Federated Learning Robustness on Real World Data in Intelligent Transportation Systems

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

Machine Learning models are widely utilized in a variety of applications, including Intelligent Transportation Systems (ITS). As these systems are operating in highly dynamic environments, they are exposed to numerous security threats that cause Data Quality (DQ) variations. Among such threats are network attacks that may cause data losses. We evaluate the influence of these factors on the image DQ and consequently on the image ML model performance. We propose and investigate Federated Learning (FL) as the way to enhance the overall level of privacy and security in ITS, as well as to improve ML model robustness to possible DQ variations in real-world applications. Our empirical study conducted with traffic sign images and YOLO, VGG16 and ResNet models proved the greater robustness of FL-based architecture over a centralized one.

Index Terms—**Federated Learning, Data Quality, Intelligent Transportation Systems, Data Privacy**

I. Introduction

MACHINE Learning (ML) models, or Foundation Models (FM), are incorporated across a wide diversity of domains, ranging from civil implementations for traffic and transportation systems [15], medicine, social media, to military applications [12]. While Zhou *et al.* [28] identified three primary areas of FM application, namely Natural Language Processing, Computer Vision, and Graph Learning, our study focuses on FM applications within Intelligent Transportation Systems (ITS). These systems typically integrate Computer Vision FM such as YOLO [26] and R-CNN to facilitate tasks of object detection in self-driving vehicles, traffic monitoring, and emergency response.

The complexity of ITS, characterized by a wide variety of data sources [4], [23], including different types of infrastructure objects, such as Road Side Units (RSU), vehicles, and traffic cameras, coupled with the dynamic nature of operational environments, poses challenges for ML models embedded within these systems. Data Quality (DQ) variations are common and may be caused by a plethora of reasons. Malicious physical or cyberattacks, such as sabotaging ITS components or orchestrating Denial of Service (DoS) attacks on network infrastructure, can result in data loss. Diverse data origins, such as on-board cameras and sensors of different brands and characteristics found in vehicles and RSU, may include distinct technological traits like resolution, accuracy, and lens focal length. Furthermore, fluctuating operational and environmental circumstances, such as varying weather conditions like snow or rain, can lead to anomalies in image capture [5].

Our assessment focuses on the robustness of the FM – the staple component of ITS – against the input data of varying quality, which may be caused either by adversarial network attacks, or heterogeneous operating conditions. Specifically, we investigate how FM perform when processing data affected by distortions such as noise, grayscale images, contrast alterations, and data loss. Our findings reveal that while FM generally handle DQ variations like noisy and grayscale images without a significant robustness decrease, they exhibit performance degradation when processing images impacted by data loss. This phenomenon is largely caused by the fact that FM are commonly trained in a centralized manner on high quality data, whereas real-world scenarios often entail the utilization of corrupted data during execution.

To mitigate this challenge, numerous approaches have been proposed to enhance FM effectiveness on relevant data. Among these, Transfer Learning (TL) [15] and Federated Learning (FL) [14] emerge as prominent solutions. TL is the process of adapting a previously trained model on a new target domain [17]. In the context of ITS, this new domain comprises real-world data of varying quality obtained from ITS sources. Employing of TL in ITS involves the FM training on this data with various corruptions [7]. However, this approach raises security concerns when the data from multiple local devices is collected centrally. To train FM effectively with the real data containing inherent DQ variations, samples of such data must be gathered. Consequently, images would need to be transmitted over a network from the node to the aggregation server. Transmission of a confidential data over a network gives rise to multiple privacy concerns, including the risk of third-party data breaches. Additionally, the data may be corrupted during the transmission process in case of DoS attack.

FL embodies an architecture wherein each node trains its own model using locally gathered data [17]. In this paper, we follow the use case developed by Manias and Shami [14], where RSU are considered as FL units for local training. In this setup, the collected data is not transmitted across the network, as each data source maintains its own FM. Consequently, since data is collected independently by each local unit, it retains its unique characteristics on each processing node, thereby rendering every FM more relevant to the specific data it processes, increasing the overall resilience of ITS. Following the FM training phase, the model from each node is supplied to the aggregation agent [14], where the global model

