InFL-UX: A Toolkit for Web-Based Interactive Federated Learning

Tim Maurer
s8timaur@stud.uni-saarland.de
Saarland University
Saarbrücken, Germany

Abdulrahman Mohamed Selim
abdulrahman.mohamed@dfki.de
German Research Center for Artificial
Intelligence (DFKI)
Saarbrücken, Germany

Hasan Md Tusfiqur Alam
hasan_md_tusfiqur.alam@dfki.de
German Research Center for Artificial
Intelligence (DFKI)
Saarbrücken, Germany

Matthias Eiletz
eiletz@proomea-group.com
ProOmea GmbH
Wangen, Germany

Michael Barz
michael.barz@dfki.de
German Research Center for Artificial
Intelligence (DFKI)
Saarbrücken, Germany
University of Oldenburg
Oldenburg, Germany

Daniel Sonntag
daniel.sonntag@dfki.de
German Research Center for Artificial
Intelligence (DFKI)
Saarbrücken, Germany
University of Oldenburg
Oldenburg, Germany

Abstract

This paper presents InFL-UX, an interactive, proof-of-concept browser-based Federated Learning (FL) toolkit designed to integrate user contributions into the machine learning (ML) workflow. InFL-UX enables users across multiple devices to upload datasets, define classes, and collaboratively train classification models directly in the browser using modern web technologies. Unlike traditional FL toolkits, which often focus on backend simulations, InFL-UX provides a simple user interface for researchers to explore how users interact with and contribute to FL systems in real-world, interactive settings. InFL-UX bridges the gap between FL and interactive ML by prioritising usability and decentralised model training, empowering non-technical users to actively participate in ML classification tasks.

CCS Concepts

• **Computing methodologies → Distributed artificial intelligence**; *Supervised learning*; • **Human-centered computing → Web-based interaction**; *User interface toolkits*.

Keywords

Federated Learning; Interactive Machine Learning; Browser-based Deep Learning

1 Introduction and Related Work

Traditional machine learning (ML) is often constrained by limited data, particularly in specialised domains where data acquisition is expensive or labour-intensive [12]. Moreover, technical barriers hinder direct input from domain experts, further delaying new data collection [7]. To overcome these issues, Fails and Olsen [6] introduced interactive machine learning (IML), which enables non-technical users to train ML models by manually classifying data or correcting outputs. Unlike conventional ML, IML supports real-time updates based on user input, permitting focused, incremental refinements [2, 5, 18]. Extending this work, Tseng et al. [17] developed *Co-ML*, a tablet-based application for collaboratively constructing ML image classification models that emphasises shared dataset design. In this paper, we build on these concepts by proposing a browser-based tool that facilitates collaborative IML using federated learning (FL).

ML models are traditionally trained on centralised datasets. However, in fields such as healthcare, data are distributed across multiple devices and cannot be shared due to privacy constraints. FL mitigates this issue by enabling decentralised training of a shared model while retaining data on client devices [10]. FL aggregates local updates on a central server and comes in two main forms: synchronous and asynchronous. Synchronous FL [10] requires all clients to train concurrently with the latest global model and submit updates together. In contrast, asynchronous FL [19] allows clients to train and submit updates independently, with the server updating the global model upon each submission. Due to its flexibility, asynchronous FL is more suited to our application. Recent advances in JavaScript-based deep learning frameworks, such as *TensorFlow.js* [9, 16], have made browser-based deep learning (DL) feasible. For example, Google's *Teachable Machine* offers a no-code interface for local model training; however, it is limited to local training and does not incorporate FL. In contrast, browser-based FL frameworks proposed by Lian et al. [8] and Ángel Morell and Alba [20] support FL but lack interactive elements, requiring users to supply data in predefined folders.
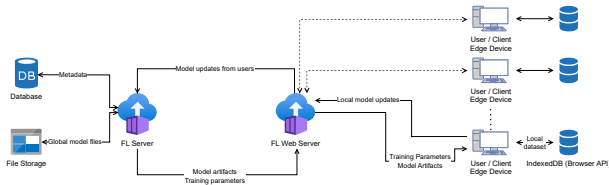
Figure 1: The system architecture overview.

