Evaluation of Thermal Stress on IoT-based Federated Learning

Yi Gu
Kennesaw State University
Marietta, Georgia, USA
ygu5@students.kennesaw.edu

Liang Zhao
Kennesaw State University
Marietta, Georgia, USA
lzhao10@kennesaw.edu

Bobin Deng
Kennesaw State University
Marietta, Georgia, USA
bdeng2@kennesaw.edu

Shaoen Wu
Kennesaw State University
Marietta, Georgia, USA
swu10@kennesaw.edu

ABSTRACT

Federated Learning is a novel paradigm allowing training of a global machine-learning model on distributed devices. It shares model parameters instead of the private raw data during the entire model training process. While Federated Learning enables machine learning processes to take place collaboratively on the Internet of Things (IoT) devices, compared to data centers, IoT devices with limited resource budgets typically have less security protection and are more vulnerable to potential cyber-attacks. Current research on the evaluation of Federated Learning is mainly based on simulation of multi-clients/processes on a single machine/device. However, there is a gap in understanding the performance of Federated Learning under cyber-attacks in real-world distributed low-power IoT devices. In this paper, we are among the first to evaluate the performance of Federated Learning under thermal stress on real-world IoT-based distributed systems. We conducted comprehensive experiments using the CIFAR-10 dataset and various performance metrics including training time, CPU and GPU utilization rate, temperature, and power consumption. The experimental results demonstrate that thermal stress is effective on IoT-based Federated Learning systems as the entire global model and device performance degrade when even a small ratio of IoT devices are being impacted.

CCS CONCEPTS

• **Computing methodologies** → *Distributed computing methodologies*; • **Hardware** → **Thermal issues**; • **Computer systems organization** → **Embedded systems**.

KEYWORDS

Thermal Stress, Federated Learning Systems, Internet of Things

ACM Reference Format:
Yi Gu, Liang Zhao, Bobin Deng, and Shaoen Wu. 2024. Evaluation of Thermal Stress on IoT-based Federated Learning. In *2024 ACM Southeast Conference (ACMSE 2024), April 18–20, 2024, Marietta, GA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3603287.3651222

ACMSE 2024, April 18–20, 2024, Marietta, GA, USA

ACM ISBN 979-8-4007-0237-2/24/04.
https://doi.org/10.1145/3603287.3651222

1 INTRODUCTION

Training a machine learning model conventionally requires transferring the raw data to a central place for computation and analytics. However, certain circumstances can limit the usage of the centralized training. Examples of that can be a too-big-to-centralize data set, data that users prefer not to share, or certain regulations restricting the collection of data. Federated Learning was first initiated by Google [18] in April 2017. Originally tested in Gboard on Android phones, Federated Learning makes it possible to carry out machine learning processes collaboratively on distributed entities such as the Internet of Things (IoT) devices. In Federated Learning, machine learning models are trained locally on clients and only the gradients/model parameters will be sent to the server for aggregation. After years of real-world practice, Federated Learning is proven to allow model training in distributed and heterogeneous devices. In addition, the feature of sharing model parameters instead of the raw data made Federated Learning a promising solution when protecting user privacy during machine learning is becoming more important recently. However, as Federated Learning is performed in distributed systems with multiple potentially untrusted devices, it is easier for the Federated Learning System (FLS) to be impacted and influenced [11]. In particular, when clients in FLS are low-end IoT devices with limited resources, impacts like thermal stress can easily prevent clients from offering their best performance.

Computing devices consume energy and produce heat as side effects during computation. However, if excessive heat is generated and cannot be dissipated in time, then it can cause reductions in performance or even physical damage to computing devices. Thermal stress, is a sort of impact which has various ways of implementation of creating excessive heat and using the heat to cause damage to the performance or even construction of the impacteded system. When it comes to Federated Learning, thermal stress will try to prevent the FLS from performing normally, thus leading to adverse consequences including dropping in CPU/GPU frequency, extended training time and increased power consumption, leading to reduction in the system performance. In this paper, we constructed a real-world FLS on IoT devices based on Flower framework [4]. We simulated thermal stress on our FLS clients using Jetson Benchmarks [2]. We evaluated the performance with various measurements and metrics including Jetson-Stats [6], and analyzed the influence of thermal stress on our FLS. Our prominent contributions in this paper are summarized as follows.