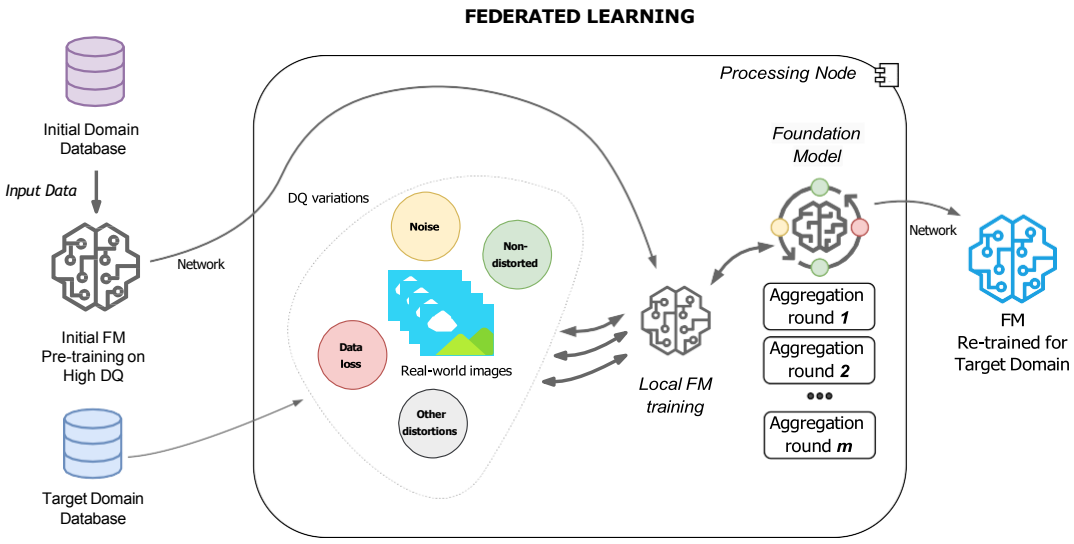

Fig. 1. Representation of the Federated Learning training process. Initially pre-trained FM is then aggregated with local FM that are trained on the real-world data of various quality. Each client pre-trains FM on the real-world data samples that are acquired into its own target dataset from a corresponding data source

is produced. Additionally, FL offers better privacy protection [14]. Rather than transmitting actual confidential images over the network, only the model updates are typically shared, allowing significant privacy enhancement [14].

In this paper, we propose FL as the way to improve ITS security and robustness against DQ variations in input images. Not only does the FL approach enhance the overall privacy protection of ITS, but our experiments also demonstrate that FL significantly improves FM's object detection performance when processing the images affected by data losses. Our study reveals that centralized FM experience decrease in their accuracy when processing images affected by data losses. Results of our experiments show that employing of FL-based setup coupled with the mixed data FM training allows to mitigate performance issues associated with the processing of corrupted images. Figure 1 illustrates our FL setup, where the Processing Node represents the image source at which the local model is trained. The images may contain various corruptions, such as noise, contrast increase and data losses. These images then participate in the local FM training since they are integrated in the training dataset. Local model updates are then transmitted over a network.

II. Related Security and Robustness Improvement Techniques in Machine Learning Applications

Systems that incorporate ML models can be very complex, comprising multiple data sources and computation nodes that are often widely spread among networks of various topologies. These systems can also be implemented using various ML architecture approaches. The abundance of components of highly different nature make these systems vulnerable to threats of discrete origins.

One direction of research endeavors focus on improvement of system aspects not directly related to refining ML architecture or FM themselves. In such works, authors investigate possibilities for upgrades in technologies that accompany integration of ML in a particular case. Such measures often aim at mitigating threats that stem from the possibility of reverse engineering attacks on ML.

Lu *et al.* [13] in their study aimed at improving the robustness of ML models in Industrial Internet of Things (IoT) applications by utilizing FL along with the blockchain technology. In this setup, the blockchain module serve as the component for establishing secure connections among participating IoT devices. Employing of FL architecture allows better data privacy because the data itself remains decentralized, while blockchain ensures the secure ML model transmission to aggregation server. While introduction of an additional security driven by blockchain to an IoT application may increase the overall system resilience to possible attacks, it also introduces additional concerns typical of blockchain systems. Among them is the increased complexity of the deployment and maintenance of systems with such architecture.

In [25], authors approached the challenge of data management in varying network conditions from a network exchange perspective, introducing a new broadcast protocol for the purpose of adapting ML models to constantly changing environments. Authors focus their investigation on Vehicular Ad-Hoc Networks (VANET) which are commonly utilized in ITS domain. They as well underscore that in such systems the network environment is constantly changing over time. The proposed protocol design leverages a fuzzy logic-based approach to determine suitable network nodes for data transmission and reception, enhancing adaptability and efficiency in VANET networks.

