WFSL: Warmup-based Federated Sequential Learning

Mohamad Arafeh, Ahmad Hammoud, Mohsen Guizani, Azzam Mourad, Hadi Otrok, Hakima Ould-Slimane, Zbigniew Dziong, Chang-Dong Wang, and Di Wu

***Abstract*—Federated learning gained importance in sensitive IoT environments by creating a privacy-preserving ecosystem where participants share machine-learning models instead of raw data. However, federated learning shifts data control away from the server, exposing it to Non-Independent and Identically Distributed (non-IID) problems caused by biased clients (IoT devices). This hinders the learning process by increasing execution time and cost. Current solutions alter the federated learning structure or compromise privacy by offloading clients' raw data to an external server. To mitigate these limitations, this paper proposes a solution to the non-IID problem by introducing an initialization phase, orchestrated by the server, that constructs high-quality initial models. These models can boost federated learning accuracy and convergence, regardless of whether IoT participants exhibit non-IID properties. Our proposed initialization scheme involves clients training over the same model sequentially, lessening the impact of aggregation, a primary cause of model degradation in federated approaches. Furthermore, a regulator algorithm deployed on the server maintains model integrity and mitigates catastrophic forgetting, enhanced by a client selection process that emphasizes the compatibility of IoT clients to cooperate effectively. Moreover, we devise an optimization scheme based on clustering and genetic algorithms to reduce the selection time while ensuring optimal performance in IoT networks. Experiments on MNIST, KDD, and CIFAR10 datasets show promising results in terms of initial model resiliency against catastrophic forgetting and non-IID settings. Additionally, our findings suggest that our approach can significantly enhance federated learning training in IoT applications by achieving higher accuracy more quickly compared to conventional methods.**

***Index Terms*—Federated Learning, Sequential Learning, Non-IID Problem, Catastrophic Forgetting, Internet of Things.**

I. Introduction

Federated learning (FL) is a practical solution for addressing restrictions on collecting user data for training machine learning models. It involves multiple entities collaborating to produce models, with the main advantage being user privacy protection (Figure 1). This allows training machine learning (ML) models in restricted environments, such as under the General Data Protection Regulation policy in Europe, and in healthcare [1], vehicular [2], [3], and IoT [4].

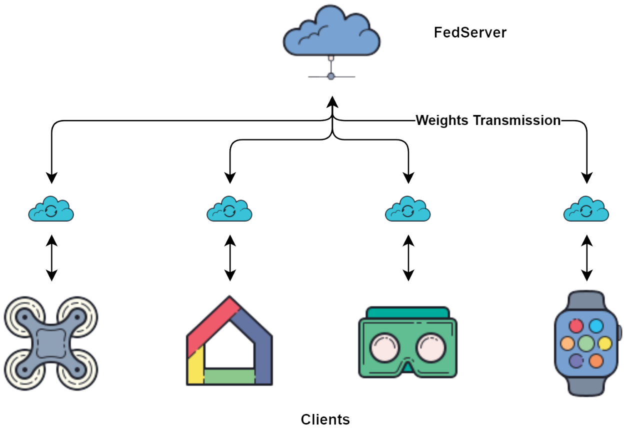


Fig. 1: Federated Learning Architecture.

Despite its advantages, federated learning faces challenges in resource requirements [5] and data distribution [6], which are particularly pronounced in IoT environments. FL imposes demands on IoT devices in terms of processing power, energy consumption, and resource allocations, limiting its performance in resource-constrained IoT devices. Furthermore, unlike centralized approaches where data is distributed independently and identically, federated learning scenarios in IoT often have uneven data distribution between devices. The Non-Independent and Identically Distributed (non-IID) nature of the IoT data

M. Arafeh is with the Department of Software and IT Engineering, Ecole de Technologie Superieure (ETS), Montreal, QC, Canada. He is also with the Department of ML, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE, as well as the Artificial Intelligence & Cyber Systems Research Center, Lebanese American University, Beirut, Lebanon (e-mail: mohamad-arfah.dabberni.1@ens.etsmtl.ca).

A. Hammoud is with the Department of Electrical Engineering, Ecole de Technologie Superieure (ETS), Montreal, QC, Canada. He is also with the Department of ML, Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE, as well as the Artificial Intelligence & Cyber Systems Research Center, Lebanese American University, Beirut, Lebanon (e-mail: ahmad.hammoud.1@ens.etsmtl.ca).

M. Guizani is with the Mohammad Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE (e-mail: mguizani@gmail.com).

A. Mourad is with the KU 6G Research Center, Department of CS, Khalifa University, Abu Dhabi, UAE. He is also with the Artificial Intelligence & Cyber Systems Research Center, Lebanese American University, Beirut, Lebanon (e-mail: azzam.mourad@lau.edu.lb).

H. Otrok is with the Center of Cyber-Physical Systems (C2PS), Department of CS, Khalifa University, Abu Dhabi, UAE (e-mail: Hadi.Otrok@ku.ac.ae).

H. Ould-Slimane is with the Department of Mathematics and Computer Science, Universite de Quebec a Trois-Rivieres (UQTR), Canada (e-mail: Hakima.Ould-Slimane@uqtr.ca).

Z. Dziong is with the Department of Electrical Engineering, Ecole de Technologie Superieure (ETS), Montreal, Canada (e-mail: zbigniew.dziong@etsmtl.ca).

C. Wang is with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China (e-mail: wangchd3@mail.sysu.edu.cn).

D. Wu is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China (e-mail: wudi27@mail.sysu.edu.cn).

negatively affects federated learning performance in terms of accuracy and convergence [7], increasing the time and cost of completing training tasks.

A potential solution for the aforementioned problem is explored in [8], where the authors propose using pre-trained models such as mBERT and M2M-100 for natural language processing tasks in FL contexts. Their results were highly accurate, as they managed to achieve outcomes similar to those of centralized models. However, a major issue stems from the initialization phase, where the model is trained on a server for several rounds before its deployment to FL clients. Such training requires access to data on the server, which is often not feasible due to data dynamicity or the unavailability of data in privacy-sensitive cases, such as medical records. Another technique was presented in [9] to lessen the effect of non-IID using a warmup model trained from the clients' data. In that work, clients collaborate by sharing a portion of their raw data with the server to train an initial model, which is later used for subsequent rounds. Such an approach has demonstrated success in mitigating the challenges of non-IID scenarios. However, it requires clients to share their raw data, which is inadequate for an FL approach and a breach of the client's privacy. Several other approaches follow a similar structure, wherein the initial model shared with clients is partially trained rather than starting from random parameters, as shown in a recent study [10]. While the authors demonstrate the positive impact of pre-training and its effectiveness on FL, the need for initial data remains a significant challenge for this approach.

To date, none of the existing methodologies in the literature have specifically aimed to eliminate the necessity of clients sharing sensitive data with the server while also addressing the dynamic challenges introduced by the ever-evolving landscape of Federated Learning environments. The main objective of this paper is to find a scheme that could address the Non-IID problem without affecting the paradigm's performance while preserving its characteristics. To this end, we propose Warmup-based Federated Sequential Learning (WFSL), a novel warmup-based scheme based on a sequential FL architecture. The architecture of our system involves a server and multiple client devices collaborating in a federated learning framework. The workflow begins with a sequential initialization phase, followed by the federated learning rounds. In this scheme, we introduce (1) an initial pre-FL step in which the server selects clients to train in sequence for a set of cycles (rounds), generating the initial model. As such, our approach aims to aid the proceeding federated learning rounds to achieve faster convergence and higher accuracy by leveraging the capabilities of the initial model pre-trained parameters. Additionally, we propose (2) a regulator algorithm designed to address the substantial shifts in weights caused by the catastrophic forgetting problems [11] that commonly occur when a neural network is used to learn a sequence of tasks. This issue similarly appears when we try to learn from clients sequentially; the learning of the later clients may degrade the performance of the models learned for the earlier ones.

Moreover, we enhance the regulator with (3) a divergence-aware client selector, which seeks to diversify weights in each training cycle while safeguarding the model against extreme weight divergence and catastrophic forgetting problems. The selector employs a combination of k-means clustering, quality assessment, and weight analysis to identify the most optimal subset of clients, which can improve the training quality during the sequential phase. Moreover, due to the complex nature of our selector, we have implemented a genetic algorithm (GA) to improve the efficiency of the client selection process. This approach aims to reduce the selection time while maintaining performance within a predetermined range, adapted to the complexity of the dataset.

After completing the initialization phase, the federated learning phase begins using the initial model generated earlier. This model serves as a global model distributed to FL clients, aiming to reduce the adverse effects of their non-IID data.

Our main contributions can be summarized as the following:

- Addressing the weight divergence issue by creating a warmup model trained sequentially with clients. With the refined sequential aggregation, we mitigate bias and promote the generation of balanced initial model weights.
- Introducing a regulator algorithm activated after each sequential training phase. Such an algorithm addresses both catastrophic forgetting and the absence of shuffling issues.
- Proposing a client selection (CS) algorithm to select the most suitable combination of clients to run in sequence for each training cycle. Using clustering and genetic algorithms, we built our CS to address sequential weaknesses such as catastrophic forgetting and longer execution times by diversifying the weights of selected clients while ensuring a certain level of model divergence among them.
- Conducting empirical experiments on MNIST, KDD, and CIFAR10 datasets. The results showcase the efficiency of our approach in building the initial model and its resilience against catastrophic forgetting compared to the Elastic Weight Consolidation (EWC) algorithm. Additionally, we demonstrate the impact of employing our sequential approach prior to the federated learning stage, which leads to achieving higher accuracy more quickly.

The remainder of the paper is structured as follows: The literature review in Section II discusses current approaches in federated learning with a non-IID environment, focusing on initialization approaches and their limitations. Section III details the non-IID problem in federated learning. Our proposed scheme, including federated learning architecture and our regulator addressing catastrophic forgetting, is presented in Section IV while Section V describes our client selection algorithm for reducing execution time. The experimental results are presented in Section VII. Finally, Section VIII concludes the paper and summarizes our findings.

