Designing Explainable AI for TinyML Systems

by
Paul Kim
Professor Iris Howley, Advisor

A thesis submitted in partial fulfillment
of the requirements for the
Degree of Bachelor of Arts with Honors
in Computer Science

Williams College
Williamstown, Massachusetts

May 20, 2024

Contents

1 Introduction **8**
- 1.1 The Problem . . . 8
- 1.2 Goals & Plans . . . 9
- 1.3 Thesis Outline . . . 11

2 Background & Prior Work **12**
- 2.1 Explainable AI . . . 12
  - 2.1.1 Types of Explanations . . . 13
  - 2.1.2 Shapley Values . . . 14
  - 2.1.3 What is a Good XAI? . . . 14
- 2.2 Tiny Machine Learning . . . 15
  - 2.2.1 Advantages and Disadvantages of TinyML . . . 16
  - 2.2.2 Motivations for TinyML . . . 17
  - 2.2.3 TinyML Pipeline . . . 17
  - 2.2.4 TinyML Datasets . . . 18
- 2.3 Combined XAI Pipelines . . . 19
  - 2.3.1 TinyML-XAI Pipeline . . . 19
- 2.4 Understanding & Evaluating Systems . . . 19
- 2.5 Summary . . . 21

3 User Information **22**
- 3.1 Identifying and Selecting Users . . . 22
  - 3.1.1 How To Identify Users . . . 22
  - 3.1.2 Selecting Users . . . 23
- 3.2 Finding Demographic Information . . . 24
  - 3.2.1 Web Scraping . . . 24
  - 3.2.2 Results of Web Scraping . . . 24
  - 3.2.3 Target Demographic . . . 25
- 3.3 Summary . . . 27

4 Methodology **28**
- 4.1 Explainable Design . . . 28
  - 4.1.1 Empathize & Define . . . 28
  - 4.1.2 Ideate . . . 29
  - 4.1.3 Swimming Dataset . . . 30
  - 4.1.4 Prototype & Test . . . 31
- 4.2 Final User Studies . . . 33
  - 4.2.1 Participants . . . 33
  - 4.2.2 Materials . . . 34

4.2.3 Design . . . 36
4.2.4 Procedure . . . 36
4.3 Summary . . . 36

5 Results & Discussion **37**
5.1 Results . . . 37
5.1.1 Pre-Test Analyses . . . 37
5.1.2 Post-Test Exploratory Analyses . . . 38
5.1.3 Pre- and Post- Test Changes . . . 39
5.2 Discussion . . . 40
5.2.1 Limitations . . . 44
5.3 Summary . . . 45

6 Conclusion **46**
6.1 Contributions . . . 46
6.2 Future Work . . . 48
6.2.1 Diverse Population Sample . . . 48
6.2.2 Experimentally Designed Explainable . . . 48
6.3 Summary . . . 48

Appendices **50**

A Pre- and Post-Test Questions **51**
A.1 Pre-Test . . . 51
A.1.1 Demographic Information . . . 51
A.1.2 Math and CS Background . . . 52
A.2 Post-Test . . . 53
A.2.1 Explainables Quiz . . . 53
A.2.2 Explainable Evaluation . . . 54

List of Figures

1.1 A dual-pronged information approach for explainables. Necessary for tackling both the main concepts of TinyML use and interpreting output during application use. . . 9
1.2 Design steps for creating and evaluating explainables. It follows a user-centered design process with iterative prototyping to create an explainable that is usable and effective. 10

2.1 Examples of various explainables 13
2.2 TinyML pipeline that serves to highlight key steps in working with TinyML 18
2.3 TinyML-XAI pipeline that improves upon the previous TinyML pipeline by incorporating a post-hoc model explanation for increased understandability 20

3.1 Harvard EdX Course for TinyML that we datascraped user information from 23
3.2 TinyML Users by Continent 25
3.3 TinyML Users by Motivation 26
3.4 TinyML Users by Occupation 26

4.1 Stanford D-School Bootleg Design Process that we use to guide our design thinking 29
4.2 Explainables that use various metaphorical narratives 30
4.3 Swimming Stroke Examples Used in Explainable 31
4.4 Paper prototype for the explainable that was created as an outline that is critiqued . 32
4.5 Screenshot 1 of Completed Web Application 35
4.6 Screenshot 2 of Completed Web Application 35
4.7 Procedure that users undertook during user studies 36

5.1 Box-and-whiskers plot of the six questions on AI perspectives 38
5.2 Responses to SUS Questions in Post-Test 40
5.3 Changes in Response to AI View Questions 41
5.4 Changes in Response to AI and TinyML Confidence 41

List of Tables

2.1 User Interface Evaluation Methods 21

3.1 User Demographic Response Rates 24

4.1 Nielsen's 10 Usability Heuristics 33

5.1 Nine questions asked based on the System Usability Scale 39

Abstract

We see an emerging field of artificial intelligence (AI) called tiny machine learning (TinyML) becoming increasingly popular. As TinyML is a more accessible and cost-friendly version of machine learning (ML), we expect to see more novice users using ML for the first time through TinyML. Accordingly, this novice user base still needs to understand how to effectively and efficiently use TinyML but needs some framework to guide them. One popular way of explaining AI systems is with the use of post-hoc explainables, or visualizations that use graphics and user interfaces, to create user understanding of a system. In addition to this understanding of a system, users must also be able to interpret the output of an ML model, be it through inherently interpretable models or post-hoc explanations with explainable AI (XAI).

Given these needs, we are interested in creating a explainable as a medium to increase user understanding of a TinyML system both in understanding essential concepts of TinyML and interpreting output. We propose a combined TinyML-XAI pipeline to introduce to users as this sufficiently encapsulates our explaining goals. In order to effectively convey this information to users, we utilize user-centered processes to create an explainable that is suitable for our goals.

Following the user-centered design process, we first datascraped a forum from an online TinyML course so that we could capture some basic characteristics of what we expect to be typical TinyML users. We use characteristics we define from our users to guide how we create our explainable. We then use iterative prototyping throughout our design process to ensure that the explainable we were building was usable and effective.

After designing and creating the explainable for TinyML, we ran user studies with pre- and post-tests we created to determine the overall usability and effectiveness of our TinyML explainable. The user studies indicated that our explainable was effective in usability and effectiveness in informing participants about TinyML, particularly with increasing user confidence in knowledge about AI and TinyML. The results of the user studies indicate effectiveness in our ability to create an effective explainable by using user-centered design processes and iterative prototyping to explain essential concepts of TinyML and its outputs.

Acknowledgments

Thank you to my family and friends for supporting me throughout my four years at Williams I would also like to thank South Science and Jesup for being there all those nights. A big shoutout to the Driscoll Breakfast Club for supporting me through this thesis as well. Also a thank you to the students who agreed to participate in my user studies. I'd also like to thank the CS department for guiding me throughout the last four years—I never would've done a thesis without them!

I would also like to thank my second reader. Bill Jannen, for his support, feedback, and advice throughout this whole process. And finally, I would like to thank Iris Howley for advising me in not only this thesis but also my journey at Williams. I've spent a lot of time with her since meeting her Junior fall and I would not be where I am now without her!

Chapter 1

Introduction

1.1 The Problem

A current trend within artificial intelligence (AI) is to build increasingly large and complex models. This trend has resulted in the size of machine learning (ML) models accelerating by several orders of magnitude within the last several years [81], as well as the creation and use of a massive amount of data [3]. This trend has so far proved effective as developments in computing power and the availability of big data created an environment suitable for large ML models [76]. However, using these large models comes with several drawbacks, such as environmental costs [7, 45], financial costs of running these models [7], and privacy concerns for the data used to train the models [73]. In response to these issues, a new field of study investigating smaller ML models arose.

This new field, dubbed tiny machine learning (TinyML), specifically sought to fix the above issues, promising lower energy costs—which results in a lower environmental footprint, lower economic costs, and an increase in data privacy [28]. These benefits have led to growth in the TinyML field, with researchers and engineers alike looking to build increasingly smaller and more efficient models [63]. Because TinyML as a field is centered around the minimization of resources in ML computing, it lends itself towards an application based study as opposed to a theoretical base, given that much of the underlying TinyML theory is translated from that of traditional ML systems [40]. This application-based study allows for many users to begin using TinyML systems without a strong theoretical background; without this knowledge, however, it is difficult for users to comprehend and effectively use these systems.

Currently, there are a few resources guiding non-ML experts on how to use TinyML systems, mostly with the assistance of pre-existing packages and software [16, 65, 79]. These resources include books covering the role of TinyML and how to use existing software to create TinyML systems [83] and online courses that instruct individuals on how machine learning broadly and TinyML specifically work [37]. However, there is no widespread consensus on what should be considered key concepts of using TinyML. One such concept is creating datasets, where users will have to create or curate datasets to build TinyML models. One other concept is how to build TinyML models. These key concepts are important for novice users to fundamentally understand and effectively use TinyML.

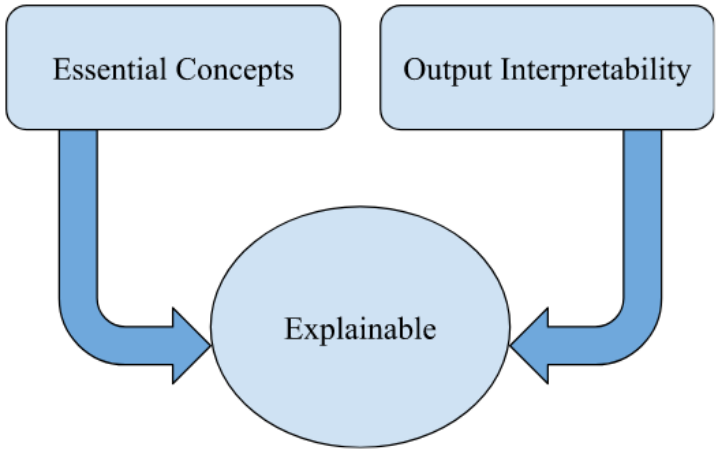


Figure 1.1: A dual-pronged information approach for explainables. Necessary for tackling both the main concepts of TinyML use and interpreting output during application use.

Another issue we run into is that we are unable to control for trust among the users of TinyML systems. It is one thing for users to understand TinyML as a concept; it is another for users to understand and have confidence in the output of TinyML systems.

So, we must tackle user understanding of TinyML from two sides. The first is that users must understand what TinyML is at a high level and understand key concepts for TinyML applications. We must ensure that the concepts that users are learning about are essential for real-world applications of TinyML without overloading the user with unnecessary information [11, 43]. Unfortunately, there is no preset list of concepts that is deemed essential for machine learning, much less one for TinyML, that exists. An important issue is figuring out whether a concept is important, how to measure understanding of said concept, and how to develop user skills in the concept.

The second aspect of understanding we must contend with is user understanding of the TinyML application at run time. While it is vital for the user to understand key concepts of how TinyML works, it is equally important to ensure that the user understands what the TinyML model means. This involves ensuring user trust and understanding of TinyML model output, as well substantive interpretability of the output. Like the first aspect of understanding, we must balance between necessary and unnecessary levels of information [25]. The important issue here is giving users a level of understanding of TinyML output that is appropriate given their knowledge of machine learning. This level of understanding is context dependent so knowing who we're designing for is important to achieve this balance.

1.2 Goals & Plans

This thesis will focus on developing a framework that allows experienced ML users and AI novices alike to effectively navigate the TinyML pipeline, although our focus will be on AI novices. A key concept at the center of our framework should be the difference between TinyML systems and traditional ML systems. Since the creation and application of TinyML models is often inherently

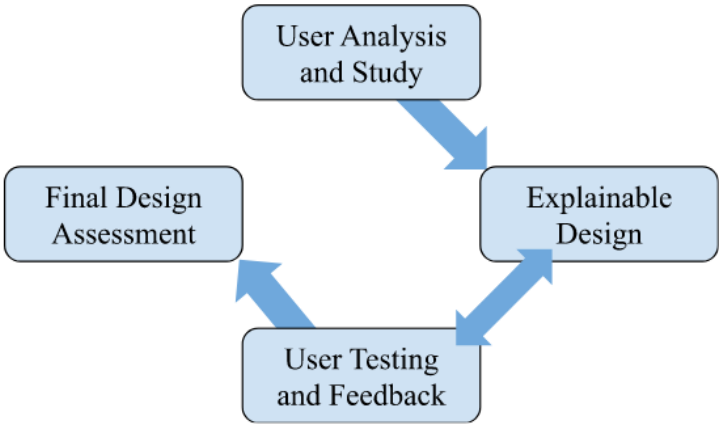


Figure 1.2: Design steps for creating and evaluating explainables. It follows a user-centered design process with iterative prototyping to create an explainable that is usable and effective.

different from traditional ML models, we have different goals and objectives in each stage of our ML pipelines. Keeping these goals in mind, our approach is based on Backward Design [84] as to not treat TinyML models purely as modified ML models.

Like previously mentioned, this framework needs to tackle both explaining the main concepts of TinyML use and application as well as interpreting output at application run-time. In order to effectively explain these concepts, we organize our ideas into an ML pipeline that covers all topics necessary. All topics necessary includes essential high-level concepts of TinyML as well as essential information about output interpretability seen in Figure 1.1 as a two-pronged information approach.

One method of effectively explaining a topic is the use of explainables. Explainables use visualizations, typically in the form of graphics and interactive user interfaces, to explain various concepts [32]. In addition to using explainables, it remains important to understand who is being taught. The beginning of this thesis will involve finding and classifying different demographic information about users who are looking to use TinyML. From this initial information about users who are looking to use TinyML, we create an explainable that is tailored closer to the needs of a real-world user base. This helps ensure that our explainable has real world applications for users who are looking to get familiar with TinyML. By keeping the users at the center of our design process, we create explainables that are more effective for our user base [66].