Another group of security concerns in ML systems is data poisoning. This process takes place when the data used in an ML system gets corrupted, which leads to DQ variation. This can happen either due to adversarial attacks, such as DoS, or because of the unintentional harmful conditions, e. g. a poor connection due to limited network bandwidth, or a storage damage. In the case of DQ variation, problems related to the

ML model performance and robustness arise.

Another direction of research studies the possibilities for improving the security and robustness of ML systems by exploring and evaluating various approaches to ML itself. Otoum *et al.* [17] conducted the overview and comparison of three approaches to ML, namely FL, Transfer Learning (TL) and Split Learning (SL). Authors employed a dataset provided by Canadian Institute of Cybersecurity Intrusion Detection System (CICIDS2017) in order to assess accuracy and detection rates, power consumption, packet loss ratio and quality of experience.

In this work, we assess the robustness of the FL-based architecture against DQ variations and compare it with the centralized ML method. We propose FL as the solution for, on the one hand, enhancing the security of the ML system by eliminating the need for extensive data transfer over a network. On the other hand, we show that employment of FL can mitigate the robustness issues caused by DQ variations.

III. Empirical Study Background

A. Factors Resulting in Data Quality Degradation

ITS is the example of a real-time service that incorporates basic Internet of Things (IoT) elements [3]. However, transportation systems are often operating in highly unstable environments, which cause variations in DQ [16]. Maintaining data coherence during transmission is crucial for the seamless functioning of such services [27], as reduced network Quality of Service (QoS) can lead to integrity breaches or corruptions in transmitted data segments [24], which ultimately may result in incorrect decisions critical in ITS. Wireless networks may be subject to data loss due to a number of reasons, such as radio frequency interference [20], extensive distances between nodes, and network congestion. In addition, unstable network characteristics, node dynamics, and high bit error rates can affect data delivery rates in cellular networks [22].

Although ITS are built using their own type of networks, such as VANET, these networks still incorporate the transport layer of TCP/IP, deriving all the limitations typical to it. The transport layer of the TCP/IP stack provides three widely utilized protocols that are commonly used in applications nowadays: Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Stream Control Transmission Protocol (SCTP). TCP and SCTP guarantee reliable data delivery, wherein lost data is re-transmitted. However, reliable data transmission requires a store and forward network infrastructure, where each transition node is able to accumulate a significant number of dropped packets awaiting for the proper delivery to the recipient [1]. In ITS, the accumulation of data is often infeasible due to limitations in persistent storage capacity, power consumption constraints, and computing overhead. If the command and control protocol is applied in the network, it is able to redistribute traffic flows by reducing the particular application's bandwidth to mitigate data loss, albeit negatively affecting network QoS and latency, potentially rendering it unsuitable for real-time services. Therefore, ITS often opts in UDP for data streaming.

Nevertheless, data loss stemming from UDP usage can significantly influence the FM performance [8]. In ML-driven applications like self-driving vehicles, instantaneous and accurate object detection is paramount for the safety of its users and other traffic participants. In addition to unstable network and environmental conditions, another group of factors contributing to DQ variations originates directly from input devices and is related to image processing. Transportation data may be affected by malfunctioning sensors or other interference [16]. For instance, a dirty camera lens in the detection unit, common in RSU operating in open environments, may result in increased noise level of a resulting image. Malfunctioning sensors within camera units may lead to production of grayscale images, while exposure to extreme weather conditions can provoke automatic contrast adjustments during image processing, such as when recording against the sun's direction. In the section below we describe the process of establishing datasets affected by these factors.

B. Image Datasets

1) Non-Distorted Image Datasets

To study image FM performance, we employed the "Traffic Sign" (TS) and "Stop Sign" (SS) image subsets from the Open Images V6 dataset [2], which incorporates images labeled for classification, object detection and semantic segmentation. In order to train the centralized FM, we utilized the original images from the employed subset.