II. Literature Work

This section outlines various approaches addressing the non-IID clients in FL while showing their limitations. In the context of shifting model weights, the authors in [12] propose a Federated Learning Framework Robust to Affine distribution shifts (FLRA). Such an approach assumes that each participant's data undergoes distinct transformations that deviate in model weights from the base distribution. As a solution, the authors propose a game-theoretic optimization to determine the global model by minimizing localized weight divergences. However, one key challenge lies in the feasibility of calculating participant correlation parameters, which might not be possible in most cases.

In parallel, an approach proposed in [9] involves initializing federated learning with a pre-trained model. The server model's gradient trains on raw data shared by the clients to localize the weight divergence. Furthermore, the collected data is shared with future clients to reduce the impact of weight divergence. However, this approach mandates the server to possess raw data portions from participants while sharing them with others, which breaches their privacy. Given the significance of model initialization, the authors in [10] analyzed its impact on the overall federated learning process. The authors show the positive impact of addressing the issues of non-IID clients while also improving convergence and accuracy. Moreover, the work emphasizes the benefits of following such a mechanism due to its possible adaptability to existing approaches. However, while such a mechanism is mostly positive, the mentioned approaches assume data is available on the server for the pretraining procedure, which is not always the case considering the dynamic nature and scale of the data.

In another study [13], the authors present weight divergence as a drifting problem. Their observation highlights that biased client training produces a skewed model with a local weight drifting aligned to their data distribution. Their solution introduces a control variable, mirroring the global model's shape, to mitigate local weight drift. Each participant utilizes the variance between their control variable and the centrally received one for countering the local model weight drift. The authors in [14] propose an enhanced aggregation algorithm called Federated Learning with Matched Averaging [14] (FedMA). Following this approach, the algorithm seeks the optimal arrangement of weights in each ML model layer rather than directly averaging model weights. In [15] the authors proposed a hierarchical clustering scheme for federated learning. Their clustering techniques measure the distance between the model weights and group clients. Hence, they put similar clients within the same group, reducing their non-IIDness. In such a scenario, each cluster is considered a separate federated learning process. Google has proposed a similar architecture integrating the federated learning of cohorts "FLOC" strategy to employ models instead of raw data, which limits advertisers' access to users' personal information. The main issue with these works is the structural changes in the federated learning scheme. For instance, only specific cases can endure the separate model for each cluster.

In [16] the authors recently proposed a sequential-based approach for federated learning addressing its non-IID limitation. Their approach employs a clustering scheme where clients are grouped randomly or based on specific parameters such as location. Subsequently, Within each cluster, training is performed sequentially, followed by aggregation between clusters. However, a significant challenge of this approach lies in its susceptibility to the issue of catastrophic forgetting. Moreover, the aggregation step between clusters may introduce biased model aggregation if the data among randomly clustered clients exhibits non-IID characteristics [7].

Various approaches have been proposed to address catastrophic forgetting problems. For instance, the authors in [17] proposed Elastic Weight Consolidation as a solution for catastrophic forgetting. They primarily utilize a Fisher information matrix to analyze the importance of parameters in the current task while aiming to protect these parameters from extreme shifts in the subsequent task. However, such a mechanism calculates the information matrix following each training epoch, significantly extending the training procedure. Moreover, the characteristics of federated learning, such as the dynamic nature of clients' data size and impact, diminish the effectiveness of the EWC application.

In conclusion, the surveyed approaches for handling non-IID challenges in federated learning demonstrate various techniques, from addressing weight divergence to proposing novel aggregation algorithms or training workflows. While improvements are noticeable in these approaches, factors like privacy, feasibility, and potential problems such as catastrophic forgetting remain, compelled us to seek alternative and more effective solutions in this domain.

III. Problem Illustration

In the context of federated learning with non-IID data, the objective is to train a global model w across K clients, each possessing its local dataset of $C = \{0, 1, ..., C\}$ classes. The focus is to effectively address challenges posed by non-IID data distributions without the need for central data aggregation.

For each client k, the training objective is expressed as follows:

$$\min_{w} \sum_{k=1}^{K} \sum_{i=1}^{C} p(k, y = i) \mathbb{E}_{x|y=i}[\log f_i(x, w)] \tag{1}$$

where in this equation, we aim to minimize the loss incurred on client k's data distribution, where $p(k, y = i)$ represents the probability of class i on client k.

Furthermore, federated learning introduces aggregation to the process in order to enable parallel training. After each client k performs local updates on its model parameters, we aggregate the updates by computing the average. The aggregation process can be expressed as:

$$w^{(t)} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{(t)} \tag{2}$$

where $w^{(t)}$ represents the global model parameters after aggregation at iteration t, $w_k^{(t)}$ represent the clients model after training, and $\frac{n_k}{n}$ signifies the averaging operation which takes into consideration the clients contribution based on their sample size. The aggregated model benefits from the collective knowledge of all clients without sharing their raw data with a third party the server.

The iterative update process for client k is detailed as follows:

$$w_k^{(t)} = w_k^{(t-1)} - \eta \sum_{i=1}^{C} p(k, y=i) \nabla \mathbb{E}_{x|y=i}[\log f_i(x, w^{(t-1)})] \tag{3}$$

where $w_k^{(t)}$ represents the updated model parameters for client k at iteration t after the local training process. The learning rate η determines the step size for parameter updates, ∇ calculates the gradient to guide these updates based on client k's data distribution, and f_i represents the loss function.

The main issue arises from clients' skewed training data distribution during the execution of Equation 3. As a result of training on such data, the model loss and the updated gradient become biased toward the training client. Such occurrences are visually illustrated in Figure 2, where each line represents an individual client weight after completing local training. To facilitate visualization, we employ principal component analysis to compress and flatten the client weights, enabling their representation in a two-dimensional graph. Applying Equation 2 to these weights results in the global model losing many of its training parameters due to a biased average aggregation. This is evident in Figure 2, where the black plot represents the global model weights. Hence, the aggregation method emerges as a significant contributor to the inefficiency of federated learning, particularly when confronted with non-IID client data, affecting the global model's accuracy.

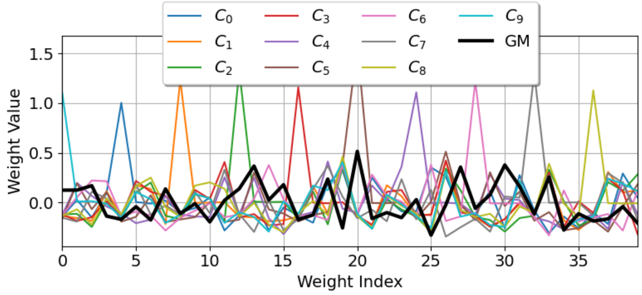


Fig. 2: Weight Divergence between non-IID clients and their Global Model

IV. Proposed Scheme

Recent federated learning developments aim to enhance efficiency in an environment where control over clients' data is unreachable. Despite the aggressive developments in the field, the non-IID issues of the clients limit their applicability due to the increase in cost and time of building ML models. Therefore, a solution should take into account the non-IID behaviour while keeping data privacy intact. Hence, we propose an approach that uses the initialization phase to build a model capable of mitigating the clients' non-IID issues.

The primary objective of this work is to initialize federated learning with a partially trained model to mitigate the impact of non-IID clients during the initial startup phase, which is particularly susceptible, without relying on any existing data on the server side. However, in order to achieve such results, it is important to lessen the impact of the aggregation step during the initialization phase. As indicated in [9], [18], [19], aggregation on non-IID clients significantly contributes to weight degradation due to aggregating biased weight parameters. Moreover, unlike current approaches that require data on the server for the pretraining procedure, our approach maintains the client's data locally on their devices without sharing it with the server.

Our scheme involves clients training sequentially during initialization while coordinating with the server. Client privacy remains intact, as clients only communicate the model to the server. Nonetheless, adopting the sequence approach exposes the model to a degree of catastrophic forgetting depending on the data divergence between clients. Hence, we enhance our scheme with a regulator algorithm to reduce this issue's impact.

This section presents our proposed architecture addressing the aforementioned problem. Moreover, we present a component diagram (Figure 3) illustrating the key components of our architecture.

A. Federated Learning Components

Figure 3 shows a diagram illustrating the architecture components. Our proposed scheme comprises two main blocks: clients and the server, each with defining modules. We explore each module, focusing on two key components: the regulator (client-side) and the initialization (server-side) modules.

Each client block has three modules: *Object Transfer*, *Trainer*, and *Regulator*. The *Object Transfer* module manages communication within both client and server blocks. This involves tasks like model synchronization and transition of training parameters, such as epochs, model configuration, and learning rate. The *Trainer* selects a framework corresponding to the client's platform, converts weights into an ML model, and handles the training process. The *Regulator* module is specific to initialization. It introduces a specialized algorithm into the training process, effectively addressing catastrophic forgetting and sequential learning challenges. A more detailed explanation on this topic can be found in section IV-C.

The server block comprises four modules: *Object Transfer*, *Client Selector*, *Aggregator*, and *Initializator*. Our scheme extends the traditional FL architecture and introduces an initialization step to produce a pre-trained model. This step mainly seeks to enhance the initial accuracy by starting from a boosted model capable of mitigating the impact of non-IID clients [9]. Further details are provided in Section IV-B.

