FedResilience: A Federated Learning Application to Improve Resilience of Resource-Constrained Critical Infrastructures

Ahmed Imteaj 1,2,†, Irfan Khan 3, Javad Khazaei 4 and Mohammad Hadi Amini 1,2,*

1 Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA; aimte001@fiu.edu
2 Sustainability, Optimization, and Learning for InterDependent Networks Laboratory (Solid Lab), FIU, Miami, FL 33199, USA
3 Clean and Resilient Energy Systems (CARES) Lab, Texas A&M University at Galveston, Galveston, TX 77554, USA; irfankhan@tamu.edu
4 Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18015, USA; khazaei@lehigh.edu
* Correspondence: moamini@fiu.edu
† Current address: 11200 SW 8th St, Miami, FL 33199, USA.

Abstract: Critical infrastructures (e.g., energy and transportation systems) are essential lifelines for most modern sectors and have utmost significance in our daily lives. However, these important domains can fail to operate due to system failures or natural disasters. Though the major disturbances in such critical infrastructures are rare, the severity of such events calls for the development of effective resilience assessment strategies to mitigate relative losses. Traditional critical infrastructure resilience approaches consider that the available critical infrastructure agents are resource-sufficient and agree to exchange local data with the server and other agents. Such assumptions create two issues: (1) uncertainty in reaching convergence while applying learning strategies on resource-constrained critical infrastructure agents, and (2) a huge risk of privacy leakage. By understanding the pressing need to construct an effective resilience model for resource-constrained critical infrastructure, this paper aims at leveraging a distributed machine learning technique called Federated Learning (FL) to tackle an agent’s resource limitations effectively and at the same time keep the agent’s information private. Particularly, this paper is focused on predicting the probable outage and resource status of critical infrastructure agents without sharing any local data and carrying out the learning process even when most of the agents are incapable of accomplishing a given computational task. To that end, an FL algorithm is designed specifically for a resource-constrained critical infrastructure environment that could facilitate the training of each agent in a distributed fashion, restrict them from sharing their raw data with any other external entities (e.g., server, neighbor agents), choose proficient clients by analyzing their resources, and allow a partial amount of computation tasks to be performed by the resource-constrained agents. We considered a different number of agents with various stragglers and checked the performance of FedAvg and our proposed FedResilience algorithm with prediction tasks for a probable outage, as well as checking the agents’ resource-sharing scope. Our simulation results show that if the majority of the FL agents are stragglers and we drop them from the training process, then the agents learn very slowly and the overall model performance is negatively affected. We also demonstrate that the selection of proficient agents and allowing them to complete only parts of their tasks can significantly improve the knowledge of each agent by eliminating the straggler effects, and the global model convergence is accelerated.

Keywords: power system resilience; disaster; Federated Learning; edge intelligence; resilience management systems; resource-limitations; demand response

1. Introduction

In this section, we present the motivation for the development of an application to improve the resilience of critical infrastructures using a novel FL model. We discuss the
prior works that have recently been published and explain how our proposed FL model can be beneficial for improving the resilience of critical infrastructures. We also show the novelty of this paper, which is followed by a brief description of the paper’s organization.

1.1. Motivation

Critical infrastructures such as power systems, transportation, fuel, water, and gas are interconnected and are all parts of distinct or interdependent networks that operate cooperatively to produce and distribute essential services [1]. When a sudden interruption occurs in any of the infrastructures, the resilience technique can assist in the continuation of the operations through its ability to resist, avoid, adapt, and recover swiftly from a disastrous situation. However, it is vital to ensure the resilience of the critical infrastructures as the sectors are interdependent. Using an inter-network communication scheme, entities within the same network can exchange resources, and in an intra-network communication infrastructure, clients residing in different network domains can provide support to each other to enhance the resilience of the system [2]. For instance, most of the critical infrastructures—e.g., transportation networks, telecommunication, and finance and banking sectors—rely on a continuous and stable power supply. However, the prolonged disruption of the operations of critical infrastructures can incur a significant economic loss. One of the recent surveys found that power outages occur for a minimum of one out of four companies in every month [3]. Specifically, in large companies, power outage loss costs over a million dollars an hour and around 150 USD annually [4]. Several works have already been conducted with the aim of improving the resilience of critical infrastructures by applying machine learning (ML) [5], deep learning (DL) [6], distributed edge computing [7,8], and transfer learning [9]. All these works constructed their prediction model either by collecting data from the agents (e.g., ML, DL) or receiving an update for some data from the distributed agents (e.g., distributed ML). However, sharing such sensitive data could be privacy-intrusive. An attacker can expose or tamper with data, which may cause the failure of the whole resilience system, and a company may thus face a huge loss. Besides, in a distributed system, we may observe straggler clients that learn very slowly due to resource-limitation issues and degrade the overall performance of the prediction model [10].

The conventional resilience approaches that are constructed on the theme of the learning and forecasting of probable outages consider that all critical infrastructure agents (CIAs) have available resources and can perform an assigned computational task. However, in a real-world scenario, any agent may possess low system configurations or may run out of resources. In consequence, some agents may not be able to complete the assigned computational tasks due to the shortage of resources. Therefore, our main motivation for this research is to mitigate the outage loss of the CIAs by developing a novel FL-based prediction model that can preserve privacy and handle the straggler issues in the case of resource-constrained network agents. We propose a novel FL-based strategy that consists of a local CIA, which acts as an intelligent decision-making entity or an FL agent; e.g., a smart factory or an autonomous micro-grid can act as a CIA. A CIA can generate a model based on its available local data (i.e., power demand and resource availability) and share the model with a central fusion center that acts as an FL server. Similarly, the neighboring CIAs share their model to pursue a common goal, and the coordinator generates a global model that learns the outage information and resource sharing scope of all the CIA agents. In case any CIA agent has limited computational resources (e.g., low processing capability, bandwidth) and cannot generate a learning model, the central coordinator is enabled to select proficient clients for the training rounds, allowing partial amounts of work from the resource-constrained CIAs considering their available resources. Therefore, the resource-constrained issues of the grids related to model training would be resolved, and distributed resources could be supplied from the neighbors in case any CIA fails to continue its operation.
1.2. Literature Review