2) Distorted Image Datasets

We utilized a set of images with various DQ to train and evaluate the performance of FM. In order to simulate the distortions that may occur in case of the network data losses, we used network utility tools such as *iptables* and *nftables* for Linux operating system [9]. These utilities allow to set network parameters for data losses based on rules defined by statistical or probability measures. Below is the code snippet with the definition of rules for the network node:

```
iptables -A INPUT -m statistic
--mode random
--probability 0.05 -p udp
--destination-port 2020 -i eth0
-j DROP
```

These rules allowed us to establish datasets with images affected by the varying percentages of data losses.

In order to comprehensively assess the impact of other factors, we created a separate set of images introducing the following DQ variations: heightened noise levels, amplified contrast, and conversion to grayscale. Noise was introduced by generating a random tensor with dimensions identical to the input image, multiplying it by a scaling factor, adding it to the input image, and then clipping the resulting values to fall within the acceptable range for an RGB image (0 to 255). Contrast modification was achieved using the "Contrast enhancer" from the Python's Pillow library [10] with a coefficient of .01. Converting to grayscale involved computing the average brightness of each pixel's RGB components. Examples of distorted images are depicted in Figure 2.

Fig. 2. Examples of distorted images: (a) – original SS image; (b) – SS image with increased contrast; (c) – SS image with 2% data loss; (d) – SS image affected by noise (200) filter; (e) – original TS image; (f) – TS image with increased contrast; (g) – TS image with 2% data loss; (h) – TS image affected by noise (200) filter

C. Foundation Models Setup

In order to evaluate the effect of corrupted images on a centralized FM, we conducted tests on two image detectors widely employed in practice: YOLO [19] and VGG16 [21]. The YOLO object detection system is characterized by its one-stage detection algorithm, employing a singular neural network across the entire image. This network partitions the image into regions, predicting bounding boxes and probabilities for each region, with these bounding boxes weighted by the predicted probabilities. The authors claim that YOLO has several advantages over classifier-based systems, including its holistic image analysis at test time, leveraging global contextual information on the image. It also generates predictions with a single network evaluation, unlike RCNN systems which require thousands evaluations for a single image, making it much faster than RCNN systems [19].

On the other hand, Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG16) stands out as a dependable and robust FM for the image classification [21]. Unlike YOLO, VGG16 follows a sequential operational flow, incorporating multiple layers consisting of 3×3 shaped filters, aimed at reducing the input image feature dimensions. Each activation layer utilizes ReLU as the activation function. The network ultimately incorporates 138 million parameters.

Both centralized setups involved training the FM using the original image files, enabling an assessment of the FM performance degradation when processing data of varied quality. The decision to train the centralized models solely on the non-distorted data is driven by the urge to closely align with a real-world setups, where such models are normally trained on data without any corruptions. For the centralized FM, we assessed their performance across two tasks: image classification and object detection. The dataset employed contained images labeled as SS and TS, with the SS utilized for evaluating object detection performance and the TS label for evaluating image classification performance.

TABLE I
YOLO OBJECT DETECTION PERFORMANCE [9]

	YOLO
Distortion Type	*AP/mAP*
No distortion	84.46
Noise (100)	99.32
Noise (200)	92.57
Grayscale	99.03
Contrast increase	99.26
2% data loss	41.38
5% data loss	30.77

In order to assess FL-based setup, we built up the FL framework in Python using PyTorch. For the ML image classification model, we employed the ResNet50 architecture [11]. The FL architecture allows various model aggregation strategies [6], and we examined the performance of the FL-based ITS utilizing the following: Geometric Median (GM) and Federated Average (FedAvg). The first strategy, GM, is referred to as more robust to outliers and deviations in the parameters of models [18].

FL architecture introduces the possibility for the training on the real-world data with various corruptions, since this data is gathered on each processing node. This allowed us to studied three training strategies for the FL setup. Initially, models were trained on a mixed training dataset comprising uniformly distributed non-distorted images and images featuring the following data losses: 1, 2, 5, 10, and 20%. Additionally, two other training datasets consisted of images affected solely by 2 and 5% data loss, respectively, establishing the basis for the remaining two training strategies. This approach not only enabled an evaluation of both model aggregation strategies, but also facilitated the identification of the optimal combination of DQ variations in the training dataset for enhanced FM performance and robustness.

IV. RESULTS EVALUATION

A. Centralized Foundation Model Performance

Table I illustrates the impact of various image distortion types on the performance of YOLO object detection on the SS labeled data. The accuracy of the model is computed as the ratio of average precision to the median average precision. Surprisingly, the first four factors – namely, “Noise (100)”, “Noise (200)”, “Grayscale”, and “Contrast increase” – resulted in the increased YOLO performance. This improvement could be attributed to the filtration of insignificant image features, thereby preserving more significant ones and enhancing overall performance.

