$ be random permutation of \mathcal{D}_k

for $t = 1, \dots, \phi_k$ **do**

$\mathbf{w}_k[n+1] = \mathbf{w}_k[n] - \alpha_k [\nabla F^{i_t}(\mathbf{w}_k[n]) - \nabla F^{i_t}(\mathbf{w}^t[n]) + \nabla F(\mathbf{w}^t[n])]$

end

end

$\mathbf{w}^t[n] = \mathbf{w}^t[n] + \sum_{k=1}^K \frac{\phi_k}{\phi} (\mathbf{w}_k[n] - \mathbf{w}^t[n])$ » Model aggregation

end

From a practical perspective, in FSVRG-OACC, it is postulated that all devices possess a randomly allocated initialization value for the parameter vector, \mathbf{w}_k . This assumption holds significant importance in the practical execution of the algorithm. The proposed algorithm involves two communication rounds, which results in increased communication costs. However, it offers advantages in terms of the convergence algorithm. Algorithm 2 introduces the FSVRG-OACC, a modified FSVRG variant tailored for over-the-air analog aggregation.

During the initial communication round (distributed loop 1), all devices compute the complete gradient of the entire function and subsequently determine the internal gradient as $g_k = [\nabla F_k^{i_t}(\mathbf{w}^t[n]) - \nabla F_k(\mathbf{w}[n])]$, where i_t is sampled uniformly from the local dataset \mathcal{D}_k . These gradients are derived using SGD, rendering their computation relatively inexpensive. The computed internal gradient is then transmitted over the air to the PS, while the estimated aggregated internal gradient is sent back to the devices via the analog medium, as shown in Figure 2.

$$\hat{g} = \frac{1}{K} \sum_{k=1}^K \frac{h_k[n] p_k[n] g_k}{\sqrt{\eta}} + \frac{z[n]}{\sqrt{\eta}}. \quad (17)$$

The estimated aggregated internal gradient, denoted as \hat{g} , compels all devices in the second round of communication (Distributed loop 2) to move in the same direction. In this communication round, the updated gradient is denoted as \hat{G} , which was estimated in the first round as $\hat{G} = \nabla F^{i_t}(\mathbf{w}_k[n]) - \hat{g}$ for device k . Subsequently, the PS aggregates this gradient over the air from all devices and transmits it back to them for the remaining iterations. Each device k uploads the following gradient over the air for aggregation.

$$G_k = \nabla F^{i_t}(\mathbf{w}_k[n]) - \hat{g}, \quad (18)$$

$$G_k = \nabla F^{i_t}(\mathbf{w}_k[n]) - \left[\frac{1}{K} \sum_{k=1}^K \frac{h_k[n] p_k[n] [\nabla F_k^{i_t}(\mathbf{w}^t[n]) - \nabla F_k(\mathbf{w}[n])]}{\sqrt{\eta}} + \frac{z[n]}{\sqrt{\eta}} \right]. \quad (19)$$

By aggregating the gradient G_k at the PS and transmitting it back to each device, a uniform descent direction can be achieved across all devices.

$$\hat{G} = \frac{1}{K} \sum_{k=1}^K \frac{h'_k[n] p'_k[n] G_k}{\sqrt{\eta'}} + \frac{z'[n]}{\sqrt{\eta'}}, \quad (20)$$

$$\hat{G} = \frac{1}{K} \sum_{k=1}^K \frac{h'_k[n] p'_k[n] \left[\nabla F^{i_t}(\mathbf{w}_k[n]) - \left[\frac{1}{K} \sum_{k=1}^K \frac{h_k[n] p_k[n] [\nabla F_k^{i_t}(\mathbf{w}^t[n]) - \nabla F_k(\mathbf{w}[n])]}{\sqrt{\eta}} + \frac{z[n]}{\sqrt{\eta}} \right] \right]}{\sqrt{\eta'}} + \frac{z'[n]}{\sqrt{\eta'}}. \quad (21)$$

Algorithm 2: FSVRG With Over-the-Air Communication and Computation (FSVRG-OACC)

Parameters: ϕ = number of data points, ϕ_k = number of data points store on device k , α = stepsize, data partition $\{\mathcal{D}_k\}_{k=1}^K$, Randomly initialize \mathbf{w}_k on each device.

for $n = 0, 1, \dots$ **do**

» Overall iterations

for $k = 1$ to K **do in parallel over device** k **do**

1: Let $\{i_t\}_{t=1}^{\phi_k}$ be random permutation of \mathcal{D}_k » Distributed loop 1

2: Compute $\nabla F_k^{i_t}(\mathbf{w}^t[n])$

3: Compute $\nabla F_k(\mathbf{w}[n]) = \frac{1}{\phi_k} \sum_{i=1}^{\phi_k} \nabla F_k^{i_t}(\mathbf{w}^t[n])$