- To the best of our knowledge, the presenting work is among the first to evaluate the performance of Federated Learning under thermal stress on real-world IoT-based systems.
- We conducted experiments of various scenarios when Federated Learning clients are under thermal stress with measuring metrics including CPU utilization rate, GPU utilization rate, temperature, and power consumption.
- We varied the proportion of clients under stress in each group of experiments and systematically quantified the effectiveness and real-world impact of thermal stress on the low-end IoT-based Federated Learning System.

2 RELATED WORK

Federated Learning. Federated Learning was first initiated by Google [18] in April 2017 and was applied on Gboard on Android phones for test and training. IoT devices like mobile phones, pads, and others tend to have data, especially privacy-related data, contained in their memory. If such data is collected in one place for traditional machine learning, not only will the collection process be resource-consuming, but there is also the risk of privacy vulnerability to those data. Different from traditional machine learning, Federated Learning trains the global model collaboratively on IoT devices in which data are located, transferring only trained model parameters and gradients instead of data to the server. Parameters and gradients will then be processed at server using certain algorithm. McManhan et al. [17] was the first to initiate the Federated Average algorithm (aka FedAVG), which later became a basic and important method to aggregate updates received at the server. Google proposed TFF [10] and Bonawitz et al. proposed Fedscale [5], both frameworks support single-node simulation. Ryffel et al. proposed PySyft [20] and Beutel et al. proposed Flower [4]. They support more functions like multi-node execution and heterogeneous computation.

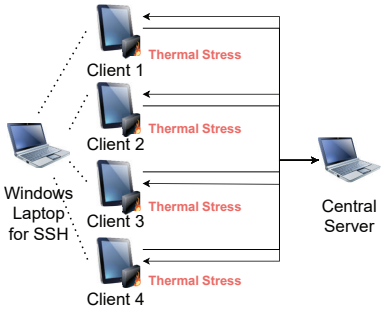


Figure 1: Schematic Diagram of Federated Learning System Under Thermal Stress

Thermal Stress. Thermal stress can weaken the performance or even damage the structure of the stressed system. Certain consequences can be communication conversion, excessive energy consumption, heat-led hardware damage, and so on. Previous research has shown different unfavorable results that can be caused by thermal stress from different perspectives. Masti et al. [16] performed RSA decryption on specific CPU cores on edge devices, as well as using thermal side channels for communication, showing thermal stress and its possibility of influencing the work of edge devices and communication between edge devices. Tian et al. [23] also found out similar phenomenon that between users who are renting the same FPGA over a period of time, certain thermal channels can be manipulated to converted channels and lead to manipulation in communication. Kong et al. [13] found that certain malicious commands can lead to fine-grained and specified over-temperature spots, thus causing certain physical damage in the instruction cache. Gao et al. [8] used workloads that can cause excessive thermal to rise temperature in data centers to a terribly high situation, conducted measures on thermal and proposed effective thermal stress vectors, all to reveal the vulnerability of data centers and likely areas when facing thermal stress. In a follow-up work, Gao et al. conducted additional testing on thermal stress under various scenarios and with thermal-related metrics measured [9]. Duchatellier et al. [7] studied the effect of thermal stress on edge devices and the vulnerability of cloud-edge systems. Jaspinder et al. [12] studied RSPP and related thermal side-channel attacks and their influence.

In the literature, only a certain amount of work focuses on the FLS which is deployed on real-world IoT devices, and even less work has been done to evaluate the influence of thermal stress on FLS so far. Our work is among the first attempts of carrying out thermal stress on heterogeneous IoT-based FLS. We consider both FLS and thermal stress aiming to understand the effect and impact of thermal stress on real-world IoT-based FLS.

