T4Train: Rapid Prototyping of ML-Driven Interactive Applications

Yasha Iravantchi
University of Michigan
Ann Arbor, Michigan, USA
yiravan@umich.edu

Alanson P. Sample
University of Michigan
Ann Arbor, Michigan, USA
apsample@umich.edu

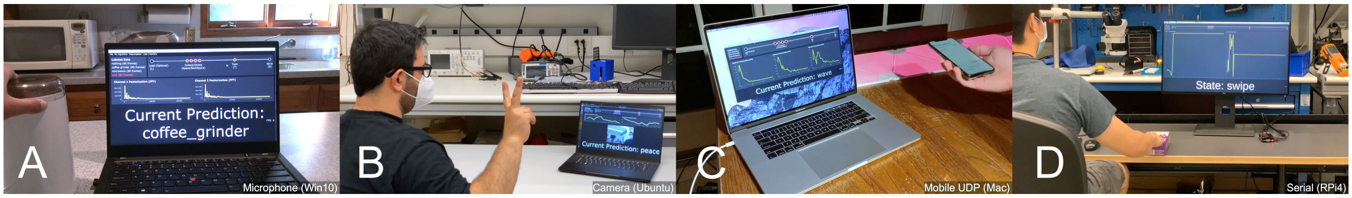

Figure 1: T4Train's extensible open-source and cross-platform framework includes 7 integrated sensor interfaces. Demonstrated above are A) Microphone on Windows B) Camera on Linux, C) UDP via smartphone on Mac, D) Serial on Raspberry Pi

ABSTRACT

Pairing real-time ML with sensor data drives many interactive applications. However, the tools to prototype these applications are often proprietary and not open-source. This course instructs how to build interactive sensing applications using T4Train, an open-source and user-friendly framework for rapid prototyping. Participants will learn sensor interfaces (e.g., on a laptop/Arduino), signal processing, ML, and the T4Train tool. Afterward, they will build real-time, interactive sensing systems, such as LED lighting that reacts to different sounds or hand movements. This course builds on 4 semesters of instruction using T4Train and multiple HCI research contributions, including 3 award-winning papers at CHI.

CCS CONCEPTS

• **Human-centered computing** → **User interface toolkits**; **Ubiquitous and mobile computing systems and tools**.

KEYWORDS

Machine Learning Tools, Interaction, Sensors, Open Source, Visualization, Sensing, Course

ACM Reference Format:
Yasha Iravantchi and Alanson P. Sample. 2024. T4Train: Rapid Prototyping of ML-Driven Interactive Applications. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24), May 11–16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3613905.3636269

CHI EA '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0331-7/24/05
https://doi.org/10.1145/3613905.3636269

1 BENEFITS

As interactive sensing researchers, we utilize a simple yet effective 6-step approach in developing new and valuable sensing applications: 1) identify a potentially valuable signal for exploration, 2) design or select a sensor to capture this signal, 3) digitize the signal and send to a computing device, 4) visualize the data for human interpretation, 5) use signal processing and machine learning (ML) to interpret the signals, and 6) perform an action based off of ML predictions. While the democratization of open-source commodity sensing hardware (like Arduinos) or open-source ML frameworks (like SciKit-Learn/Tensorflow) have made Steps 2 and 5 in this 6-step approach easier to develop and more reproducible across practitioners and researchers, the software that acts as a "glue" to connect sensor data to machine learning, as seen in many HCI research works, is largely proprietary and challenging to create from the ground up. This means that starting researchers, practitioners, and makers must implement their own framework, creating a high barrier to entry to prototype and experiment with novel interactive sensing applications. Furthermore, without a set of common tools that are open-source, interactive sensing research cannot benefit and grow in the same way that the areas of electronic hardware and machine learning have from the accessibility and availability of Arduinos and SciKit-Learn/Tensorflow.