4: Over-the-Air Gradient Aggregation: Each device k uploads

$g_k = [\nabla F_k^{i_t}(\mathbf{w}^t[n]) - \nabla F_k(\mathbf{w}[n])]$ Over-the-Air.

end

Aggregated signal in PS (Server Side)

$\hat{g} = \frac{1}{K} \sum_{k=1}^K \frac{h_k[n] p_k[n] g_k}{\sqrt{\eta}} + \frac{z[n]}{\sqrt{\eta}}$ » Model aggregation 1

Aggregated signal \hat{g} is received by all edge devices

for $k = 1$ to K **do in parallel over device** k **do**

1: Initialize: $\alpha_k = \alpha / \phi_k$ » Distributed loop 2

2: Compute $\nabla F^{i_t}(\mathbf{w}_k[n])$

3: Aggregated signal \hat{G} is received by the edge device k

$n = 0 \rightarrow$ randomly initialize \hat{G}

for $t = 1, \dots, \phi_k$ **do**

| $\mathbf{w}_k[n+1] = \mathbf{w}_k[n] - \alpha_k \hat{G}$

end

4: Over-the-Air Gradient Aggregation: Each device k uploads

$G_k = [\nabla F^{i_t}(\mathbf{w}_k[n]) - \hat{g}]$ Over-the-Air.

end

Aggregated signal in PS (Server Side)

$\hat{G} = \frac{1}{K} \sum_{k=1}^K \frac{h'_k[n] p'_k[n] G_k}{\sqrt{\eta'}} + \frac{z'[n]}{\sqrt{\eta'}}$ » Model aggregation 2

end

This approach is motivated by the primary goal of maintaining gradient step consistency among all clients by utilizing the aggregated gradient over-the-air signal in stochastic first-order methods. Therefore, the algorithm's complexity is exceptionally low due to the simplicity of first-order gradient calculations at each step. The only cost involved is the number of communication rounds.

In summary, the gradient update $[\nabla F^{i_t}(\mathbf{w}_k[n]) - \nabla F_k^{i_t}(\mathbf{w}^t[n]) + \nabla F_k(\mathbf{w}^t[n])]$ is divided into two parts for FSVRG-OACC. In the first distributed loop, the last two gradients $\nabla F_k^{i_t}(\mathbf{w}^t[n])$ and $\nabla F_k(\mathbf{w}^t[n])$ are calculated, and afterward, the distance between these two gradients is aggregated across all edge devices. In the second distributed loop, we can access the aggregated value of $[\nabla F_k^{i_t}(\mathbf{w}^t[n]) - \nabla F_k(\mathbf{w}^t[n])]$ for all devices. In the first iteration of this loop ($n = 1$), we calculate the stochastic gradient for each edge device and subtract it from the aggregated gradient obtained from the first distributed loop $[\nabla F^{i_t}(\mathbf{w}_k[n]) - \text{aggregation}(\nabla F_k^{i_t}(\mathbf{w}^t[n]) - \nabla F_k(\mathbf{w}^t[n]))]$. This gives us G_k , the overall gradient update for device k , which is then aggregated with all other gradients over the air.

Starting from the second iteration ($n = 2$) and beyond, we can utilize the whole aggregated gradient \hat{G} to directly update the model parameters of each device.

4. Performance Evaluation of FSVRG-OACC

In this section, we evaluate the performance of the proposed algorithm on the task of anomaly detection for a PM application in an FL manner with over-the-air analog aggregation. It aims to investigate the convergence characteristics of four optimization algorithms, GD, SGD, FSVRG, and FSVRG-OACC, in the context of over-the-air analog aggregation. The primary focus is to analyze the performance of these algorithms in terms of both model accuracy and convergence time, which are crucial performance metrics in the field. Similar to prior research in the domain of Federated Machine Learning Algorithm for Collaborative Predictive Maintenance [1], we employ the widely recognized and benchmarked CMAPSS [28] dataset in this study.

The C-MAPSS model, developed by NASA and implemented in MATLAB/Simulink, represents a nonlinear dynamic model of a commercial turbofan engine. By manipulating the input parameters of this simulation model, it becomes possible to simulate diverse degradation patterns under various engine conditions. In order to generate datasets reflecting different fault modes, four distinct time series (FD001, FD002, FD003, FD004) were produced using these simulation tools. These datasets comprise multivariate time series, each of which is further divided into training and testing subsets. For the purpose of this study, we selected and analyzed two datasets: FD003 and FD004. The FD003 dataset comprises 100 test trajectories and 100 train trajectories, focusing on a single fault mode and two types of degradation: high-pressure compressor degradation and fan degradation. On the other hand, the FD004 dataset includes 248 test trajectories and 249 train trajectories, covering six fault modes and two types of degradation. The time series data in each dataset include 21 sensor observations, three operating settings, a trajectory ID, and a cycle count. The Remaining Useful Life (RUL) of an engine is estimated based on the number of operation cycles remaining before the engine fails.