In this research, we employ an iterative design process where we design an explainable, test it with users, and make necessary modifications on the explainable, before testing it with users again as seen in Figure 1.2. This process will be repeated as necessary until our final iteration, which then undergoes one final assessment. By constantly testing explainables with users, we follow a user-centered design process so that our final explainable is as applicable for our user base. However, there are many costs associated with creating multiple explainables, mainly time and effort costs. To correct for this, we create explainables at varying levels of fidelity so that we obtain user information and feedback while keeping costs low with user testing [27].

We also obtain empirically grounded information about how effective the explainable is. This is

investigated by means of a pre- and post- test examination to procure both subjective assessment data as well as a more objective measure for assessing the data for whether our explainable was effective in explaining TinyML to users.

1.3 Thesis Outline

- **Chapter 1** begins to outline the problem that this project looks to address. This leads into a discussion of the goals of this project, which includes finding a target population for the explanation, creating an explainable for TinyML, and determining how effective the explainable is in having users gain an appropriate level of understanding for TinyML.
- **Chapter 2** covers two main topics: explainable AI (XAI) and tiny machine learning (TinyML). The first section covers what explainable AI is, different types of explainable AI, and what an effective XAI system should look like. We go more in-depth into Shapley Values, a type of post-hoc XAI method that is implemented in the explainable. The section covers what TinyML is, why it is used, and current issues within the field. This includes a discussion of machine learning pipelines and why they are necessary for optimal systems. These two sections are combined when proposing a combined XAI-Tiny pipeline.
- **Chapter 3** discusses the data scraping of TinyML forums. Then, we discuss the creation of the explainable itself; this involves the various prototyping stages. Here, we discuss the design choices that went into the TinyML-XAI pipeline explainable. Finally, we discuss how we are evaluating the effectiveness of the explainable based on how much the user understands and feels confident in using TinyML in the future and whether the explainable itself is usable.
- **Chapter 4** discusses the results of the evaluations that were laid out at the end of the methodology section. We discuss aspects such as trust in machine learning systems and how our explainable impacts it, as well as areas where the explainable is limited.
- **Chapter 5** provides a quick recap of why this study exists and future directions this research could take. The reasons on why this study exists includes what this thesis contributes to existing literature and possible applications of the research.

Chapter 2

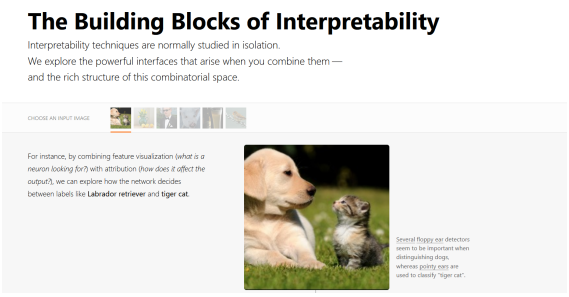
Background & Prior Work

Explainable AI is an increasingly common system used to increase transparency in machine learning systems. This explainable framework is applied to tiny machine learning (TinyML) for users to gain a deeper understanding of how to utilize the system.

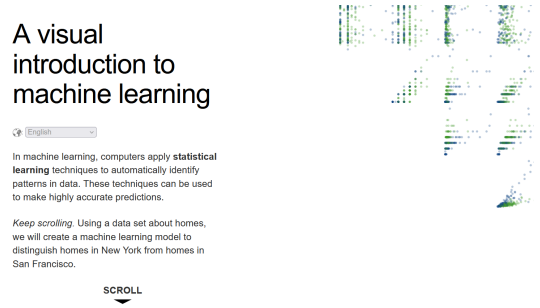
2.1 Explainable AI

Varying ML models have varying levels of interpretability and transparency. Decision trees, for example, rank high in interpretability as they mimic the human decision making process [49]. Neural networks and deep learning, on the other hand, are opaque systems that are much more difficult to interpret. We cannot fully interpret and understand why these models make certain decisions as a result of their opaqueness. This lack of transparency has also led to distrust in ML models as we cannot properly hold them accountable for their outputs [4]. These outputs could be filled with implicit biases from the datasets and other systematic failures—and no one would detect them or know how to rectify these issues. In response to these consequences created by deep learning and other opaque systems, there has been rapid growth in the field of **Explainable AI (XAI)** [86].

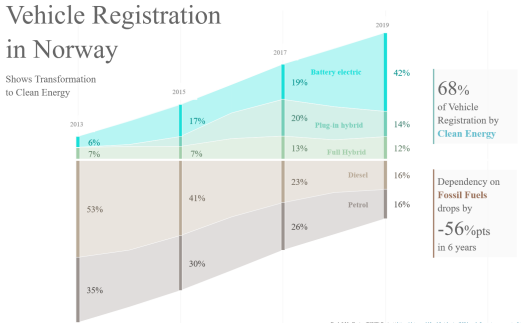
XAI looks to produce more explainable models as it enables people to understand, trust, and manage the AI systems they interact with [6]. This understanding is important for non-ML experts as many users of ML systems today are non-experts—typically subject matter experts in other fields—that don't need to understand every technical detail of machine learning and AI systems but do need to know how to effectively use ML models. Therefore we must consider three main points with XAI: how to produce an explainable model, how to design an explainable model interface, and what is required for an effective explainable [26]. An effective XAI leaves users with a deeper understanding of the ML model; satisfaction in its explanation; belief that the explanation was complete, useful, accurate, and trustworthy; as well as confident in the model's output [31]. There are many ways, however, to create an effective XAI model.



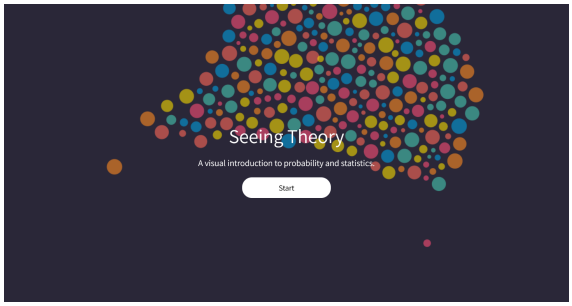
(a) The Building Blocks of Interpretability



(b) A Visual Introduction to Machine Learning



(c) Norway Vehicle Registration



(d) Seeing Theory

Figure 2.1: Examples of various explainables

2.1.1 Types of Explanations

Currently in literature, there are two large classifications of XAI: inherently interpretable design and post-hoc explanations [74]. The former inherently reveals how a model works—including model structure, parameters, algorithms, and optimizations, which is helpful for developers of ML models [86]. Transparency design models necessitate an entirely different workflow, where the designer must be proactive in being transparent throughout the entire development process, integrate transparency gradually in the development process in accordance with its complexity, and be understanding of the audience perspective so that the explanations are effective [23]. Transparency design ultimately works to improve mental models of the ML model so that individuals are able to understand the model as a whole [20].

The second classification of XAI is post-hoc explanations, which is helpful for users of ML models who do not need to fully understand the how of an ML model [86]. Post-hoc explanations look to reveal why a model produces specific outputs. Several classifications of post-hoc explanations include natural language explanations, visualizations of models, local explanations, and explanations by example, where a combination of these is typically used in a post-hoc explainable [48]. Additionally, there are sub-classifications of factual, counterfactual, and semi-factual examples, with factual explanation being the most prominent of the group [41]. Regardless of type, post-hoc explanations work by providing an in-depth explanation and/or examples of why certain inputs into a model result in specific outcomes.

One example of a post-hoc explanation are AI explainables. AI Explainables use visualizations,

typically in the form of graphics and interactive user interfaces, to explain AI to users [32]. Currently, most AI explainables are created by participants in the IEEE VIS Visualization for AI Explainability Workshop [58] as shown by the examples in Figure 2.1. One advantage of post-hoc explainables is the robustness of model data. Because the user adjusts the ML model with respect to the features, the user tests the model for varying inputs, allowing them to reach a deeper understanding of the causal relationships inherent to the model [47]. Explainables also are designed to use cognitively active tasks with specific goals in mind for the interactions. By appropriately using performance-approach goals in the explainable, optimal motivation is promoted so that the user's focus on the explainable is maximized [29].

Ultimately, choosing what type of explanation to use for an ML model depends on the designer's goals and intended audience. As stated before, ante-hoc explanations work well for ML model developers while post-hoc explanations work well for ML model users. If we look at post-hoc explanations' subcategories, we see varying possible use cases for each category. For instance, natural language explanations have the advantage and disadvantage of being very malleable to the designer's wishes [48]. Visualization explainables have the benefit of being able to clearly depict relationships between objects and information digestibly [68]. This would make it easier for a non-ML expert to grasp the concepts necessary to effectively use the ML model. Or, we could use explanations by examples to more clearly depict causal relationships. Evidently, there isn't a single way to effectively use explainables.

2.1.2 Shapley Values

One notable example of a post-hoc explanation method is Shapley values, a concept that comes from cooperative game theory. At a high level, Shapley values give a clearer understanding of how each feature contributes towards some prediction [34].

In more precise terms, Shapley values take as input a set function $v : 2N \rightarrow R$. The Shapley value produces attributions s_i for each player $i \in N$ that add up to $v(N)$. The Shapley value of a player i is given by [77, 67]:

$$\varphi_i(v) = \sum_{S \subset N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} (v(S \bigcup \{x_i\}) - v(S)) \tag{2}$$

Shapley values are useful in having users understand the contribution of each feature, be it positive or negative, in a prediction in comparison with the average prediction, something that is helpful for machine learning algorithms with low interpretability and/or transparency [34]. Having an explainable system like Shapley values for more opaque systems such as neural networks is one strategy for having ML novices and ML users at large gain a clearer understanding of how to interpret the output of an opaque model.

2.1.3 What is a Good XAI?

While using XAI lets users to gain a better understanding of the ML models they use, there are certain pitfalls that we must avoid. One of these is creating misleading explanations for opaque

systems [50]. By misrepresenting the explainable for the ML system being explained, we run the risk of perpetuating the biases inherent in the opaque system. These risks include failing to capture the causal relationship in the model, using an explanation that is quantitatively sound but qualitatively incorrect, or creating an explainable that isn't robust and only applies to the specific model used [46]. Another issue that XAI faces is over-explanation where excessive transparency of the model could lead to cognitive overload, resulting in a critically flawed understanding of the model [60, 70]. By knowing these possible concerns from XAI, we consider what an XAI system needs to be deemed effective to the standards described in Hoffman (2018) [31]. These standards should serve to rectify the possible errors present in XAI design.

If we consider the first risk of incorrectly capturing the model being described, there are some user-end considerations to be made. People do not make rational decisions one-hundred percent of the time which must be taken into consideration. Regardless of the model being described, an effective XAI mitigates representativeness bias, availability bias, anchoring bias, and confirmation bias [82]. There are also designer-end considerations to be made; it is the duty of the XAI designer to faithfully translate the model that is interpretable for a wider audience. An XAI system could be completely accurate but inaccessible to a wider audience. It could also be widely accessible but misrepresentative of the ML model it explains. A user-centered design approach to XAI keeps the audience in mind while also highlighting the biases of the designer to mitigate misrepresentation [1].

In regards to the other risk of cognitively overloading the user, we must make sure the information we give sufficiently augments human reasoning about risks and errors in order to translate the ML system's behavior into a form that aids individuals [70]. On the other hand, we must make sure not to cognitively underload the user, as underspecification leads to complications about understanding ML [15]. A study by Shen (2020) looked into whether or not certain explanations empirically increase or decrease understanding of ML models through a user study. Empirical evidence is used in conjunction with learning sciences theory to create an effective XAI [82]. Empirical evidence we gather includes objective accuracy, perceived coverage of information, complexity, and human friendliness [24]. Theory from the learning sciences could include user-centered design [1], backward design [84], and communicative visualizations [2]. Ultimately, it remains of paramount importance to create an effective XAI system as without it, we have little to present users.

2.2 Tiny Machine Learning

Tiny Machine Learning (TinyML), broadly speaking, encompasses machine learning models that are deployed on tiny devices, like microprocessors or sensors [40]. While some posit exact requirements for a model to be considered TinyML, such as having an energy cost below 1mW [83], there is no widely agreed upon definition for what constitutes TinyML and what does not. Given that tiny devices are necessary for TinyML, we often see TinyML and edge computing in the same conversation as each TinyML localizes to its own hardware. Edge computing is a computational paradigm that performs computing near the edge of the network or the source of data [10]. One common error seen when using TinyML is the usage of TinyML as simply just smaller traditional ML. By neglecting key differences in TinyML and traditional ML and treating the two unequivocally

the same, we miss out on effectively using TinyML systems. We must be able to distinguish between TinyML and tiny ML.

2.2.1 Advantages and Disadvantages of TinyML

TinyML is similar to traditional ML with several key distinctions. These distinctions are both advantageous and disadvantageous as we must trade-off features compared with traditional ML. Currently, TinyML is gaining popularity as it promises improvements in energy and economic cost efficiency, privacy, responsiveness (low latency), and autonomy [5]. This energy and economic cost efficiency applies for both training the model and deploying the model; in addition, the devices TinyML systems are deployed on are typically low cost to create and to maintain [78]. This exists in stark contrast to the current trend of mining massive amounts of data to build larger ML models [39]. TinyML also promises greater privacy and data security as once the data for training the model is collected, no new data is transmitted. Because TinyML models are localized to the device it is built on, newly collected data cannot be transmitted to external devices and data servers, resulting in a safer, more private data collecting experience [78]. This also exists in contrast to many traditional ML models that collect data to not only customize their services towards an individual but also to train future ML models [39]. The high responsiveness from TinyML models is caused by locality to the hardware they are trained on. As mentioned earlier, no data needs to be transmitted to an external device or cloud, decreasing network latency. In addition, because the storage and processing power we work with is also tiny, we decrease our processing latency [63]. Finally, we have the benefit of autonomy. Because TinyML is deployed on edge devices, they must have the ability to function without human interference for some time. This becomes especially useful when our TinyML model is self-adaptive without having to be manually monitored [62].