3 SYSTEM DESCRIPTION AND STRESS MODEL

In an FLS, clients train a global model collaboratively under the coordination of a central server. Each client trains its local data for a local model, and the central server carries out weighted aggregation of local models to formulate a global model. An iteration of Federated Learning is as follows. (1) All clients download the global model W_{t-1} from the central server. (2) Client k trains its local data to obtain a local model $W_{t,k}$ (local model for client k in the t-th round of communication). (3) All clients upload updated local model parameters and gradients to the central server. (4) After receiving all data, the central server carries out a weighted aggregation using algorithms like FedAVG or FedBN to obtain the global model named W_t (global model in the t-th round of communication). After multiple rounds of iterations, a final model W will be produced, which is close to the results of centralized machine learning under the same model as the global model using the same dataset.

Figure 2: Our Federated Learning System Experiment Setup

Our thermal stress aim to produce excessive heat within the FLS and use the heat or other side-effects of the heat to influence system

performance or damage the system. In FLS, clients are typically deployed on devices with more limited resources than the central server, thus making clients more vulnerable than the server in terms of stress. Heat is generated on IoT devices mainly when computing is performed. On the same device and for the same length of time, the more resources-demanded one program is, the more excessive its thermal will be produced. To produce excessive heat, we choose to use certain programs which are highly resource-demanded. Such programs are loaded on IoT devices that serve as clients and run on the clients at the same time the FLS is running. By doing so we simulate thermal stress to our FLS. Considering the four steps in an iteration of Federated Learning, the simulated thermal stress will influence steps one to three mentioned above. Figure 1 demonstrates the general framework of our FLS and simulated thermal stress on the system. Using the methods described above, we can have thermal stress on FLS can be simulated and evaluated.

4 EXPERIMENTAL EVALUATION

4.1 Evaluation Methodology

To create a real-world multi-node FLS and see how it performs under thermal stress, we choose to use the Flower framework. In this paper, we chose four Jetson Nanos as clients and a Lambda Laptop as the central server, using gRPC Protocol to conduct c-s communication. We also chose another Windows laptop to run SSH on the four Jetson Nanos for the convenience of our control. Figure 2 shows the real-world picture of our FLS clients, with each Nano in the picture reflecting one of the clients.

As we can see from the figure, we name the four Jetson Nanos as Nano1, Nano2, Nano3, and Nano4. We name the Lambda Laptop as Lambda. The configurations of the five devices are as in Table 1. Note that Nano1 and Nano2 share the same configuration, Nano3 and Nano4 also share the same configuration. We just list out Nano1 and Nano3 in the following list to remove duplicates. Also, considering the Windows laptop serves only for SSH, its configuration is of no effect to our framework and thus won't be mentioned.

Table 1: The Configuration of Devices Running Federated Learning System

	Lambda	Nano1-2	Nano3-4
CPU	Intel i7-12800H	ARM A57	ARM A57
CPU Core Num	14	4	4
GPU	RTX 3080 Max-Q	Maxwell	Maxwell
GPU Core Num	6144	128	128
FAN	Yes	Yes	No
MEM	32GB	4GB	4GB
Disk	1TB	128GB	128GB

In this paper, we choose to use the "embedded-devices" example on Flower's GitHub official website [1] as our FLS example. For the server, the only thing to do is get server files ready and have certain environments installed properly as required in requirements.txt. For an FLS client, the first thing to do is to install JetPack 4.6.1 on the 128GB microSD card. Note that the storage of the micro SD card should not be below 64 GB, or certain issues of not having enough storage left will occur in later stages. Set the device following system instructions step by step. Create a new user group in the complimentary docker with NVIDIA jetson and add a new user to it. After those, get the FLS client's files ready and run them in the terminal to create a docker image for FLS clients. Later, FLS clients will be running in the docker images. Also, considering the latest version of CUDA supported on our Jetson Nanos is outdated, we chose to use CPU to run clients.