This course intends to teach attendees how to engineer their own interactive systems with this 6-step approach using T4Train, an open-source, cross-platform framework that makes it easier for users to experiment and prototype with various approaches at each step of the process. The course materials, including the course outline, will be available at this **link**. At the end of the course, attendees will gain an intuition of how to interface and visualize data from real-time sensors (from the ones provided as part of the course or one they bring themselves), learn simple yet powerful signal processing techniques, collect data, quickly train a real-time live ML model, and use this framework to rapidly prototype interactive applications. Attendees can choose to develop their own applications or select guided prototype examples such as hand-gesture-based LED light controls or a playful Arduino robot that dances different moves to whistles, hums, and claps.

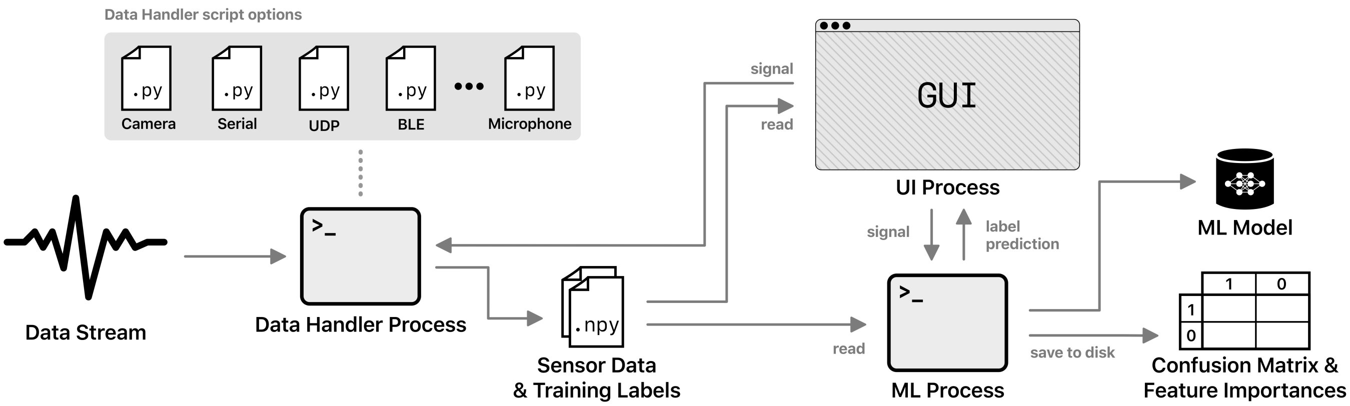

Figure 2: T4Train's process flow demonstrates how the tool handles each step in prototyping interactive applications.

2 INTENDED AUDIENCE(S)

This course is intended for a general computer science audience and anyone who wishes to learn how to quickly prototype interactive applications using sensors, machine learning, and prototyping hardware (e.g., Arduinos and sensors); we hope a diverse group of attendees will make for a more interesting collaborative course. This course would be of particular interest to makers, practitioners, and researchers in this area.

3 PREREQUISITES

No background is required to understand the course beyond a passing familiarity with Python (for T4Train) and C++ (for Arduino); if a potential attendee is unsure, please contact the instructors. Attendees are asked to bring a laptop if they can, as the instructors will only have a limited number of spare laptops. The instructors will group attendees into teams and provide each group with an Arduino and sensor kit. Attendees will receive a survey beforehand so that the instructors can tailor the content to them.

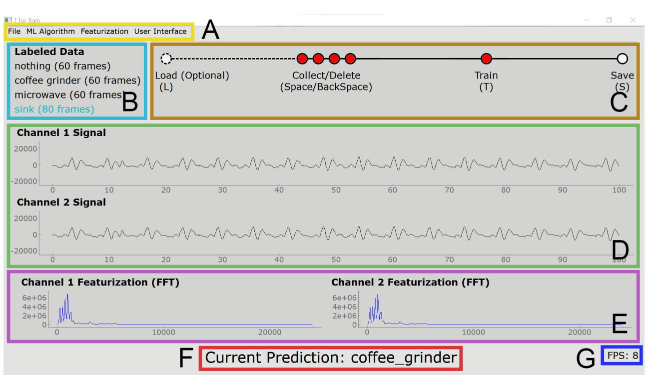

Figure 3: A screenshot of T4Train's UI capturing microphone data. Highlighted are the UI elements that visualize data and features and guide the user to collect data and train ML models for interactive applications: A) context menu B) labels C) progress bar D) raw data plots E) featurized data plots F) current prediction G) FPS tracker