There are some downsides to these benefits. Namely, the major tradeoff that occurs with TinyML is the performance-efficiency tradeoff. Because we are working with tiny devices, we do not have much computing power to work with [5]. This limitation becomes more apparent when compared with traditional ML systems, where the solution is often increasing computing power. TinyML models will have lower accuracy results and will typically underfit the dataset it was trained on. We have this quality and accuracy tradeoff as it is difficult to gain enough computational complexity to overfit our dataset on these devices [5]. While the low energy cost was an advantage when taking an environmental and economic viewpoint, we see that it becomes disadvantageous when viewing it from a computational perspective. This low energy and processing power forces us to make additional tradeoffs. Another we must make is the latency and energy consumption. While this tradeoff exists entirely in the hardware—not involving our TinyML models—it is still important to consider how the benefits we gain from TinyML are in a balanced state where we cannot overly draw from one [30]. Despite these unavoidable tradeoffs, we create highly effective TinyML models by carefully optimizing our models for situations where the downsides of these tradeoffs become negligible.

2.2.2 Motivations for TinyML

TinyML by design is well-suited for certain tasks given its ability to perform computations on small devices. These use cases include audio processing (audio wake words, context recognition, etc.), image processing (visual wake words, object recognition, etc.), behavioral metrics (activity detection, forecasting, etc.), and industry telemetry (sensors, anomaly detection, etc.) to list a few [5]. TinyML's small size and low costs are able to bring AI to locations previously not possible. One such possible use case is for wearable medical devices, such as pacemakers, that are difficult to remodify once they have been implemented [78]. Within these contexts, these devices should be able to function at a high level without modification for some extended time frame without worry that the device itself will fail.

TinyML is thought of as a response to the current trend of mining massive amounts of data and building large ML models [39]. One current issue that plagues large ML models is the environmental cost of utilizing excessive energy and resources to build [7]. The use of many processors and compute time, as well as the need to send data to external data center locations has a major impact on the carbon emission quantity [45]. TinyML's locality and low costs directly counter this issue, where we are still able to utilize ML without as heavy an environmental cost. In addition, localizing ML models has the added benefit of reducing network traffic by offloading tasks off the grid [63]. Additionally, many applications of TinyML are well-suited for improving the sustainability of the future and helping combat climate change [61]. However, something that must also be considered is the climate cost of creating tiny devices, like micro-controllers, for a climate-focused technological future [61]. TinyML reduces overall carbon emissions in the implementation of the ML models but must be balanced with the carbon emissions of creating devices.

In addition, as we enter an increasingly digitized world, more and more regular appliances are becoming "smart" devices. Using edge computing has been one method of coping with the challenge of massive-scale computing and storage [42]. By pairing TinyML with the edge devices that already exist on appliances, we further augment its effectiveness at its designated task.

2.2.3 TinyML Pipeline

The lifecycle of many ML models is defined by multistep ML pipelines which serve to clarify the workflow of working with ML. These phases include data analyzing, model training, model evaluation, and model deployment [88]. Unfortunately, many ML pipelines are blind to the context they were created in, creating conflict between desirable outcomes in expectation and actual behavior in deployment [15]. TinyML and traditional ML systems, while overlapping, have different use cases and expectations. If we used the standard ML pipeline of model specification, training data, and an independent and identically distributed evaluation [15], we would be neglecting a whole class of issues inherent to TinyML. Therefore, a new pipeline, a TinyML pipeline 2.2, must be implemented to cover these cases.

The first stage in the pipeline is drawn from traditional ML pipelines and involves collecting and cleaning data. TinyML models are not created on tiny devices and do not necessitate being tiny from the start. However, that does not mean that "tiny" shouldn't be on the designer's mind. Given

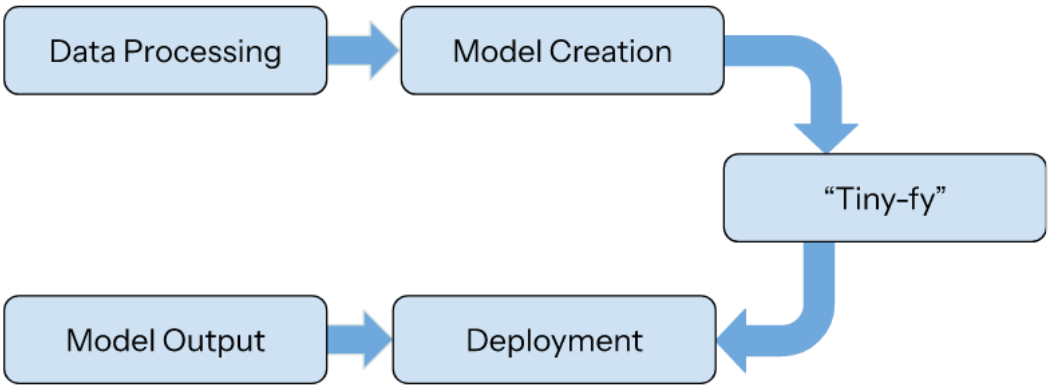


Figure 2.2: TinyML pipeline that serves to highlight key steps in working with TinyML

that TinyML models are deployed on limited processing power and memory, features of their data should reflect these limitations. TinyML models could end up having a limited number of features; our datasets should be robust so that the models are able to function with a limited number of features regardless of use case [63].

The second stage involves creating the ML model. Like the first stage, this stage draws heavily from the traditional ML pipeline. Most TinyML models are neural networks that are built and trained like traditional ML models. These models are then modified to become TinyML models in the third stage, where we "tiny-fy" the model. There are multiple ways to achieve this, with the most popular methods being quantization, pruning, and sparsity fusion [56, 63]. Quantization involves using fewer bits for computation (i.e going from 32 bits integers to 8 bits), pruning involves the removal of unncessary features, and fusion involves combining multiple features into one [63].

The fourth stage deploys the model onto tiny devices, such as edge devices. The model needs to be small enough to fit into the device as well as simple enough that any operation is feasibly completed on the device. Finally, the model's output must be interpreted by some user to determine whether the device is running effectively. The user should be made aware that accuracy is a subjective measure and that the threshold for passable accuracy depends on the context, such as detecting cancer cells in a patient or deciding recidivism for prison inmates, where precision and recall metrics become necessary [36]. Understanding the results of the model is important in deciding next steps for the TinyML model, such as whether to redeploy it or update the model.

2.2.4 TinyML Datasets

We discussed why users might want to use TinyML for certain contexts; TinyML datasets should reflect the tasks that TinyML is well suited for. We largely classify four categories of TinyML use cases: audio, image, physiological measures, and telemetry [8]. There are subcategories for this classification, including audio wake words, gesture recognition, and activity detection.

Currently, many datasets that are being used to train TinyML models are open-source data sets designed for traditional ML models; many datasets that are currently being used for TinyML

purposes are proprietary and not publicly available [5, 63]. Regardless of this deficit, a few open source datasets that are well-suited for TinyML use exist. ToyADMOS, or anomaly detection in machine operating sounds for toys, is well suited for testing TinyML systems for anomaly detection. CIFAR-10 and CIFAR-100, datasets of tiny images, are well suited for image detection for TinyML. Regardless, there is still a lack of a standard set of datasets used for testing TinyML models [63].

2.3 Combined XAI Pipelines

As mentioned earlier, ML pipelines are a useful tool for clarifying machine learning workflows. These pipelines ultimately results in a more optimized ML models for those utilizing the pipelines [8]. While improving development efficiency, many workflows is still lacking in overall transparency and interpretability. One way to resolve this issue is to adapt ML pipelines to target users, increasing interpretability by various means as necessary for the user group to gain a clear understanding of the ML process as a whole [80]. One possible mean of adapting the ML pipeline is by incorporating XAI methods into the pipeline [8]. By incorporating an XAI method into our pipeline, we are able to directly benefit from the interpretability it adds while also being given context in the ML workflow as a whole. This knowledge allows users to understand ML models as a whole, understand and appropriately trust the models they use, as well as understand optimizing the model further [75].

2.3.1 TinyML-XAI Pipeline

We present one possible pipeline that combines XAI and the TinyML pipeline presented in Figure 2.3, a framework that is based off the XAI-AutoML pipeline developed by Bifarin [8]. Here, we have three main portions of the pipeline: data preparing, machine learning, and explaining, with machine learning taking up a majority of the pipeline stages. By embedding XAI into the machine learning pipeline, we ensure that interpretability and transparency is not a supplemental feature but is an inherent attribute [52]. With this framework, we address explainability on both a micro- and macro-level. On a macro level, we explain each stage of the combined TinyML-XAI pipeline to address essential, high-level concepts. On a micro level, we have explainability embedded into how the user should understand using the system, improving output interpretability. We ultimately address the two-pronged approach for explainables we raised earlier with this framework.

2.4 Understanding & Evaluating Systems

There are several methods we use to analyze effectiveness, as well as different ways to quantify effectiveness of explainable or TinyML systems.

As seen in Table 2.1, there are four broad categories of ways to evaluate a framework or user interface: there is formal analysis, automatic computerized procedures, empirical experiments, and heuristically. Each of these four categories have their own distinct advantages. For instance, formal analysis techniques, such as cognitive task analysis, have a history of effective use and are generally well categorized, making implementation easier [12]. However, this method is costly in time and

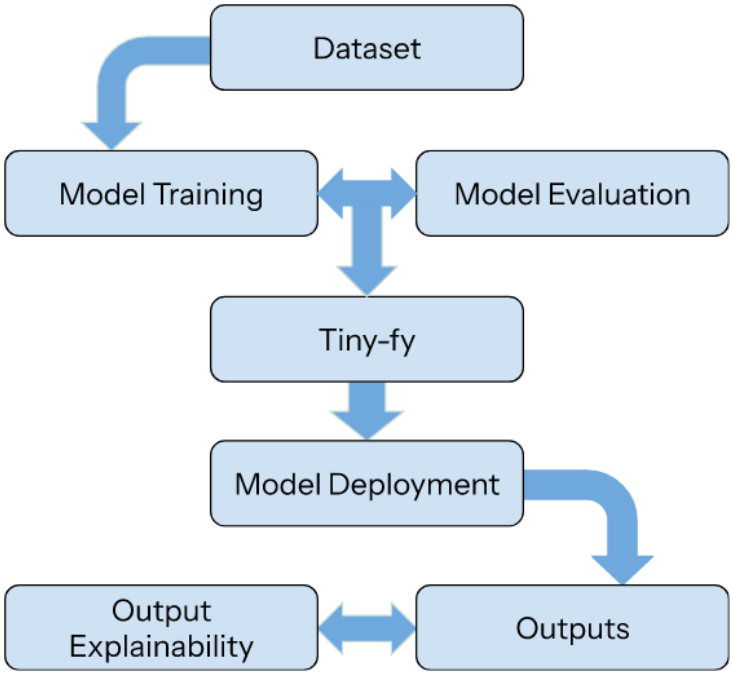


Figure 2.3: TinyML-XAI pipeline that improves upon the previous TinyML pipeline by incorporating a post-hoc model explanation for increased understandability

Method	Description
Formally	formal analysis techniques such as cognitive task analysis
Automatically	computerized procedure for automatic and objective evaluation
Empirically	experiments like user studies
Heuristically	looking at the interface and passing judgement based on set standards

Table 2.1: User Interface Evaluation Methods

human resources. The heuristic evaluation technique is cheap, intuitive, does not require advanced planning, and is used early on the in development process, something that is helpful in the user-centered design process. However, this method may identify problems without providing solutions and is limited if there is only one evaluator [54]. Automatically evaluating a user interface also saves time and human resources and is relatively cheap. This method however lacks qualitative and subjective measures and is also inefficient in that it is only evaluated once the system is deployed [35]. Empirically evaluations such as user testing obtains results that are most similar to real world user interactions. However, running user studies is costly in time and human resources [17]. Given the tradeoffs, we must choose evaluation methods that are most effective for our user interface given the working situation.

2.5 Summary

Explainable AI is an essential tool for modern artificial intelligence, where understanding and appropriately trusting how machine learning works are necessary. In particular, post-hoc explainables are helpful for users to understand both how a machine learning model works and to be able to interpret useful information from its outputs. We investigate how TinyML works and why it is an essential form of modern artificial intelligence. In order to effectively explain TinyML, we construct a pipeline that highlights important concepts of TinyML that are necessary for a useful understanding of TinyML. This pipeline is the combination of XAI and TinyML to develop a framework for users to gain the most understanding from a single explainable. We discuss how and what an effective evaluation of an explainable might look like.

Chapter 3

User Information

We are looking to use the user-centered design process [82] to design and built the explainable. The first necessary step of user-centered design is finding and analyzing the users themselves as illustrated in Figure 1.2.

We will be focusing on users of TinyML so that we this explainable is useful for those looking to learn more about TinyML. From our users, we are looking to collect several key demographic information. This information must be relevant to our explainable where we cater the explainable towards users through the information we find.

3.1 Identifying and Selecting Users

There are two main steps in getting usable user information. The first is obtaining the necessary information. This involves knowing how to find a good possible user base as well as having a good source of users. Once we have collected this information, we distill it into information that will be pertinent for creating the explainable.