The concept of resilience can be generalized for any discipline as a system’s capacity to predict and withstand forthcoming shocks, restore the system’s normal state swiftly, and adapt with an improved action for handling future catastrophic events. Managing and improving the infrastructure resilience of critical infrastructures has recently attracted the attention of several researchers and, in consequence, several studies related to the modeling and upgrading of systems and networks resilience have been proposed [11–13]. The authors in [14] focused on reducing the peak load of the critical infrastructure of power systems by considering multi-agent-based power generation, network grids, and relative demand response status. In their proposed approach, the agents could share their local information only with a central fusion center and were unable to interact with neighboring agents to exchange local resources. Besides, a comprehensive study on modeling the resilience of large-scale critical infrastructure was presented in [15]. However, centralized resilience systems become overly complex when a large amount of data is stored, processed, analyzed, and shared from a central fusion center [16,17]. The drawbacks of centralized resilience systems (e.g., scalability, computational power, storage) can be handled with distributed systems and learning resilience schemes [18]. The authors of [19] presented a detailed analysis of multi-agent systems (MAS), leveraging distributed intelligence among the network agents through peer-to-peer communication and sharing demand and load status to achieve a common goal. Besides, the authors of [20] proposed an adaptive synchronization approach for heterogeneous MAS against actuator fault by developing a multi-objective optimization technique to measure the installation capacity of network agents considering power-resilience against disasters [21,22]. Moreover, several works have adapted the strategy of utilizing infrastructure resources to improve the resilience of relative operations [23,24]. Further, some recent works developed resilience management systems for power systems [8,25–28], transport [29–32], urban areas [33–35], healthcare [36–38], and production systems [39–41] by adding intelligence to the CIAs so that the agents could make autonomous decisions by analyzing the demand–response state. In summary, all the prior works proposed the improvement of resilience either by passing local sensitive data of infrastructure agents to a central fusion center or by sharing such local sensitive data with neighboring agents. However, sharing the sensitive data that reside in the CIAs leads to the risk of privacy violation and can also interrupt the infrastructure operation through data falsification. To prevent that, a recently invented distributed ML technique called Federated Learning (FL) was proposed that can generate a smart model by utilizing edge resources and keeping an agent’s information private. As the FL process is completely dependent on the agent’s local model update, one of the challenges that the FL process presents is the straggler issues that arise due to the heterogeneity of the systems. System heterogeneity can be referred to as the heterogeneous nature of the agents in terms of their computational power, memory, battery life, or bandwidth. If we apply the FL process considering Internet of Things (IoT) devices, then there is a high chance of observing straggler agents during a training process [42]. This is because IoT devices are resource-constrained and vulnerable [43]. If we consider the state-of-the-art FedAvg algorithm [44], then it simply drops the straggler agents from the training process. However, dropping the stragglers can degrade the model performance, and also some agents may have valuable data. Instead of this approach, we need a strategy that can effectively handle the stragglers by counting every contribution, irrespective of its size. The authors of [45] proposed the FedProx algorithm, which can enable partial amounts of work to be collected from the agents; however, they randomly selected agents for the training round. According to the authors of [46], FedMax outperforms FedProx in terms of communication rounds by applying a strategy of limiting activation-divergence across multiple devices.

To tackle the above-mentioned issues in the context of the resilient operation of critical infrastructures and analyzing the existing works of FL, this paper is the first to propose a novel FL-based strategy that can predict the probable outages and resource-sharing capabilities of the network agents with the aim of improving resilience. Our FedResilience
algorithm can select proficient agents by examining their resources and handle the stragglers by assigning feasible local computational tasks based on their capabilities. Our proposed technique relies on sharing local models of the infrastructure agents instead of sharing sensitive data and, finally, exploits the collaboratively learned knowledge on the probable outages and resource availability status of the whole FL network to enhance the resilience operations.

1.3. Contribution

The main contributions of this paper are given below:

- To the best of our knowledge, this is the first FL application that can improve the resilience of critical infrastructures through early prediction;
- We present a pathway of collaborative learning for CIAs that enables on-device learning without sharing any data and by exchanging only model information;
- We choose only the proficient agents for the FL training process and enable partial works to be collected from the resource-constrained agents to resolve the straggler issues;
- To demonstrate the effectiveness of Federated Learning in improving resilience by the early prediction of the outages and resource-sharing scope of the agents, we evaluate the prediction performance considering a varying number of stragglers and compare the model with the popular FedAvg [44] algorithm.

1.4. Organization

The rest of this paper is organized as follows. Section 2 presents the overview of FL and explains how our developed FL model can be effective in improving the resilience operations of critical infrastructures in detail. Section 3 presents our experiment results and is followed by Section 4, which concludes the paper.