For this paper, we have clients training a MobileNet-v2/3 model under Pytorch [19] framework. We also choose to use CIFAR-10 [14] as our dataset. Designed for image classification, the dataset contains 60k images in 10 classes, with 50k of them as the training set. The 50k image dataset is then evenly split into 50 partitions, each assigned to a different client. The training rounds are assigned at the server side and are set as 3, the training epoch in each round is set to 2. Also, we set it on the server side to make sure all clients connected to the server will be sampled.

4.2 Thermal Stress Simulation

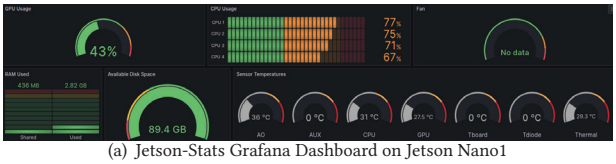
Of all the parts in an IoT device, the CPU and GPU are technically parts that carry out high-computing-resource-demanded programs, thus making them the majority of parts to conduct excessive heat. To simulate our thermal stress, we can choose to deploy high-CPU-consuming or high-GPU-consuming programs while the FLS is working. However, during our preliminary pilot experiments, we found that high-CPU-consuming programs while the FLS is working always lead to overload CPU and system breakdown for our Jetson Nanos. A possible reason for that might be our FLS clients and high-CPU-consuming programs all work on our CPU and this led to insufficient computing resources left for the system to operate normally.

In order to have thermal stress and our FLS running at the same time for better evaluation, we decided to choose jetson-benchmarks [2] as our high-GPU-consuming programs for simulation. Jetson benchmarks are official benchmarks provided by Nvidia, including Inception V4, ResNet-50, OpenPose, VGG-19, YOLO-V3, Resolution and Unet. All benchmarks work using GPU+2DLA and are originally designed to test the extreme performance of Jetson devices, which makes Jetson benchmarks the perfect choice for our simulation. To execute these benchmarks, first, we have to get the requirements for benchmarks to run ready. The next step is to download models and a CSV file that contains all parameters for models. The last step is to run the benchmark scripts using the terminal and all is set. A typical running time for running all benchmarks on Jetson Nano shall be no less than two hours. One more thing to note is that those high-computation-resources consuming programs will only be deployed on our clients of the FLS.

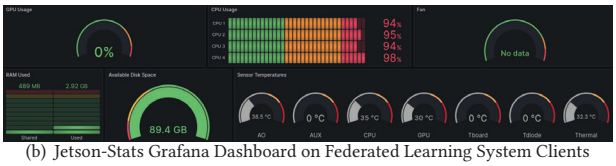
4.3 Measurements of Metrics

Several metrics of FLS Clients have been taken into consideration and tools have been chosen to monitor these metrics. The goal of such measurements is to find out the performance of FLS clients running normally or the performance of FLS clients under thermal stress to see the influence of thermal stress.

4.3.1 Training Time. As a complimentary FLS client function, FLS clients will automatically print the training time of every epoch and the evaluation time of every epoch on its training log. By



(a) Jetson-Stats Grafana Dashboard on Jetson Nano1



(b) Jetson-Stats Grafana Dashboard on Federated Learning System Clients

Figure 3: Jetson-Stats Grafana Dashboards

documenting the training log, we can get the training time of every client under or not under thermal stress and see the difference.

4.3.2 CPU and GPU Utilization Rate and Total Energy Consumption. To find out the effect of the thermal stress, the utilization rate of FLS clients can be documented and analyzed. As our FLS runs on CPU and thermal stress runs on GPU, the utilization rate of CPU and GPU can be key value to see how thermal stress will affect our clients. Also, considering thermal stress is highly computing-resources-demanded and thus might cost extra energy consumed, total energy consumption (TOT) can be another key metric to see the influence of thermal stress. All three can be documented by jtop-logger provided by jetson-stats [6], which is a Python file and automatically logs the condition of our FLS client's system into a CSV file on a one-second per log basis.