3.1.1 How To Identify Users

Users come with a wide variety of backgrounds, personalities, and expectations. It is infeasible to try and design for all types of users, a task that would result in an explainable that satisfies no one. Instead, it is necessary to design with a subset of possible users in mind so that we satisfy some population. This process of designing for a subset includes:

1. define the characteristics of the user population and
2. work with a representative sample of the user group [55].

But what characteristics do we use from the user population? Some possible user characteristics include age, gender, physical abilities, education, cultural or ethnic background, training, motivation, goals, personality, user communities, different countries, and location (urban vs. rural), economic profile, disabilities, and attitudes toward using technology [44]. While it would be of massive benefit

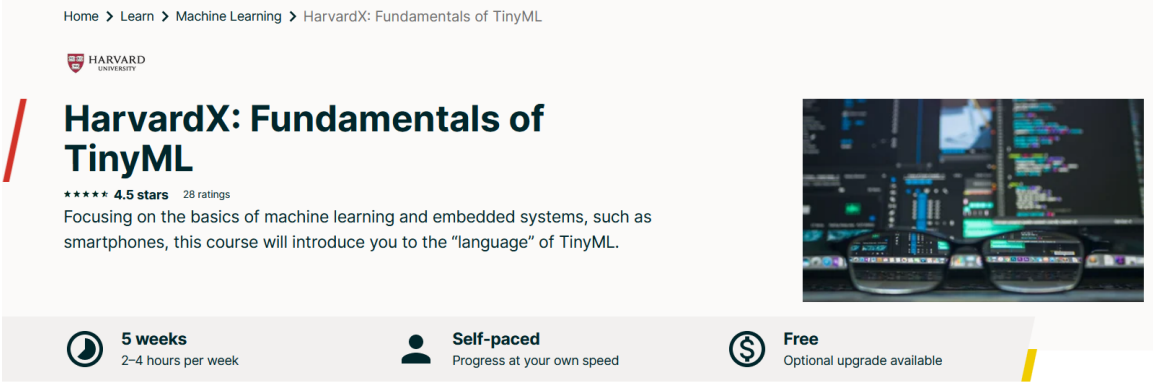


Figure 3.1: Harvard EdX Course for TinyML that we datascraped user information from

to obtain all of these characteristics, we might find that not all of them are relevant to the design of our explainable. In addition, this information should be readily accessible as surveying users for each characteristic would be a large time and human resource commitment. Therefore, we should use characteristics that overlap with this list, characteristics that are deemed relevant to the topic, and characteristics that are readily available from whatever source the user information is obtained from. We use these characteristics to help define our typical user for our explainable. However, something we must also consider is how many defining characteristics we use. Too little and we underfit our users. Too many and we overfit our users. Previous literature suggests that it's safer to overfit our users as we will end up pleasing some subset of users [13].

3.1.2 Selecting Users

We obtained a representative sampling of users as we had to first identify a source of possible users. More specifically, we needed a sampling of users representative of the average TinyML learner. We conducted a search online to see if there were any large groups of TinyML users that we could readily survey for demographic information. Some notable searches that came up was the TinyML Foundation[1] and Meetup groups for TinyML interest [2]. However, the most useful finding was the Harvard EdX course on TinyML, seen in Figure 3.1 [37] [3]. In addition to specifically catering to a user base that was looking to learn about TinyML, the EdX course had the added benefit of a convenient discussion board that we could pull demographic information from. The first page on this forum was an introductory board, where students were encouraged to introduce themselves to their peers. This came with three specific questions for students:

1. Where are you from?
2. What are you excited to learn about TinyML? and
3. How do you hope to use TinyML in your life and career?

[1] https://www.tinyml.org/
[2] https://www.meetup.com/pro/tinyml/
[3] https://learning.edx.org/course/course-v1:HarvardX+TinyML1+1T2023/home

Demographic Info	Number of Responses	Percent of Responses
Location	671	76.4%
Motivation	661	75.3%
Occupation	339	38.6%

Table 3.1: User Demographic Response Rates

From the posts on this board, we retrieved basic demographic information as well as motivational goals and information. This basic demographic information and motivational information sets up the basis for defining our typical user.

3.2 Finding Demographic Information

3.2.1 Web Scraping

To more effectively process all of the post data from the introductory discussion board, we created a web scraper in Python to collect all the posts into a readable comma-separate values (csv) file. The forum was scraped on January 22, 2024 at 5:30AM EST, collecting 878 unique posts. However, this came in the form of messy data that was unstructured and lacking in information as some users did not answer all the questions.

From the collected data, we looked to capture three variables that were most commonly seen in every post: location, motivation, and occupation as seen in Figure 3.1. To encode most of this data, we manually sifted through each post description.

The one exception to this was encoding location. We used the Python packages GeoPy[4] and PyCountry[5] to find and extract countries from each post. However, we manually checked all posts that did not initially have a country to check for typos and re-updated the list accordingly. From countries, we extrapolated to continent as this gave us an easier task of handling locations as we no longer had to handle outlier countries. Motivation was unique where we had to encode the post into a value. The possible values that we created for motivation were: community, future aspects, hobby, knowledge, and projects. To encode occupation, we manually inputted the occupation or implied occupation—such as referencing grade level for schooling—for each post.

3.2.2 Results of Web Scraping

The first demographic variable we investigated was the country of origin for the TinyML users. We found that 229 of these users were from North America, 206 from Asia, 88 from Europe, 75 from Africa, 48 from South America, and 7 from Oceania as seen in Figure 3.2. If we look closer at our data at the country-level, we find that the top five countries are: the United States with 179 users, India with 104, Mexico with 25, Spain with 23, and Nigeria with 20 users. While we see that the

[4]https://pypi.org/project/geopy/
[5]https://pypi.org/project/pycountry/

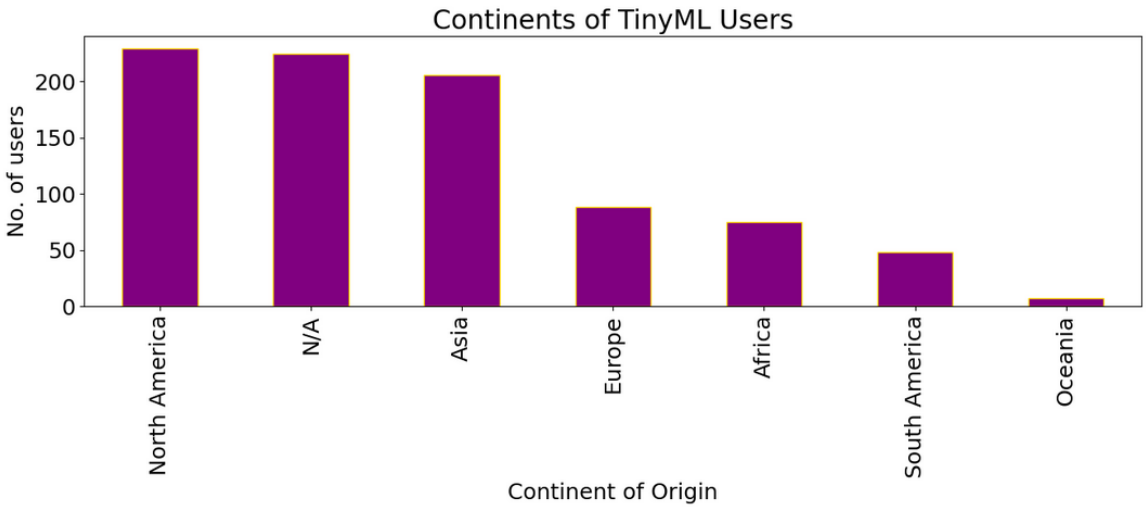


Figure 3.2: TinyML Users by Continent

majority of our users are from the United States, it is important to note that this may be due to the EdX course being taught in English, as well as the discussion board being in English as well.

The second demographic variable we investigated was the motivations for using and learning about TinyML. From coding the discussion posts into the five possible categories, we found that 390 users were looking to gain more general knowledge about TinyML—and machine learning in general, 131 were curious to see how TinyML would impact the future, 116 were looking to use TinyML is personal projects, 12 were looking for a machine learning community, and 11 users were looking to learn TinyML as a hobby as seen in Figure 3.3. This last category of hobby could be misconstrued to overlap with personal projects but hobby refers to mostly users who indicated learning as a hobby, not necessarily TinyML as the hobby.

Finally, the third demographic variable we looked at was the occupation of these users. Compared with location and motivation, there were much fewer responses for this variable. However, there were still enough responses to find some patterns. Due to the low frequency of some occupations, we grouped any occupation with three or less users as "other". We found that 138 users were students, 127 were engineers of varying background (software, mechanical, electrical, etc.), 16 were teachers, 8 were researchers, 5 worked in the medical field, 4 were consultants, 4 were artists, and 36 various other occupations. These various other occupations included a firefighter, a construction worker, and a banker just to list a few.

3.2.3 Target Demographic

From the three demographic variables, we found that the most common responses were users from the United States, users who are looking to gain general knowledge about TinyML, and users who are students. Thus, we use United States students as the user archetype for TinyML. Keeping our user base in mind, we create an explainable that would line up with the goals of students from the United States.

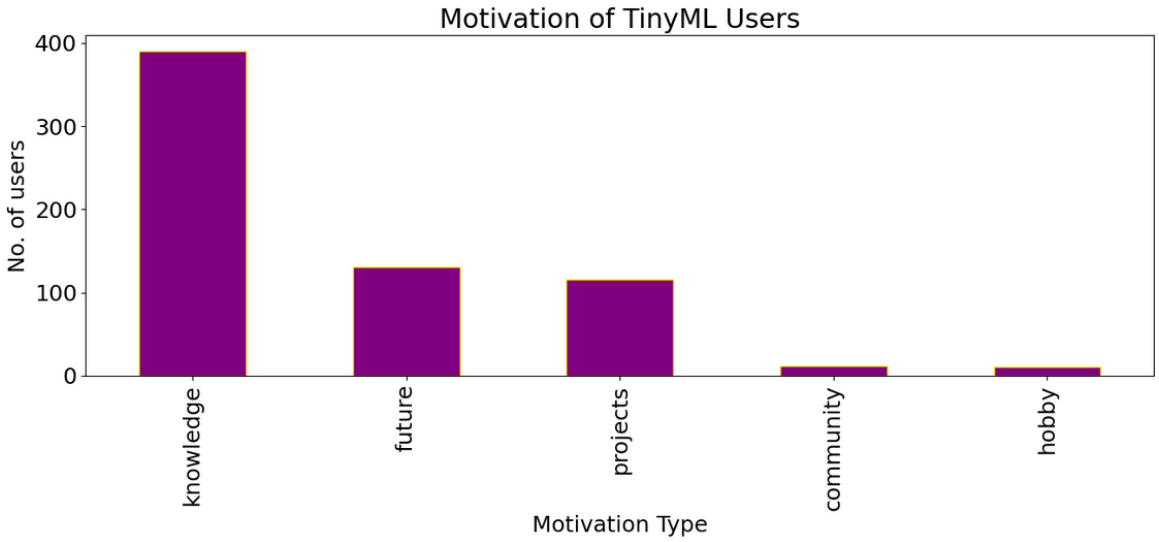


Figure 3.3: TinyML Users by Motivation

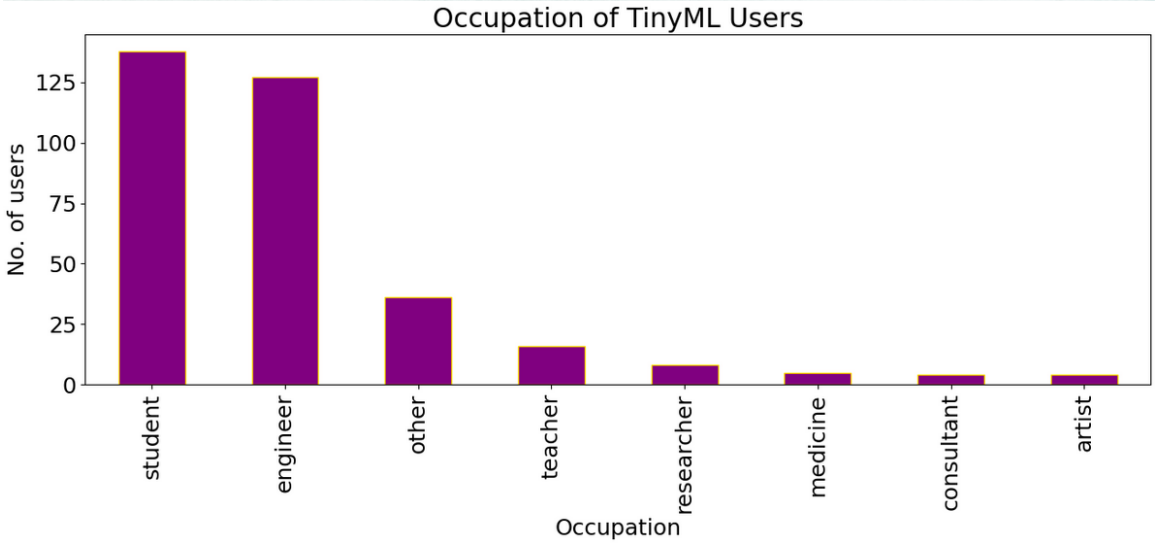


Figure 3.4: TinyML Users by Occupation

3.3 Summary

The first step of the user-centered design process was to find and analyze possible users for the explainable. We found users on an online Harvard EdX course for TinyML and data scraped a discussion post to gain demographic information about who was looking to learn more about TinyML. We collected three demographic variables: user location, motivation, and occupation. From these variables, we found the most common responses to be the United States, general knowledge, and student respectively. So, we use these characteristics to guide how we create our explainable.

