Robust Training on the Edge: Federated vs. Transfer Learning for Computer Vision in Intelligent Transportation Systems

Sergei Chuprov, Raman Zatsarenko, Dmitrii Korobeinikov, and Leon Reznik
Rochester Institute of Technology, Rochester, NY, USA,
Email: sc1723@rit.edu, rz4983@rit.edu, dk9148@rit.edu, leon.reznik@rit.edu

2024 IEEE World AI IoT Congress (AIIoT) | 979-8-3503-8780-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIIoT61789.2024.10578970

***Abstract*—The synergy between the Internet of Things and Edge AI is revolutionizing industries by enabling real-time data processing on devices at the network's edge, like sensors in Intelligent Transportation Systems (ITSs). However, a key challenge arises when deploying foundation Machine Learning (ML) models, trained on high quality data, on these Edge AI systems often subjected to Data Quality (DQ) variations. In this paper, we address this challenge by leveraging Transfer Learning (TL) and Federated Learning (FL) as strategies to mitigate the impact of DQ variations on ML application performance. While these strategies were not originally designed for this purpose, our findings demonstrate that both TL and FL can effectively enhance the robustness of ML applications in ITS scenarios that involve running ML processes on edge devices. We showcase this through a real-world traffic sign detection application, analyzing how TL and FL can be employed to improve model robustness against variations in DQ typically encountered by edge devices in ITS. We found that when high-quality data only is available for re-training, FL with Geometric Median aggregation allowed to train models performing on average by 20% better than in the TL scenario. Our results demonstrated that employing Geometric Median aggregation in FL allowed to increase accuracy by 6.7% on average across the all the considered cases in comparison to Federated Average aggregation. Additionally, employing varying DQ for re-training helps to further enhance ML performance if the application's operation scenario involves high dynamics in the quality of input data.**

***Index Terms*—Federated learning, data quality, Internet of Things, Edge AI, computer vision**

I. Introduction

The current increasing employment of Machine Learning (ML) applications on various Internet of Things (IoT) devices has become renowned as Edge AI concept. Within the domain of Intelligent Transportation Systems (ITSs), ML systems integrated with edge devices have found their applications to address a wide variety of problems, such as energy management strategies for hybrid electric vehicles [13]; road traffic management to reduce accidents and environmental impact [1], [6], [19]; and pedestrian routing assessment to accommodate better design and planning of urban environments [24]. The synergy between IoT and Edge AI plays a critical role in maintaining various ITS applications and processes. The edge devices, running ML applications, heavily rely on the data collected in real-time by distributed sensors, such as cameras or LiDARs. In order to get deployed in ITSs, which often involve complex, dynamic, and uncertain operation scenarios, ML applications have to be resilient, adaptable, and robust to Data Quality (DQ) variations [4], [30]. Ensuring the robustness of these applications during their execution stage becomes crucial for their proper operation, as it significantly impacts the performance and reliability of Edge AI systems. In this paper, we refer to robustness as to the ability of an ML model application to maintain its performance over its execution even in the case when the input DQ drops down after an initial training was completed.

Unfortunately, the foundation computer vision models employed in today's Edge AI systems typically exploit models pre-trained on good quality data only [14]. Their performance, while executed in real-life scenarios with varied DQ, may demonstrate a significant decline. To ensure the robustness of ML applications, employed in Edge AI systems, against DQ variations, it is necessary to handle and adapt to these effects while minimizing their impact on learning and inference performances. In order to enhance learning efficiency and ML generalizability, the approach to adapt already pre-trained models to a target knowledge domain have become favorable [31]. Generalizable and robust ML models are crucial for Edge AI applications as they can adapt well to input DQ shifts, which are not rare in dynamic sensor and IoT networks.

An investigation of improving ML robustness in Edge AI systems' design does deserve further research. In [3], we presented the employment of Federated Learning (FL) in design of applications that rely on edge devices for training the ML models for ITS. In this paper, we methodologically extend our research by examining, verifying, and analyzing Transfer Learning (TL) in addition to FL as a strategy to address DQ variations in real-life data. Though both FL and TL techniques were not originally designed to mitigate DQ drifts, but as our empirical study proves, can effectively enhance ML applications robustness. Our goal in this paper is to demonstrate that despite TL and FL distinct natures and original purposes, they both are effective in improving ML application robustness against DQ variations in applications that involve edge devices for training and running ML models. However, their selection will depend on the available resources and other conditions.