2. Proposed System Description

FL is a distributed machine learning technique that allows the on-device training of network clients with their local data instead of sharing raw data with the server. Each client generates a local model by optimizing its local objective function that is shared with the FL server. After receiving local models from all participating FL clients, the FL server performs aggregation on the received models and updates a global model which is initialized as well as shared with all network clients at the initial stage of FL training. After that, the updated global model is disseminated to all FL clients, and each FL client tunes their local model by learning from the global model. The FL client–server interaction process is continued until the global model achieves a desired accuracy; hence, the model reaches a target convergence. The overall FL process is presented in Figure 1.
In critical infrastructure networks, we may observe heterogeneous agents with varying system configurations and data volumes. Therefore, it is not viable to assign a uniform number of tasks to all the agents that participate in the FL process. The authors in [10] conducted a comprehensive survey on leveraging FL for IoT devices, where they discussed the possible challenges faced while applying FL on resource-constrained agents. Due to the varying and limited resource statuses, while one agent could perform a given computational task efficiently, another might turn into a straggler. An agent may become a straggler if the assigned computational task is overwhelming compared to its available resources. If the majority of the participating agents in the FL process turn into stragglers, then target convergence may never be obtained. Besides this, the IoT-enabled infrastructure agents are generally more prone to attacks that may cause divergent local model updates [47]. To improve the resilience, it is crucial to predict the infrastructure’s outage, and in distributed systems, the main hindrance in the agent’s learning process is stragglers. Therefore, it is essential to monitor the resource status and ensure that all agents are effectively operating by avoiding straggler issues. The typical FedAvg algorithm [44] assumes that all FL agents are resource-proficient and capable of accomplishing any given computational tasks. However, if a real-world FL-based IoT scenario is considered, then the majority of the agents may possess very few resources. Therefore, it is not effective to randomly select a fraction of agents for the training process. If the agents’ resource availability statuses are tracked and the weak agents are filtered out from the training process, then it may possible to move one step closer towards resolving straggler issues. To infuse resource-awareness functionalities into the FL process, the task publisher (i.e., FL server) needs to acknowledge each network agent’s minimum requirements for accomplishing a published task. After that, all interested agents share their resource information (e.g., memory, processing ability, bandwidth, and battery-life) with the task publisher. By examining the interested agents’ resources, the task publisher prepares a list of proficient agents and randomly selects a subset of agents for that task.

2.1. Handling Systems and Statistical Heterogeneity of Critical Infrastructure Agents (CIAs)

In this segment, we discuss how a strategy of allowing partial works from the FL agents can be adopted through a generalization of the FedAvg algorithm [44]. In Section 2, we explain how the comparatively proficient agents can be selected for an FL process. However, it is possible that, among the selected agents, some agents would not be able to accomplish their entire task. Particularly, this can occur when all the interested and available FL agents have constrained resources and there are no other options without considering a subset of those agents for the training phase. Now, if the conventional FedAvg
algorithm is applied [44], which instructs the center to assign uniform local computational tasks to all the selected agents, then straggler effects may be observed that can slow down the model convergence, or we may never be able to reach the target convergence. Instead, if we allow the selected agents to perform computational tasks based on their resources, then we would not require the straggler agents to be dropped, and every agent could contribute towards constructing a global model. In Figure 2, the high-level view of allowing partial works from the FL agents is presented. From the figure, it can be observed that the water network agent and transportation agent have limited resources while the power network has sufficient resource availability. Considering the resource status, the water network and transportation agents are performing 30% and 48%, respectively, of the overall computational tasks, while the power network is performing the entire task. Let us assume that the task publisher defines a local epoch of 100 that needs to be performed by all chosen FL agents. However, some of the agents are not capable of performing 100 local epochs on their data to generate a local model. In such a case, if an agent is capable of performing 30% of the overall computational task (i.e., 30 local epochs on their own data) rather than the whole task, then the agent would be allowed to perform that amount of the computational task and send back the model to the server. This proposed strategy solves two issues: first, the FL server does not need to wait a long time for a straggler agent, and second, every individual contribution from the agents can be counted.

![Figure 2. Allowing partial amounts of work from the FL agents.](image)

To reduce communication overheads, a popular strategy in federated optimization is that for each iteration period, each agent tries to achieve a local objective function that is used as a replacement of a global objective function. In each training round, a subset of agents is chosen, and each agent uses its resources to optimize the local objective function. After that, the agents share their model with the FL server, which performs aggregation and updates the global model. Allowing a flexible amount of work helps to solve the inexact nature of local objectives and assists in tuning the number of communications vs. local computations. While too many local epochs can overfit the model, a smaller number of local epochs increases communication overheads as well as the convergence time [47]. Therefore, it is required to set local epochs through proper tuning to ensure robust convergence. The concept of an inexact solution can be stated as follows:

**Definition 1** \((q\text{-inexact solution})\). Let us consider a function \(R(\omega; \omega_0) = F(\omega) + \frac{\zeta}{2}\|\omega - \omega_0\|^2\), and \(q \in [0, 1]\), it can be said that \(\omega^*\) is a \(q\) - inexact solution of \(\min \theta R(\omega; \omega_0)\) if \(\|\nabla R(\omega^*; \omega_0)\| \leq q\|\nabla R(\omega_0; \omega_0)\|\), where \(\nabla R(\omega; \omega_0) = \nabla F(\omega) + \zeta(\omega - \omega_0)\).
To leverage the proper tuning of local computations by handling system heterogeneity, we use the concept of the inexact solution [45], which allows us to collect variable numbers of local epochs from the participated agents according to their resource availability. The ϱₐ⁻inexactness for a CIA a at training round t can be defined as follows:

**Definition 2** (ϱₐ⁻inexact solution). Let us consider a function $Rₐ(\omega;\omega₁) = Fₐ(\omega) + \frac{1}{2}\|\omega - \omega₁\|^2$, and $\varrho \in [0,1]$, we call $\omega^{*}$ is a $\varrhoₐ⁻$-inexact solution of $\min_{\omega} Rₐ(\omega;\omega₁)$ if $\|\nabla Rₐ(\omega^{*};\omega₁)\| \leq \varrhoₐ\|\nabla Rₐ(\omega;\omega₁)\|$, where $\nabla Rₐ(\omega;\omega₁) = \nabla Fₐ(\omega) + \xi(\omega - \omega₁)$.