Chapter 4

Methodology

We studied who our desired users were in Chapter 3. Continuing with the user-centered design process, seen in Figure 1.2, we must design our explainable, conduct user-testing to obtain feedback on the design, create our final design, and assess the effectiveness of our explainable. Designing the explainable and obtaining feedback from user-testing was an iterative process before the final design and assessment.

Our final design is an interactive website that users use to learn more about tiny machine learning. After we create our final explainable, we assess the effectiveness of the explainable, both in its usability and its information content. We gather both quantitative and qualitative information from our users to analyze so that we obtain an in-depth review of what worked and what did not. Ultimately, this reflects what the designer thinks is important regarding TinyML and how well this knowledge is conveyed for users.

4.1 Explainable Design

Within the explainable design process, we follow the iterative methodology that we outlined in Figure 1.2. As mentioned earlier in Chapter 1, this iterative process involves designing an explainable and testing it with users with varying levels of fidelity for each explainable. To be more precise, however, we will be following Stanford D-School's design principles. These principles outline five modes as the components of design thinking necessary for effective user design: empathize, define, ideate, prototype, and test [19]. By following this design process, we create a more effective and user friendly explainable.

4.1.1 Empathize & Define

Empathizing is key to the user-centered design process as the problem we are trying to solve is not for us but for the users we are designing for [19]. We already identified our users in Chapter 3 and our typical TinyML user as students in North America. To be more precise, we were aiming to work with undergraduate students with minimal exposure to AI and machine learning in the classroom.

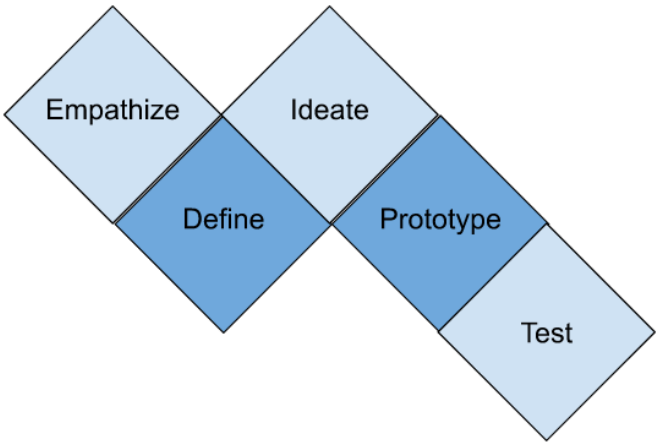


Figure 4.1: Stanford D-School Bootleg Design Process that we use to guide our design thinking

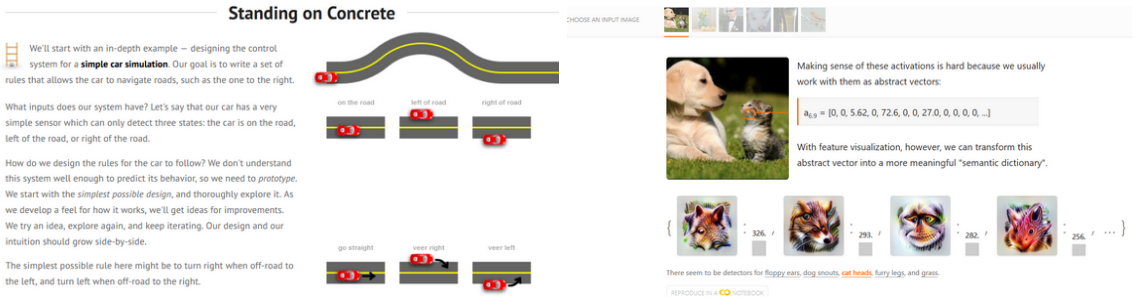
With these users in mind, we had some key considerations when designing the explainable:

1. What are concepts we should implement in our explainable? What information is vital?
2. What are concepts that undergraduate students would be familiar with? Should we start with a baseline knowledge of computer science topics?
3. Is our goal purely educational? Or should users take away enjoyment from the explainable?
4. How much knowledge should we impart to users? How can we prevent cognitively overloading users?

We conducted a literature review, seen in Chapter 2, as a response to some of these considerations. These were used to define and develop concepts used in our explainable. With this knowledge, we define what an effective TinyML explainable looks like, reflecting what we and the literature suggest is important for TinyML. This knowledge was gathered, synthesized, and constructed into forms that were more readily available for a wider audience, such as the TinyML-XAI pipeline defined in Figure 2.3. This design vision includes not only how to use TinyML effectively but also looks to address trust and expectancy of TinyML systems as well.

4.1.2 Ideate

This design vision could be implemented in various ways, or to ideate different designs. One aspect of brainstorming different design implementations was what medium the message would be conveyed. While the information that was synthesized during the literature review process was constructed into readily available forms, we had to ensure that users would fully understand the topic. One method of doing so is the use of metaphorical narratives in explanations. Metaphorical narratives have been shown to sufficiently simplify and abstractify complex topics into simpler models [38]. This has the added benefit of fitting users into a singular perspective; regardless of previous topic



(a) This explainable uses a driving narrative (b) And this explainable uses a cat and dog narrative

Figure 4.2: Explainables that use various metaphorical narratives

knowledge, all users must contend with the topic from the metaphorical narrative's viewpoint. The mechanism behind metaphorical effectiveness is how they "nudge" users to conclusions about the topic. All metaphors have some implicit biases to them that swing people's opinions about a topic [22]. Additionally, most explainables online use a metaphorical narrative to guide the topic at hand. The explainable in Figure 4.2 (a) uses a driving narrative that abstractifies an algorithm with a simple car simulation. The explainable in Figure 4.2 (b) uses cats and dogs to guide an explanation on computer vision. Because this is a common practice for individuals creating explainables, it made the most sense to implement a narrative with our explainable.

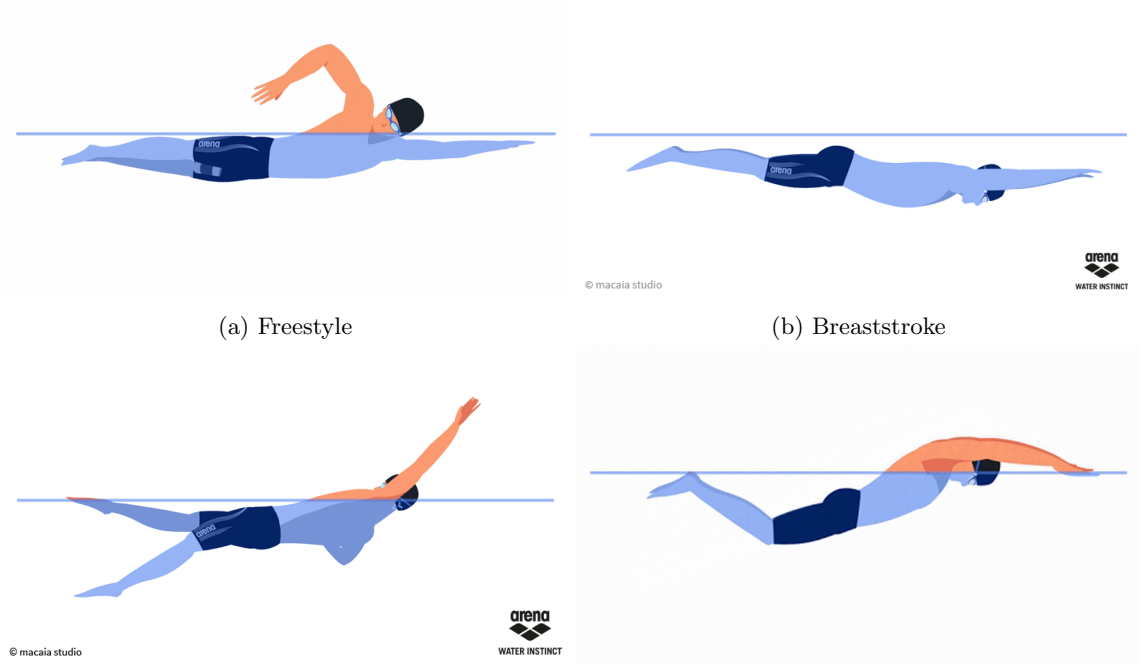
The focus of our TinyML explainable was the TinyML pipeline. Therefore, we decided to use a narrative that fit with the general unidirectionality of the pipeline. After a brainstorming process that included considering flowcharts, sewer systems (going down the "pipeline"), driving..., and navigating an electrical grid, the metaphor that was chosen was swimming through a river. This metaphor was chosen as it was deemed memorable but also simplistic enough to not misconstrue any information.

In addition to having a metaphorical narrative to shape the flow of the explainable, we also needed some mechanism to more concretely ground the information we were looking to present. This is done in the form of example-based explanations, which explains algorithmic results using surface examples of some dataset [9]. This, in conjunction with metaphorical narratives, only serve to further better explain complex concepts to a novice user base. We looked to use a dataset to fit the following criteria:

1. Be related to swimming conceptually,
2. Be a task realistically solvable by TinyML, and
3. Ideally something that has previous literature.

4.1.3 Swimming Dataset

With these in mind, we found that using sensor and accelerometer data to identify swimming strokes was an ideal candidate as an example to use. As discussed in Chapter 2, there are certain use cases



(a) Freestyle (b) Breaststroke

(c) Backstroke (d) Butterfly

Figure 4.3: Swimming Stroke Examples Used in Explainable

that are ideal for TinyML, one of which is activity detection. So, classifying swimming strokes is one such task that is well suited for TinyML.

However, sensory data from previous literature regarding the classification of swimming strokes was not publicly available for use [14, 57]. Instead, we created an example dataset using a free accelerometer phone app[1], a waterproof phone case, and wristbands to gather rudimentary swimming stroke data. Similarly to the Costa paper, we looked to classify a small subset of swimming strokes, limiting ourselves to four [14]: freestyle, breaststroke, backstroke, and butterfly as seen in Figure 4.3 [2]. These are also the four strokes used in the Olympics, which works in our favor as more users will be familiar with them. Accordingly, we collected accelerometer data for the four strokes so that users had a concrete example to ground more complex topics in TinyML. Ultimately, by using a combination of metaphorical narratives and example-based explanations, we create an explainable that is much more engaging for users to use and understand.

4.1.4 Prototype & Test

The next stage of the design process entailed creating prototypes of the explainable so that we could evaluate them in a user study. As previously mentioned, creating prototypes is a labor and time intensive process. So, we created several prototypes of varying fidelities: a low fidelity paper

[1] https://play.google.com/store/apps/details?id=com.chrystianvieyra.physicstoolboxsuite
[2] Gifs from and permission to use gifs granted by **arenasport**

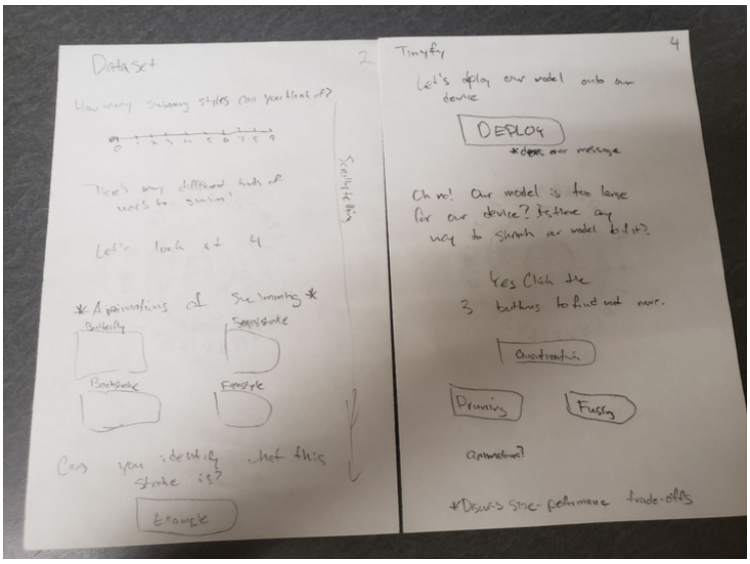


Figure 4.4: Paper prototype for the explainable that was created as an outline that is critiqued

prototype, a medium fidelity slides prototype, and a high fidelity web application prototype that is akin to the final product.

The low fidelity paper prototype was designed to quickly decide what information was necessary to include in the explainable [72]. In addition, this allowed us to get a basic idea for how the overall explainable would be structured (i.e. figuring out the sequence of events) and what it would look like seen in Figure 4.4. The paper prototype is necessary in that it is not a labor intensive process and gives a space to create an outline that is critiqued to create higher fidelity prototypes.

There are several different methods of critiquing prototypes, each with their own benefits. The method of critique used on the low fidelity paper prototypes was comparing the design against Neilsen's 10 Usability Heuristics, outlined in Table 4.1. Nielsen's usability heuristics were a useful tool to use early on in the process of designing as it allowed us, the designer, to quickly identify several issues that may be negatively impacting the usability of the explainable. After identifying which heuristic was violated, we then determine the severity of the violation and how to rectify it.

One such error was a violation in user control and freedom. In the prototype, we lacked a way for users to easily navigate between different pages, being forced to use a side task bar, a design that was unintuitive. This violation was quite severe as it negatively impacts the user experience of the explainable. In the best case, it becomes a stumbling block for the user; in the worst case, it prevents the user from continuing through with the explainable altogether. Another error was violating consistency and standards. We implemented several different graphs throughout the prototype, which were each designed separately, creating visual discrepancies in the walk-through. This violation was less severe as it does not detract from the user being able to complete the explainable but could still negatively impact their experience.