We also have another timely displayed Jetson-Stats-Grafana-Dashboard [22] to show the real-time utilization rate of CPU, GPU, TOT, and other metrics. The dashboard first collects Jtop gathered data using certain API provided and uses certain scripts to regulate data into certain forms that can be accepted and then transfers those data to the host running Prometheus [3]. Then it has another host running Grafana [15], which is the platform for our dashboard. By importing certain dashboard distribution files and pulling data from Prometheus, Grafana can offer a timely display of utilization rate in certain metrics. We also have modified the Jetson-Stats-Grafana-Dashboard for it to fit our Jetson Nano clients. Figure 3 is our Jetson-Stats-Grafana-Dashboard on Jetson Nano1 without any other program running, and Figure 4 is the dashboard of our FLS clients while training local model.

4.3.3 Temperature. As we are conducting thermal stress, temperature plays a vital part in all the metrics. By evaluating the temperature of FLS clients under or not under thermal stress, we can easily get to know how FLS clients are influenced under thermal stress from this perspective. We also choose to use Jtop to log temperature-related data and use Jetson-Stats-Grafana-Dashboard for real-time display.

4.4 Experiment Process and Outcomes

In this paper, we set four groups of experiments to see how thermal stress influence the FLS and if changes in the number of clients under stress have any influence on the FLS. Each group has four clients connected to the server, running three rounds of training and two epochs in each. After every epoch, an evaluation will be processed. The dataset we chose is CIFAR-10 [14], set and distributed evenly into 50 shares and have one share distributed to one client as described above. For the first group, all clients are running Federated Learning without thermal stress. For the second group, one client (Nano1) is running Federated Learning under thermal stress while the other three clients are running Federated Learning without thermal stress. For the third group, two clients (Nano1 and Nano2) are running Federated Learning under thermal stress while the other two clients are running Federated Learning without thermal stress. For the fourth group, three clients (Nano1, Nano2, and Nano4) are running Federated Learning under thermal stress while the other client is running Federated Learning without thermal stress. Another group of experiments was also planned, in which all clients are running Federated Learning under thermal stress. However owing to the vulnerability of edge devices, the experiment failed to execute under thermal stress. Have jtop-logger and Jetson-Stats-Grafana-Dashboardand ready for every device to log and display real-time data of metrics of clients, and we are all set for our experiments. One more thing to mention is that we need to download the log of each client in their terminal for the running time after every group of experiments has ended. We also have a 5V 4A powerline for every client.

From the first to the fourth group of experiments, all experiments were carried out successfully. However, we failed in running Grafana dashboard on any client under thermal stress, with any attempt ending in stuck and crash condition. For utilization rates and temperate, we selected the results from the first two epochs in the first round and listed them in a group of figures from Figure 4 to Figure 7. For training time and accuracy, we have detail training and evaluation time of epoch 1 and 2 from round 1 listed in Table 2, and have general time and accuracy data listed in Table 3.

5 DISCUSSION AND ANALYSIS

5.1 Impact On CPU and GPU Utilization Rate

Figure 4 shows that as long as Federated Learning System clients are running their training rounds, their CPU utilization rates, with or without thermal stress, are all nearly 100%. If clients are not under thermal stress, then typically they will have a much lower CPU utilization rate while running the evaluation round.

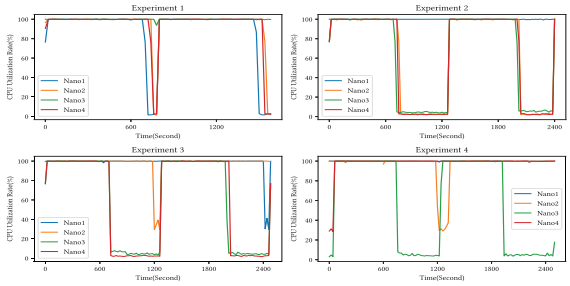


Figure 4: CPU Utilization Rate of Round 1

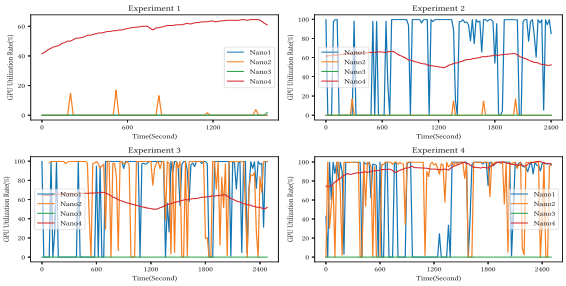


Figure 5: GPU Utilization Rate of Round 1

Note that clients without thermal stress have to wait for clients under thermal stress to finish their training, so during the waiting time, clients without thermal stress will also have less CPU utilization rate. If a client is under thermal stress, then most of the time its CPU utilization rate will be around 100%, no matter whether it is running training rounds or evaluation.