Here, the convenience of $\varrho$-inexactness is that it allows variable local computations to be accomplished by the selected CIAs in each training round. As the system heterogeneity causes heterogeneous progress from the agents while solving local objective functions, it is vital to enable adaptive $\varrho$ considering agents’ resource availability. We can consider a scenario from our real-life perspectives. Suppose we have a few power agents that agree to participate in an FL process and utilize their edge resources. Each agent may have some outage information about some past events and also can possess resource information about its neighboring agents. Now, if an agent wants to gather knowledge about outage events that were never seen by that agent and store resource information from the agents that are not its neighbors, then it needs to adopt a method so that it can obtain the collective knowledge of the whole network. We can infuse the collective knowledge to each agent through the power of FL. In case, if a power agent does not have sufficient resources to complete an assigned computational task, we allow that agent to perform partial works. In this way, we do not ignore any agent’s local knowledge. As a consequence, each agent is more capable of predicting an outage event and can locate an agent that needs a power supply.

### 2.2. Proposed FedResilience Algorithm

The proposed FedResilience algorithm is presented in Algorithm 1. The goal of this algorithm is to predict the outages and resource-sharing scope of CIAs without sharing any agent’s local data, utilizing the computational resources of the CIAs. Applying the FL strategy for critical infrastructures mainly involves two entities: the critical infrastructure server (CIS) and available CIAs within the networks. At the beginning of the FL process, the server initializes a global model that is disseminated to all available CIAs within the networks specifying task requirements (line 1–2). Each interested CIA shares its current resource status with the CIS (line 3). In each training round, the CIS examines the resource information (i.e., processing power, memory, bandwidth, battery-charge status, and data volume) of the interested CIAs by calling the CheckResource() function (line 4–5). The CheckResource() function receives a CIA’s information upon calling, stores the information in a list, and compares the resource availability status with the task requirements (line 13–15). If the CIA’s available resources satisfy the minimum task requirements, then that CIA’s information is stored in another list and sent back from where the CheckResource() function is called (line 16–18). Upon receiving the resource information from all the interested CIAs, the CIS sorts the eligible CIAs based on their resource status, selects a fraction from those CIAs, and randomly chooses a subset of proficient CIAs for the training phase (line 6–8). After that, the CIS calls the selected agents to perform on-device training using the AgentLocalUpdate() function and shares the latest global model (line 9–10). It is assumed that the total number of data samples within the network is $n$, which are distributed among the CIAs with a set of indexes $Dₐ$ on CIA a, where $Nₐ = |Dₐ|$. Each CIA’s local data in a communication round $t$ is referred to by $Nₜ$. During FL training, each selected CIA utilizes its local solver to determine the inexact minimizer $\varrhoₐ$ to solve the local objective function (line 19–20). Further, each CIA splits its local samples into batches, performs SGD to achieve an optimal local solution, and shares the model with the CIS (line 21–25). The CIS aggregates the local models to generate an updated global model, and the same iteration period is continued until the global model reaches convergence (line 11–12).
Algorithm 1: FedResilience: An FL-based approach to predict the outages and resource-sharing scopes of critical infrastructures. The $A$ eligible clients are indexed by $a$; $B$ = local minibatch size, $F$ = client fraction, $E$ = local epoch, and $\eta$ = learning rate.

1. **CIS executes:** initialize global model $w_0$
2. Disseminate task requirements to all CIAs
3. Collect resource status of interested CIAs
4. For each round $t = 1, 2, \ldots$
   - $R_t = \text{CheckResource} (P_t, M_t, B_t, C_t, V_t)$ for all interested CIAs
   - Sort CIAs based on $R$ and store in a list $L$
   - $E \leftarrow \text{Top} L \cdot F$ CIAs
   - $A_t \leftarrow$ (random set of $E$ CIAs)
   - For each client $a \in A_t$ in parallel
     - $w_{t+1} \leftarrow \text{AgentLocalUpdate} (a, w_t)$
     - For each CIA $a \in A_t$
       - $w_{t+1} \leftarrow w_{t+1} + \frac{1}{N_a} w_a^t$
   - $\text{CheckResource} (P_a, M_a, B_a, C_a, V_a)$:
     - Store $(B_a, M_a, E_a, V_a)$ into a list $Q_a$
     - Compare $Q_a$ with $L_{\text{Req}}$
     - If $Q_a$ satisfies $L_{\text{Req}}$ then
       - Add $Q_a$ to list $R$
   - Return $R$

5. **AgentLocalUpdate** $(a, w)$ : // Run on CIA $a$
   - Each CIA $a$ finds a $w_a^{t+1}$ which is a $\varrho_a^t$-inexact minimizer of: $w_a^{t+1} = F_a(w) + \frac{\xi}{2} \|w - w_t\|^2$ and measures maximum feasible round of local epochs $E$
   - $B \leftarrow$ (split $D_a$ into batch size $B$)
   - For each CIA’s local epoch $e$ from 1 to $E$
     - For batch $b \in B$
       - $w \leftarrow w - \eta \nabla \ell(w; b)$
   - Return $w$ to server