This work was supported in part by the National Science Foundation, USA, award #2321652

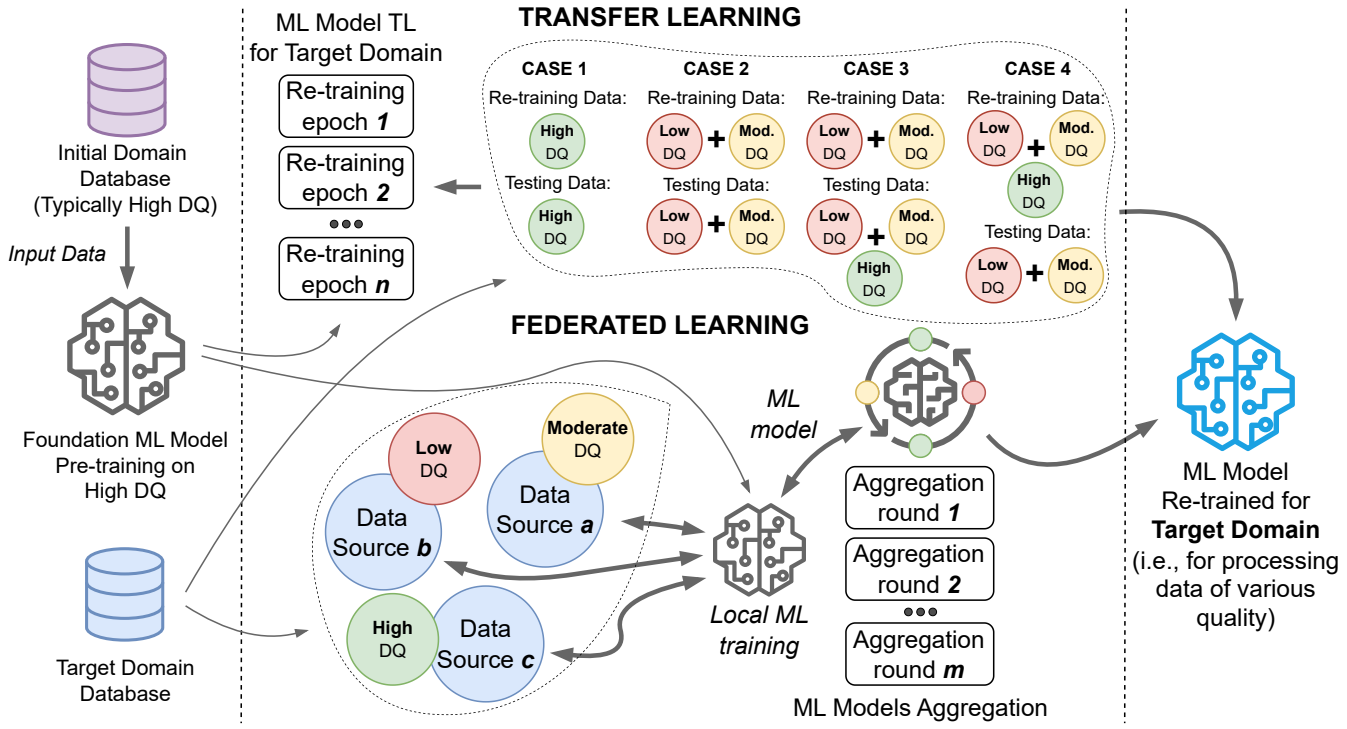


Fig. 1. Schematic representation of the conceptual differences and similarities between the TL and FL empirical study setups, employed in this paper. The upper part of the Figure depicts the studied TL cases on using the various DQ combinations for ML model re-training. The bottom part portrays how the training progress is organized in the FL setup. Both TL and FL techniques employ the pre-trained foundation computer vision model, and then this model is re-trained in order to achieve higher ML robustness to DQ variations in the ML application scenario with dynamic input DQ

As contributions additional to [3], we extend our empirical study by investigating the effect of TL against DQ variations in ITS scenario, where Road Side Units (RSUs) act as edge devices. We focus on a real-world use case with the ML computer vision application to detect traffic signs, specifically dealing with varied DQ due to network Quality of Service (QoS) decrease. Figure 1 demonstrates similarities and differences between TL and FL in our ITS application setup. In contrast to centralized training, when all the data is accumulated by a single unit and then a single ML model is trained, training in a FL manner allows to significantly reduce the communication burdens by lowering the amounts of data transmitted over the network in ITS. Switching to FL makes edge devices responsible for the data collection and local model training, which allows to avoid DQ variations caused by the network disruptions while transmitting all the data to a central processor.