Regarding the remaining modules in the server block, the *Client Selector* and the *Aggregator* are specific to the federated

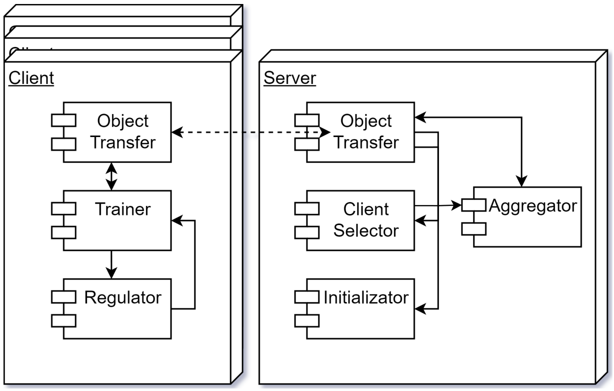


Fig. 3: WFSL Architecture.

learning process. The role of the *Client Selector* is to choose a portion of the clients. This is mainly done to alleviate the client and server burden. Furthermore, it is essential to recognize that assuming all clients are consistently available for training is challenging [7]. Then, the *Aggregator* module combines the received weights to create the global model using FL aggregation algorithms such as FedAVG or FedSGD [14].

Figure 3 illustrates communication between modules through arrows. For instance, *Object Transfer* facilitates communication between the server and clients by exchanging weights and training parameters. The *Trainer* on the client side receives weights from the *Object Transfer* module and conducts further training using local data. On the server side, the *Object Transfer* receives selected clients from the *Client Selector* modules, dispatching the most recent weights—either from the Initializator after the initial phase or from the *Aggregator* during the FL process.

B. Initializator Module

Our scheme leverages the initialization step to build a pre-trained model. During this stage, a single client is selected to train a model using his local data. Upon completing its task, the client transmits the update to the server, which, through random or algorithmic selection, elaborated in section V-C, designates another client to continue training on the same model. The server determines the termination of the initialization stage by using all or several selected clients to complete the training task for a specific number of iterations, given a cost or a specified amount of time to finish the initialization. This approach ensures that the resulting initial model is well-trained across all its weights, regardless of whether clients exhibit non-IID properties. Choosing a sequential strategy over a parallel one avoids the necessity for aggregation among clients, a primary factor contributing to weight divergence and accuracy degradation in federated learning.

Furthermore, our scheme employs the initialization phase solely as a one-time setup instead of per-round execution. This decision is primarily driven by the considerable time cost incurred from mandatory synchronization among clients and the server. In traditional federated learning approaches, clients operate in parallel with the server, reducing the overall time required to complete a given task. However, parallelism necessitates an aggregation mechanism to combine the models of parallel clients. By opting for an initialization-based approach, we eliminate the need for parallelism and aggregation during this phase, enabling a time-controllable scheme for the initial buildup of the ML model.

C. Regulator Module

During the sequential phase and when the model is transferred to a different client, the weights are updated to fit the new biased information. This process can cause the model to lose its ability to accurately predict or classify data it was previously trained on, especially in our non-IID case, where the new client's data is significantly different from the old ones. Such concerns can be identified as catastrophic forgetting problems [11]. To address this issue, we propose a regulator module that can be applied during the initialization phase without increasing execution time. Such a module tries to mitigate the impact of catastrophic forgetting and the deterioration in model accuracy. Our proposed algorithm tries to maintain the knowledge of the previous clients by integrating it into the model generated by the new clients after completing their training procedure. In the following, we assume that the previous client possesses all the information acquired from the one preceding it, and our algorithm aims to uphold this information by incorporating it into the newly trained weights. Hence, the algorithm performs a form of dilution on the old clients' weights while merging them into the new ones. This is formalized in the following Equation:

$$w_{c+1}^{i'} = (1-\alpha)w_{c+1}^{(i)} + \alpha w_c^{(i)} \tag{4}$$

where α represents the regulation rate, $w_c^{(i)}$ the model weights that are trained directly by the previous client and sent to the current client, $w_{c+1}^{(i)}$ refers to the new model after being trained by the current client.

The main motivation behind our proposed architecture is the fact that we preserve the characteristics of the federated learning scheme proposed by [18]. Furthermore, we extend the traditional FL architecture with our initialization strategy, combined with our regulator components aims at addressing the non-IID limitation of the training clients.

V. Sequential Client Selector

A. Problem Definition

In this work, we propose an initialization phase that does not rely on any existing form of raw data. This involves a sequential approach that circumvents aggregation limitations to enhance the federated learning process, albeit at the expense of longer execution time. Unlike parallel execution of clients' local training, sequential training mandates clients to synchronize with the server to update the global parameter when a client finishes its local epochs. Such a mechanism significantly increases execution time, particularly considering all clients are equally important and thus must be included in each cycle to achieve optimal results. However, by considering sequential learning properties, it is possible to tailor a client selection strategy to enhance the regulator's efficiency and reduce its lengthy execution time. To this end, three crucial properties can guide us in reducing execution time by reducing the number of participating clients during the initialization phase without compromising the model performance. These properties are elaborated below:

- Non-IID definition: Since the clients are essentially continuing the work rather than updating their parameters from an aggregated model, as is typically done in federated learning, the definition of non-IID may differ. In this scenario, clients simulate a level of central training where non-IID describes the data as a whole. Thus, having more diversified data with more samples becomes more important than having clients with similar weights training together.
- Reduction of weight divergence impact: This is primarily achieved by refining the aggregation phase. Consequently, selecting clients with divergent samples in either size or model parameters will not significantly impact the procedure as in federated learning.
- Catastrophic forgetting: This is induced by the sequential nature of training, wherein the learning process for clients working in sequence can deteriorate the performance of models learned from earlier clients [17].

Building upon the mentioned properties, we aim to develop a client selector capable of achieving comparable performance to selecting all clients but while choosing fewer ones, with the main objective of reducing the overall execution time. Subsequent sections (V-B and V-C) offer deeper insights into our proposed algorithm.

B. Formulation

In the system model, we assume having n clients in total, represented by the set N. In sequential training, each training cycle begins with selecting a subset of m clients, chosen to train on a model provided by the server sequentially. The main reason that hinders us from selecting all clients for the sequential phase is the time required to complete such a task. For simplicity, we represent the selected client subset by Ch, where $Ch = l_0, l_1, ..., l_i, ..., l_n$ such that l_i is a binary operator indicating whether the client i is included in the selected subset if its value is 1. The problem turns into choosing Ch' of size m. During the selection, the server takes into consideration clients' trained parameters in addition to their sample size to build a subset submitting to the following objective:

- Maximizing selected client diversity: which benefits from the reduced impact of non-IID implications following a sequential-based approach.
- Minimizing selected client divergence: designed to limit the extent of the catastrophic forgetting impact. While we attempt to diversify the learning parameter, we also seek to maintain a level of similarity to avoid the model going into the range of catastrophic forgetting.
- Maximizing learning quality: such as we focus in our case on the client's sample size. As such, clients with more samples will have a slight boost of being selected.

Following up on the first objective, maximizing diversity, we aim to cluster clients based on their similarity index. Therefore, tailoring the selection process to focus on diversifying training parameters by ensuring each client's subset contains clients from every cluster. This way, we can selectively engage different clients in terms of features to include them in the sequential learning process. This objective is presented in equation 5

$$\begin{aligned} &\textbf{Objective:} \\ &\textbf{O1:} \quad \text{Minimize} \sum_{i=1}^{n} \sum_{k=1}^{K} z_{i,k} \|w_i - \mu_k\|^2 \\ &\textbf{Constraint:} \\ &\sum_{k=1}^{K} z_{i,k} = 1, \quad \forall i \in \{1, \ldots, n\} \end{aligned} \tag{5}$$

where n is the number of clients, K represents the number of clusters, w_i is the model parameter for client i, μ_k is the centroid of data points in cluster k, and $z_{i,k}$ is a binary indicator that determines whether the data point w_i is assigned to the cluster k. The goal is to find a configuration of clusters where clients have the minimum squared Euclidean distance between their model parameters and the cluster centroid. This equation is subject to a constraint ensuring each client belongs to exactly one cluster. The minimization is conducted over compressed weights via Principal Component Analysis (PCA) to enhance optimization and reduce computation time. The KMeans algorithm is utilized for practical implementation due to its fast execution and reliable results. We denote the set of clusters $Z = Z_1, Z_2, ..., Z_k$ is obtained after executing the KMean algorithm, and Z_i denotes the set of clients in a single cluster.

Building on the first objective $O1$, we assume that clients selected for the subsequent objectives and constraints are those identified in the preceding phase.

Objective $O2$: maximizing distant clients' similarity. This objective primarily focuses on reducing the impact of catastrophic forgetting and improving learning efficiency as provided in equation 6.

$$\text{SI}(W_i, W_j) = \frac{\sum_k W_{i,k} \cdot W_{j,k}}{\sqrt{\sum_k W_{i,k}^2} \cdot \sqrt{\sum_k W_{j,k}^2}} \tag{6}$$

where SI represents the similarity index function utilizing the cosine similarity method. W_i and W_j denote the model weights of two specified clients to assess their similarity. Cosine Similarity yields values within the $[-1, 1]$ range, with higher values indicating greater similarity between the given inputs. Thus, our objective is to maximize the similarity between chosen clients as outlined in equation 7

$$\textbf{O2:} \quad \max\left(\sum_{i,j}^{Ch'} l_i \cdot l_j \cdot \text{SI}(W_i, W_j)\right) \tag{7}$$

Objective $O3$: maximizing the average data size si of clients selected to improve the learning quality of the training cycle following equation 8.

$$\textbf{O3:} \quad \max\left(\sum_{i}^{Ch'} s_i \cdot l_i\right) \tag{8}$$

Furthermore, to optimize the effectiveness of the preceding objectives, it is essential for a subset to adhere to the following constraints:

C1 - Selection of m clients: aligned with our primary objective of reducing overall execution time, this constraint aims to minimize the number of clients participating in sequential training while, in parallel, other objectives focus on maintaining high accuracy levels. $C1$ is presented in equation 9 as follow:

$$\sum_{i} l_i = m \tag{9}$$

C2 - Non-repeating selection pool: as such, a selection should not contain the same client twice to avoid the possibility of exhibiting a bias towards a specific client's data and not overwhelm the same client with the training burden. $C2$ is presented in equation 10

$$l_i \cdot t_i = 1 \quad \forall i \in Ch \tag{10}$$

where t_i indicates the number of times client l_i is selected for training.