4 CONTENT

This course will consist of two components: a lecture-style overview of the building blocks for developing interactive applications and a hands-on workshop where attendees, with instructor assistance, will select sensors, prototype with different signal processing and ML algorithms, and develop their own interactive applications.

4.1 Lecture: Developing Interactive Applications

The course's lecture component will briefly introduce various common sensing approaches (e.g., microphones, cameras, IMUs) and some effective rules of thumb to identify whether the selected sensor is appropriate for a given task. Then, the course will introduce common approaches to connect these sensors to computers (e.g., USB Serial, WiFi UDP) and ways to ensure data integrity. Finally, the course will gently introduce signal processing, featurization, and simple yet effective classical machine learning approaches suitable for realtime interactive applications.

4.2 Workshop: Prototyping Interactive Applications

Brainstorming an Application: First, in groups (size depending on course attendance), attendees will brainstorm an interactive application by identifying a set of detectable events (e.g., whistling, clapping, humming) as inputs and a set of actions (e.g., flashing a color on the screen, playing a sound, sending a command to an Arduino to move a servo motor) as outputs. These inputs and outputs can then be used to design an interactive application. For example, an example application could be when a user whistles, a toy car moves in the forward direction, and when the user hums, the toy car moves in reverse.
Selecting a Sensor: Second, based on the groups' applications, they will select a sensor. This could include a laptop's microphone or camera, any of the sensors in the provided Arduino Nano 33 Sense, a sensor brought by the instructors, or their own. Once having selected a sensor, they will connect the sensor to T4Train by selecting an appropriate data handler from the drop-down menu. If an Arduino-based sensor is selected, the group will also program the Arduino to transmit the sensor data.

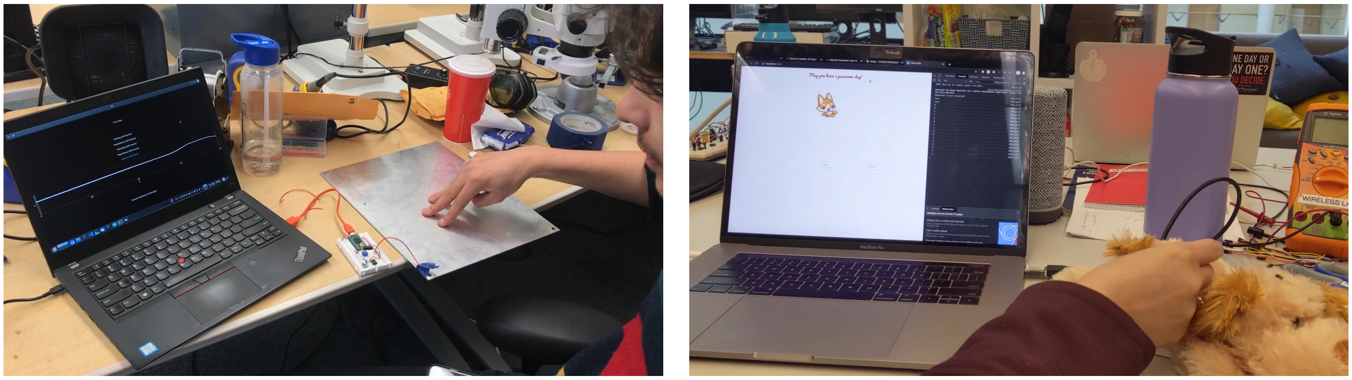

Figure 4: Example student projects: (left) a music controller where different touch gestures on a surface paused/played music and increased/decreased volume and (right) a stuffed animal that enjoyed head petting but disliked when its paws were touched.