The scope of our research is to investigate how such learning and re-training techniques as TL and FL contribute to improving the robustness of ITS computer vision applications for traffic sign detection and classification to DQ variations. We discuss how each of the techniques is applicable to various DQ conditions in ITS. The importance of our study lies in its potential to enhance the robustness of ITS that involve ML applications integrated with edge devices. By investigating techniques addressing the performance degradation due to DQ variations in the ML applications' execution stage, this research contributes to the development of more robust systems that can function effectively in vulnerable ITS cyberinfrastructure conditions.

II. Related ML Techniques to Improve ML Robustness

TL is a highly powerful technique in ML that leverages knowledge from one domain to enhance ML model's learning performance in another, commonly referred as a target domain. Weiss *et al.* [28] classify TL approaches as homogeneous or heterogeneous, based on the similarity or disparity between the source and target domains' feature spaces. Zhuang *et al.* [31] categorize TL approaches based on the elements employed for knowledge transfer in the target domain: instant-based, feature-based, parameter-based, and relational-based ones. Distribution matching aims at minimizing the divergence between the source and target domain distributions, such as the Maximum Mean Discrepancy [9] or the Wasserstein distance [27]. Adversarial training incorporates the process of domain discriminator training, which is responsible for distinguishing between the source and target domain while the feature extractor aims at confusing the discriminator by generating domain-invariant representations [8]. Another approach is ML model fine-tuning that involves adapting a pre-trained ML model to a source domain in order to enhance its performance on a target domain [26]. There are other TL approaches designed and developed for the specific cases, such as combining TL with

multi-task learning [20] and leveraging ML model ensembles [31] to improve the performance on the target domain.

The fundamental FL feature is preserving the local data privacy by communicating only the model updates instead of raw data to the aggregation unit [21]. One of the key challenges is how to aggregate the updates from local clients in a way that allows to balance the generalizability, robustness to the local data variations, and performance of the resulting model. Various aggregation functions have been proposed, such as FedAvg [18], which incorporates averaging of all the local updates submitted for aggregation; Geometric Median [22], which is more robust to the outliers in the data; Krum [10]; trimmed mean [29]; and FedMGDA [12]. Some studies leverage the robust aggregation strategies to train ML model more robust to heterogeneous or shifted data distributions [15].

Real-world ML applications, especially those that involve data collection and transmission in sensor and IoT networks, and running ML processes on edge devices, pose several challenges to achieve ML robustness against the DQ variations [16], such as limited access to the data on the target domain, and dynamically generated data of varied quality. In this paper, we investigate the two approaches not initially designed to enhance ML robustness against the DQ variations, but practically appeared to demonstrate effectiveness in this manner: TL and FL. However achieving the similar goal, these approaches drastically vary in the means to achieve this goal. TL makes use of adjusting the last layers of the ML model's architecture and re-training it on the new training collection relevant to the target domain, which enhances generalizability but commonly decreases the performance [25]. In contrast, FL utilizes the mechanism of ML local models' aggregation, which employs the specific aggregation function to produce the global model based on the majority of submitted local updates. In sec. V, we analyze and compare the effects of employing these two approaches to enhance ML application robustness to DQ variations.

III. Intelligent Transportation System Use Case Description

To facilitate our use case, we design a prototype of an ITS computer vision-based facility, which is responsible for transmitting the images from the data sources over a wireless network to a cloud-based ML application and classifying them into stop sign or non-stop sign categories. To establish the real wireless network connection, we employ similar POWDER platform [2] setup we described in [5]. We use a subset of images from Open Images V6 Dataset[1] and transmit them over a network with unstable QoS to produce the data of varied quality. Specifically, as the network QoS parameters, we study the effect of packet loss and available buffer resources on the receiving unit. Examples of the images received by the ML application after transmitting them over the network with various QoS can be seen in Figure 2. For our study, we employ the stop sign and traffic sign image categories from the dataset. There are around 600 stop sign images and over 3000 traffic sign images. We employ these images in both TL and FL setups to re-train and test the ML model on various DQ.