With the low fidelity paper prototype and its proposed improvements, we moved on to creating the medium fidelity slides prototype. This version of the application was created using Google Slides

#	Heuristic	Description
1	Visibility of System Status	Keep user informed about what goes on
2	Real World Conventions	Speak the user's language
3	User Control and Freedom	Undo and redo should be supported
4	Consistency and Standards	Consistency: express same thing same way
5	Error Prevention	Prevent errors from occurring in the first place
6	Recognition Rather than Recall	See-and-point instead of remember-and-type
7	Flexibility and Efficiency of Use	Accelerators should be provided
8	Aesthetic and Minimalist Design	Provide only necessary information
9	Help Error Recognition/Recovery	Help users recognize, diagnose & recover from errors
10	Help and Documentation	Use proactive & in-place hints to guide users

Table 4.1: Nielsen's 10 Usability Heuristics

[3] that would serve to more closely resemble the final web application in both what information was being presented, as well as design implementation.

The slides prototype was then critiqued with a light pilot study. We chose two Williams College undergraduate students selected as the pilot users; these participants were chosen as individuals with some exposure to computer science topics but with no formal training with higher level topics, specifically TinyML and machine learning in general. We conducted a think-aloud review process, a common practice in usability testing, to collect general sentiment about the usability of the prototype. The think-aloud review process makes processes more explicit by having participants think-aloud while they complete a task, giving salience to the problem-solving process [85].

We then created the final web application using HTML, CSS, and JavaScript, with screenshots of the application in Figure 4.5 and Figure 4.6. We split the web application into five separate pages, one for each stage of the TinyML-XAI Pipeline: Datasets, Model Training, Tiny-Fy, Model Deployment, and Output & Explainability.

4.2 Final User Studies

Once we finished with the creation of our final web application and conducted light testing with users, we had to conduct one final design assessment in line with our design steps for creating and evaluating the explainable in Figure 1.2. This is a more formal process than the pilot studies conducted with the medium fidelity slides prototype and the high fidelity web application prototype. This would give us information about the usability and the usefulness of the explainable.

4.2.1 Participants

The participants of this study were undergraduates from Williams College recruited through email. All participants were students who had taken the introductory computer science course (CSCI 134)

[3] slides.google.com

in the Fall of 2023. This was to ensure that participants were familiar with computer science topics while also making sure that they had little to no experience with higher level topics like machine learning. Each participant signed up for the study voluntarily for monetary compensation in the form of a $10 online Amazon gift card. We recruited five participants for the full study and the studies were conducted in-person in a private lab setting.

The age range of the participants in the study was 18-21, with the median age being 19. Three participants were female-identifying and two participants were male-identifying. All of the participants were planning on majoring in either computer science or math, with three participants majoring in computer science and two in mathematics (with one participant double majoring in both and two other participants double majoring with sociology).

4.2.2 Materials

This study was approved by the College's Institutional Review Board. The materials for this study includes the consent form, the interactive web application explainable, the pre-survey, and the post-survey. Both the pre-survey and the post-survey were created and filled out using Google Forms [4]. The consent form was filled out on paper and both the pre- and post-tests, as well as the interactive web application explainable, were conducted in-person.

The consent form and pre-test questions were used to gain consent to the experiment and demographic information about the participants, as well as get some bearing on how familiar the users are with machine learning and their opinions on ML. We additionally obtained some basic information about the participants' backgrounds with computer science and mathematical concepts, as well as to obtain a more objective measure of this information with a brief quiz. We based the questions regarding participant opinions on machine learning from previous papers [59, 69]. Like the aforementioned papers, the questions concerning participant background with computer science and views on machine learning were mostly based on self-report scales. This included both questions about TinyML and machine learning in general. These questions were scored on a 5-point Likert scale for all questions that were self-reports. We also gave participants a brief quiz on mathematical concepts. The mathematical concepts we chose to quiz were deemed to be related to machine learning, which were linear algebra and probability topics [18, 51, 64]. These questions were multiple-choice as seen in Appendix A.

The creation and details of the interactive web application explainable are explained in-depth in the first half of Chapter 4. The participants were told to follow the web application as it was designed and to ask any questions for help if necessary.

The post-test questions used many of the same questions as the pre-test questions, specifically questions concerning user familiarity with machine learning as well as views on machine learning. This was designed so that in analysis, we could determine if there were any interesting shifts in views after using the explainable. Like the pre-test, these questions continued to be scored on the 5-point Likert scale.

[4] forms.google.com

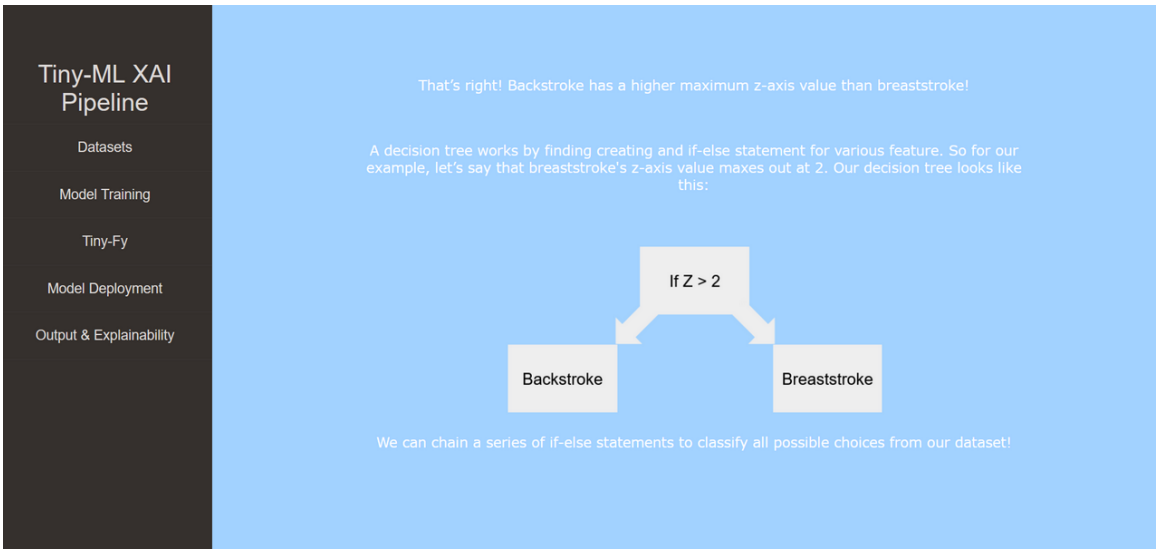


Figure 4.5: Screenshot 1 of Completed Web Application

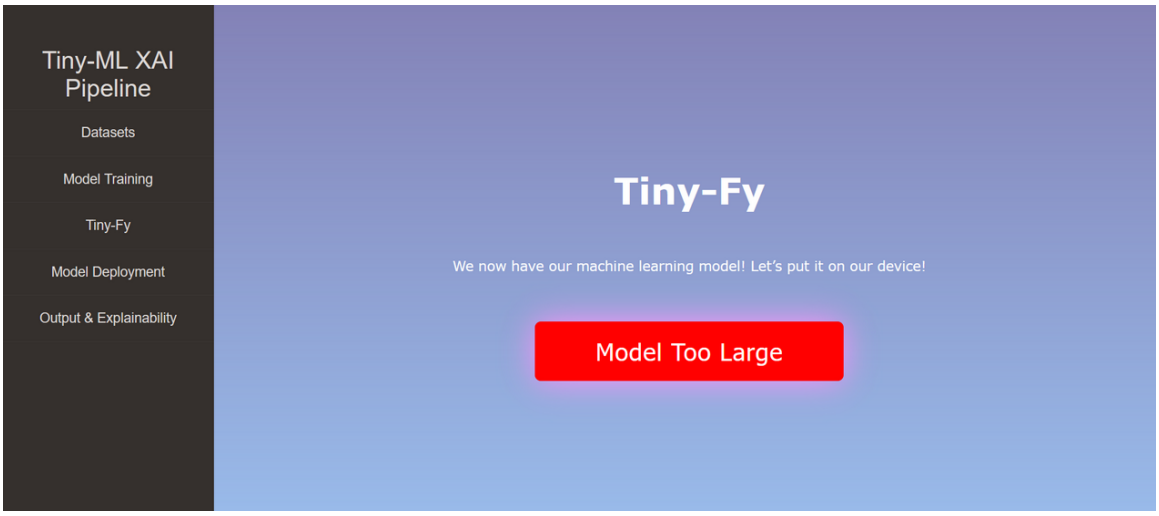


Figure 4.6: Screenshot 2 of Completed Web Application

Figure 4.7: Procedure that users undertook during user studies

In addition to these questions, we added some questions regarding the usability of the explainable itself. We used an abridged version of the questions from the System Usability Scale (SUS) to determine overall usability of the system [33]. These usability questions were also scored on the 5-point Likert scale. We also had several questions pertaining to knowledge about TinyML that was provided within the explainable. These questions were a mixture of multiple choice questions as well as some short answer questions. Ideally, if the explainable worked, the participant answers for these questions would be accurate.

4.2.3 Design

This study was designed to evaluate whether our initial TinyML explainable could be effective in providing necessary user understanding of TinyML. In addition, there are some exploratory analyses on the usability of the explainable specific to the post-test as we cannot test users on this before they interact with the explainable.

4.2.4 Procedure

Participants completed the study individually during their allotted 30 minute time slot in-person with the presence of a researcher conducting the study. All subjects completed a consent form, followed by the pre-test, the web application explainable, and the post-test. Once the study was completed, participants were provided additional information regarding compensation.

4.3 Summary

To complete our study, we must first create an explainable for TinyML. We followed an iterative process to create prototypes of varying fidelity levels and testing these prototypes so that they could be improved for future variations. Within this iterative process, we also followed the Stanford D-School's design principles to help ensure that we created a usable and effective explainable. This iterative process started with a low fidelity slides prototype that we ran Nielsen's Usability Heuristics against. Then, we moved onto a medium fidelity slides prototype that we ran pilot studies with users. Finally, we created the high fidelity web application prototype that we also evaluated with users to create the final web application explainable. With this final version, we ran a small user study to determine the usability and effectiveness of the explainable.

Chapter 5

Results & Discussion

This research began with an investigation into who the users of TinyML were. With this investigation, we were able to determine an expected user group that we could cater our explainable towards. With this knowledge, we began the process of creating an explainable for the use of TinyML. We analyze the responses from the pre- and post- tests to determine the overall effectiveness of our explainable. We conduct an exploratory analyses for questions on the pre- and post- tests.

5.1 Results

We analyzed descriptive data for many of the questions in the pre-test. This was to provide context for the users who participated in using the explainable. We then explored the quantitative data for interesting initial trends. Finally, we analyze for descriptive data for the questions in the post-test for some exploratory questions. All questions that were self-reported were rated on the 5-point Likert scale.

5.1.1 Pre-Test Analyses

After collecting demographic information from the users, we first asked about their mathematical background. Participants were fairly confident in their math skills, with a mean value of 3.8 (SD = .45) and a median value of 4.0. The brief math quiz reflected this high confidence fairly with an average score of 80% (SD = 20.9) and a median value of 75%. We also calculated Cronbach's alpha for the three questions regarding personal confidence in technology and computer science ($\alpha = 0.952$). More specifically, we had a mean confidence in technology skills of 4.0 ($SD = 1.22$) and a mean confidence in computer science of 3.2 ($SD = 0.84$). The high Cronbach's alpha value is a positive indication that the participants were paying attention during the tests [21]; this is because we have high correlation in expected columns.

Participants were generally hopeful about the usage of AI, with a mean value of 4.6 ($SD = 0.55$) when thinking about the beneficial applications of AI in society. Participants, however, had mixed opinions about whether the prevalence of AI in daily life was a positive or a negative with a mean

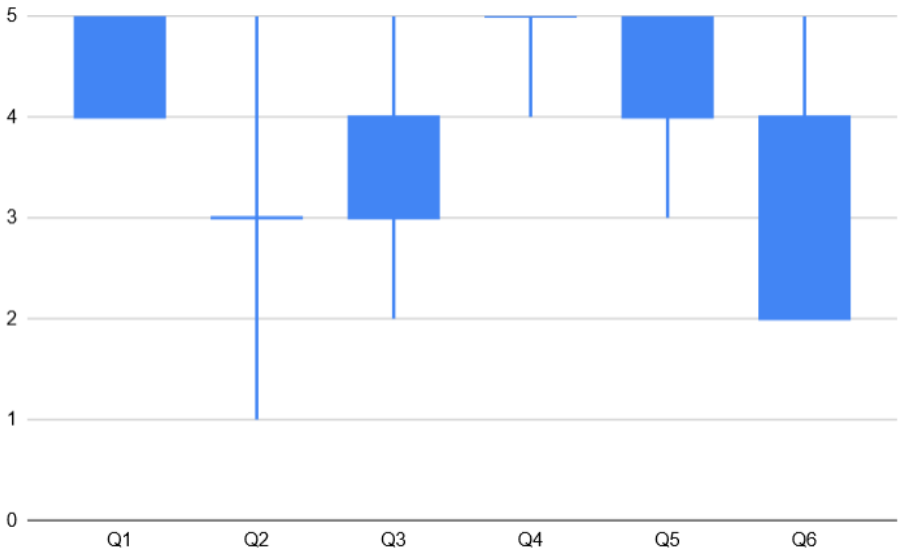


Figure 5.1: Box-and-whiskers plot of the six questions on AI perspectives

value of 2.7 ($SD = 1.25$). In addition, they also held mixed opinions about whether we should be more cautious or excited about AI with a mean value of 3.2 ($SD = 1.30$). Participants did hold strong opinions about the importance of AI education, with a mean value of 4.6 ($SD = 0.70$).