3. Experimental Results

To evaluate the performance of the proposed FedResilience method, various distributed mobile robots are considered as critical infrastructure agents that possess heterogeneous resources in terms of processing power, battery life, memory, and data volume. To simulate the straggler effects and the effectiveness of the proposed FedResilience algorithm, the Electro-Maps dataset [48] is used to predict the power outages and resource-sharing scopes of the agents. The dataset is preprocessed considering temperature, number of weeks, hour, holiday, and population, and an additional column of resource availability is generated from the information regarding the population, holiday, and temperature. Using the information, we target the prediction of the outages and resource-sharing scopes of the critical infrastructure agents. A similar transmission rate is set for all the distributed agents for the simplicity of the FL implementation process. To simulate the effectiveness of allowing partial works from the distributed agents, different numbers of weak distributed agents are deliberately considered to create straggler effects; i.e., some of the agents fail to generate local models due to their constrained resources. It is assumed that there remains a global cycle that is followed by each agent, and each selected agent measures the amount of the local computational task it can perform in training round $i$ as a function of its available resources and clock cycle. The code is publicly available and has been uploaded to a GitHub repository (https://github.com/Imteaj10/FedResilience, accessed on 31 July 2021). In a conventional FL approach, a global epoch $E$ is defined for all the participating agents to perform a particular task, and if any of the agents fail to generate a local model on
time, the model simply drops that agent from the training process (no partial tasks are allowed). However, dropping slow clients from the training process may prolong the model convergence, or the model may even never reach the target convergence. To handle such issues, we adapt a generalization of the FedAvg algorithm that enables each agent to perform part of a computational task by considering the agent’s resource limitations. To present the motivation behind this research, we applied the FedAvg algorithm [44] for predicting the outages and resource-sharing scopes of CIAs and presented the straggler effects. We considered a varying number of CIAs and assumed that a majority of those agents would be stragglers. At first, we considered three agents (where two were stragglers) and computed the training loss and testing accuracy during the prediction of a probable outage by applying the state-of-the-art FedAvg [44] algorithm. In Figure 3, we can see that the training loss started to decrease in the initial few communication rounds and remained almost unchanged for further communication rounds due to the dropping of the majority of clients. In contrast, in Figure 4, it is clear that the improvement of testing accuracy was quite steady and each agent learned very slowly.

Figure 3. Straggler effects on participating FL agents’ (three agents and two stragglers) model loss for the prediction of critical infrastructure outage.

Figure 4. Straggler effects on participating FL agents’ (three agents and two stragglers) model accuracy for the prediction of critical infrastructure outage.

After that, we simulated the straggler effects by increasing the number of agents (three stragglers out of five agents) and computing the training loss to predict a probable outage by applying the state-of-the-art FedAvg [44] algorithm (see Figure 5). We can see a small decrease in training loss for communication round 500. In contrast, in Figure 6, it is observable that some agents had very low accuracy while other agents had compara-
tively high accuracy. However, none of the agents achieved satisfactory improvements in their accuracy.

![Figure 5](image1.png)

**Figure 5.** Straggler effects on participating FL agents’ (five agents and three stragglers) model loss for the prediction of critical infrastructure outage.

![Figure 6](image2.png)

**Figure 6.** Straggler effects on participating FL agents’ (five agents and three stragglers) model accuracy for the prediction of critical infrastructure outage.

We also simulated the straggler effects for eight agents (where six of them were stragglers) and generated the training loss for a predicted outage by applying the state-of-the-art FedAvg [44] algorithm (see Figure 7). We can see that both of the non-straggler agents had a very slow learning process in spite of a higher communication round. In Figure 8, we can see that the clients barely learned from each other and consequently were not able to improve their model quality significantly.
Next, we simulated the straggler effects during the prediction of the agents’ resource-sharing scope by applying the state-of-the-art FedAvg [44] algorithm. Similar to the outage prediction, we considered three agents (where two were stragglers) and generated the training loss and testing accuracy to predict the resource-sharing scope of the agents. In Figure 9, we can see that though the training loss was comparatively lower than the outage prediction loss for the three agents–two stragglers scenario, a similar training loss was observed for all agents (i.e., the agents did not learn through collaboration). On the other hand, we can see from Figure 10, that agent 2 had a comparatively lower accuracy than other agents, but it slightly improved its accuracy through the FL process. However, after 200 communication rounds, all agents’ testing accuracies improved very slowly.

Besides, we simulated the straggler effects during the prediction of the agents’ resource-sharing scope by increasing the number of agents (five agents, where three of them were stragglers). We generated the training loss and testing accuracy by applying the state-of-the-art FedAvg [44] algorithm. In Figure 11, we can see that though the training loss dropped significantly in the few initial communication rounds, almost constant training loss was observed for all agents. Moreover, in Figure 12, all the agents failed to obtain a marginal improvement in their accuracy.
Figure 9. Straggler effects on participating FL agents’ (three agents and two stragglers) model loss for the prediction of agents’ resource-sharing capability.

Figure 10. Straggler effects on participating FL agents’ (three agents and two stragglers) model accuracy for the prediction of agents’ resource-sharing capability.

Figure 11. Straggler effects on participating FL agents’ (five agents and three stragglers) model loss for the prediction of agents’ resource-sharing capability.
Similarly, we simulated the training loss and accuracy by considering eight agents (where six were stragglers) and applied the FedAvg algorithm [44] during the prediction of the agents’ resource-sharing scope. From Figures 13 and 14, we can see that the training loss and accuracy improved as we increased the number of agents; however, both accuracies improved little with the increment of communication rounds due to the straggler effects. In a summary, for all the considered cases, the agents struggled to minimize loss and remained very steady in terms of improving accuracy due to the straggler effect.
To eliminate the straggler effects during the prediction of the power outage and resource-sharing information, we proposed partial works to be allowed from the straggler agents; i.e., we assigned computational tasks based on the agents’ available resources. To evaluate the performance of FedResilience, we considered the same number of agents (three, five, and eight agents) for the training rounds and observed their learning process. At first, we considered three agents (where two agents were stragglers) and checked their loss (Figure 15) and accuracy (Figure 16) during the prediction of power outages. Though the training loss increased due to the deviation of the local model updates as the stragglers performed low computational tasks, the agents started to reduce their training loss by learning from the global model and from their own data. On the contrary, the accuracy of the agents started to increase after 320 communication rounds because of the low number of resource-sufficient agents (Figure 16). We also simulated the loss (Figure 17) and accuracy (Figure 18) during the prediction of resource-sharing scope by considering the same number of agents and achieved better performance than the FedAvg [44] algorithm.
After that, we considered five agents (where two agents were stragglers) and checked their loss (Figure 19) and accuracy (Figure 20) during the prediction of power outages. For these simulations, we observed similar patterns to those in Figures 15 and 16, but obtained better performance due to the higher number of active clients. The accuracy of the agents started to increase after 100 communication rounds because of the comparatively higher number of resource-sufficient agents (Figure 20).
We also simulated the loss (Figure 21) and accuracy (Figure 22) during the prediction of resource-sharing scope by considering five agents and achieved better performance than the FedAvg [44] algorithm.