[1]https://blog.research.google/2020/02/open-images-v6-now-featuring-localized.html

(a) (b) (c) (d)

Fig. 2. Examples of the original images from Open Images Dataset V6 and their corrupted versions obtained through employed for empirical study: (a) – original traffic sign image. Stop sign image transmitted with 512B buffer and: (b) – 5% packet loss; (c) – 10% packet loss; (d) – 20% packet loss

Algorithm 1 Transfer Learning with VGG16 Model

Input: M_{base} – base VGG16 model without the top classification layer; M_{pre} – parameters of VGG16 model pre-trained on ImageNet; D_{target} – target dataset (our data cohorts of varying quality)

Output: M_{ft} – fine-tuned VGG16 model after re-training on the target domain

1: $M_{base} \leftarrow$ Load M_{pre} excluding the top classification layer (FC)
2: **for** each convolutional layer l in M_{base} **do**
3: Freeze weights (W_l)
4: **end for**
5: $FC_{new} \leftarrow$ Initialize new fully connected layer with output size equal to the number of target classes (2 in our case)
6: $M_{ft} \leftarrow$ Replace FC in M_{pre} with FC_{new}
7: Compile M_{ft} with loss function L (binary cross-entropy in our case) and optimizer O (stochastic gradient descent in our case)
8: Train M_{ft} on D_{target} with batch size b (20 in our case), for e (20 in our case) epochs
9: Evaluate M_{ft} on validation set from D_{target}
10: Deploy M_{ft} for target task

A. Transfer Learning Setup

In the TL case, we employ VGG16 image classifier [23] as an example to examine how the network QoS decrease affects the quality of the input data and the performance of the ML application. We first transmit our data over a real network in the unstable QoS conditions, and obtain the images of varied quality. We then submit these corrupted data to the employed classifier, pre-trained on the ImageNet [7] dataset and re-trained on the good quality images from our target domain. We measure the ML baseline performance and find that the DQ variations reduce the ML model's ability to correctly classify the provided samples due to the lack of robustness to the low quality input data. To investigate ML robustness enhancement ways, we further re-train our baseline model on the images of varied quality. In total, we study four re-training

and testing data combinations: re-trained on original images and tested on distorted; re-trained on distorted images and tested on distorted; re-trained on distorted and tested on a mix set of various quality images; and re-trained on a mix set of various quality images and tested on distorted only. A mix set of various quality images represents a uniform distribution sampled from each of the considered DQ cohorts. We evaluate how the employed re-training strategies contribute to the performance and robustness of the produced ML model. We re-train the model for 20 epochs and 74 steps per epoch in each experimental scenario with the Learning Ratio (LR) of 0.001. The particular steps employed to re-train the VGG16 model using TL strategy for each of the DQ cohorts are represented in Algorithm 1.

B. Federated Learning Setup

In this study, we follow-up the work of Manias and Shami [17], who suggested using Road Side Units (RSUs) as edge devices to train the models in a FL manner for ITS applications. We envision a scenario where RSUs obtain data from mobile and static nodes over the network and leverage this data to train a local ML model, which is then transmitted for aggregation with the updates from other RSUs. The data obtained by RSUs originates from various sensor devices (e.g., embedded into vehicles or road infrastructure), and is conveyed over a wireless communication channel, which might induce the DQ variations. This data of diverse quality is then used to train the local models on each RSU. In our experiments, we employ the images of various quality to train the ML model in the FL manner, and then cross-evaluate the trained model performance on various DQ cohorts. We analyze the obtained results and compare them with the ones obtained in the TL scenario. Furthermore, we propose our recommendations on boosting the ML computer vision classification system robustness for the considered ITS industrial use case.

To facilitate our empirical study, we developed the FL framework in Python using PyTorch. As the ML image classification model, we selected ResNet50 architecture [11], pre-trained over the ImageNet [7] data. As a data collection, we utilized the original (high quality) and corrupted (varied quality) labeled traffic sign images. The corrupted images are represented by five cohorts corresponding to the network packet loss ratio during their communication: 1, 2, 5, 10, and 20%. In this case, we did not change the buffer size and employed images transmitted with 512B buffer. We performed several experiments with the models trained on a single image cohort influenced by a certain packet loss percentage, and tested on all other image cohorts. For instance, we trained the FL model over the high quality data distributed over 10 clients, and then we evaluated the resulting model on image sets of diverse quality (influenced by the packet loss of 1, 2%, etc.). The image corpus assigned to each client is split into 66 of training and 33% of testing data in order to perform a local training iteration. In our experiments, we performed 10 successive FL training rounds with 10 local training epochs for each client with the LR of 0.001. After the local training, the acquired models are transmitted to the aggregation unit, where the aggregation procedure is performed. We contrasted two FL aggregation strategies: FedAvg [18] and Geometric Median (GM) [22].