4.1. Algorithm Implementation

In this study, a federated SVM model was employed to predict the RUL using the provided time series data. Specifically, FD003 and FD004 datasets were distributed among ten devices located at the factory site, with the purpose of participating in a collaborative PM scenario. It is crucial to emphasize that the data distribution was non-independent and not identically distributed (non-IID). This implies that the instances of failures were not randomly distributed among the edge devices, and the data distribution was not evenly spread across all edge devices. Instead, specific edge devices were linked to particular types of failures, while others experienced different types. We intentionally opted for this non-IID distribution as a worst-case scenario to thoroughly assess the effectiveness of our proposed method. This collaboration aimed to achieve low latency through over-the-air analog aggregation. During all the simulations, the optimizer parameters were configured to achieve optimal performance. In particular, the learning is kept fixed at 0.01, and the ℓ_2 -regularization parameter λ is set to $\lambda = 0.1$. The momentum factor is kept equal to 0.9 and the number of local epochs is set to 1. The definition of the loss function for the federated SVM is as follows.

$$f(\mathbf{w}) = \frac{1}{\phi_k} \sum_{j \in \mathcal{D}_k} f_j(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2 \quad (22)$$

$$f_j(\mathbf{w}) = \max(0, 1 - y_j \mathbf{w}^T x_j),$$

The input data, denoted as x_j , are structured in a 2D format resembling an image, where one dimension corresponds to the sequence length, and the other dimension represents the number of sensor measurements. The output variable, y_j , indicates the condition of the engine. Specifically, if the input data are associated with a particular condition whose

RUL is below a specified threshold, the output value is assigned as $\{-1\}$, indicating the detection of an anomaly. Conversely, if the RUL exceeds the threshold, the output value is assigned as $\{1\}$. In all the conducted experiments, the reported convergence and accuracy averaged over five independent runs were calculated over 500 communication rounds.

4.2. Performance of FSVRG-OACC

4.2.1. General Performance

We first conduct a comparison between FSVRG-OACC and standard FSRVG, SGD, and GD methods to illustrate the notable enhancements in convergence rate and accuracy achieved in the context of over-the-air analog aggregation with a noisy environment. Throughout the experiments, we assessed the performance of each method and observed the significant gains achieved by FSVRG-OACC. We assume the environment noise variance to be $\sigma_z^2 = 1$ for $\mathbf{Z}_k \sim \mathcal{CN}(0, \sigma_z^2 \mathbf{I})$, and the probability of noise presence as $p = 1$. In this study, our primary focus is not on the transmission policy. Therefore, we make the assumption that the channel inversion policy is employed for estimation. Additionally, we assume that each component of $Z[n]$ has a zero mean and a variance of σ_z^2 .

The performance is demonstrated on the FD003 and FD004 datasets. In this experiment, for all of $K = 10$ devices at the factory site, this implies that the transmitting power is set to $P = 300$ mW and the receiving scale factor, $\sqrt{\eta}$, is set to 0.1. The accuracy formula used is as follows:

$$\text{accuracy} = \frac{\mu_P + \mu_N}{\mu_P + \mu_N + \Gamma_P + \Gamma_N}, \quad (23)$$

where μ_P represents the number of true positives, Γ_P represents false positives, μ_N represents true negatives, and Γ_N represents false negatives. The results have been plotted in Figure 4 for FD003 and Figure 5 for FD004. We observed the following results.

1. In the case of FD003, it is evident that both the GD-MT and FSVRG algorithms fail to converge under the given noise environment and transmission power settings. Consequently, these algorithms are excluded from the accuracy analysis. On the other hand, the GD-GT and SGD-GT algorithms demonstrate convergence, but they exhibit significant fluctuations during the convergence phase. In contrast, our proposed FSVRG-OACC method shows excellent convergence performance under the same environmental conditions and noise levels.

As depicted in the accuracy plot for FD003, both SGD-DT and GD-GT algorithms experience a considerable drop in accuracy. However, our proposed algorithm achieves an average accuracy of 91% and demonstrates higher stability compared to the other algorithms.

2. In the case of FD004, we observed similar results, although this dataset presents greater challenges due to its inclusion of six fault modes, making the prediction algorithm significantly more complex compared to other CMAPSS datasets. Despite these difficulties, our proposed FSVRG-OACC algorithm demonstrates robust convergence, with only minimal fluctuations occurring after the convergence stage. These fluctuations can be attributed to the estimation anomalies present in the most challenging instances of the dataset.

In the accuracy plot, our proposed method achieved a commendable accuracy of 61% on the model. In contrast, the average accuracy of the other two converged methods is lower than that of the proposed method, and their accuracy values exhibited less fluctuation during the communication rounds.