Figure 19. FedResilience’s impact on participating FL agents’ (five agents and three stragglers) model loss for the prediction of CIA outages.

Figure 20. FedResilience impact on participating FL agents’ (five agents and three stragglers) model accuracy for the prediction of CIA outages.

Figure 21. FedResilience’s impact on participating FL agents’ (five agents and three stragglers) model loss for the prediction of CIAs’ resource-sharing capability.
Further, we simulated the performance of eight agents (where six agents were stragglers) and checked their loss (Figure 23) and accuracy (Figure 24) when predicting power outages. Here, it was clear that the agents improved their knowledge base (i.e., the training loss decreased and a significant accuracy improvement is observed) due to the improved quality of the global model.

In a similar fashion, we simulated the loss (Figure 25) and accuracy (Figure 26) during the prediction of resource-sharing scope by considering eight agents and achieved a remarkable performance improvement compared to the FedAvg [44] algorithm. The training...
loss became close to 0.1 (Figure 25), and some of the agents achieved higher accuracy within 250 – 300 communication rounds (Figure 26).

From the simulation results, it is observable that FedResilience has better performance than the conventional FedAvg model. As we increase the number of agents and count partial works from each of them, then agents can learn quickly, and the global model accuracy also increases. When we considered eight agents and six stragglers and applied the FedAvg algorithm, then the global model only contained the knowledge of the two active agents. As a result, the agent could not upgrade its knowledge base and showed a steady learning curve. However, when we counted the partial computational tasks by those stragglers, then the accumulation of those partial works generated an upgraded global model. As the quality of the global model improved and each agent tuned their local model by learning from the latest global model, the agents’ learning process was accelerated. Our simulation results demonstrate two trends: first, the FedAvg algorithm is not suitable to predict outages or the resource-sharing information of resource-constrained CIAs as the algorithm cannot handle the straggler effects, which eventually slows down the agents’ learning process; second, the FedResilience algorithm can handle straggler effects and is suitable even when we have a large number of stragglers within the network. In Figure 27, we can see that the FedResilience algorithm outperforms the FedAvg algorithm [44], achieving higher global model accuracy (cumulative updates of all the participating agents’ local models) while predicting the outages and resource-sharing information of CIAs even with a large number of stragglers.
In Figure 28, a linear approximation of the real system and the performance of the proposed FedResilience algorithm for a disaster event is presented. It can be seen that the real system performance index decreases after time $t_d$ and reaches a minimal index at time $t_m$. In the beginning, the performance index is stable due to a preventive outage; however, as soon as the preventive outage is finished, the curve starts to move downwards and reaches a minimal performance index ($P_{\text{min}}$). The low-performance index remains until a certain time interval, and after that, the system starts to recover. In contrast, when the FedResilience algorithm is applied during an outage, the performance index does not move down at minimal performance index ($P_{\text{min}}$); instead, using the power of edge intelligence, the system can recover swiftly and a remarkable performance index can be achieved.

Figure 28. Eight-point linear approximation of the performance of the FedResilience algorithm during a disaster event.

4. Conclusions

This paper proposes a strategy to improve the resilience operations of critical infrastructures even when the network agents have limited resources. To evaluate our approach, the impact of straggler agents on the overall learning process is presented by considering resource-constrained distributed agents. After that, the effectiveness of our proposed FedResilience algorithm is evaluated, demonstrating the acceleration of the distributed agents’ learning process despite heterogeneous system resources and model updates. By choosing proficient agents, performing on-device training, transferring knowledge, and allowing
partial works, a robust and consistent FL model is achieved with higher global model accuracy compared to the state-of-the-art FedAvg algorithm; the model can also accelerate the learning process of unreliable IoT-enabled heterogeneous environments. The proposed concept can be applied to any resource-constrained heterogeneous IoT environment that is disrupted by straggler effects and struggles to reach convergence due to slow learning.

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

**Funding:** This research received no external funding.

**Acknowledgments:** A.I. and M.H.A. acknowledge the resources and support of Sustainability, Optimization, and Learning for InterDependent networks laboratory (solid lab) members at the Florida International University (www.solidlab.network) (accessed on 20 June 2021).