IV. Intelligent Transportation System Use Case Results

A. Transfer Learning Re-training Results

1) Baseline Model: To adapt the pre-trained VGG16 to our specific knowledge domain, we re-train it on the set of high quality images. Then, we test this re-trained model on samples corrupted by the unstable QoS while transferred over a network. For the model evaluation, we employ images transmitted with various buffer sizes and packet loss percentages: buffer size of 128B and packet loss of 5%; buffer size of 256B and packet loss of 5%; buffer size of 512B and packet loss of 20%. The results of the baseline model testing classification accuracy are shown in Table I, which provides mean values of the classification accuracy demonstrated by the model, over 20 re-training epochs, and Standard Deviation (SD) of these values. According to the results obtained, the ML model re-trained on original images demonstrates on average the lowest performance in comparison to other data re-training cases. Hence, ML models pre-trained on the high quality data need to be further trained for the selected case.

2) Re-training and Testing on Distorted Samples: To explore possible ways of enhancing ML model's robustness to the varied DQ, we continued to further re-training the baseline model on corrupted images. TL is performed separately for various DQ variations categories, such as buffer size and packet loss percentage, e.g., first we re-trained the baseline model on images distorted by communication through the channel with the 128B buffer size and 5% packet loss, and tested this model on images distorted in the same way; then we repeated this procedure on other buffer sizes and packet loss percentages. According to Table I, in comparison to the baseline model training results, the classification accuracy did not improve enough over the training process and was still not sufficient for the industry-level systems. This means that it is much more difficult for the model to learn the knowledge representations over the varied quality data. Also, the higher the image corruption degree we trained the model on (e.g., an increased packet loss or reduced buffer size), the lower classification accuracy we obtained on the testing set.

3) Re-training on Distorted Data and Testing on a Mix of Distorted and Original Data: In this case, we examined further training the baseline ML model only on the distorted samples and testing it on a combined collection of various quality samples. As in the previous scenarios, we conducted experiments for all corrupted images categories. The mean and SD values for the classification accuracy for data cohort are shown in Table I. One can see that the model re-trained on only low quality data and tested on a mix of distorted and original images demonstrated on average higher performance in the majority of cases.

TABLE I
AVERAGE CLASSIFICATION ACCURACY AND SD OVER 20 TRAINING EPOCHS, DEMONSTRATED BY THE PRE-TRAINED ML MODEL AFTER TL ON THE DATA COHORTS OF VARIOUS QUALITY

	DQ cohorts		
Re-training and testing sets	Buf. 128B, PL‡ 5%	Buf. 256B, PL 5%	Buf. 512B, PL 20%
TR*: original; TS†: corrupted	0.702 (±0.054)	0.747 (±0.097)	0.669 (±0.033)
TR: corrupted; TS: corrupted	0.821 (±0.07)	**0.934** (±0.021)	0.802 (±0.055)
TR: corrupted; TS: mixed	**0.885** (±0.032)	0.915 (±0.032)	**0.866** (±0.048)
TR: mixed; TS: corrupted	0.745 (±0.103)	0.801 (±0.101)	0.738 (±0.066)

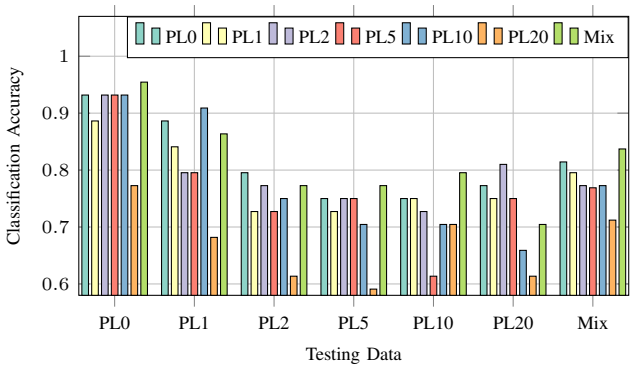
* Re-training dataset; † Testing dataset; ‡ Packet loss

4) Re-training on a Mix of Original and Corrupted Data and Testing on Corrupted Data: In this scenario, we continued to re-train the baseline model on a mixed set of high and low quality images and evaluated its performance only on the low quality samples. As in the previous case, the model was re-trained and tested separately on each DQ image cohort. According to Table I, combining high and low quality samples into a single training set helped to slightly improve the ML performance in comparison to the baseline model. However, in this case training and testing sets differed more than in the previous one, and the ML model trained on the mixed set demonstrated lower performance over the low quality data than the model re-trained only on this low DQ.