In this work, we introduce **InFL-UX**[1], an interactive, browser-based FL toolkit that demonstrates a proof-of-concept (POC) implementation using contemporary web technologies. The application employs asynchronous FL via FedAsync [19], chosen for its simplicity and single-hyperparameter configuration, to enable collaborative model training. Users can upload datasets, define custom classes, and train models for various classification tasks. While much FL research relies on simulations, **InFL-UX** integrates FL with IML to prioritise user engagement and address FL challenges from a user-centric perspective. We aim to assist FL practitioners in embedding FL into intelligent user interfaces (UIs) and evaluate these setups through user studies. Future enhancements will incorporate advanced FL aggregation methods and additional ML tasks, such as data annotation, to broaden applicability.

2 System Design

InFL-UX utilises ONNX Runtime[2], a cross-platform ML library that supports fast on-device inference and training in web browsers. It leverages modern browser APIs, including WebAssembly [13]—a low-level language offering near-native performance—and WebGPU [11], which facilitates high-performance GPU computations and supersedes WebGL[3]. This browser-based approach, which capitalises on the ubiquity of web browsers, ensures seamless compatibility with models from prevalent ML frameworks via the ONNX format. Developed in Python using Flask[4], the application integrates ONNX Runtime through JavaScript and employs Web Components[5] for UI modularity, thereby avoiding dependence on specific UI frameworks. Persistent storage is managed through IndexedDB [3], allowing user data, such as uploaded files and inference results, to be retained across sessions. The code for **InFL-UX** is publicly available at https://github.com/tmaurer42/interactive-fl-poc.

2.1 System and Application Design

The architecture (Figure 1) comprises two main components: the **FL Server** and the **FL Web Server**. The FL Web Server hosts the web application and delivers required files to client devices, while the FL Server manages client updates, maintains the global model, and provides the latest model version and training instructions on demand. This structure underpins the POC, which primarily addresses image classification; it can also accommodate tasks such as object detection and image segmentation through abstract classes

and supports the simulation of multiple independent clients. An administrator client configures the system via the configuration page (Figure 2a) by specifying the use case, selecting the aggregator and ML model, and setting training parameters. Clients then upload images, receive model-generated label suggestions, and may accept or adjust these labels (Figure 2b). Once reviewed, a training session is initiated locally, and upon completion, updates are automatically forwarded to the central server for aggregation. Clients can evaluate the global model using separate testing datasets (Figure 2c).

Currently, the system implements a single FL aggregation strategy, i.e., FedAsync, and one ML model, i.e., MobileNetV2 [14]. Adding new aggregation strategies or ML models is straightforward, requiring only minimal code modifications: a new aggregation strategy is defined as a Python function with corresponding configurable parameters added to the administrator interface, and additional ML models are integrated via an expanded dropdown menu featuring extra PyTorch models[6]. The system is deployed using Docker[7]. Although designed for extensibility, **InFL-UX** does not support simulation tasks, which are handled by established frameworks such as Flower [4]; instead, it focuses on validating aggregation methods in real-world settings.

2.2 Limitations

Several limitations emerged during development. The relatively new ONNX Runtime integration for the web offers incomplete training functionalities; for example, the optimiser's learning rate is fixed at 0.001, and only a limited set of loss functions is available. Furthermore, while WebGPU currently supports inference, training is restricted to the WebAssembly backend. Browser-specific constraints also affect the application: the local dataset stored in IndexedDB is limited in size (varying by browser and available disk space), and the WebGPU API is presently available only in developer builds of modern browsers, which may be restricted by security policies.

3 Conclusion and Future Work

InFL-UX, demonstrated the feasibility of an interactive, browser-based FL system that integrates state-of-the-art technologies from the UI to the server-side aggregation process. This approach allows embedding FL capabilities into IML systems, particularly in privacy-sensitive domains. Despite its limitations, the application effectively utilises ONNX Runtime and modern browser features (WebAssembly and WebGPU) to support client-side FL training, enabling collaborative training across devices while ensuring compatibility and encouraging user adoption.

Future work should involve user studies to assess real-world adoption, performance, and scalability and extend the application to additional computer vision tasks, such as image segmentation and object detection. Moreover, incorporating explainable AI techniques (e.g. class activation maps [15], and concept-based modelling [1]) could improve transparency and trust. Finally, enabling administrators to deploy new training tasks and allowing clients to participate in multiple tasks with distinct datasets would significantly enhance the system's utility.