Conversely, the remaining two factors – both “2% data loss” and “5% data loss” – resulted in a drastic performance decline for the YOLO object detector. Furthermore, the performance

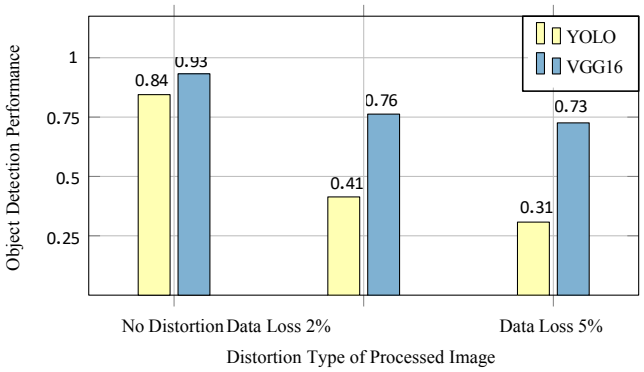

Fig. 3. Centralized FM performance decrease when processing images affected by data losses. FM were trained on data with no distortions

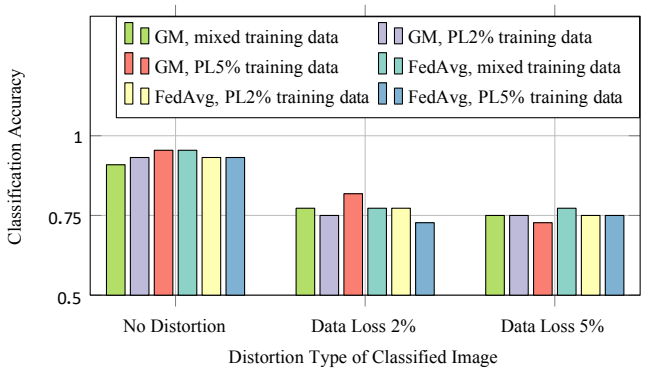

Fig. 4. Image classification performance of ResNet50 FM in for various training datasets and aggregation strategies [6]

TABLE II
IMAGE CLASSIFICATION PERFORMANCE (YOLO)

Distortion Type	*AP/mAP*
No distortion	70.69
Noise (100)	61.29
Noise (200)	43.55
Grayscale	63.25
Contrast increase	68.04
2% data loss	27.78
5% data loss	10.14

drop observed with 2% and 5% data loss equates to half and nearly one-third of the performance achieved with original images. However, such an effect on images affected by data loss is not surprising. This decline occurs because the distorted images become excessively corrupted and unrecognizable for the detector, as can be seen on Figures 2(c) and 2(g).

Figure 3 displays the comparison of the performance deterioration between YOLO and VGG16 FM during object detection on images affected solely by data loss. As one can see from this figure, VGG16 suffers from a somewhat lesser yet still notable performance reduction. It is clear that in such scenarios both centralized FM experience a significant decrease in object detection accuracy.

Additionally, we conducted a series of experiments to assess the image classification performance of YOLO FM when processing corrupted data, utilizing the TS labeled data for this purpose.

The results are presented in Table II. As one can see from this table, the decrease in image classification performance is more pronounced compared to the decrease observed in the previous series of experiments focusing on object detection. Particularly, the deterioration in performance when processing images affected by data loss factors prevails in the case of image classification as well.

Given these findings, we choose to evaluate FL-based FM robustness against images distorted by data loss. Our rationale stemmed from the observation that this type of DQ degradation significantly affects FM performance in a negative way, thereby heightening the risk of overall malfunctioning within the ITS.

B. FL Model Performance

Figure 4 showcases the image classification accuracy results demonstrated by our FL model across the image testing sets featuring data loss distortions. We present the outcomes achieved through the employment of both FedAvg and GM aggregation strategies. In each experiment, FM trained on datasets with varying DQ were subsequently assessed.

It is noteworthy that models trained on the mixed DQ dataset exhibit on average better performance on the non-distorted images. Interestingly, despite being trained on corrupted data, FM consistently demonstrates the highest performance on the original images across all cases. Regardless of the dataset on which the FM was trained, both aggregation strategies showcase the highest classification accuracy on non-distorted images.