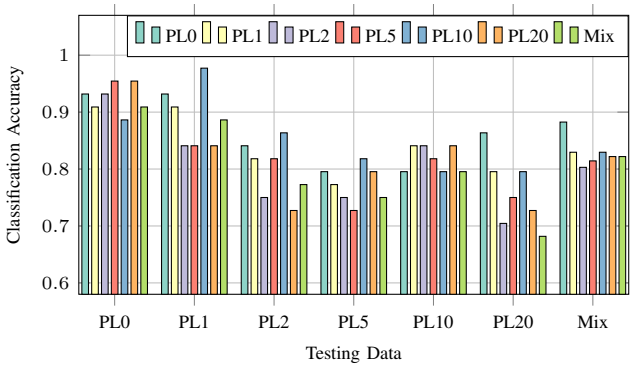
B. Federated Learning Re-training Results

Figure 3 illustrates the results for the image classification accuracy attained by our FL model over the image testing sets of diverse quality. In each experiment, we evaluated the ML models trained on various DQ and measured their performance against the images corrupted by real unstable network conditions. To examine the case when the data produced by a single local unit might be of varied quality, we incorporated the cohort that contained the combination of all employed DQ, i.e., the combination of original images and images affected by 1-20% packet losses sampled in a uniform distribution manner. This image testing set is marked as "Mix" in Figure 3.

Figure 3(a) reveals the ML performance results obtained with the application of FedAvg as the FL aggregation strategy. The models trained on the mixed DQ set showed on average better performance on the original images. Interestingly, despite the training on the low DQ categories, the model exhibited the highest performance on the original images in all cases. The model trained on the "Mix" image cohort allowed to achieve more stable results in terms of classification performance. It surpassed models trained on other DQ categories in four testing cases, while the models trained on other cohorts excelled only once: trained on "PL10" and tested on PL1", trained on "PL0" and tested on PL2", and trained on "PL2" and tested on "PL20". Figure 3(b) demonstrates our FL model image classification accuracy with the GM aggregation



(a)



(b)

Fig. 3. FL model's performance after 10 consequent FL training rounds demonstrated over various DQ using two aggregation strategies: (a) – FedAvg; and (b) – Geometric Median (GM). Various colors represent the employed training data, and the labels on the horizontal axis correspond to the data the models were tested on. PL is a packet loss, and the number after PL corresponds to the packet loss percentage, for example, PL0 corresponds to 0% packet loss percentage during the image transmission over the network

strategy. As one can see from the results, the models trained with this FL aggregation strategy attained on average better performance than with FedAvg in almost all the considered cases. Like the FedAvg case, the models classified the original images on average better than the other categories. However, the model trained on "Mix" cohort did not follow the same trend and did not even dominate in any experiment. The model trained over "PL10" showed better performance in "PL1", "PL2'', and "PL5" cases, and surprisingly, the model trained over the original data demonstrated the higher classification accuracy in "PL20" and "Mix" testing categories.

V. DISCUSSION AND ANALYSIS

A. Transfer Learning Case Results Analysis

As we learned in our investigation, pre-trained foundation ML models may not perform sufficiently well in applications, where DQ may vary. ML models are usually trained only on good quality images (e.g., state-of-the-art datasets such as ImageNet [7]), but in reality, images can be distorted by various factors, such as vulnerable cyberinfrastructure. Further re-training is needed to demonstrate better performance on both high and low quality samples from the target domain.

If the DQ is not expected to vary much during the ML application operation, a short re-training with some additional samples can rapidly help in ML performance improvement. However, this may not work for more complicated cases, where the data undergoes multiple stages and is processed by various ML cyberinfrastructure components before being classified. In this case, the DQ might be affected by noise, interference, data loss or malicious attacks. Our investigation showed that training on good quality images only does not improve the ML performance on the varied DQ.

In this paper, we studied TL from a source domain (high DQ) to a target one (varied DQ) as one of the ways to enhance ML robustness against DQ variations. TL can involve retraining on the varied DQ only or on the mixed set of varied quality images. Our results demonstrated that TL allowed improving the ML performance on both high and low quality images. With re-training on the low quality data, the ML performance on both DQ types was commensurable. Re-training the ML model on the mixed set resulted in less efficient training process, as it required more time for the ML performance to converge because of the dynamic and inconsistent patterns in the training data. Based on our investigation, we can recommend to perform further ML model re-training to the target domain data in order to enhance its robustness to the varied DQ, and to improve the performance on both high and varied quality data.

Based on our results, one can see that current foundation computer vision models are needed to be re-trained on lower quality data samples to achieve acceptable performance in real-world applications, especially those that possess dynamic DQ nature. The classifier's performance robustness to possible DQ variations can be improved with TL by further re-training on bad quality images. TL on low DQ only appeared to be more effective than extending the training base by combining high and low DQ.