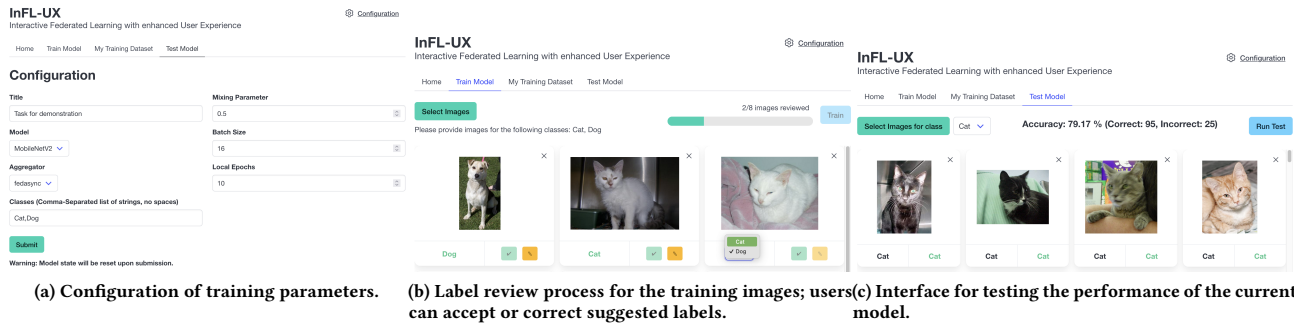[1]The name is a combination of Interactive Machine Learning, Federated Learning, and User Experience.
[2]https://onnxruntime.ai/ (Accessed January 03, 2025)
[3]https://www.khronos.org/webgl/ (Accessed January 03, 2025)
[4]https://flask.palletsprojects.com/en/stable/ (Accessed January 03, 2025)
[5]https://www.webcomponents.org/ (Accessed January 03, 2025)

[6]https://pytorch.org/ (Accessed January 03, 2025)
[7]https://www.docker.com/ (Accessed January 03, 2025)

(a) Configuration of training parameters.

(b) Label review process for the training images; users can accept or correct suggested labels.

(c) Interface for testing the performance of the current model.

Figure 2: Screenshots from InFL-UX showing (a) the configuration page, (b) the label review process, and (c) the model testing interface. The images used in the demo were taken from the Cats and Dogs Classification Dataset © Bhavik Jikadara: https://www.kaggle.com/datasets/bhavikjikadara/dog-and-cat-classification-dataset, which is licensed under Apache 2.0.

Acknowledgments

References

[1] Hasan Md Tusfiqur Alam, Devansh Srivastav, Md Abdul Kadir, and Daniel Sonntag. 2025. Towards Interpretable Radiology Report Generation via Concept Bottlenecks Using a Multi-agentic RAG. In *Advances in Information Retrieval - 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6-10, 2025, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 15574)*, Claudia Hauff, Craig Macdonald, Dietmar Jannach, Gabriella Kazai, Franco Maria Nardini, Fabio Pinelli, Fabrizio Silvestri, and Nicola Tonellotto (Eds.). Springer, 201–209. https://doi.org/10.1007/978-3-031-88714-7_18

[2] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine* 35, 4 (Dec. 2014), 105–120. https://doi.org/10.1609/aimag.v35i4.2513

[3] Joshua Bell. 2024. *Indexed Database API 3.0.* Technical Report. W3C. https://www.w3.org/TR/IndexedDB/ Version Number: 3.0.

[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, and Nicholas D. Lane. 2022. Flower: A Friendly Federated Learning Research Framework. https://doi.org/10.48550/arXiv.2007.14390 arXiv:2007.14390 [cs].