Visualizing the Sensor Data: Once the sensor is selected, the realtime data can be visualized in T4Train. The groups will test various signal processing approaches to find an effective way to featurize the signal to be visually distinguishable. For example, by selecting the FFT featurization module, T4Train can visualize incoming microphone data as its frequency components and experiment with the number of bins: too few bins and similarly pitched whistles will be lumped together; too many bins and the system becomes fragile. By visualizing the sensor data, groups can confirm that their detectable events, sensing approach, and signal processing are effective before collecting data and training a model.
Collecting Training Data and Training a Realtime Model: The groups will then collect training data for each detectable event by defining the labels in T4Train. They can select the current label using the "Up Arrow" and "Down Arrow" keys and press the "Spacebar" to store a data sample for that event. Groups can then select an integrated ML algorithm and press the "T" key to train based on their collected data, hence the platform's name. They can test the performance of the trained model by performing their events and decide to collect more data for each event and retrain by pressing "T" again or select a different ML model and retrain. Once the ML model has expected performance, they can save the collected data and model state by pressing "S" to save.
Performing an Action Off Predictions: Finally, to finish off their applications, groups will tie in their expected actions to T4Train's realtime predictions. Once they have modified T4Train's code to send commands based off the predictions (e.g., send a command to the toy car to move forward), they can load their previously trained model using the "L" key. They can further refine their application by exploring the tradeoff between responsiveness and smoothness. For example, adding a window can smooth the transition between predicted events (e.g., reduce the jitter of the toy car's movement) but will increase the latency between the event and the action. At the end, groups will showcase their applications, and the instructors will close with final remarks.

5 PRACTICAL WORK

Outside of the short lecture component of this course, attendees will have hands-on instruction in building interactive systems. This includes: programming the Arduino to properly send sensor data to the laptop; programming T4Train to correctly capture and visualize the incoming sensor data; selecting or implementing their own signal processing scheme to convert time-domain signals into features; collecting labeled data samples for training an ML model; evaluating the performance of different ML models; and using the prediction of the ML models to perform an action. Attendees can use the integrated T4Train features or implement their own components to drive their applications.

6 INSTRUCTOR BACKGROUND

Yasha Iravantchi is a Ph.D. Candidate at the University of Michigan, advised by Prof. Alanson Sample, and a Meta Ph.D. Research Fellow. His research focuses on developing novel sensing hardware and has been featured in venues such as CHI [2–5], UIST [7], and MobiCom [1, 6]. At CHI, his sensing research has received three paper awards [2, 4, 5], each using a version of T4Train to support realtime interactivity. As a Teaching Assistant, he has utilized T4Train as an instructional tool in Prof. Sample's *EECS 598: Engineering Interactive Systems* course across four semesters, where students used the tool for an instructional lab and final projects. More information can be found at his website.

Dr. Alanson Sample is an Associate Professor at the University of Michigan (UM), where he leads the Interactive Sensing and Computing Lab. His research interests lie broadly in the areas of Human-Computer Interaction (HCI), Mobile Systems, and wireless technology. Prior to joining the UM, he was a Lab Director at Disney Research, Los Angeles, where he led his team in creating new interactive experiences by applying novel approaches to electromagnetics, embedded systems, and HCI. Prior to Disney, he was a Research Scientist at Intel Labs in Hillsboro, working on energy harvesting for wearable and Internet of Things applications.

Alanson received his Ph.D. in Electrical & Computer Engineering in 2011 from the University of Washington. Throughout his graduate studies, he worked at Intel Research Seattle on projects related to wireless power delivery using magnetically coupled resonance, energy harvesting as well as ubiquitous computing. He is currently an editor of the Proceedings of the ACM on Interactive, Mobile, Wearable, and Ubiquitous Technologies, and previously served as a subcommittee chair for the CHI Building Devices subcommittee. More information can be found at his website.