B. Federated Learning Case Results Analysis

Based on our investigation results, we can offer the following suggestions on how to increase the foundation computer vision models robustness to the DQ variations when trained in a FL manner. One suggestion is to use ML models initially pre-trained on comprehensive datasets (e.g., ImageNet [7]) and re-train them for the target domain rather than training a model from scratch only on the local data. This pre-training allows improving the model's generalizability on new data. Another suggestion is to consider the dynamic ML application operation environment and changing image DQ used for training, which may affect the ML training performance. If the operational conditions are stable, only the available high quality data may be used for the re-training. Otherwise, the better strategy is to mix the data received from various edge devices for the local training in order to improve the robustness of the model. Moreover, in almost all the investigated cases, the FL models trained with the GM-based aggregation strategy showed higher robustness against the DQ variations. Hence, we can recommend employing GM as the aggregation strategy whenever the corresponding resources are available.

C. Discussion on Transfer Learning and Federated Learning Capabilities to Enhance ML Robustness

As our study has shown, the performance of real-world ML applications, which operate in dynamic environments and employ edge devices for running ML processes, is significantly influenced by the quality of the training data. When ML models, trained on original data, are tested on the lower DQ, their performance tends to decrease. This observation emphasizes the importance of maintaining high input DQ for achieving the required ML performance in the operational stage. However, in real-world applications, the quality of data may vary due to multiple reasons. As a way to address this problem and to enhance ML robustness to DQ variations, we investigated two approaches that were not directly designed for this task: TL and FL.

TL strategies might be successfully employed to address DQ variations. In cases when DQ variations were a concern, training the model solely on the low DQ, without including the high quality data into the re-training set, allowed to achieve higher performance. This can be attributed to the initial pre-training of the model on the original data, which establishes the necessary knowledge for recognizing patterns and capturing relevant features. When re-training the ML model only on the data of degraded quality, it better adapts to the characteristics and challenges posed by varied DQ, resulting in improved performance in the presence of DQ variations.

When dealing with DQ variations, FL with the GM as the aggregation strategy was found to be more robust compared to FL with FedAvg, provided the necessary computational resources are available. By employing GM, the FL aggregator is better equipped to produce more effective global model under the adverse local data conditions, making FL with GM a preferable approach when DQ is a concern. When considering training or re-training on the high quality data, FL demonstrated greater robustness to DQ variations compared to TL. FL's inherent ability to leverage distributed data sources and aggregate models from multiple participants enabled it to handle variations in data distributions more effectively. FL proved to be a robust approach for mitigating the impact of DQ variations, surpassing TL in terms of overall performance and adaptability in the case of employing high training DQ.

Despite the advantages of FL, TL can still achieve comparable results when trained or re-trained on the varied DQ. However, TL in a conventional centralized fashion implies high data communication loads, which might be too expensive for the ITS network facilities and raises DQ variation issues due to possible network disruptions. In the scenarios examined, FL and TL demonstrated similar performance levels. This suggests that both FL and TL are viable strategies for addressing DQ variations, with each approach offering different advantages and trade-offs.

VI. Conclusion

In this paper, we investigated methods to enhance the robustness of pre-trained foundation computer vision models deployed in IoT systems that involve running ML processes on edge devices. Specifically, we concentrated on an ITS scenario with DQ variations due to network disruptions serving as a real-world use case. The key challenge we addressed is maintaining robustness to the quality variations in the data received by edge devices. Our study demonstrated that pre-trained, foundation models can be effectively leveraged in Edge AI applications, but require re-training to maintain performance in real-world scenarios. We explored TL and FL as strategies for re-training these models on edge devices. Through empirical evaluation, we found that both TL and FL could effectively mitigate the impact of such variations, and the employment of the particular strategy depend on the resources available. When high-quality data only is available for retraining, FL with Geometric Median aggregation demonstrates superior robustness compared to TL. In applications operating within dynamic input DQ landscapes, leveraging available data of varying quality for model re-training using FL is advantageous. Our findings highlight the importance of re-training pre-trained models within Edge AI and IoT systems, and offer the community valuable approaches to address input DQ variations.

References

[1] A. Adams, A. M. Abu-Mahfouz, and G. P. Hancke, "Machine learning-imaging applications in transport systems: A review," in *2023 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. IEEE, 2023, pp. 1–7.