In conclusion, we provide the average local training runtime and accuracy of FSVRG-OACC, SGD-GT, and GD-GT at the client level, as summarized in Table 2. It is noteworthy to mention that the computational runtime is primarily governed by the computation of gradients in all 500 iterations. Additionally, we assume that the communication round is negligible in comparison to the gradient computation. As is evident, the runtime in FSVRG-OACC exceeds that of the other algorithms. However, this method, employed in

FLOACC, demonstrates substantial reduction in communication latency while achieving commendable convergence and consistent accuracy.

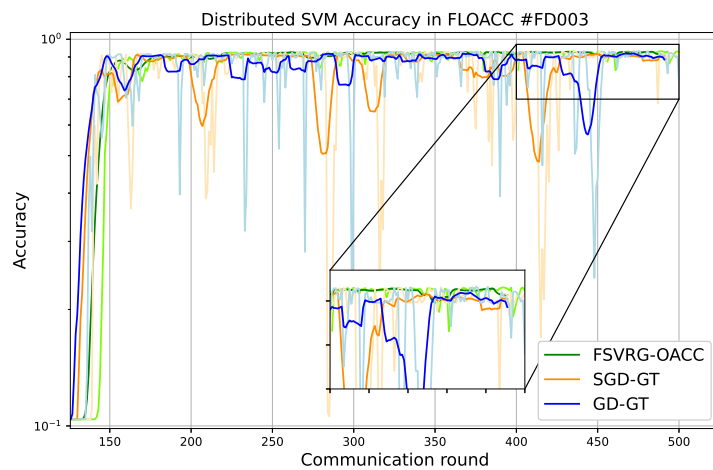
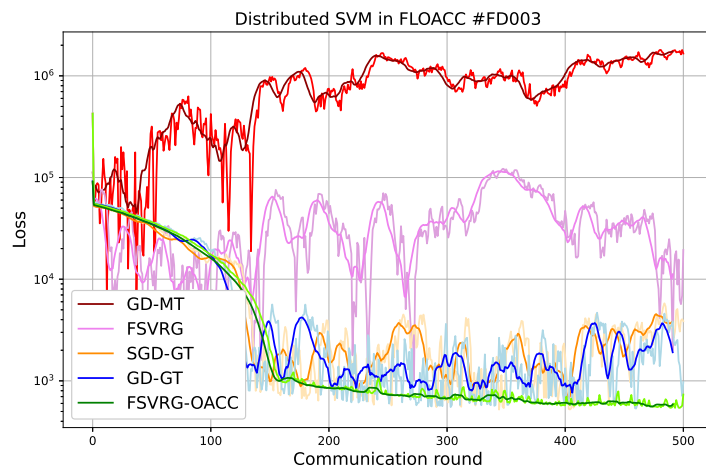


Figure 4. Convergence and accuracy versus communication round for different algorithms in Over-the-Air analog aggregation ($\sigma_z^2 = 1$, $p = 1$, and dataset: FD003). The pale graphs represent the signal, while the bold graphs depict the windowed average of the signal.

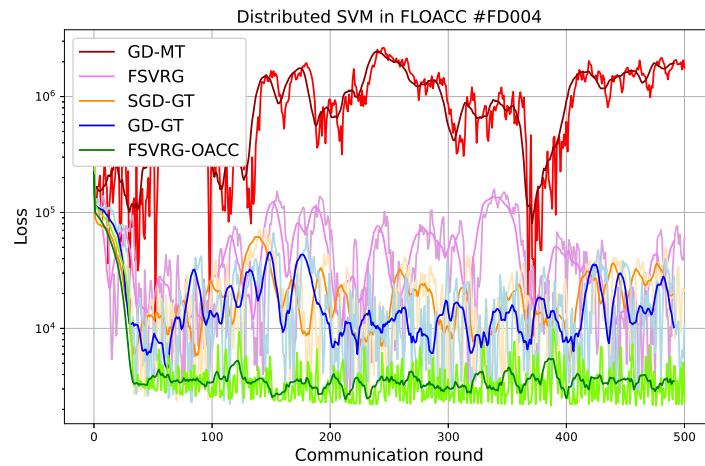


Figure 5. Cont.

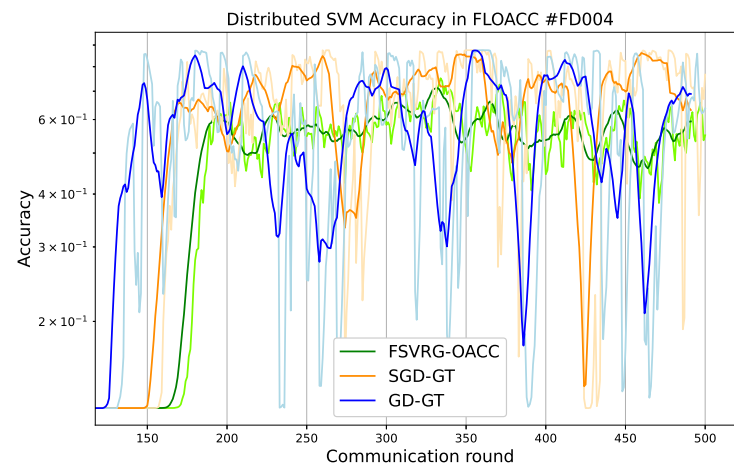


Figure 5. Convergence and accuracy versus communication round for different algorithms in Over-the-Air analog aggregation ($\sigma_z^2 = 1$, $p = 1$, and dataset: FD004). The pale graphs represent the signal, while the bold graphs depict the windowed average of the signal.