C3 - Increased diversity: forcing selection to include as much as possible an equal number of clients from each cluster given by objective O_1. $C3$ is detailed in equation 11:

$$|\sum_{i}^{n} z_{i,a} - \sum_{j}^{n} z_{j,b}| <= 1 + ||Z_a| - |Z_b||\,, \qquad \forall Z_a, Z_b \in Z \tag{11}$$

where $\sum_i^n z_{i,a}$ count the total number of clients selected from cluster a, and $\sum_j^n z_{j,b}$ count the total number of clients selected from cluster b. Following this equation, our aim is to ensure that the difference in the total number of clients selected from each cluster can only exceed that of others by 1[1] unless the clusters differ in size, as indicated by $(|Z_a| - |Z_b|)$.

C. Genetic Algorithm

Following the aforementioned objectives, client selection in sequential learning involves seeking a compatible subset of clients while navigating over contradictory and constrained objectives with properties of combinatorial decision-making. This process is fundamentally NP-hard due to the exponential number of possible solutions, similar to the combinatorial explosion seen in the Knapsack problem. Equation 12 presents the solutions obtained when selecting a specific number of clients q from each cluster Z of total size K.

$$P = \binom{|Z_1|}{q} \times \binom{|Z_2|}{q} \times ... \times \binom{|Z_k|}{q} \tag{12}$$

Identifying an optimal solution involves combinatorial iteration, which requires substantial time. This, along with the need to balance objectives such as maximizing diversity and minimizing divergence, introduces multi-objective optimization, which is notably challenging to solve due to opposing goals.

Given the complexity of the objectives and the necessity to reduce selection time, we opt to optimize our search using Genetic Algorithms. This approach excels in locating a suboptimal solution within a manageable timeframe. Moreover, genetic algorithms aid in efficiently exploring a substantial portion of quality solutions, leveraging clients' weights to exploit our sequential approach properties. Algorithm 1 outlines our genetic implementation for client selection, while subsequent sections provide comprehensive insights into population setup, fitness calculation, and evolutionary processes.

Population: Genetic algorithms operate on chromosomes composed of a series of genes. In our context, each gene corresponds to an individual client. Therefore, the primary goal of the genetic algorithm is to identify the most optimal chromosome $Ch' = l1, l2, ...ln$ representing n clients, maximizing the specified fitness function Fn outlined in a subsequent paragraph. Additionally, to minimize the occurrence of invalid chromosomes and enhance the generation of high-quality solutions, we enforce constraints $C1$, $C2$ & $C3$ during population initialization, whether it be during the initial setup or any subsequent generation of new chromosomes.

Fitness calculation: In the evaluation phase, we quantify each chromosome according to the objectives outlined in section V-B by using the following Equation:

$$Fn(Ch') = W1 \cdot norm_1(O2(Ch')) + W2 \cdot norm_2(O3(Ch'))$$

where $W1 + W2 = 1$ denote the weights assigned to objectives $O2$ and $O3$. $Fn(Ch')$ represents the score computed for chromosome Ch'. Our objective is to maximize Fn by evaluating the compatibility of client weights through the cosine similarity index in objective $O2$, while considering the aggregated sample size of the chromosome in objective $O3$. Additionally, we employ the $norm$ function, as described in equation 13, to project both objectives onto the same dimension, ensuring that their influence is primarily regulated by the values of their respective weights.

$$norm = \frac{score - min\_score}{max\_score - min\_score} \tag{13}$$

[1] if m is different than K, this forces the system to choose an uneven number of clients from the clusters. Therefore, we force the selection to allow for only 1 client difference between the least and the most selected clients from clusters.

```
Algorithm 1: Genetic Algorithm Implementation
input : $genes = clients$
$population \leftarrow$
  $initial\_population(genes, population\_size,$
                              $chromosome\_size)$
$solution \leftarrow \emptyset$
$n\_iter \leftarrow 0$
while $n\_iter < max\_iter$ do
    $scores \leftarrow fitness(population)$
    $solution \leftarrow argmax(scores)$
    $population \leftarrow wheel(population, scores)$
    $children \leftarrow \emptyset$
    for $i$ in range(start=0, stop=$|population|$, step=2)
     do
        $child1 \leftarrow population[i]$
        $child2 \leftarrow population[i + 1]$
        for $c$ in $crossover(child1, child2, r\_cross)$ do
            $mutation(c, genes, r\_mut)$
            $children \leftarrow children + c$
    $n\_iter \leftarrow n\_iter + 1$
return $solution$
```

Evolution: We start by evaluating the initial population of chromosomes using the designated fitness function fn. Subsequent to evaluation, we utilize the Roulette Wheel method for chromosome selection, taking into account their fitness scores and following the specified constraints. Crossover and mutation follow separately when a randomly generated float number between 0 and 1 exceeds the specified rate. Crossover involves the exchange of segments between two chromosomes, enabling the exploration of new solutions, while mutation introduces diversity by randomly altering chromosome genes. These processes iterate until a predefined stopping criterion is met, which can be specified to regulate the time allotted for the Genetic Algorithm. The optimal outcome is a chromosome representing a collection of clients aligned with the stated objectives.

VI. Architecture Workflow

The architecture of our system involves a server and multiple client devices collaborating in a federated learning framework. The workflow begins with a sequential initialization phase, followed by the federated learning rounds. Algorithm 2 shows the detailed execution of the workflow, while Figure 4 presents a detailed UML sequential diagram illustrating the architecture's progression, encompassing the integration of both phases.

Building upon the representation depicted in Figure 4, the scheme starts with the initialization phase, comprising the following procedures:

1) The server starts a training cycle, preselects the required clients to train based on the provided client selector and places them in an idle buffer.
2) The server assigns a client to initiate the training process and sends it the latest model weights.
3) The client proceeds with the training using its local data, updating the received model weights.
4) After training, the client returns its updated weights to the server.
5) The server selects a new client from the idle buffer.
6) The server forwards the last received model's weights to a new client for training.
7) Once training is complete, the client employs the regulation algorithm following Equation 4, integrating the new model weights with the ones received from the server.
8) Once weights regulation is complete, the client returns the updated model to the server.
9) The server receives the new model and caches it. Furthermore, it becomes the new reference for the next client selected by the server.
10) Steps 4 to 8 are repeated iteratively till the stopping criteria are reached.

In this context, the stopping criteria can be determined by iteratively looping on a predefined number of clients for a specific number of rounds. Hence, controlling the duration required to complete the initial phase is possible. Expedited progression can be achieved by involving a limited number of clients for a marginal number of rounds. Alternatively, getting a better initial model can be achieved by increasing the initial participating clients while increasing the number of initial rounds. Ultimately, it depends on how much cost can be allocated into the initial phase and the minimum threshold deemed necessary prior to the start of federated learning, which mainly relates to the non-IID level between clients and the complexity of the model and its dataset.

```
Algorithm 2: WFSL Initializer Algorithm
Require: clients $C$, model_configs, cycles,
         regulator_factor
return weights

server.trained_clients ← $\emptyset$;
for $r$ in range(cycles) do
    $l$ = server.select($C$);
    server.last_model ← None;
    for each client $c$ in $l$ do
        $c$.model ← download();
        $c$.train();
        $c$.model ← regulator($c$.model, server.last_model,
          regulator_factor);
        server.last_model ← $c$.upload();
        server.trained_clients ← trained_model;

weights ← regulator(trained_clients);
return weights;
```

Following the initialization phase, the federated learning rounds start. Each round follows a similar sequence of earlier steps, involving client training and model transmission. However, clients work in parallel during federated learning while the weights are combined using an aggregation algorithm.

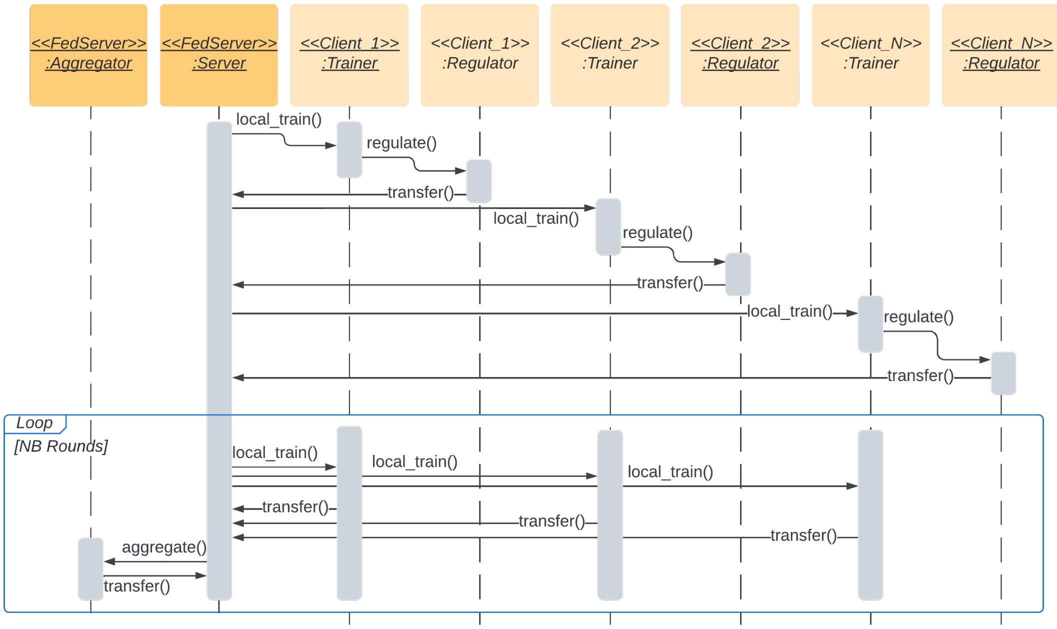


Fig. 4: WFSL Workflow.