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

**Abbreviations**

The following abbreviations are used in this manuscript:

- FL Federated Learning
- CIA Critical infrastructure agent

**References**

1. Guidotti, R.; Chmielewski, H.; Unnikrishnan, V.; Gardoni, P.; McAllister, T.; van de Lindt, J. Modeling the resilience of critical infrastructure: The role of network dependencies. *Sustain. Resilient Infrastruct.* 2016, 1, 153–168. [CrossRef]
2. Amini, M.H.; Imteaj, A.; Pardalos, P.M. Interdependent networks: A data science perspective. *Patterns* 2020, 1, 100003. [CrossRef]
3. S & C Electric Company, F.S. S & C’s 2018 State of Commercial & Industrial Power Reliability Report. 2013. Available online: https://www.sandc.com/globalassets/sac-electric/documents/sharepoint/documents---all-documents/technical-paper-100-t120.pdf (accessed on 23 April 2018).
4. Hussain, A. A Day Without Power: Outage Costs for Businesses. 2013. Available online: https://www.bloomenergy.com/blog/a-day-without-power-outage-costs-businesses (accessed on 8 October 2019).
5. Salehi, V.; Veitch, B.; Musharraf, M. Measuring and improving adaptive capacity in resilient systems by means of an integrated DEA-Machine learning approach. *Appl. Ergon.* 2020, 82, 102975. [CrossRef]
6. Dick, K.; Russell, L.; Souley Dosso, Y.; Kwamena, F.; Green, J.R. Deep learning for critical infrastructure resilience. *J. Infrastruct. Syst.* 2019, 25, 05019003. [CrossRef]
7. Alqahtani, A.; Abhishek, R.; Tipper, D.; Medhi, D. Disaster recovery power and communications for smart critical infrastructures. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
8. Amini, M.H.; Imteaj, A.; Mohammadi, J. Distributed Machine Learning for Resilient Operation of Electric Systems. In Proceedings of the 2020 International Conference on Smart Energy Systems and Technologies (SEST), Istanbul, Turkey, 7–9 September 2020; pp. 1–6.
9. Ilgen, S.; Sengers, F.; Wardekker, A. City-to-city learning for urban resilience: The case of water squares in Rotterdam and Mexico City. *Water* 2019, 11, 983. [CrossRef]
10. Imteaj, A.; Thakker, U.; Wang, S.; Li, J.; Amini, M.H. A Survey on Federated Learning for Resource-Constrained IoT Devices. *IEEE Internet Things J.* 2021. [CrossRef]
11. Cai, H.; Lam, N.S.; Qiang, Y.; Zou, L.; Correll, R.M.; Mihunov, V. A synthesis of disaster resilience measurement methods and indices. *Int. J. Disaster Risk Reduct.* 2018, 31, 844–855. [CrossRef]
12. Pursiainen, C. Critical infrastructure resilience: A Nordic model in the making? *Int. J. Disaster Risk Reduct.* 2018, 27, 632–641. [CrossRef]
13. Alemzadeh, S.; Talebian, H.; Talebi, S.; Dueñas-Osorio, L.; Mesbahi, M. Resource Allocation for Infrastructure Resilience using Artificial Neural Networks. In Proceedings of the 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Baltimore, MD, USA, 9–11 November 2020; pp. 617–624.
14. Amini, M.H.; Nabi, B.; Haghifam, M.R. Load management using multi-agent systems in smart distribution network. In Proceedings of the 2013 IEEE Power & Energy Society General Meeting, Vancouver, BC, Canada, 21–25 July 2013; pp. 1–5.
15. Wang, J.; Zuo, W.; Rhode-Barbarigos, L.; Lu, X.; Wang, J.; Lin, Y. Literature review on modeling and simulation of energy infrastructures from a resilience perspective. *Reliab. Eng. Syst. Saf.* 2019, 183, 360–373. [CrossRef]
16. Lee, E.H.; Choi, Y.H.; Kim, J.H. Real-time integrated operation for urban streams with centralized and decentralized reservoirs to improve system resilience. *Water* 2019, **11**, 69. [CrossRef]

17. Wang, Y.; Roussis, A.O.; Strbac, G. On microgrids and resilience: A comprehensive review on modeling and operational strategies. *Renew. Sustain. Energy Rev.* 2020, **134**, 110313. [CrossRef]

18. Arghandeheh, R.; Brown, M.; Del Rosso, A.; Ghatikar, G.; Stewart, E.; Vojdani, A.; von Meier, A. The local team: Leveraging distributed resources to improve resilience. *IEEE Power Energy Mag.* 2014, **12**, 76–83. [CrossRef]

19. Sujil, A.; Verma, J.; Kumar, R. Multi agent system: Concepts, platforms and applications in power systems. *Artif. Intell. Rev.* 2018, **49**, 153–182. [CrossRef]

20. Chen, C.; Xie, K.; Lewis, F.L.; Xie, S.; Davoudi, A. Fully distributed resilience for adaptive exponential synchronization of heterogeneous multistage systems against actuator faults. *IEEE Trans. Autom. Control* 2018, **64**, 3347–3354. [CrossRef]

21. Uemichi, A.; Yagi, M.; Okawa, R.; Yamasaki, Y.; Kaneko, S. Multi-objective optimization to determine installation capacity of distributed power generation equipment considering energy-resilience against disasters. *Energy Procedia* 2019, **158**, 6538–6543. [CrossRef]

22. Hossain, E.; Roy, S.; Mohammad, N.; Nawar, N.; Dipta, D.R. Metrics and enhancement strategies for grid resilience and reliability during natural disasters. *Appl. Energy* 2021, **290**, 110709. [CrossRef]

23. Hussain, A.; Bui, V.H.; Kim, H.M. Microgrids as a resilience resource and strategies used by microgrids for enhancing resilience. *Appl. Energy* 2019, **240**, 56–72. [CrossRef]

24. Gao, H.; Chen, Y.; Mei, S.; Huang, S.; Xu, Y. Resilience-oriented pre-hurricane resource allocation in distribution systems considering electric buses. *Proc. IEEE* 2017, **105**, 1214–1233. [CrossRef]

25. Imteaj, A.; Amini, M.H.; Mohammad, J. Leveraging decentralized artificial intelligence to enhance resilience of energy networks. In *Proceedings of the 2020 IEEE Power & Energy Society General Meeting (PESGM)*, Montreal, QC, Canada, 2–6 August 2020; pp. 1–5.