Table 2. Performance analysis of different algorithms on FLOACC.

Dataset	Evaluation Metrics	Optimizer		
		GD-GT	SGD-GT	FSVRG-OACC
FD003	Runtime	45.8 s	21.3 s	66.7 s
	Final accuracy	76%	61%	91%
FD004	Runtime	70 s	20 s	105 s
	Final accuracy	42%	41%	61%

We compared our results with a scenario where there is no communication noise. In a previous study, we looked at how well the FL algorithm works for PM applications when there is no communication interference. We summarized those results in Table 3. As evident from the results, there is a significant disparity in accuracy between GD-GT and SGD-GT when applied to FLOACC and noiseless communication settings. This demonstrates that employing a power transmission policy can enhance the accuracy of both GD and SGD in the context of FLOACC, bringing them closer to achieving results similar to those in noiseless communication scenarios. It is worth noting that, for the FSVRG-OACC method and FSVRG applied to noiseless communication, the difference in accuracy remains minimal. This observation is particularly noteworthy in the case of FD003, where the RUL prediction task is less challenging than FD004. These findings underscore the robustness and effectiveness of our proposed approach, which achieves these results without the need for any power transmission policies.

Table 3. Performance analysis of FL algorithm on noiseless communication channel [1].

Dataset	Evaluation Metrics	Optimizer		
		GD-GT	SGD-GT	FSVRG
FD003	Runtime	69 s	20.5 s	161 s
	Final accuracy	94.2%	92.2%	91.7%
FD004	Runtime	143.5 s	45.5 s	337 s
	Final accuracy	78.4%	71.7%	86.6%

4.2.2. Performance of FSVRG-OACC with Varying Noise Level

In this section, we conduct an analysis to assess the robustness of the proposed algorithms for Over-The-Air analog aggregation in the presence of varying levels of noise.

The performance evaluation is conducted using a part of the FD003 dataset while systematically varying the noise level σ_z . Specifically, we consider noise levels σ_z from the set $\{0.25, 0.75, 1.5, 2.5\}$ to assess the algorithm's performance under different noise conditions. For the total of 10 local devices, the receiver scaling factor, denoted as $\sqrt{\eta}$, is varied across different values $\{0.4, 0.13, 0.06, 0.04\}$. To investigate the impact of noise, we manipulate the probability of noise presence, denoted as p , throughout the experiments. The set $\{0.1, 0.25, 0.5, 1\}$ is utilized to vary the probability of noise. The results for GD-GT and SGD-GT have been plotted in Figures 6 and 7.

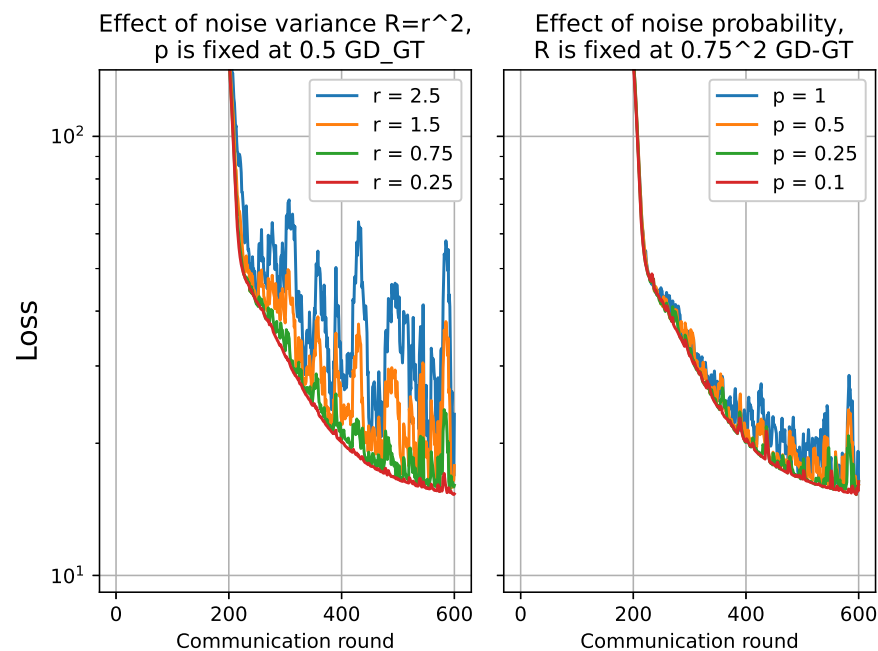


Figure 6. Convergence analysis for GD-GT by varying noise level $r \equiv \sigma_z = \{0.25, 0.75, 1.5, 2.5\}$, and the probability of noise presence $p = \{0.1, 0.25, 0.5, 1\}$.