The overall procedure is the following:

1) The server assigns the model from the initial phase as the federated learning global model.
2) The server selects a set of clients from a pool of available clients and sends them the global model.
3) Each client trains, in parallel, the model using its local data and transmits the weights back to the server.
4) The server applies an aggregation algorithm to combine the received weights.
5) Step 2-4 repeats till the stopping criteria are reached.

The framework stops the federated learning process when it reaches a preset number of rounds. Regarding the aggregation function, the framework employs the traditional FedAVG as shown in Equation 2.

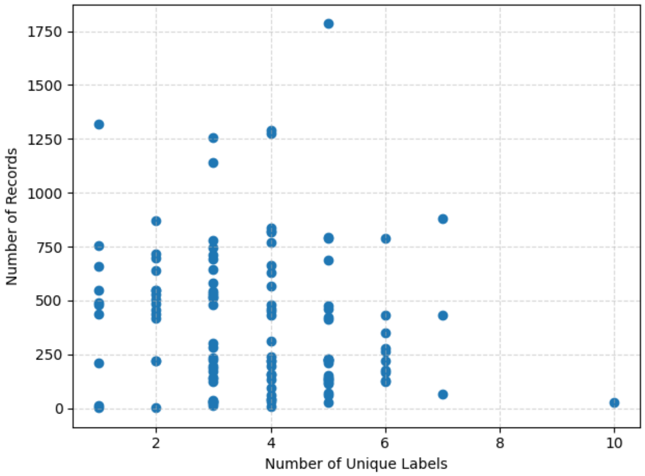


Fig. 5: Scatter Plot of the Client's Data Following a Dirichlet Distribution

VII. Experimental Results

This section outlines the empirical findings of our WFSL scheme across both the initialization and federated learning phases. In the initialization phase, our primary comparison is with Elastic Weight Consolidation (EWC), demonstrating how our approach stands in addressing the issue of catastrophic forgetting in sequential learning. Additionally, we evaluate our method against the Shared Raw approach [9] due to how promising it is to mitigate the non-IID issue. Moving to the federated learning phase, the benchmark includes both the Shared Raw approach and traditional FedAVG (basic) approach as baselines compared to ours showing the overall benefits in terms of achieved accuracy and convergence to the federated learning when boosted by sequential learning with regulators.

We conducted the initialization phase experiments on MNIST and KDD datasets. The MNIST dataset consists of 28*28 pixel images of handwritten digits from 0 to 9. It contains 60,000 train samples and 10,000 test samples. On the other hand, the KDD dataset comprises 125,973 records associated with 41 distinct features. We employed the multi-label version, which contains 23 unique labels that indicate whether the network requests are normal or an attack while identifying the type of attack.

In the federated learning experiment, we opted for a more challenging dataset paired with a complex CNN model to comprehensively evaluate the impact of employing boosting models in terms of achieved accuracy and time cost compared

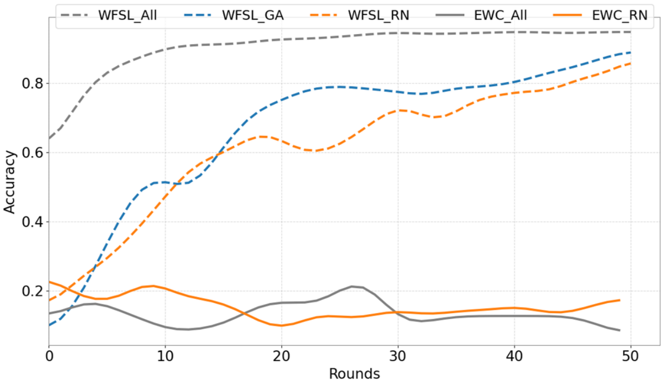


Fig. 6: MNIST Shard Accuracy/Cycle

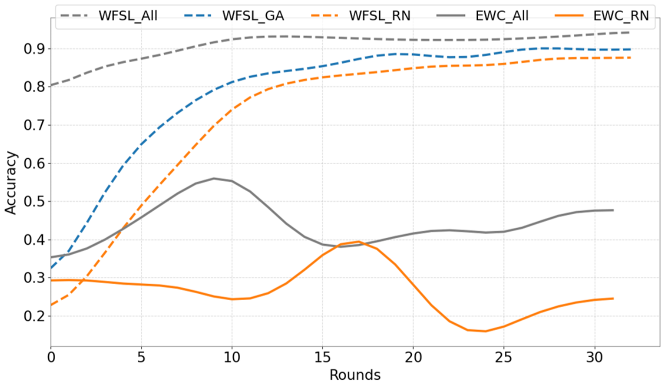


Fig. 7: MNIST Dirichlet Accuracy/Cycle

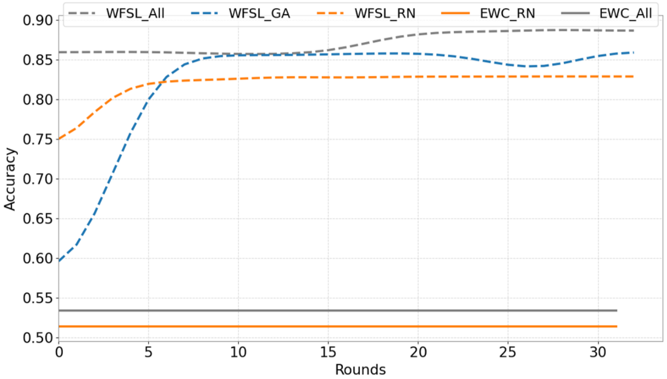


Fig. 8: KDD Shard Accuracy/Cycle

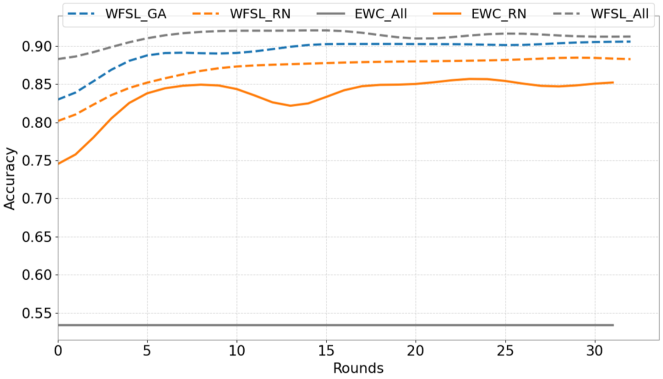


Fig. 9: KDD Dirichlet Accuracy/Cycle

to those without. Specifically, we conducted the federated experiments using the CIFAR-10 dataset with 3 layers CNN model.

Furthermore, our experimental simulations were conducted through the ModularFed framework [20]. The authors provide a framework for federated learning researchers to streamline the implementation of their approach without starting from scratch. We incorporate our approach into the framework, specifically within the initialization layer, where we incorporate the scheme algorithm responsible for generating the initial model from the participating clients. Furthermore, the framework provides the necessary tools for client simulations. Hence, clients are built based on data distributors capable of stimulating different levels of non-IID distributions.

Throughout our experiments, we opted for two distributions: Shard [18] and Dirichlet. Shard distribution requires two parameters: shard count, indicating the number of shards each client possesses, and shard size, representing the number of samples per shard. For our experiment, we set the shard count to 2, reflecting a scenario of non-IID clients. Additionally, the shard size was fixed at 400. These settings align with our use case, focusing on IoT devices with limited storage and processing capacities. Similarly, we can achieve a high non-IID scenario in the Dirichlet distribution case by selecting a small alpha value that controls client data distribution. Lower alpha values lead to a greater concentration of data in specific areas, indicating a higher non-IID environment in terms of both labels/classes and data sample sizes. Figure 5 presents a scatter plot illustrating the distribution of unique labels against the number of records allocated to each client. Through this figure, we show significant variations in data accessibility among clients. For instance, one client can access only a single label and three records, whereas another can access all ten labels, though with a limited number of records. Such variation in data distribution highlights non-IID anomalies, affecting both the diversity of labels and the volume of records available to different clients.

Parameter	**Value(s)**
Dataset	MNIST$_1$, KDD$_2$, CIFAR-10$_3$
Distributions	Shard$_{12}$, Dirichlet$_{123}$
LR	0.01$_{12}$, 0.001$_3$
Iterations	50$_{12}$, 5000$_3$
Shard Count	1
Shard Size	400
Dirichlet Alpha	0.1
Client Selectors	All, RN (10/R), GA (10/R)
Regulators	WSFL, EWC
GA Population	200
GA Iterations	50
GA Crossover Rate	0.1
GA Mutation Rate	0.05

TABLE I: Simulation Parameters

A. Initialization

This series of experiments is dedicated to evaluating the effectiveness of our regulator, focusing exclusively on the model's initialization phase compared to the EWC regulator across various iterations of client selection in each cycle. The plots legend follow the format {Regulator}_{Client Selector}. For the regulators, 'WSFL' denotes our method, while 'EWC' represents the Elastic Weight Consolidation approach. As for the client selectors, 'All' signifies the employment of all clients in each cycle, 'RN' represents random client selection, and 'GA' represents our genetic-based client selection, formulated according to the objectives outlined in Section V-B. The combinations of the plots are the following:

- WSFL_All: proposed architecture with all clients involved in the sequential training phase.
- WSFL_RN: proposed architecture with randomly selected clients involved in the sequential training phase.
- WSFL_GA: proposed architecture with our proposed client selector involved in the sequential training phase.
- EWC_All: Elastic Weight Consolidation architecture with all clients involved in the sequential training phase.
- EWC_RN: Elastic Weight Consolidation architecture with randomly selected clients involved in the sequential training phase.