As shown in Figure 5, clients without thermal stress typically have very little GPU utilization rate, with only a short period of time utilization rate peaks appearing. Clients under thermal stress tend to have up-and-downs in the GPU utilization rate, but most of the time the utilization rate is in high condition.

5.2 Temperature and Power Consumption

When it comes to temperature, as is in Figure 6, some normal FLS clients without thermal stress, like Clients 1 and 2 in Experiment 1, have a stable temperature of around 40 Celsius. However, other normal FLS clients like Clients 3 and 4 have trends to fluctuate between 40 Celsius and 70 Celsius. We can also see from Experiment 4 that while Clients 1 and 2 are under thermal stress, they tend to have temperatures around 50 Celsius and are less stable when they are not under stress. Client 4 can have 80 to 100 Celsius while under thermal stress. The possible reason could be Clients 1 and 2 are with fans while Clients 3 and 4 don't. Also, we can note that as the case shown in CPU utilization rate, while there are clients in an FLS under thermal stress, other clients without stress tend to have temperature drops while waiting for the under-stress clients to finish their training rounds and while running evaluations.

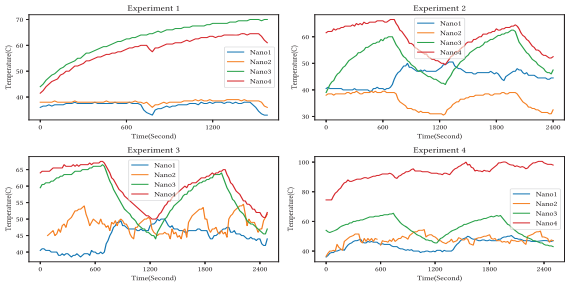


Figure 6: Temperature Trend of Round 1

For power consumption (see Figure 7), normal clients running Federated Learning Syetem training rounds tend to be around 5000 mW power while working. Power consumption can go down to 2000 mW if normal clients are running evaluations or are waiting for other under-stress clients to finish their training. Clients under thermal stress, however, tend to have a power consumption of more than 5000 mW, fluctuate between 5000 and 9000 mW typically, and sometimes it can go to 10000 mW and even 12000 mW. We can also see from Experiment 4 that while under the same thermal stress, Client 4 consumes more power than Clients 1 or 2, and the possible reason could be Clients 1 and 2 are with fans while Client 4 doesn't.