It is also worth mentioning that, across all investigated scenarios, the model trained on the mixed dataset and utilizing the FedAvg aggregation strategy consistently shows the best performance, except for the scenario of processing images impacted by 2% data loss. In this particular case, the model trained on the dataset containing 5% data loss images and employing the GM aggregation strategy outperforms.

C. Comparison of Centralized and FL-based Models

As one can see in Figure 5, employment of the mixed dataset in the process of FM trained in a FL manner allows better system robustness against data losses leading to DQ variation. Both FL aggregation strategies demonstrated significantly lesser classification accuracy degradation when handling corrupted real-world data compared to the centralized FM.

Figure 6 illustrates the comparison of the FL-based FM performance across various training datasets featuring different corruptions. Specifically, we compare the model's performance when trained on images affected by 2% data loss, 5% data loss, and a mixed dataset. These results are presented for the GM aggregation strategy.

Remarkably, FM trained on images distorted by 5% data loss outperforms when classifying both non-distorted images and those affected by 2% data loss compared to FM trained on mixed images.

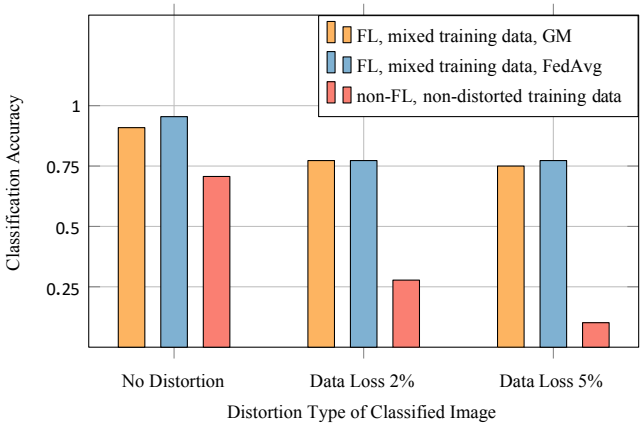

Fig. 5. Comparison of FM performance when processing non-distorted images and images affected by data losses. Here we compare FL-based FM when using GM and FedAvg aggregation strategies with the centralized YOLO FM

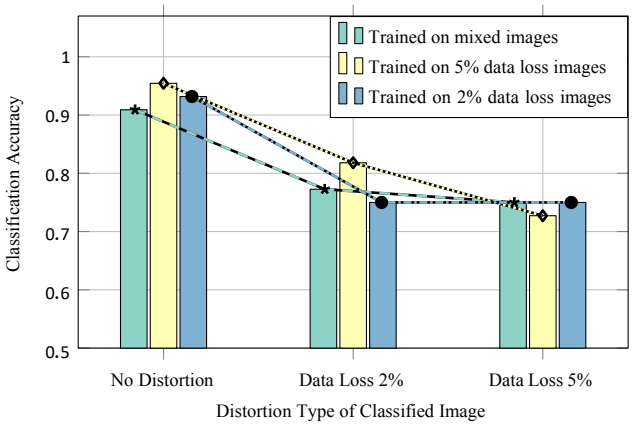

Fig. 6. Comparison of the performance demonstrated by FM trained in a FL manner on various datasets

However, the FM trained on the "mixed" dataset demonstrates the highest robustness among the three training strategies, showcasing the least steep decline in classification accuracy when processing corrupted data. This underscores the efficiency of training on real-world data with DQ variations in enhancing FM performance in ITS.

V. Conclusion

Among reasons that result in DQ variation in ITS are data losses that may be caused by DoS attacks on the network infrastructure. Others are increased noise and contrast levels, grayscale images, that may be caused by environmental factors. We evaluated centralized FM robustness against images affected by these factors and identified that they experience a significant decrease in performance when classifying and detecting objects on images with distortions caused by data losses. We proved that the FL-based approach is an effective solution to mitigate challenges related to data loss that may happen in the real-world ITS. Not only the FL architecture allows enhanced data privacy by preserving confidential client information locally, but also our experiments demonstrated that in all of the investigated cases FL-based ML models that were trained on mixed data demonstrated higher image classification and detection accuracy than ML models trained in a centralized manner.

References

[1] M. Beck, T. Moore, J. Plank, and M. Swany, "Logistical networking," in *Active Middleware Services*. Springer, 2000, pp. 141–154.