[2] J. Breen, A. Buffmire, J. Duerig, K. Dutt, E. Eide, A. Ghosh, M. Hibler, D. Johnson, S. K. Kasera, E. Lewis *et al.*, "Powder: Platform for open wireless data-driven experimental research," *Computer Networks*, p. 108281, 2021.

[3] S. Chuprov, K. M. Bhatt, and L. Reznik, "Federated learning for robust computer vision in intelligent transportation systems," in *Proceedings of the 2023 IEEE Conference on Artificial Intelligence (IEEE CAI)*, 2023, to be published.

[4] S. Chuprov, S. Mahajan, R. Zatsarenko, L. Reznik, and A. Ruchkan, "Are industrial ml image classifiers robust to withstand adversarial attacks on videos?" in *2023 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*. IEEE, 2023, pp. 1–4.

[5] S. Chuprov, L. Reznik, A. Obeid, and S. Shetty, "How degrading network conditions influence machine learning end systems performance?" in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2022, pp. 1–6.

[6] S. Dalal, B. Seth, M. Radulescu, T. F. Cilan, and L. Serbanescu, "Optimized deep learning with learning without forgetting (lwf) for weather classification for sustainable transportation and traffic safety," *Sustainability*, vol. 15, no. 7, p. 6070, 2023.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[8] Z. Deng, L. Zhang, K. Vodrahalli, K. Kawaguchi, and J. Y. Zou, "Adversarial training helps transfer learning via better representations," *Advances in Neural Information Processing Systems*, vol. 34, pp. 25 179–25 191, 2021.

[9] A. Farahani, B. Pourshojae, K. Rasheed, and H. R. Arabnia, "A concise review of transfer learning," in *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2020, pp. 344–351.

[10] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[12] Z. Hu, K. Shaloudegi, G. Zhang, and Y. Yu, "Fedmgda+: Federated learning meets multi-objective optimization," *arXiv preprint arXiv:2006.11489*, 2020.

[13] J. J. Jui, M. A. Ahmad, M. I. Molla, and M. I. M. Rashid, "Optimal energy management strategies for hybrid electric vehicles: A recent survey of machine learning approaches," *Journal of Engineering Research*, 2024.

[14] A. Kolides, A. Nawaz, A. Rathor, D. Beeman, M. Hashmi, S. Fatima, D. Berdik, M. Al-Ayyoub, and Y. Jararweh, "Artificial intelligence foundation and pre-trained models: Fundamentals, applications, opportunities, and social impacts," *Simulation Modelling Practice and Theory*, vol. 126, p. 102754, 2023.

[15] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[16] L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, and H. H. Olsson, "Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions," *Information and software technology*, vol. 127, p. 106368, 2020.

[17] D. M. Manias and A. Shami, "Making a case for federated learning in the internet of vehicles and intelligent transportation systems," *IEEE Network*, vol. 35, no. 3, pp. 88–94, 2021.

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

[19] M. Megnidio-Tchoukouegno and J. A. Adedeji, "Machine learning for road traffic accident improvement and environmental resource management in the transportation sector," *Sustainability*, vol. 15, no. 3, p. 2014, 2023.

[20] T. Mehmood, A. E. Gerevini, A. Lavelli, and I. Serina, "Combining multi-task learning with transfer learning for biomedical named entity recognition," *Procedia Computer Science*, vol. 176, pp. 848–857, 2020.

[21] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.

[22] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] A. Stanitsa, S. H. Hallett, and S. Jude, "Investigating pedestrian behaviour in urban environments: A wi-fi tracking and machine learning approach," *Multimodal Transportation*, vol. 2, no. 1, p. 100049, 2023.

[25] D. Vela, A. Sharp, R. Zhang, T. Nguyen, A. Hoang, and O. S. Pianykh, "Temporal quality degradation in ai models," *Scientific Reports*, vol. 12, no. 1, p. 11654, 2022.

[26] G. Vrbančič and V. Podgorelec, "Transfer learning with adaptive fine-tuning," *IEEE Access*, vol. 8, pp. 196 197–196 211, 2020.

[27] X. Wang and Y. Yu, "Wasserstein distance transfer learning algorithm based on matrix-norm regularization," in *2022 2nd International Conference on Algorithms, High Performance Computing and Artificial Intelligence (AHPCAI)*. IEEE, 2022, pp. 8–13.

[28] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[29] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[30] R. Zatsarenko, C. A. Marathe, S. Chuprov, M. Hyland, and L. Reznik, "Are industrial ml image classifiers robust to data affected by network qos degradation?" in *2023 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*. IEEE, 2023, pp. 1–4.

[31] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.