Figures 6 and 7 depict the progression of accuracy over training cycles using Shard and Dirichlet distributions, respectively. As evident from the figures, our regulator significantly mitigated the impact of non-IID clients across both distributions. For MNIST shard distribution, applying the regulator to all clients allowed us to attain an initial accuracy of 95.3%, a crucial element in boosting federated learning and increasing its tolerance to newly introduced non-IID data across clients. While achieving such accuracies comes with high costs, our selection-based regulator achieves comparable accuracy levels while utilizing fewer clients. For instance, employing a GA-based selector achieves 89.2% accuracy at the end of the cycles compared to 86% for the random-based selector. Similar results are shown in the Dirichlet distribution with WSLF_ALL ending at 94.4% followed by WSFL_GA 89.2% and WSFL_RN 86.1%. In parallel, the EWC approach struggled to attain better accuracy, ending with 17% for EWC_ALL and 9% for EWC_RN, as shown in Figure 6. This is primarily due to the limited data samples distributed among clients. Unlike the traditional EWC applications, which address catastrophic forgetting in central server tasks with abundant data, federated learning presents challenges where each client possesses only a small fraction, affecting the evolution of EWC-based approaches. Following the Dirichlet distribution, some clients have access to larger sample sizes, positively impacting the overall accuracy achieved by the EWC regulator, achieving accuracies of 24% and 48% as shown in Figure 7. To conclude, the sequential approach employing 'All' as a client selector achieved the highest accuracy, closely followed by our genetic algorithm (GA) selection on both distributions. This highlights the potential for achieving similar accuracy without burdening all clients with training, which increases the processing costs, as shown in later figures.

Results similar to the MNIST dataset were obtained for the KDD dataset as illustrated in Figures 8 and 9. We notice WSFL_All reaching 95.3% and 88.75% on the Dirichlet and Shard distribution, respectively, followed by WSFL_GA 89.2% and 86.08% and WSFL_RN reaching 86.1% and 82.5%. Notably, the major difference in this experiment is given to the EWC_RN approach achieving higher accuracy values of a minimum of 52% due to its access to a larger sample size given by the KDD dataset. However, it is still noticeable how our approach outperforms the other one in terms of initializing the model.

Our second set of experiments evaluates accuracy in terms of time taken rather than cycles, as shown in Figures 10, 11, 12, and 13. These experiments were conducted within the same environment and with access to the same device capabilities. The Y-axis indicates accuracy, while the X-axis represents the cumulative time between recorded accuracies, presented in Time Units (TU), measured at the end of each cycle.

We note that EWC_All entails the highest time cost due to Fisher measurements occurring after each epoch, prolonging the experiment's duration by the number of cycles. On the MNIST dataset (Figures 10, 11), the time cost incurred using EWC_All is recorded as 0.84TU for the shard distribution and 0.38TU for the Dirichlet distribution. WSFL_ALL achieved the highest accuracy, reaching 95% and 94% in both Shard and Dirichlet distributions, respectively. However, it required a longer execution time than the rest of the WSFL approaches, taking 0.206TU on Shard distribution and 0.11TU on Dirichlet distribution due to involving all clients in the initialization training. Comparable accuracies were achieved using a genetic algorithm for client selection, yielding an accuracy of 89.1% and 90% with a significantly reduced execution time of 0.0222TU and 0.03TU compared to the other approaches. This was followed by random client selection WSFL_RN, achieving 85% accuracy in 0.0172TU for shard distribution and 86% in 0.019TU for Dirichlet. The variance in time between WSFL_GA and WSFL_RN is attributed to the genetic selection process, which requires time to generate populations and calculate fitness. However, this process was fully optimized utilizing the device's full GPU and CPU capabilities, typically available on a server with abundant resources. Furthermore, WSFL_GA time cost is further enhanced by utilizing low-cost hyperparameters for genetic algorithms. We used a population size of 200 and a maximum of 50 iterations in our simulations to enhance accuracies without incurring excessive costs.

As shown in Figures 12 and 13, results similar to the MNIST dataset were achieved with the KDD dataset. Following shard distribution 12, EWC_ALL achieved 53% accuracy in 0.65TU, followed by WSFL_ALL reaching an accuracy as high as 89% in 0.29TU, compared to WSFL_GA completing cycles in 0.0154TU while achieving 85.8% accuracy. When projecting the WSFL_GA accuracy onto the WSFL_ALL plot, we observe that 85.8% could be attained in 0.05TU, a significantly higher time cost than with GA (0.0154TU). For the Dirichlet

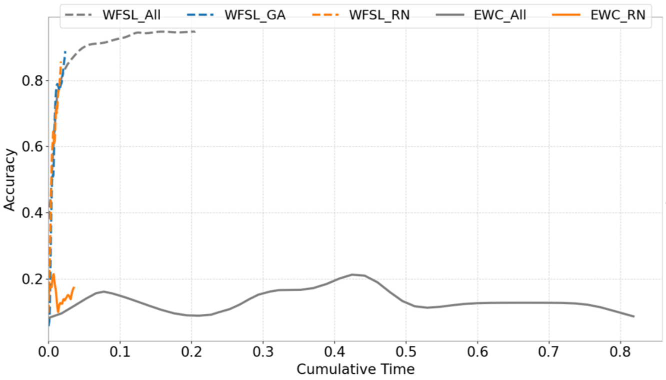


Fig. 10: MNIST Shard Accuracy/Time

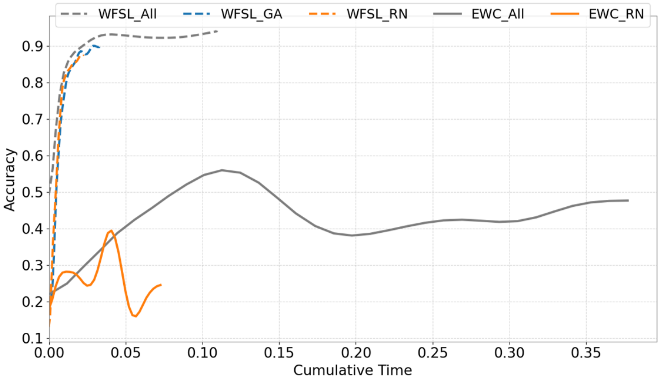


Fig. 11: MNIST Dirichlet Accuracy/Time

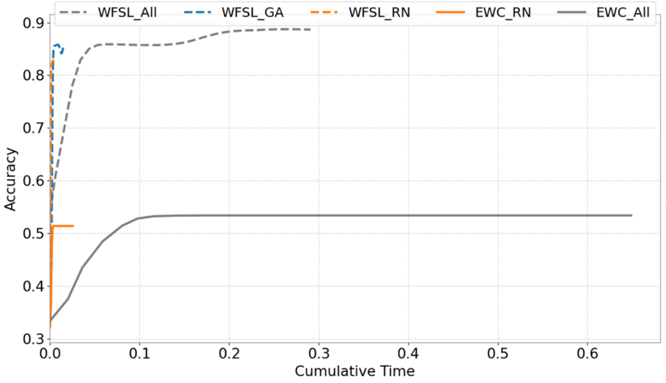


Fig. 12: KDD Shard Accuracy/Time

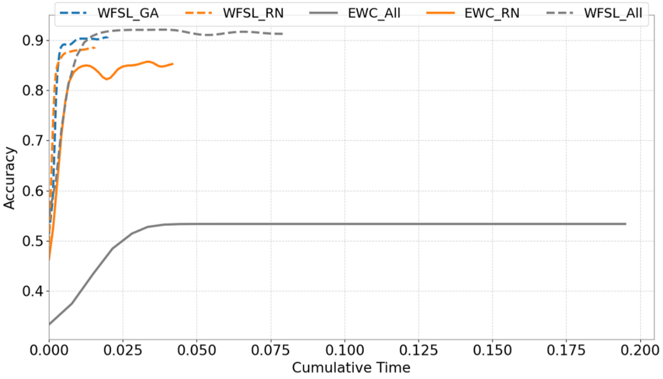


Fig. 13: KDD Dirichlet Accuracy/Time

distribution, considering all clients, WSFL_ALL can achieve an accuracy as high as 91% in 0.079TU, completing all cycles, compared to 90% for WSFL_GA completing cycles in 0.0197TU. Projecting the 90% accuracy of WSFL_ALL onto WSFL_GA reveals it can be achieved in 0.015TU compared to 0.0086TU with GA. While achieving higher accuracy faster with WSFL_ALL is possible, tracking accuracies on the server becomes challenging with limited access to the data used for testing. In these experiments, accuracy was solely used for benchmarking, assuming no test data are available on the server. This emphasizes our focus on measuring the end of the cycle rather than the time accuracy was achieved.

Regarding the Shared Raw approach, given its unique characteristics, we separately illustrate its performance on the previously mentioned datasets and distributions in Figure 14. Such an approach operates entirely on the server, utilizing 5% of each client's data to enhance the global model. Furthermore, it's worth noting that this approach is measured in rounds rather than cycles, with each round equivalent to one epoch in a machine-learning context. Therefore, assessing its time efficiency would be unfair, as it operates at the speed dictated by the server's capabilities. Hence, we present the experiment separately to evaluate the final accuracies of the collected data. After 500 rounds, most Shared Raw approaches achieved notable accuracies, such as 49% for shard distribution on the KDD dataset and 48% for both MNIST and KDD on the Dirichlet distribution by the end of the run. These accuracies aim to improve the global model parameters, enabling them to accommodate the non-IIDness of clients during the federated rounds.