For both GD-GT and SGD-GT, as the noise level increases, there is a significant degradation in performance, accompanied by amplified fluctuations in the loss function. Additionally, we observe a parallel trend in the cost function when the probability of noise presence is elevated. Furthermore, we conducted the same analysis on the FSVRG-OACC method, and the corresponding results are presented in Figure 8. Notably, the proposed algorithm demonstrates significantly enhanced robustness against higher noise levels and increased probability of noise presence.

All of these analyses provide evidence supporting the suitability of the proposed FSVRG-OACC method for analog aggregation and FL over the air. It is important to note that all of these analyses were conducted under the same transmission power conditions. However, we anticipate that employing a power control policy in communication with this method would yield even more accurate results in convergence and model accuracy.

On the other hand, implementing our proposed approach in real-world industrial settings has practical implications and challenges. A significant part of this approach is adjusting the local gradients in large edge devices using an analog waveform and sending them through the same wireless channels. Our main challenge is achieving perfect waveform superposition, which is crucial for our algorithm's success. This task is complicated because of frame timing offset and carrier frequency offset. To overcome these issues, we require high-performance devices with substantial computational capabilities. Another important challenge emerges when we only have access to partial information, resulting in some edge devices being unable to effectively join the FLOACC aggregation process. This led to a deviation in the gradient descent regime. This deviation highlights

the necessity for robust device selection and strategies to manage situations where certain devices may have limited participation capabilities.

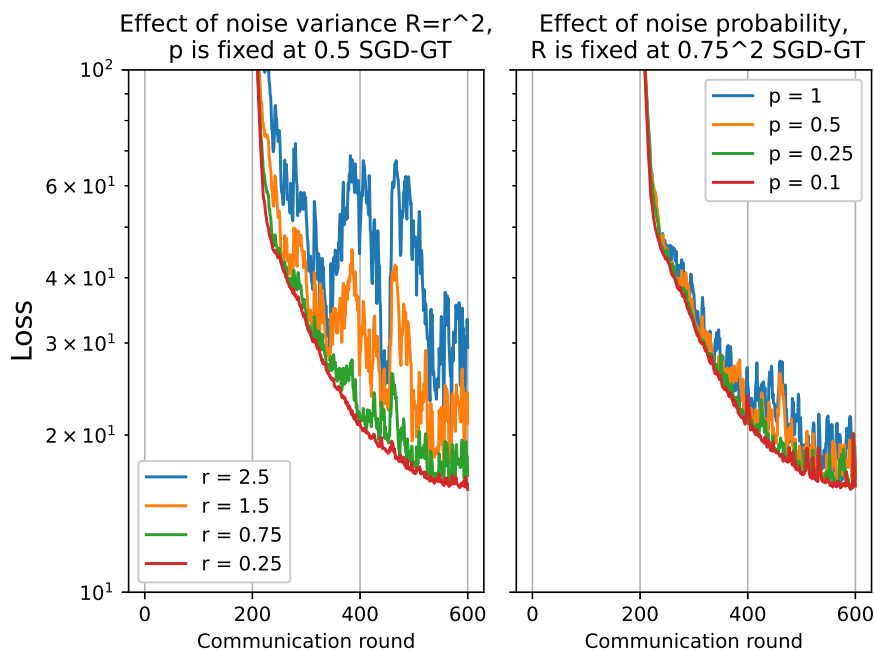


Figure 7. Convergence analysis for SGD-GT by varying noise level $r \equiv \sigma_z = \{0.25, 0.75, 1.5, 2.5\}$, and the probability of noise presence $p = \{0.1, 0.25, 0.5, 1\}$.