In addition, none of the participants had taken a formal class on machine learning. Accordingly, the mean value of confidence in knowledge about AI was 2.2 ($SD = 1.3$). Specifically for TinyML, we had a mean confidence in TinyML knowledge of 1.4 ($SD = 0.55$). However, there was no strong correlation between confidence in AI and confidence in TinyML, as we only observed a low Cronbach's alpha values ($\alpha = 0.57$) between the two features. This distinction is important because it may reveal that that users are not conflating TinyML and AI.

5.1.2 Post-Test Exploratory Analyses

After successfully stepping through the web application explainable, users then proceeded to answer questions about the explainable, usability, and opinions about AI and TinyML. We first investigate the results of the questions pertaining to the explainable. These first six post-test questions were designed to test and obtain a more objective measure on whether users actually gained information about TinyML. The participants scored an average of 92% on this portion of the test, indicating that they could recall most of the information learned in the explainable. If we look at the self-report for TinyML confidence, however, we only see an average value of 2.6 ($SD = 1.34$). Previous literature suggests that individuals is unreliable with self-reports and are bad at self-assessing their knowledge [87, 71].

We also analyze how usable the final iteration of the explainable ended up being by analyzing the results of the System Usability Scale questions. We had nine questions on the form that pertained to this group. We see a chart illustrating the average responses to each of these questions in Figure

#	Question
1	I thought the explainable was easy to follow
2	I found the various sections in this explainable were well integrated
3	I found the explainable to be well-designed
4	I found this explainable engaging and interesting
5	I enjoyed completing this explainable
6	I thought this explainable provided an effective explanation of TinyML
7	I could generally explain the TinyML pipeline to another person
8	I would feel comfortable using TinyML after completing this explainable
9	I am interested in learning more about TinyML

Table 5.1: Nine questions asked based on the System Usability Scale

5.2. When we run a correlation test across these questions (excluding Question 9), we see that we have weak correlation ($\alpha = 0.70$). We excluded Question 9 is because this question does not directly pertain to the usability of the explainable but instead focuses on future user interest in TinyML; we will investigate this question separately. We average a response value of 3.53 ($SD = 0.96$) for the eight questions about usability and effectiveness in conveying knowledge. Question 9 about future user interest in learning about TinyML received an average response of 3.8 ($SD = 0.834$).

Identically to the pre-test, we asked participants about their views on AI and their confidence about AI and TinyML. Participants were generally hopeful about the use of AI, with a mean value of 4 ($SD = 1.00$) when considering the beneficial applications of AI in society. There were mixed opinions about the prevalence of AI in daily life as a positive or negative with a mean value of 3.5 ($SD = 1.23$). There were also the mixed opinions about whether we should be more cautious or excited about AI with a mean value of 3.2 ($SD = 1.30$). Strong opinions were still held about the importance of AI education, with all of the responses being 5 ($SD = 0.00$).

The mean confidence in knowledge about AI was 3.0 ($SD = 0.71$) while the mean confidence in knowledge about TinyML was 2.6 ($SD = 1.34$). Again, there was no strong correlations between confidence in knowledge about AI and knowledge about TinyML when running a correlation test between the two features ($\alpha = 0.61$). Again, this is important as even after learning about TinyML, participants continued to not conflate AI and TinyML, instead treating TinyML as a distinct concept.

5.1.3 Pre- and Post- Test Changes

We now compare the results of the pre- and post- tests to see if using the explainable had any remarkable effects on the participants. We have two main points of comparison we make. The first is detecting whether there was a change in views towards AI. We asked six questions regarding this:

- Q1. There are many beneficial applications of artificial intelligence
- Q2. I support the increasing prevalence of AI in society.
- Q3. AI is used too much in daily life

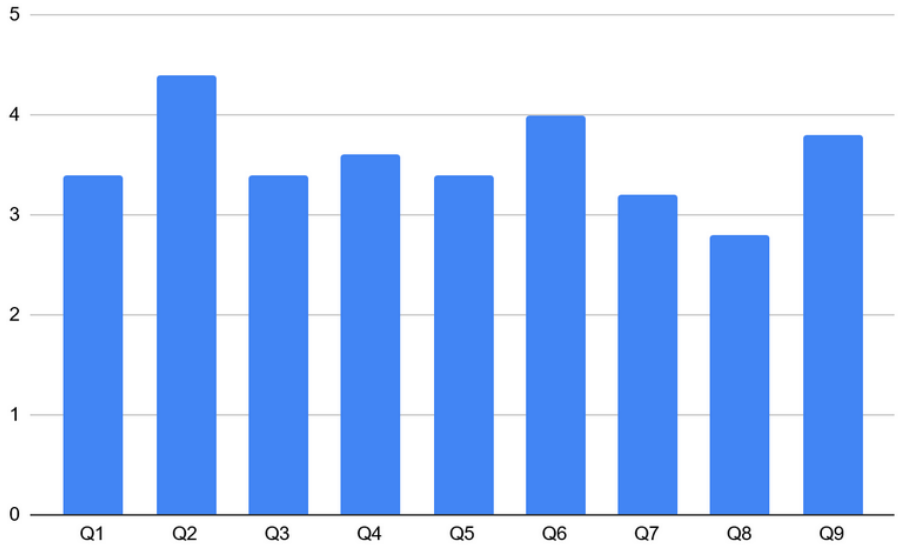


Figure 5.2: Responses to SUS Questions in Post-Test

- Q4. It is important to educate the public on how AI systems work.
- Q5. It is important to focus on the potential harms and limitations of AI systems.
- Q6. We should be more cautious than excited about AI

Ultimately, we do not observe drastic swings in views regarding AI from participants before and after they used the web application explainable, as seen in Figure 5.3. For three of the questions, we saw no changes in average responses. For the first question, we saw a slight decrease in responses, while we saw slight increases in response to the second and fourth question.

The other set of questions we investigate is how confidence in one's knowledge about AI and TinyML were affected by the explainable. From the graph in Figure 5.4, we see some increases in confidence in knowledge for both AI and TinyML from participants after they used the web application explainable. We do not see as drastic of a difference for AI confidence, with the post-test increasing 0.8 points on average, or a 36.4% increase. For confidence in knowledge about TinyML, we saw an increase in the post-test of 1.2 points on average, or an 85.7% increase.

5.2 Discussion

From analyzing the results of some questions in our pre-test, our post-test, and analyzing the changes in answers between the pre- and post- test, we observe certain trends about the explainable created. These conclusions tell us more about the usability and effectiveness and what they mean in the greater context of our design considerations and the overall goals of this project. We consider the implications of our pre-test question results, our post-test question results, and the changes we see between the pre- and post- tests.

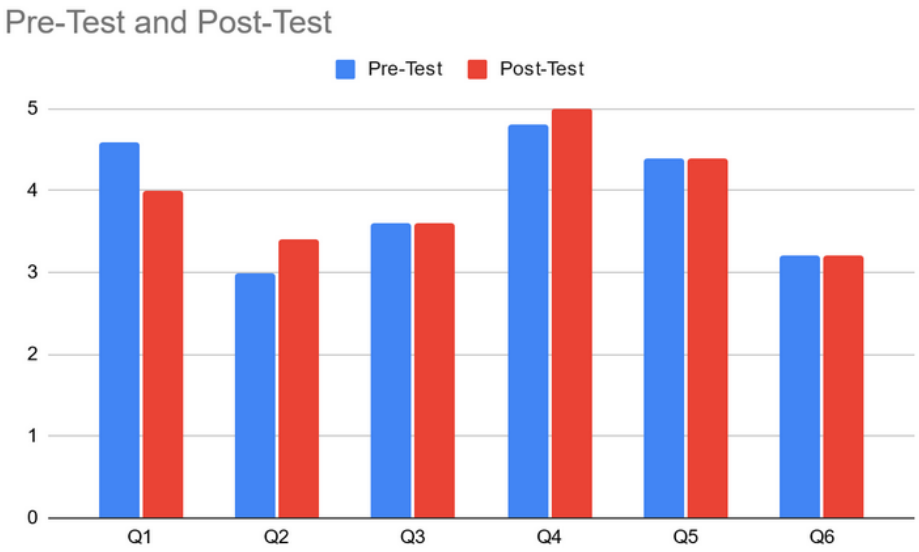


Figure 5.3: Changes in Response to AI View Questions

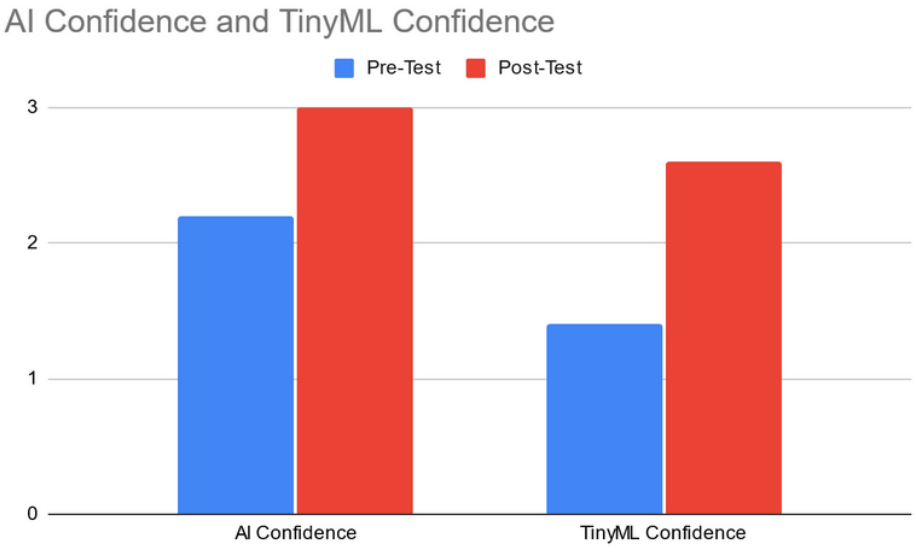


Figure 5.4: Changes in Response to AI and TinyML Confidence

Looking at our pre-test results, one finding was that participants were fairly confident in their mathematics and technology backgrounds, while being mixed on their confidence in their computer science knowledge, and even less so in their knowledge about AI and TinyML. Their belief in their mathematics was supported by their performance on the brief math quiz. With this, we address the consideration of what concepts our user base should be familiar with. Participants performed well in linear algebra and general statistics, so these are topics we build off of in our explainable and were appropriate to implement. It's important to consider the context of our users and how this base knowledge that we assume works for this specific user group of students but may not work for other user groups, such as engineers. Engineers, for example, most likely have an advanced degree in engineering or a related topic. If we used machine learning engineers as our target group for this explainable, we could assume that our users would have a higher level understanding of mathematics and machine learning. For our user group of students, however, it makes more sense and is expected that they have a more mixed confidence in computer science and AI topics, corroborated with the finding that none of the participants had taken a formal class on machine learning.

Additionally, this also tells us what information we should impart to users. For the students with little AI experience, it is more important to impart high level concepts and ideas without going too specific in the details as this risks cognitively overloading our users. For instance, the section of neural networks does not provide the exact details of how neural networks work; there is no need to explain how backpropagation works to AI and ML novices. It is more important for them to understand that a neural network is low in interpretability but is powerful nonetheless. Other audiences might appreciate an in-depth guide into specific machine learning models or tiny-fying techniques but this is again context dependent. Given our novice student user base, it would be far more effective to provide a breadth of information than having a detail-specific guide to TinyML.

The participants also had both complementary and conflicting views towards AI. The complementary views included questions about the beneficial applications of AI in society and the importance of AI education. This latter question has more interesting implications on the importance of explainables for TinyML. Almost unanimously, participants agreed that educating the public on how AI works is of high importance. Their belief in AI education could allow them to more effectively use the explainable or more likely to recommend the use of explainables in the future. This concept is further corroborated by findings in the post-test survey, which we will discuss later. The conflicting views towards AI included questions about the prevalence of AI in daily life being a positive or negative and whether we should be more cautious or excited for AI. It may be positive for us that these questions received a mixed reception as it indicates we had users with a diverse view of opinions on AI. Because we did not select for specific views on AI when considering what the TinyML user was, having a diverse range of opinions on AI allows us to more accurately reflect on a bigger population group rather than a group who might consider AI only negative or only positive.

We also extrapolate some possible implications from the results of our post-test exploratory analyses. Initially with the post-test results, we found that the participants of the study performed exceptionally well with the questions on the explainable material. This could indicate one of several possibilities. Ideally, it would indicate that the participants learned the material from the explainable successfully and were able to recall the information in the post-test survey. It could also, however,

just be that the time in between using the explainable and taking the survey was short enough where participants could simply recall the information back. One issue we run into, however, is the low self-report score for confidence in knowledge about TinyML. We could interpret this discrepancy in a variety of ways. One interpretation follows that users did not believe that the explainable was effective in explaining TinyML concepts. This, however, contradicts the finding that users reported that they found the explainable to provide an effective explanation of TinyML. Another interpretation could be that while they are now familiar with TinyML, they hold low confidence in explaining or using TinyML in other contexts. This is supported by the lower self-reports for the participants believing they could explain TinyML to another person and that they would feel comfortable using TinyML after completing the explainable.

One change we did see between the pre- and post- test was the difference in confidence for knowledge about AI and TinyML. We saw increases in both. This would make sense, given that some of the information covered in the TinyML pipeline is applicable to machine learning at large. However, we saw a more drastic increase for confidence in knowledge about TinyML. This could be caused by a combination of several factors. One is that our explainable did its function in providing users understanding about TinyML so that users felt more confident in their knowledge about the topic. Participants were also able to distinguish between AI in general and TinyML (supported by the correlation test), which is why confidence rose for TinyML more. Another possible factor is the floor effect, where because the participants had no knowledge of TinyML before the user studies—possibly to the point where they had never even heard of TinyML, the post-test survey confidence had much more room to increase.