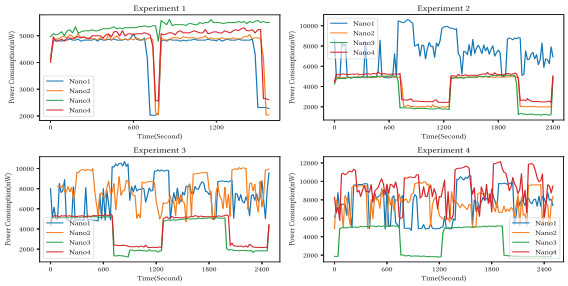


Figure 7: Power Consumption of Round 1

Table 2: Training and Evaluating Time for Round 1 Clients

	E1 TRN	E1 EVA	E2 TRN	E2 EVA
EXP1Nano1	11:30	00:12	11:21	00:11
EXP1Nano2	12:20	00:13	12:19	00:13
EXP1Nano3	12:57	00:14	13:03	00:13
EXP1Nano4	12:10	00:12	12:03	00:12
EXP2Nano1	21:11	00:19	18:16	00:23
EXP2Nano2	12:06	00:13	12:08	00:12
EXP2Nano3	11:36	00:11	11:34	00:12
EXP2Nano4	11:59	00:14	12:00	00:14
EXP3Nano1	21:10	00:20	18:11	00:20
EXP3Nano2	19:37	00:19	19:25	00:20
EXP3Nano3	11:39	00:12	11:29	00:12
EXP3Nano4	11:43	00:12	12:07	00:13
EXP4Nano1	22:05	00:26	18:47	00:20
EXP4Nano2	19:43	00:23	18:50	00:23
EXP4Nano3	11:15	00:12	11:27	00:12
EXP4Nano4	19:16	00:18	20:56	00:19

Table 3: Running Time and Accuracy for Experiments

	Running Time	Acc RND1	Acc RND2	Acc RND3
EXP1	1:09:17	0.2735	0.2950	0.3255
EXP2	2:01:41	0.2675	0.2765	0.3025
EXP3	2:06:32	0.2455	0.2555	0.2835
EXP4	2:08:28	0.2165	0.2315	0.2565

5.3 Impact On Training Time and Accuracy

We observe from Tables 2 and 3 that as long as there is any client under thermal stress in the system, the running time of each round and in general is almost doubled. We can further conclude from

Table 3 that the more ratio of clients are under thermal stress, the longer the training time is and the less accuracy the trained models have. Last but not least, while processing the data, we also found out that sometimes the log of system status on clients under thermal stress is missing for a few tiny periods. After looking into the logs of jtop-loggers, we found that it is most likely because thermal stress took too many resources in those tiny periods of time and there is so little left for jtop-loggers to function normally.

5.4 Analysis and Insights

From the experimental results, we found that thermal stress will cause nodes in FLS to have considerably unnecessary CPU and GPU utilization rates like 100%. They also cause the temperature of nodes in the FLS to rise around 80% on average and even rise to as high as 100 Celcius. Moreover, thermal stress and the excessive heat conducted could lead to a rise in node's power consumption from 60% to even 140%. Also, nodes without fans tend to have a higher temperature and power consumption compared with nodes with fans while they are all under the same thermal stress.

When any node is under stress, all other clients have to wait for the specific client to finish its training before moving to the next round, which increases training time. However, if just the ratio of clients under stress rises when clients are already under stress, the increase in time is not obvious. When only one node is under stress, in the aspect of accuracy, the performance of Federated Learning is decreased by about 8%. When the ratio of clients under stress increases to 50% or 75%, the performance will go down by 13% and even 21%. When all the clients are under stress, some nodes stop, causing our FLS to fail to work normally. We find out that thermal stress can seriously influence our FLS, thus leading to higher utilization of CPU and GPU and temperature, more consumed power, exceeding of training time, decreases in model accuracy, and even preventing FLS from performing normally and influencing the system robustness.

Based on the experimental results, we suggest adding a data-driven anomaly detection system [21] to Federated Learning Systems. It should focus on abnormal utilization rates, temperature, and power consumption to detect thermal stress.

6 CONCLUSION

In this paper, we are among the first to evaluate the performance of Federated Learning on real-world IoT-based systems under thermal stress. We used high GPU-consuming programs to simulate thermal stress and varied ratios of clients under stress in the system to see the influence of thermal stress. Extensive experiments results have shown that thermal stress can cause low-end IoT-based Federated Learning Systems to have nearly doubled training time, as much as 80% higher in temperature, more exceeded heat conducted, an average of 167% more power consumption made, and even threats to system robustness. Future work includes evaluations on larger networks and the performance under non-IID data distributions among IoT devices.