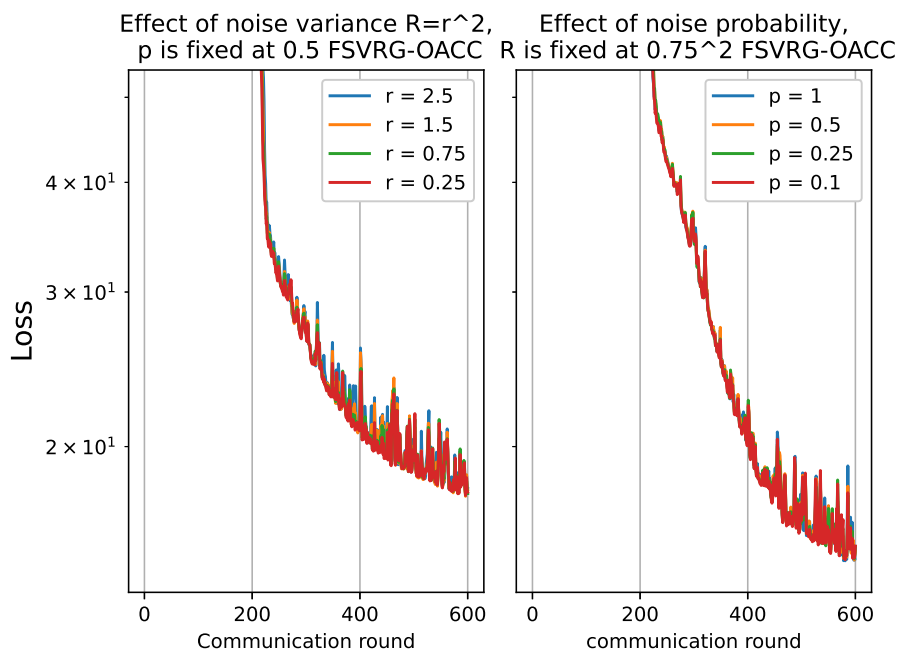


Figure 8. Convergence analysis for FSVRG-OACC by varying noise level $r \equiv \sigma_z = \{0.25, 0.75, 1.5, 2.5\}$, and the probability of noise presence $p = \{0.1, 0.25, 0.5, 1\}$.

5. Conclusions

In this paper, our focus was on FL scenarios that leverage wireless transmission channels for both communication and computation. The design presented capitalizes on the waveform superposition property inherent in a multi-access channel, which enables efficient update aggregation, optimizing the communication process. We specifically emphasized

the potential of over-the-air analog aggregation in FL for hierarchical PM scenarios. Our study focused on the factory site as the lower hierarchical component, recognizing the criticality of meeting low-latency requirements for time-sensitive applications. We have demonstrated that this challenge can be effectively addressed through the utilization of analog aggregation.

Throughout our investigation, we thoroughly examined the impact of practical channel effects, including noise and fading, on the learning algorithm's performance. To enhance the robustness of learning algorithms against channel noise effects, we proposed a novel algorithm called FSVRG-OACC. The effectiveness and superiority of the proposed algorithm were demonstrated through the application of a distributed SVM for anomaly detection using the CMPASS dataset. The simulation results validate the effectiveness of FSVRG-OACC in reducing aggregation distortion while operating at the same transmission power level. Furthermore, the learning behavior of the proposed algorithm can be further enhanced by incorporating power control and device selection policies into its design.

Author Contributions: Conceptualization, N.B. and A.B.; methodology, A.B.; software, A.B.; validation, N.B. and A.B.; formal analysis, N.B. and A.B.; investigation, N.B. and A.B.; resources, N.B.; data curation, A.B.; writing—original draft preparation, A.B.; writing—review and editing, N.B. and A.B.; visualization, N.B. and A.B.; supervision, N.B.; project administration, N.B.; funding acquisition, N.B. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the European Commission within the European Regional Development Fund, Swedish Agency for Economic and Regional Growth, Region Gävleborg, and the University of Gävle.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The CMAPSS dataset is available at <https://data.nasa.gov/dataset/C-MAPSS-Aircraft-Engine-Simulator-Data/xaut-bemq> (accessed on 19 August 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
ML	Machine Learning
PM	Predictive Maintenance
PS	Parameter Server
FL	Federated Learning
OACC	Over-the-Air Communication and Computation
SNR	Signal-to-Noise Ratio
FedAvg	Federated Averaging
OFDM	Orthogonal Frequency-Division Multiplexing
GD	Gradient Descent
SGD	Stochastic Gradient Descent
non-IID	non-independent and not identically distributed
FSVRG	Federated Stochastic Variance Reduced Gradient
FLOACC	Federated Learning Over-the-Air Communication and Computation
FSVRG-OACC	Federated Stochastic Variance Reduced Gradient-Over-the-Air Communication and Computation

References

1. Bemani, A.; Björzell, N. Aggregation Strategy on Federated Machine Learning Algorithm for Collaborative Predictive Maintenance. *Sensors* **2022**, *22*, 6252. [[CrossRef](#)] [[PubMed](#)]
2. Amiri, M.M.; Gündüz, D. Machine Learning at the Wireless Edge: Distributed Stochastic Gradient Descent Over-the-Air. *IEEE Trans. Signal Process.* **2020**, *68*, 2155–2169. [[CrossRef](#)]

3. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
4. Peng, C.; Hu, Q.; Wang, Z.; Liu, R.W.; Xiong, Z. Online-Learning-Based Fast-Convergent and Energy-Efficient Device Selection in Federated Edge Learning. *IEEE Internet Things J.* **2023**, *10*, 5571–5582. [[CrossRef](#)]
5. Liu, S.; Yu, J.; Deng, X.; Wan, S. FedCPF: An Efficient-Communication Federated Learning Approach for Vehicular Edge Computing in 6G Communication Networks. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1616–1629. [[CrossRef](#)]
6. Zhang, L.; Wang, H.; Xue, H.; Zhang, H.; Liu, Q.; Niyato, D.; Han, Z. Digital twin-assisted edge computation offloading in industrial Internet of Things with NOMA. *IEEE Trans. Veh. Technol.* **2022**. [[CrossRef](#)]
7. Zheng, S.; Shen, C.; Chen, X. Design and Analysis of Uplink and Downlink Communications for Federated Learning. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2150–2167. [[CrossRef](#)]
8. Amiri, M.M.; Gündüz, D. Federated Learning Over Wireless Fading Channels. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3546–3557. [[CrossRef](#)]
9. Krouka, M.; Elgabli, A.; Ben Issaid, C.; Bennis, M. Communication-Efficient Federated Learning: A Second Order Newton-Type Method With Analog Over-the-Air Aggregation. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 1862–1874. [[CrossRef](#)]
10. Jing, S.; Xiao, C. Federated Learning via Over-the-Air Computation With Statistical Channel State Information. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 9351–9365. [[CrossRef](#)]
11. Yang, P.; Jiang, Y.; Wang, T.; Zhou, Y.; Shi, Y.; Jones, C.N. Over-the-Air Federated Learning via Second-Order Optimization. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 10560–10575. [[CrossRef](#)]
12. Du, J.; Jiang, B.; Jiang, C.; Shi, Y.; Han, Z. Gradient and Channel Aware Dynamic Scheduling for Over-the-Air Computation in Federated Edge Learning Systems. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1035–1050. [[CrossRef](#)]
13. Konečný, J.; McMahan, B.; Ramage, D. Federated optimization: Distributed optimization beyond the datacenter. *arXiv* **2015**, arXiv:1511.03575.
14. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.Y. Communication-efficient learning of deep networks from decentralized data. *Proc. Artif. Intell. Statist.* **2017**, *54*, 1273–1282.
15. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [[CrossRef](#)]
16. Zhu, G.; Wang, Y.; Huang, K. Broadband Analog Aggregation for Low-Latency Federated Edge Learning. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 491–506. [[CrossRef](#)]
17. Chen, M.; Yang, Z.; Saad, W.; Yin, C.; Poor, H.V.; Cui, S. A Joint Learning and Communications Framework for Federated Learning Over Wireless Networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 269–283. [[CrossRef](#)]
18. Yang, H.H.; Liu, Z.; Quek, T.Q.; Poor, H.V. Scheduling Policies for Federated Learning in Wireless Networks. *IEEE Trans. Commun.* **2020**, *68*, 317–333. [[CrossRef](#)]
19. Krouka, M.; Elgabli, A.; Issaid, C.B.; Bennis, M. Communication-Efficient and Federated Multi-Agent Reinforcement Learning. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 311–320. [[CrossRef](#)]
20. Ang, F.; Chen, L.; Zhao, N.; Chen, Y.; Wang, W.; Yu, F.R. Robust Federated Learning With Noisy Communication. *IEEE Trans. Commun.* **2020**, *68*, 3452–3464. [[CrossRef](#)]
21. Amiri, M.M.; Gündüz, D.; Kulkarni, S.R.; Poor, H.V. Convergence of Federated Learning Over a Noisy Downlink. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 1422–1437. [[CrossRef](#)]
22. Wei, X.; Shen, C. Federated Learning Over Noisy Channels: Convergence Analysis and Design Examples. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 1253–1268. [[CrossRef](#)]
23. H. Guo et al., Over-the-Air Aggregation for Federated Learning: Waveform Superposition and Prototype Validation. *J. Commun. Inf. Netw.* **2021**, *4*, 429–442. [[CrossRef](#)]
24. Cao, X.; Zhu, G.; Xu, J.; Huang, K. Optimized Power Control for Over-the-Air Computation in Fading Channels. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 7498–7513. [[CrossRef](#)]
25. Liu, W.; Zang, X.; Li, Y.; Vucetic, B. Over-the-Air Computation Systems: Optimization, Analysis and Scaling Laws. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 5488–5502. [[CrossRef](#)]
26. Xiao, L.; Zhang, T. A proximal stochastic gradient method with progressive variance reduction. *SIAM J. Optim.* **2014**, *24*, 2057–2075. [[CrossRef](#)]
27. Konečný, J.; McMahan, H.B.; Ramage, D.; Richtárik, P. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv* **2016**, arXiv:1610.02527.
28. Li, X.; Ding, Q.; Sun, J.Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliab. Eng. Syst. Saf.* **2018**, *172*, 1–11. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.