26. Mishra, D.K.; Ghadi, M.J.; Azizivahed, A.; Li, L.; Zhang, J. A review on resilience studies in active distribution systems. *Renew. Sustain. Energy Rev.* 2021, **135**, 110201. [CrossRef]

27. Ghiasi, M.; Dehghani, M.; Niknam, T.; Baghaee, H.R.; Padmanaban, S.; Gharehpetian, G.B.; Aliev, H. Resiliency/cost-based optimal design of distribution network to maintain power system stability against physical attacks: A practical study case. *IEEE Access* 2021, **9**, 43862–43875. [CrossRef]

28. Bhusal, N.; Abdelmalak, M.; Kamruzzaman, M.; Benidris, M. Power system resilience: Current practices, challenges, and future directions. *IEEE Access* 2020, **8**, 18064–18086. [CrossRef]

29. Colon, C.; Hallegatte, S.; Rozenberg, J. Criticality analysis of a country’s transport network via an agent-based supply chain model. *Nat. Sustain.* 2021, **4**, 209–215. [CrossRef]

30. Bellini, E.; Bellini, P.; Cenni, D.; Nesi, P.; Pantaleo, G.; Paoli, I.; Paolucci, M. An IoE and Big Multimedia Data Approach for Urban Transport System Resilience Management in Smart Cities. *Sensors* 2021, **21**, 435. [CrossRef]

31. Argyroudis, S.A.; Mitoulis, S.A.; Winter, M.G.; Kaynia, A.M. Fragility of transport assets exposed to multiple hazards: State-of-the-art review toward infrastructural resilience. *Reliab. Eng. Syst. Saf.* 2019, **191**, 106567. [CrossRef]

32. Pan, S.; Yan, H.; He, J.; He, Z. Vulnerability and resilience of transportation systems: A recent literature review. In *Physica A: Statistical Mechanics and Its Applications*; Elsevier: Amsterdam, The Netherlands, 2021; p. 126235.

33. Nik, V.M.; Moazami, A. Using collective intelligence to enhance demand flexibility and climate resilience in urban areas. *Appl. Energy* 2021, **281**, 116106. [CrossRef]

34. Elmqvist, T.; Andersson, E.; Frantzeskaki, N.; McPhearson, T.; Olsson, P.; Gaffney, O.; Takeuchi, K.; Folke, C. Sustainability and open challenges. In *Proceedings of the 2020 IEEE Power & Energy Society General Meeting (PESGM)*, Montreal, QC, Canada, 2–6 August 2020; pp. 1–5.

35. Cariolet, J.M.; Vuillet, M.; Diab, Y. Mapping urban resilience to disasters—A review. *Sustain. Cities Soc.* 2019, **51**, 101746. [CrossRef]

36. Rieckert, A.; Schuit, E.; Bleijenberg, N.; Ten Cate, D.; de Lange, W.; de Man-van Ginkel, J.M.; Mathijssen, E.; Smit, L.C.; Stalpers, D.; Schoonhoven, L.; et al. How can we build and maintain the resilience of our health care professionals during COVID-19? Recommendations based on a scoping review. *BMJ Open* 2021, **11**, e043718. [CrossRef]

37. Hines, S.E.; Chin, K.H.; Glick, D.R.; Wickwire, E.M. Trends in moral injury, distress, and resilience factors among healthcare workers at the beginning of the COVID-19 pandemic. *Int. J. Environ. Res. Public Health* 2021, **18**, 488. [CrossRef]

38. Setiawati, Y.; Wahyuhadi, J.; Joestandari, F.; Maramis, M.M.; Atika, A. Anxiety and resilience of healthcare workers during COVID-19 pandemic in Indonesia. *J. Multidiscip. Healthc.* 2021, **14**, 1. [CrossRef] [PubMed]

39. Jung, J.; Maeda, M.; Chang, A.; Bhandari, M.; Ashapure, A.; Landivar-Bowles, J. The potential of remote sensing and artificial intelligence as tools to improve the resilience of agriculture production systems. *Foods* 2021, **10**, 497. [CrossRef]

40. Kusiak, A. Open manufacturing: A design-for-resilience approach. *Int. J. Prod. Res.* 2020, **58**, 4647–4658. [CrossRef]

41. Boyac-Gündüz, C.P.; Ibrahim, S.A.; Wei, O.C.; Galanakis, C.M. Transformation of the Food Sector: Security and Resilience during the COVID-19 Pandemic. *Foods* 2021, **10**, 497. [CrossRef] [PubMed]

42. Khan, L.U.; Saad, W.; Han, Z.; Hossain, E.; Hong, C.S. Federated learning for internet of things: Recent advances, taxonomy, and open challenges. In *IEEE Communications Surveys & Tutorials*; IEEE: Piscataway, NJ, USA, 2021.
43. Mothukuri, V.; Khare, P.; Parizi, R.M.; Pouriyeh, S.; Dehghantanha, A.; Srivastava, G. Federated Learning-based Anomaly Detection for IoT Security Attacks. *IEEE Internet Things J.* 2021. [CrossRef]

44. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Agüera Arcas y, B. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Ft. Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.

45. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *arXiv* 2018, arXiv:1812.06127.

46. Chen, W.; Bhardwaj, K.; Marculescu, R. Fedmax: Mitigating activation divergence for accurate and communication-efficient federated learning. *arXiv* 2020, arXiv:2004.03657.

47. Imteaj, A.; Amini, M.H. FedAR: Activity and Resource-Aware Federated Learning Model for Distributed Mobile Robots. In Proceedings of the 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 14–17 December 2020.

48. Hegde, R.; Mukherjee, K.; Gupta, S.T. Electro-Maps. 2020. Available online: https://github.com/sabm0hmayhai/Electro-Maps (accessed on 10 May 2021).