Both instructors have extensive experience in building real-time sensor+ML systems in both engineering and research contexts. Additionally, both have experience teaching university-level sensing-related coursework, including electronics and signal processing. This course is adapted from an *EECS 598: Engineering Interactive Systems* assignment taught by both instructors.

7 RESOURCES

Attendees can access T4Train's source code at this GitHub repo. All course materials, including an informative video figure showing the capabilities of T4Train and example applications using T4Train, will be available at this **link** in advance of the course. For any other questions about the course or requests for additional resources, the instructors can be reached via email. Since T4Train can support various data inputs, the instructors also invite potential attendees to email them in advance if there are particular use cases or sensors they would like to work with as part of the course. The instructors are happy to accommodate exciting new sensors outside of the ones detailed earlier.

8 ACCESSIBILITY

Attendees in need of accessibility arrangements are encouraged to contact the course instructors, as they will make every effort to provide an inclusive course experience. For blind or visually impaired (BVI) attendees: while T4Train has not explicitly been tested with existing BVI-assistive tools, such as screen readers, the user interface is built using QT which does suggest support for accessible UI elements. Additionally, given T4Train's modular structure, the UI elements can be modified ad-hoc to read elements when selected, announce various events and actions, or alternative ways to replace a visual UI. For deaf and hard of hearing (DHH) attendees: all lecture and spoken content can be presented alongside a live captioning system via Zoom. For the workshop, T4Train does not incorporate audible feedback mechanisms.

REFERENCES

[1] Dongyao Chen, Mingke Wang, Chenxi He, Qing Luo, Yasha Iravantchi, Alanson Sample, Kang G. Shin, and Xinbing Wang. 2021. MagX: Wearable, Untethered Hands Tracking with Passive Magnets. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking* (New Orleans, Louisiana) *(MobiCom '21)*. Association for Computing Machinery, New York, NY, USA, 269–282. https://doi.org/10.1145/3447993.3483260

[2] Yasha Iravantchi, Karan Ahuja, Mayank Goel, Chris Harrison, and Alanson Sample. 2021. PrivacyMic: Utilizing Inaudible Frequencies for Privacy Preserving Daily Activity Recognition. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 198, 13 pages. https://doi.org/10.1145/3411764.3445169

[3] Yasha Iravantchi, Mayank Goel, and Chris Harrison. 2019. BeamBand: Hand Gesture Sensing with Ultrasonic Beamforming. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 15, 10 pages. https://doi.org/10.1145/3290605.3300245

[4] Yasha Iravantchi, Yang Zhang, Evi Bernitsas, Mayank Goel, and Chris Harrison. 2019. Interferi: Gesture Sensing Using On-Body Acoustic Interferometry. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article 276, 13 pages. https://doi.org/10.1145/3290605.3300506

[5] Yasha Iravantchi, Yi Zhao, Kenrick Kin, and Alanson P. Sample. 2023. SAWSense: Using Surface Acoustic Waves for Surface-Bound Event Recognition. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 422, 18 pages. https://doi.org/10.1145/3544548.3580991

[6] Mingke Wang, Qing Luo, Yasha Iravantchi, Xiaomeng Chen, Alanson Sample, Kang G. Shin, Xiaohua Tian, Xinbing Wang, and Dongyao Chen. 2022. Automatic Calibration of Magnetic Tracking. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking* (Sydney, NSW, Australia) *(MobiCom '22)*. Association for Computing Machinery, New York, NY, USA, 391–404. https://doi.org/10.1145/3495243.3558760

[7] Yang Zhang, Yasha Iravantchi, Haojian Jin, Swarun Kumar, and Chris Harrison. 2019. Sozu: Self-Powered Radio Tags for Building-Scale Activity Sensing. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 973–985. https://doi.org/10.1145/3332165.3347952