[5] John J. Dudley and Per Ola Kristensson. 2018. A Review of User Interface Design for Interactive Machine Learning. *ACM Transactions on Interactive Intelligent Systems* 8, 2 (June 2018), 1–37. https://doi.org/10.1145/3185517

[6] Jerry Alan Fails and Dan R. Olsen. 2003. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces* (Miami, Florida, USA) *(IUI '03)*. Association for Computing Machinery, New York, NY, USA, 39–45. https://doi.org/10.1145/604045.604056

[7] Joshua Holstein. 2024. Bridging Domain Expertise and AI through Data Understanding. In *Companion Proceedings of the 29th International Conference on Intelligent User Interfaces (IUI '24 Companion)*. Association for Computing Machinery, New York, NY, USA, 163–165. https://doi.org/10.1145/3640544.3645248

[8] Zhuotao Lian, Qinglin Yang, Qingkui Zeng, and Chunhua Su. 2022. WebFed: Cross-platform Federated Learning Framework Based on Web Browser with Local Differential Privacy. In *ICC 2022 - IEEE International Conference on Communications*. 2071–2076. https://doi.org/10.1109/ICC45855.2022.9838421

[9] Yun Ma, Dongwei Xiang, Shuyu Zheng, Deyu Tian, and Xuanzhe Liu. 2019. Moving Deep Learning into Web Browser: How Far Can We Go?. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 1234–1244. https://doi.org/10.1145/3308558.3313639

[10] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, 1273–1282. https://doi.org/10.48550/arXiv.1602.05629

[11] Kai Ninomiya, Jim Blandy, and Brandon Jones. 2024. *WebGPU.* W3C Working Draft. W3C. https://www.w3.org/TR/webgpu/

[12] Yuji Roh, Geon Heo, and Steven Euijong Whang. 2021. A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. *IEEE Transactions on Knowledge and Data Engineering* 33, 4 (April 2021), 1328–1347. https://doi.org/10.1109/TKDE.2019.2946162 Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[13] Andreas Rossberg. 2022. *WebAssembly Core Specification.* Technical Report. W3C. https://www.w3.org/TR/wasm-core-2/ Version Number: 2.0.

[14] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2019. MobileNetV2: Inverted Residuals and Linear Bottlenecks. https://doi.org/10.48550/arXiv.1801.04381 arXiv:1801.04381 [cs].

[15] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 618–626. https://doi.org/10.1109/ICCV.2017.74 ISSN: 2380-7504.

[16] Daniel Smilkov, Nikhil Thorat, Yannick Assogba, Charles Nicholson, Nick Kreeger, Ping Yu, Shanqing Cai, Eric Nielsen, David Soergel, Stan Bileschi, Michael Terry, Ann Yuan, Kangyi Zhang, Sandeep Gupta, Sarah Sirajuddin, D Sculley, Rajat Monga, Greg Corrado, Fernanda Viegas, and Martin M Wattenberg. 2019. TensorFlow.js: Machine Learning For The Web and Beyond. In *Proceedings of Machine Learning and Systems*, A. Talwalkar, V. Smith, and M. Zaharia (Eds.), Vol. 1. 309–321. https://doi.org/10.48550/arXiv.1901.05350

[17] Tiffany Tseng, Jennifer King Chen, Mona Abdelrahman, Mary Beth Kery, Fred Hohman, Adriana Hilliard, and R. Benjamin Shapiro. 2023. Collaborative Machine Learning Model Building with Families Using Co-ML. In *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference* (Chicago, IL, USA) *(IDC '23)*. Association for Computing Machinery, New York, NY, USA, 40–51. https://doi.org/10.1145/3585088.3589356

[18] Hasan Md Tusfiqur, Duy MH Nguyen, Mai TN Truong, Triet A Nguyen, Binh T Nguyen, Michael Barz, Hans-Juergen Profitlich, Ngoc TT Than, Ngan Le, Pengtao Xie, et al. 2022. DRG-Net: interactive joint learning of multi-lesion segmentation and classification for diabetic retinopathy grading. *arXiv preprint arXiv:2212.14615* (2022).

[19] Cong Xie, Sanmi Koyejo, and Indranil Gupta. 2019. Asynchronous Federated Optimization. *arXiv e-prints*, Article arXiv:1903.03934 (March 2019). https://doi.org/10.48550/arXiv.1903.03934 arXiv:1903.03934 [cs.DC]

[20] José Ángel Morell and Enrique Alba. 2022. Dynamic and adaptive fault-tolerant asynchronous federated learning using volunteer edge devices. *Future Generation Computer Systems* 133 (2022), 53–67. https://doi.org/10.1016/j.future.2022.02.024