[2] G. A. Blog. (2020) Overview of Open Images V6. (Date last accessed 15-February-2024). [Online]. Available: https://storage.googleapis.com/openimages/web/factsfigures.html

[3] T. M. Bojan, U. R. Kumar, and V. M. Bojan, "An internet of things based intelligent transportation system," in *2014 IEEE International Conference on Vehicular Electronics and Safety*. IEEE, 2014, pp. 174–179.

[4] S. Chuprov, P. Belyaev, R. Gataullin, L. Reznik, E. Neverov, and I. Viksnin, "Robust autonomous vehicle computer-vision-based localization in challenging environmental conditions," *Applied Sciences*, vol. 13, no. 9, p. 5735, 2023.

[5] ——, "Robust autonomous vehicle computer-vision-based localization in challenging environmental conditions," *Applied Sciences*, vol. 13, no. 9, p. 5735, 2023.

[6] S. Chuprov, K. M. Bhatt, and L. Reznik, "Federated learning for robust computer vision in intelligent transportation systems," in *2023 IEEE Conference on Artificial Intelligence (CAI)*, 2023, pp. 26–27.

[7] S. Chuprov, I. Khokhlov, L. Reznik, and S. Shetty, "Influence of transfer learning on machine learning systems robustness to data quality degradation," in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–8.

[8] S. Chuprov, L. Reznik, and G. Grigoryan, "Study on network importance for ml end application robustness," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 6627–6632.

[9] S. Chuprov, L. Reznik, A. Obied, 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)*, 2022, pp. 1–6.

[10] A. Clark, "Pillow," 2022, (Date last accessed 15-February-2024). [Online]. Available: https://github.com/python-pillow/Pillow/

[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] S. Jha, A. Roy, A. Cobb, A. Berenbeim, and N. D. Bastian, "Challenges and opportunities in neuro-symbolic composition of foundation models," in *MILCOM 2023 - 2023 IEEE Military Communications Conference (MILCOM)*, 2023, pp. 156–161.

[13] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2020.

[14] 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.

[15] M. K. Moghimi and F. Mohanna, "Reliable object recognition using deep transfer learning for marine transportation systems with underwater surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 2515–2524, 2023.

[16] A. Neilson, Indratmo, B. Daniel, and S. Tjandra, "Systematic review of the literature on big data in the transportation domain: Concepts and applications," *Big Data Research*, vol. 17, pp. 35–44, 2019.

[17] S. Otoum, N. Guizani, and H. Mouftah, "On the feasibility of split learning, transfer learning and federated learning for preserving security in its systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7462–7470, 2023.

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

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[20] D. C. Salyers, A. D. Striegel, and C. Poellabauer, "Wireless reliability: Rethinking 802.11 packet loss," in *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*. IEEE, 2008, pp. 1–4.

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

[22] Y. Tian, K. Xu, and N. Ansari, "Tcp in wireless environments: problems and solutions," *IEEE Communications Magazine*, vol. 43, no. 3, pp. S27–S32, 2005.

[23] F.-Y. Wang, Y. Lin, P. A. Ioannou, L. Vlacic, X. Liu, A. Eskandarian, Y. Lv, X. Na, D. Cebon, J. Ma, L. Li, and C. Olaverri-Monreal, "Transportation 5.0: The dao to safe, secure, and sustainable intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 10 262–10 278, 2023.

[24] Y. Wang, V. Menkovski, I. W.-H. Ho, and M. Pechenizkiy, "Vanet meets deep learning: The effect of packet loss on the object detection performance," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–5.

[25] C. Wu, Y. Ji, X. Chen, S. Ohzahata, and T. Kato, "An intelligent broadcast protocol for vanets based on transfer learning," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015, pp. 1–6.

[26] C. Yao-Liang, "Application of an effective hierarchical deep-learning-based object detection model integrated with image-processing techniques for detecting speed limit signs, rockfalls, potholes, and car crashes," *Future Internet*, vol. 15, no. 10, p. 322, 2023.

[27] X. Zhang, X. Yang, J. Lin, G. Xu, and W. Yu, "On data integrity attacks against real-time pricing in energy-based cyber-physical systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 1, pp. 170–187, 2016.

[28] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, H. Peng, J. Li, J. Wu, Z. Liu, P. Xie, C. Xiong, J. Pei, P. S. Yu, and L. Sun, "A comprehensive survey on pretrained foundation models: A history from bert to chatgpt," 2023.