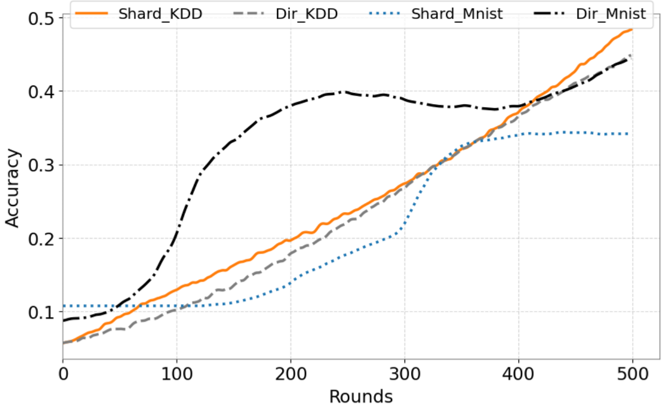


Fig. 14: Shared Raw approach [9] accuracies across both MNIST and KDD datasets

B. Federated Learning Impact

In this section, we highlight the impact of employing the initialization phase prior to federated learning using the CIFAR-10 dataset. We used Dirichlet Distribution for this experiment, allocating the 60,000 Cifar10 records across 120 clients with a

Dirichlet alpha parameter of 0.1, which indicates a highly non-IID context. Figure 15 shows the accuracy evolution in terms of cycles during the initialization and rounds during federated learning. Meanwhile, Figure 17 displays the same results but with the X-axis representing the cumulative time taken until the accuracy measurements are recorded.

The federated learning configuration includes 7000 rounds, a learning rate of 0.001, 20 epochs per round, and a batch size of 50. The WSFL approach employs our genetic algorithm to select 10 clients per cycle for the initialization phase, running for a total of 400 cycles. For the Shared Raw approach, we collected 5% of the data from all clients to train an initial model. This model, along with the collected data, was then distributed to all clients. As a result, each client starts with at least the 5% of data shared by the server, in addition to their own local data.

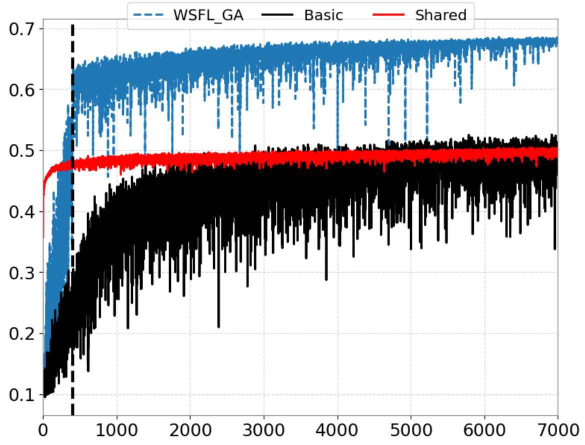


Fig. 15: Initialization Phase Followed By Federated Learning. Black Vertical Line Indicates The End of The Initialization Phase. X-Axis Represents the Cycle/FL Rounds while Y-Axis Represents the Accuracy

In Figure 15, we observe that WSFL_GA can achieve an accuracy of 56% at the initialization phase's end. Then the accuracy improved throughout the federated learning process, reaching 69.6% at the conclusion of the 7000 rounds. In contrast, the Basic and the Shared Raw approaches yield lower and similar results. The Shared Raw approach benefits from a pre-trained model trained using the 5% of data shared by the clients, achieving higher accuracy faster. However, after 7000 rounds of federated learning, the model's accuracy does not improve beyond 49.13%, which aligns with the accuracy achieved by the basic federated learning approach. Moreover, we noticed that the main difference between Basic and Shared Raw approaches is the significantly reduced accuracy oscillation of the latter, primarily attributable to parameters pretraining during the initialization phase.

Figure 17 presents the same results in terms of time rather than rounds. While the initialization phase of WSFL_GA consumed considerable time, it was capable of improving further the accuracy during the execution of federated learning compared to the stagnation observed in both the Shared Raw and Basic approaches. The experiments also highlight the extended amount of time required by the Shared Raw approach due to the increased data load of clients, thereby extending the overall training time for each federated learning round. The subplot of Figure 17 illustrates the continuation of the execution, ending at 6.4TU compared to 0.58TU for basic federated learning and 0.92TU for WSFL_GA. The main reason our WSFL approach outperforms the others is its ability to sequentially understand and recognize the patterns effectively in the dataset prior to the start of the Federated Learning phase. Moreover, the WSFL scheme empowered by Genetic Algorithm client selection has shown promising results in reducing the processing time for model convergence compared to other approaches.

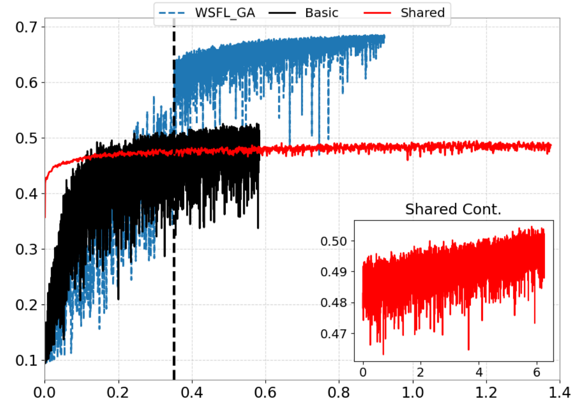


Fig. 16: Initialization Phase Followed By Federated Learning. Black Vertical Line Indicates The End of The Initialization Phase. X-Axis Represents The Cumulative Time while Y-Axis Represents the Accuracy

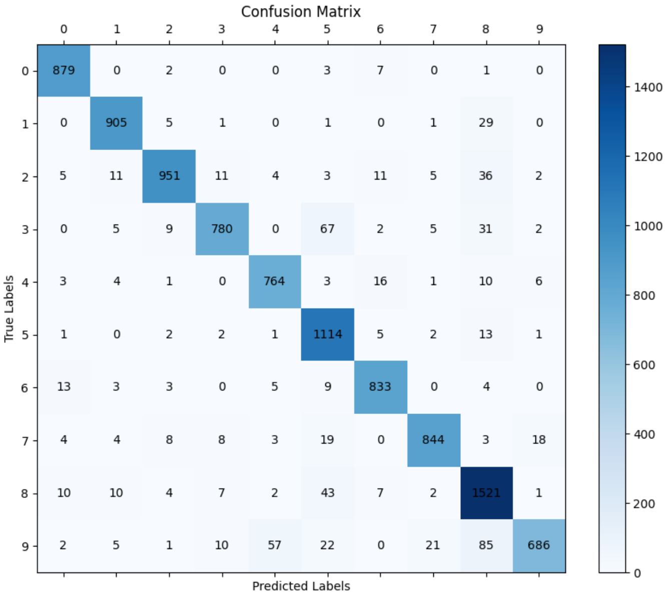
C. F1 Metrics

VIII. Conclusion

This paper focuses on addressing the impact of non-IID clients (IoT devices) in FL through warmup-based sequential learning. Our WFSL approach introduces an extra initialization layer to the conventional FL architecture, where selected clients train sequentially an initial model. This model incorporates partially trained parameters across the entire dataset, enhancing its effectiveness against non-IID and biased clients and allowing the subsequent federated learning to achieve higher accuracy and faster convergence. Our approach incorporates a regulator to mitigate the catastrophic forgetting problem, a significant challenge in sequential environments. Additionally, we introduce a genetic algorithm selector for the initialization phase, which benefits from the properties of sequential learning to select clients based on their model parameters, improving

Method	Precision	Recall	Accuracy	F1	Time Taken	Special Indicate
Shared	0.552	0.481	0.602	0.426	2.329	dr: 0.05
Shared	0.674	0.539	0.678	0.498	3.087	dr: 0.1
WSFL_ALL	0.952	0.950	0.950	0.951	748.465	-
WSFL_GA	0.898	0.878	0.878	0.879	61.401	m: 0.05, c: 0.1, p: 50, cls: 5
WSFL_GA	0.921	0.906	0.906	0.908	88.468	m: 0.1, c: 0.3, p: 100, cls: 5
WSFL_GA	0.899	0.861	0.861	0.863	58.832	m: 0.05, c: 0.1, p: 50, cls: 10
WSFL_GA	0.918	0.895	0.895	0.899	92.707	m: 0.1, c: 0.3, p: 100, cls: 10
WSFL_RN	0.906	0.877	0.877	0.882	41.496	cr: 10
WSFL_RN	0.842	0.770	0.770	0.774	14.919	cr: 5
EWC_ALL	0.084	0.208	0.208	0.109	307.587	weight: 0.1
EWC_ALL	0.420	0.447	0.447	0.394	290.471	weight: 0.5
EWC_RN	0.146	0.092	0.092	0.060	7.600	weight: 0.1, cr: 5
EWC_RN	0.440	0.394	0.394	0.287	17.858	weight: 0.1, cr: 10

TABLE II: F1 Metrics Results. In This Table, dr refers to the data ratio, which controls how much data is shared from clients in the $Shared$ approach, m is the mutation rate, c is the crossover rate, p is population, cls is the number of clusters, cr is client ratio, which indicates how much clients are selected in the RN approaches. Finally, $weight$ is the control variable for the EWC approach



	0	1	2	3	4	5	6	7	8	9
0	879	0	2	0	0	3	7	0	1	0
1	0	905	5	1	0	1	0	1	29	0
2	5	11	951	11	4	3	11	5	36	2
3	0	5	9	780	0	67	2	5	31	2
4	3	4	1	0	764	3	16	1	10	6
5	1	0	2	2	1	1114	5	2	13	1
6	13	3	3	0	5	9	833	0	4	0
7	4	4	8	8	3	19	0	844	3	18
8	10	10	4	7	2	43	7	2	1521	1
9	2	5	1	10	57	22	0	21	85	686

Fig. 17: Confusion Matrix

accuracies and reducing the overall training time cost. Experimental evaluations of our WFSL approach on MNIST, KDD, and CIFAR-10 datasets show significant improvements in model accuracy and time efficiency compared to traditional Shared Raw and basic approaches.

Acknowledgment

The authors would like to thank...

References

[1] H. Elayan, M. Aloqaily, and M. Guizani, "Sustainability of healthcare data analysis iot-based systems using deep federated learning," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7338–7346, 2022.