REFERENCES

[1] Afermarq and Tanertopal. 2023. Federated Learning on Embedded Devices with Flower. https://github.com/adap/flower/tree/main/examples/embedded-devices
[2] AsawareeBhide and Amey Kulkarni. 2022. NVIDIA-AI-IOT Jetson_Benchmarks. https://github.com/NVIDIA-AI-IOT/jetson_benchmarks
[3] Prometheus Authors. 2023. Prometheus Monitoring Systems and Time Series Database. https://prometheus.io/
[4] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, et al. 2020. Flower: A Friendly Federated Learning Research Framework. *arXiv preprint arXiv:2007.14390* (2020).
[5] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards Federated Learning at Scale: System Design. arXiv:1902.01046 [cs.LG]
[6] Raffaello Bonghi. 2023. Jetson-Stats. https://github.com/rbonghi/jetson_stats
[7] Justin Duchatellier, Tyler Holmes, Kun Suo, and Yong Shi. 2021. An Empirical Study of Thermal Attacks on Edge Platforms. In *Proceedings of the 2021 ACM Southeast Conference*. New York, United States, 175–179.
[8] Xing Gao, Zhang Xu, Haining Wang, Li Li, and Xiaorui Wang. 2017. Why "Some" Like It Hot Too: Thermal Attack on Data Centers. In *Proceedings of the 2017 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*. Urbana-Champaign, USA, 23–24.
[9] Xing Gao, Zhang Xu, Haining Wang, Li Li, and Xiaorui Wang. 2018. Reduced Cooling Redundancy: A New Security Vulnerability in a Hot Data Center. In *NDSS*. San Diego, USA.
[10] Google. 2020. Tensorflow Federated: Machine Learning on Decentralized Data. https://www.tensorflow.org/federated
[11] Ala Gouissem, Khalid Abualsaud, Elias Yaacoub, Tamer Khattab, and Mohsen Guizani. 2022. Federated Learning Stability Under Byzantine Attacks. In *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. Austin, USA, 572–577. https://doi.org/10.1109/WCNC51071.2022.9771594
[12] Jaspinder Kaur and Shirshendu Das. 2024. RSPP: Restricted Static Pseudo-Partitioning for Mitigation of Cross-Core Covert Channel Attacks. *ACM Trans. Des. Autom. Electron. Syst.* 29, 2, Article 27 (jan 2024), 22 pages. https://doi.org/10.1145/3637222
[13] Joonho Kong, Johnsy K John, Eui-Young Chung, Sung Woo Chung, and Jie Hu. 2010. On the Thermal Attack in Instruction Caches. *IEEE Transactions on Dependable and Secure Computing* 7, 2 (2010), 217–223.
[14] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. [n. d.]. CIFAR-10 (Canadian Institute for Advanced Research). ([n. d.]). http://www.cs.toronto.edu/~kriz/cifar.html
[15] Grafana Labs. 2023. Grafana: The Open Observability Platform. https://grafana.com/
[16] Ramya Jayaram Masti, Devendra Rai, Aanjhan Ranganathan, Christian Müller, Lothar Thiele, and Srdjan Capkun. 2015. Thermal Covert Channels on Multi-core Platforms. In *24th USENIX security symposium (USENIX security 15)*. Washington D.C., USA, 865–880.
[17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*. PMLR, Ft. Lauderdale, USA, 1273–1282.
[18] Brendan McMahan and Daniel Ramage. 2017. Federated Learning: Collaborative Machine Learning without Centralized Training Data. https://blog.research.google/2017/04/federated-learning-collaborative.html
[19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Curran Associates Inc., Red Hook, NY, USA.
[20] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. 2018. A Generic Framework for Privacy Preserving Deep Learning. *arXiv preprint arXiv:1811.04017* (2018).
[21] Betty Saridou, Gueltoum Bendiab, Stavros N Shiaeles, and Basil K Papadopoulos. 2021. Thermal Management in Large Data Centres: Security Threats and Mitigation. In *Security in Computing and Communications: 8th International Symposium, SSCC 2020, Chennai, India, October 14–17, 2020, Revised Selected Papers 8*. Springer, Chennai, India, 165–179.
[22] Svcavallar. 2021. Jetson-Stats-Grafana-Dashboard. https://github.com/svcavallar/jetson-stats-grafana-dashboard
[23] Shanquan Tian and Jakub Szefer. 2019. Temporal Thermal Covert Channels in Cloud FPGAs. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. Seaside, USA, 298–303.