[2] "Vanet qos-olsr: Qos-based clustering protocol for vehicular ad hoc networks," *Computer Communications*, vol. 36, no. 13, pp. 1422–1435, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366413001710

[3] A. Hammoud, H. Otrok, A. Mourad, and Z. Dziong, "On demand fog federations for horizontal federated learning in iov," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 3062–3075, 2022.

[4] A. Alagha, S. Singh, H. Otrok, and R. Mizouni, "Influence-and interest-based worker recruitment in crowdsourcing using online social networks," *IEEE Transactions on Network and Service Management*, 2022.

[5] W.-N. Chen, D. Song, A. Ozgur, and P. Kairouz, "Privacy amplification via compression: Achieving the optimal privacy-accuracy-communication trade-off in distributed mean estimation," in *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023. [Online]. Available: https://openreview.net/forum?id=GiUuJlogu0

[6] K. K. Patel, M. Glasgow, L. Wang, N. Joshi, and N. Srebro, "On the still unreasonable effectiveness of federated averaging for heterogeneous distributed learning," in *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023. [Online]. Available: https://openreview.net/forum?id=vhS68bKv7x

[7] M. Arafeh, A. Hammoud, H. Otrok, A. Mourad, C. Talhi, and Z. Dziong, "Independent and identically distributed (iid) data assessment in federated learning," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 293–298.

[8] O. Weller, M. Marone, V. Braverman, D. Lawrie, and B. V. Durme, "Pretrained models for multilingual federated learning," 2022.

[9] 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.

[10] J. Nguyen, J. Wang, K. Malik, M. Sanjabi, and M. Rabbat, "Where to begin? on the impact of pre-training and initialization in federated learning," 2022.

[11] I. J. Goodfellow, M. Mirza, X. Da, A. C. Courville, and Y. Bengio, "An empirical investigation of catastrophic forgeting in gradient-based neural networks," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014. [Online]. Available: http://arxiv.org/abs/1312.6211

[12] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie, "Robust federated learning: The case of affine distribution shifts," in *Advances in Neural Information Processing Systems*, vol. 2020-December, 2020.

[13] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: stochastic controlled averaging for on-device federated learning," *CoRR*, vol. abs/1910.06378, 2019. [Online]. Available: http://arxiv.org/abs/1910.06378

[14] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni,

"Federated learning with matched averaging," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=BkluqlSFDS

[15] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–9.

[16] Z. Chen, D. Li, R. Ni, J. Zhu, and S. Zhang, "Fedseq: A hybrid federated learning framework based on sequential in-cluster training," *IEEE Systems Journal*, vol. 17, no. 3, pp. 4038–4049, 2023.

[17] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," 2016.

[18] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[19] M. Arafeh, H. Ould-Slimane, H. Otrok, A. Mourad, C. Talhi, and E. Damiani, "Data independent warmup scheme for non-iid federated learning," *Information Sciences*, vol. 623, pp. 342–360, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025522015407

[20] M. Arafeh, H. Otrok, H. Ould-Slimane, A. Mourad, C. Talhi, and E. Damiani, "Modularfed: Leveraging modularity in federated learning frameworks," *Internet of Things*, vol. 22, p. 100694, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660523000173

Arafeh received the B.S. and M.S. degree in business computing from Lebanese University in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree with École de Technologie Supérieure, Montreal, QC, Canada. His main research interests include federated learning, artificial intelligence, and split learning.

Ahmad Hammoud received the B.S. degree in business computing from Lebanese University in 2016, the M.S. degree in computer science from Lebanese American University in 2019, and the Ph.D. degree from École de Technologie Supérieure (ETS), Canada in 2023. He is a PostDoc Fellow at ETS. His current research interests include Metaverse, cloud and fog federations, federated learning, Internet of Vehicles, game theory, blockchain, artificial intelligence, and security

Mohsen Guizani received the BS (with distinction), MS and PhD degrees in Electrical and Computer engineering from Syracuse University, Syracuse, NY, USA in 1985, 1987 and 1990, respectively. He is currently a Professor of Machine Learning and the Associate Provost at Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE. Previously, he worked in different institutions in the USA. His research interests include applied machine learning and artificial intelligence, Internet of Things (IoT), intelligent systems, smart city, and cybersecurity. He was elevated to IEEE Fellow in 2009 and was listed as a Clarivate Analytics Highly Cited Researcher in Computer Science in 2019, 2020 and 2021. Dr. Guizani has won several research awards including the "2015 IEEE Communications Society Best Survey Paper Award", the Best ComSoc Journal Paper Award in 2021 as well five Best Paper Awards from ICC and Globecom Conferences. He is the author of ten books and more than 800 publications. He is also the recipient of the 2017 IEEE Communications Society Wireless Technical Committee (WTC) Recognition Award, the 2018 AdHoc Technical Committee Recognition Award, and the 2019 IEEE Communications and Information Security Technical Recognition (CISTC) Award. He served as the Editor in-Chief of IEEE Network and is currently serving on the Editorial Boards of many IEEE Transactions and Magazines. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer.

Azzam Mourad received his M.Sc. in CS from Laval University, Canada (2003) and Ph.D. in ECE from Concordia University, Canada (2008). He is currently a Visiting Professor at Khalifa University, a Professor of Computer Science and Founding Director of the Artificial Intelligence and Cyber Systems Research Center at the Lebanese American University, and an Affiliate Professor at the Software Engineering and IT Department, École de Technologie Supérieure (ETS), Montreal, Canada. His research interests include Cyber Security, Federated Machine Learning, Network and Service Optimization and Management targeting IoT and IoV, Cloud/Fog/Edge Computing, and Vehicular and Mobile Networks. He has served/serves as an associate editor for IEEE Transactions on Services Computing, IEEE Transactions on Network and Service Management, IEEE Network, IEEE Open Journal of the Communications Society, IET Quantum Communication, and IEEE Communications Letters, the General Chair of IWCMC2020-2022, the General Co-Chair of WiMob2016, and the Track Chair, a TPC member, and a reviewer for several prestigious journals and conferences. He is an IEEE senior member.

Hadi Otrok received his Ph.D. in ECE from Concordia University. He holds a Full Professor position in the Department of Computer Science (CS) at Khalifa University. Also, he is an Affiliate Associate Professor at the Concordia Institute for Information Systems Engineering at Concordia University, Montreal, Canada, and an Affiliate Associate Professor in the Electrical Department at Ecole de Technologie Superieure (ETS), Montreal, Canada. His research interests include the domain of blockchain, reinforcement learning, crowd sensing and sourcing, ad hoc networks, and cloud security. He co-chaired several committees at various IEEE conferences. He is also an Associate Editor at IEEE Transactions on Services Computing, IEEE Transactions on Network and Service Management (TNSM), Ad-hoc Networks (Elsevier), and IEEE Network. He also served from 2015 to 2019 as an Associate Editor at IEEE Communications Letters.

Hakima Ould-Slimane received the Ph.D. degree in computer science from Laval University, Québec City, QC, Canada, in 2011. She is currently a Professor with the Department of Mathematics and Computer Science, Université de Québec à Trois-Rivières, Trois-Rivières, QC, Canada. Her research interests include information security, cyber resilience, homomorphic encryption, federated learning, preserving data privacy in smart environments, machine learning-based intrusion detection, access control, optimization of security mechanisms, and security of social networks.

Zbigniew Dziong received the Ph.D. degree from the Warsaw University of Technology, Poland, where he also worked as an Assistant Professor. From 1987 to 1997, he was with INRS-Telecommunications, Montreal, QC, Canada. From 1997 to 2003, he worked with Bell Labs, Holmdel, NJ, USA. Since 2003, he has been with the École de Technologie Supérieure (University of Quebec), Montreal, as a Full Professor. He is an expert in the domain of performance, control, protocol, architecture, and resource management for data, wireless, and optical networks. He has participated in research projects for many leading telecommunication companies, including Bell Labs, Nortel, Ericsson, and France Telecom. He won the prestigious STENTOR Research Award (1993, Canada) for collaborative research. His monograph ATM Network Resource Management (McGraw Hill, 1997) has been used in several universities for graduate courses.

Chang-Dong Wang received the Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2013. From January 2012 to November 2012, he was a visiting Student with the University of Illinois at Chicago, Chicago, IL, USA. In 2013, he joined Sun Yat-sen University, where he is currently an Associate Professor with the School of Data and Computer Science. His research interests include machine learning and data mining. He has authored or coauthored more than 70 scientific papers in international journals and conferences such as IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Knowledge and Data Engineering, IEEE Transactions on Cybernetics, IEEE Transactions on Neural Networks and Learning Systems, ACM Transactions on Knowledge Discovery from Data, IEEE Transactions on Systems, Man, and Cybernetics, IEEE Transactions on Industrial Informatics, IEEE Transactions on Systems, Man, and Cybernetics Part C, KDD, AAAI, IJCAI, CVPR, ICDM, CIKM, and SDM. His ICDM 2010 paper was the recipient of the Honorable Mention for Best Research Paper Awards. He was also the recipient of the 2012 Microsoft Research Fellowship Nomination Award. and 2015 Chinese Association for Artificial Intelligence (CAAI) Outstanding Dissertation. He is an Associate Editor for Journal of Artificial Intelligence Research.

Di Wu received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2000, the M.S. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 2007. He was a Post-Doctoral Researcher with the Department of Computer Science and Engineering, Polytechnic Institute of New York University, Brooklyn, NY, USA, from 2007 to 2009, advised by Prof. K. W. Ross. Dr. Wu is currently a Professor and the Associate Dean of the School of Computer Science and Engineering with Sun Yat-sen University, Guangzhou, China. He was the recipient of the IEEE INFOCOM 2009 Best Paper Award, IEEE Jack Neubauer Memorial Award, and etc. His research interests include edge/cloud computing, multimedia communication, Internet measurement, and network security.