For the rest of the System Usability Scale questions, we see several patterns emerge. Participants reported mixed opinions on the explainable being easy to follow. This is corroborated by two of the five users asking for in-person clarification with the explainable during the user studies. Participants found the sections in the explainable to be cohesive. This may give some credence to the metaphorical narrative and grounded example-based dataset that we used in our explainable that we proposed in the design step of our explainable. This could also be the participants being very nice and not wanting to hurt the experimenter's feelings regarding the explainable. Opinions were generally moderate about the design of the explainable and whether participants enjoyed using the explainable. Again, this could be the participants holding back their true opinions to be considerate when rating the explainable. It may be more likely that the participants viewed the explainable as a medium for information about TinyML and found the explainable overall inoffensive. This is corroborated by three of the five participants informing the experimenter that they found the explainable and TinyML "interesting and somewhat fun." Even incidentally, this addresses the design consideration of whether the goal of the explainable was to be purely educational or if the user should derive additional enjoyment from it. However, it is difficult to ascertain the source of the enjoyment, whether it came from learning or from interacting with the explainable.

One last important finding from the SUS questions was that participants reported that they were interested in learning more about TinyML. Even if we disregard every other finding that we have found about this explainable, our explainable would still be able to contribute introducing TinyML to users and making them more interested in the topic. This explainable would still hold some value

for the education of TinyML for users with solely this finding.

We must finally contend with the changes, or lack thereof, in results between the pre- and post-tests. As stated earlier, we did not see any dramatic differences in results, with several of the results being the same, for the questions about personal views towards AI. This could mean that the explainable was not well designed to appeal to any personal views about AI. This doesn't have to detract from the explainable though. The explainable could have been effective in being usable and effective in explainable users about TinyML—which the results seem to weakly indicate—without having to provide worldview shattering insights about AI. These questions may have ultimately been irrelevant to the goals of the explainable.

Ultimately, the results from the user studies seem to point towards the conclusion that our explainable was effective in informing participants about TinyML and the combined TinyML-XAI pipeline. We were able to address the design considerations we raised when defining and designing the explainable. The explainable was limited in how many effects it left for users, but we were still able to convey the topics we wanted to users.

5.2.1 Limitations

There are some limitations that come with the problem and defining the scope of this project but most of the limitations discussed comes from the methodology of this project, defined in Chapter 4.

One issue that is inherent with the user studies is the small sample size used. Although we collected descriptive information about the results of the pre- and post- tests, as well as ran light statistical analysis on the results, our small sample size limits the generalizability of the explainable. We are lacking statistical power with our studies, so that the results end up acting more like a suggestion rather than a rule for explainability. If we were to run the study with more participants, we would be able to run more statistical tests to generalize our results further. Additionally, the demographic information of our participants was very limited. We pulled the participants from exclusively Williams College undergraduate students who had taken an introductory computer science course in the Fall of 2023. This is a hyper specific group of people to sample from so this could create issues with external validity when generalizing to a wider audience. By running studies on a larger and more diverse population, we ensure that our study has greater external validity.

Another limitation of this study was with the question of "How do we support users in understanding and appropriately trusting TinyML models?" The idea of "appropriately trusting" becomes hard to define as this changes with different contexts [53]. We more easily support users in understanding TinyML, as this involves putting information into the explainable in a concise manner. However, "appropriately trusting" becomes hard to quantify. We try and address this by describing different scenarios of TinyML contexts so that users understand trust is something that is context-dependent. However, our pre- and post-tests don't address this as we would need multiple tests to effectively quantify this information. Solving this would help provide insights into how we effectively explain to users a topic generalizable to different topics.

One final limitation that must be addressed is the issue of internal validity within the methodology of this study. We are explaining to users about TinyML and the TinyML-XAI pipeline, with many

individual steps and concepts to learn. One question that is never addressed is "Are we explaining what we should be explaining?" The design of this study involved a large literature review done by a single researcher. Ideally, we would employ a robust process to ensure this information is relevant to TinyML, such as by using Cognitive Task Analysis (CTA) with users to determine if the information they learned in the explainable match up to concepts that we have pre-determined should be learned. Fixing this limitation would allow us to trust our results more and actually believe that our explainable is working as it is expected to work as CTA extracts implicit and explicit knowledge from experts that we use in the explainable knowing that it is relevant knowledge [12].

5.3 Summary

Analyzing the results of the user studies provided us with deeper insights into the usability and effectiveness of the web application explainable we created. By analyzing the pre-test results, post-test exploratory results, and changes in pre- and post-test results, we found that the explainable was effective in its ability to inform users about TinyML and the TinyML-XAI pipeline. We found that while the explainable did not alter users' preconceived opinions about AI, we did increase confidence in knowledge regarding both AI and TinyML. Additionally, we found out what level of background information our typical user group roughly had, so that we could more closely understand the learning needs of our users. We also ran usability testing, which found that the explainable was mostly usable, with participants generally enjoying the explainable; one promising result found that users were likely to want to learn more about TinyML in the future. In addition, we cover several limitations of this study, which includes the low sample size of the user studies, the issue with trying to define what "appropriately trusting" in a single context, and the lack of internal validity with our explainable caused by not checking the topics taught with predetermined concepts.

Chapter 6

Conclusion

6.1 Contributions

With this thesis, we contributed the following to the body of work on TinyML Explainables:

- Proposed a new TinyML-XAI Pipeline to streamline TinyML work processes and to provide additional interpretability built into the framework.
- Datascraped online forums to gather demographic information about TinyML users. Provides insight into who the typical users of TinyML are and what their needs are.
- Created an accelerometer sensor dataset of various different swimming strokes. Provides a great example of a dataset that is ideal for TinyML use cases.
- Built an explainable that explains how to use and navigate the TinyML-XAI Pipeline.

TinyML is a quickly growing field with a user base that is eager to learn more about TinyML and ML in general. As TinyML is relatively cheap compared with larger machine learning models, it becomes the everyperson's entry into using machine learning. Due to its accessibility and low entrance cost, we expect to see more and more people utilizing TinyML in various contexts. However, we want people to use machine learning efficiently, effectively, and ethically. We want people to be able to learn these skills quickly and with low cost, as not everyone has time or can afford to take classes on machine learning. So, this thesis raises the question "How do users know how to effectively use and appropriately trust these models?" Ultimately, we decided we utilize explainables as a method to provide understanding to users about these concepts. Explainables are effective as they are designed for users to gain understanding of a system so that they appropriately trust it. There are many explainables that exist for various topics.

However, there currently are no explainables for TinyML, nor are there effective frameworks on how to use TinyML. So, we looked to create an explainable that effectively taught users the ins-and-outs of TinyML. The explainable should be able to provide necessary understanding for how to effectively use TinyML while avoiding the pitfalls of TinyML interpretability simultaneously.

For example, currently an issue we face with TinyML is that many resources for TinyML learning are geared towards traditional ML. This is an issue that affects many aspects of TinyML, ranging from the datasets used to models explained. Creating an explainable that is focused on TinyML use is a necessity for the field to grow. While an explainable is useful for users to understand how TinyML works at a high level, it does not give information about specific outputs. Thus, I propose a combined Tiny-XAI pipeline that integrates the post-hoc XAI framework. By explaining this framework to users, the hope is that they understand and appropriately trust the TinyML systems they use.

Although we know that we should build an explainable, the question lingers of how do we create an effective explainable. We want to address both the usability of the explainable and the effectiveness of the explainable. In order to ensure that we created a good explainable, we employed a user-centered design process. With this design framework, we began to design the explainable.

One of the first steps of the user-centered design was to find out who our user is by conducting user research and analysis. So, we datascraped an online forum to gather demographic information about TinyML users. We scraped the forum for information about user location, user motivation for learning TinyML, and user occupation. We found that most common demographic for these features, respectively, was the United States for user location, gaining knowledge for user motivation, and student for user occupation. So, these became the characteristics of the typical user, which we kept in mind as we designed the explainable.

Within our explainable, we wanted to implement an example-based dataset as this provides more clarity for understanding the topics in the explainable. Given the requirements of being related to the metaphorical narrative used in our explainable, being a dataset for a use case that is inherently related to TinyML, and having previous literature back up the possible dataset, we decided upon using accelerometer data classify swimming strokes as our example-based dataset of choice. This served multiple purposes, including being used in our example as well as being a useful dataset for TinyML uses.

With all of this knowledge, we then created the actual explainable. This was an iterative process with prototyping and testing. This began with a low fidelity paper prototype; this stage was necessary to get rough ideas out and to start planning out what the eventual end product will look like. We then compared this prototype against Nielsen's 10 Usability Heuristics to fix any violations for the next prototype. We then created the medium fidelity slides prototype. This prototype served mainly as the gap between the low and high level prototypes, where it closely resembled the high fidelity version while being less labor intensive. We conducted a light pilot study, where two individuals tested the prototype and conducted a think-aloud review. With the comments from these individuals, we began to create the final end product. Ultimately at the end of this process, we created an explainable for users to learn more about TinyML.

We then had users use the explainable to see how it affected their opinion on AI and various views they held. We found that using the explainable had little to no effect on pre-held views about AI for participants in the study. We did see a slight increase in confidence in knowledge about AI and TinyML from the participants of the study after using the explainable. Additionally, participants generally rated the explainable as fairly effective in explaining TinyML.

6.2 Future Work

6.2.1 Diverse Population Sample

One direction this project could be further improved is with a larger, more diverse population sample as its participants. The researchers wished to be able to perform more quantitative analysis with the data obtained from the user studies in Chapter 4, but the small sample size held back this possibility. Additionally, one limitation and concern of this project was its external validity. We could rectify these issues in the future by running an identical study but with more participants from varying backgrounds. We could compare this to our original study to see if there are any quirks that are associated with the sample population we pulled from.

Additionally, we could add background as another variable to analyze with our results. For instance, all the users that we had were undergraduates majoring in either computer science, math, or both. It could be interesting to see how users majoring in a social science or humanities topic respond to the explainable as compared with our computer science and math users. This could provide more insight on how we create an explainable that is truly novice friendly as we apply our explainable to different user groups.

6.2.2 Experimentally Designed Explainable

In addition to applying our explainable to different user groups, it could also be interesting to apply different explainables to the same user group. With a more robust understanding of what questions lend itself towards what concepts, we could create an experimental explainable, where we have slight variations in information and/or the way the information is presented that we test against users. By creating an experimental setup, we draw stronger correlations between certain concepts, as well as identifying causational relationships with our design choices. If we incorporate this into the iterative design process, we could create a more well-informed final explainable by determining what design choices have better outcomes.

We inversely also apply the same explainable to different user groups to see what areas of the explainable are effective for some groups and not for others. While we were running our user analysis for this project, we had determined that the occupation we would be focusing on was students. If we look at the graph in Figure 3.4, however, we see that engineers were almost just as prevalent. However, using the same explainable for a group of novice students and more experienced engineers would lead to some discrepancies. So, by analyzing how different user groups respond to the same explainable, we more easily determine effective knowledge gaps for certain groups.

6.3 Summary

The goal of this thesis was to build a framework for working with TinyML that allows users to understand and appropriately trust the models they work with. To do this, we drew from the learning sciences to use explainables as a medium for users to interact with so that they could reach this knowledge. We datascraped online forums to discover user information regarding TinyML users

so that we could more effectively create an explainable that was catered to a specific demographic. We then used an iterative design process to create various levels of prototypes of our explainable and test them against users so that we could reach a high level of usability and effectiveness in explaining TinyML to users. The user studies we ran showed that our explainable was effective in usability and effectiveness in informing participants about TinyML, with the added benefit of increasing confidence in knowledge about AI and TinyML for users. Ultimately, the results of the study showed that TinyML is a concept that be introduced to users for the betterment of their understanding of AI and trust in AI systems.

Appendices

Appendix A

Pre- and Post-Test Questions

This is the full list of all the questions used in the pre- and post-tests in the user study described in Chapter 4. Some things to note about the tests are that

- These questions were asked on a Google Form.
- Every single question was required to be filled out.
- All questions asking about a self-report was scored on the 5-point Likert scale.
- Questions were repeated in the pre- and post-test.

A.1 Pre-Test

The following text was included at the top of the test:

This survey is to be taken before using the interactive web application about TinyML. The answers to this survey will be completely anonymous and is for collecting demographic information and knowledge.

A.1.1 Demographic Information

1. Age
    A. < 18
    B. 18
    C. 19
    D. 20
    E. 21
    F. > 21

2. Gender
    A. Male
    B. Female
    C. Non-binary
    D. Transgender Male
    E. Transgender Female
    F. Genderfluid
    G. Other
3. Nationality
    A. Asian or Pacific Islander
    B. Hispanic or Latino
    C. Native American or Alaskan Native
    D. White or Caucasian
    E. Multiracial or Biracial
    F. Other
4. (Prospective) College Major(s) Area[1]
    A. Div I
    B. Div II
    C. Div III
5. Specific Major Name

A.1.2 Math and CS Background

The following text was included at the top of the test:

Please continue with answering all questions. For the math problems, please do the work without any external help. You are NOT being evaluated for correct answers.

1. I am confident in my knowledge about math/statistics
2. If we multiple a 2x3 matrix by a 3x4 matrix, what dimension matrix do we end up with?
3. If we have the 2x4 matrix A, what dimension is the transpose of A
4. If we flip a fair coin twice, what is the probability that we get heads both times?
5. If we roll two dice, what is the probability that they sum up to 5?

[1]The answers to this question are sorted in line with Williams College division of academic majors

6. I would consider myself a tech-savvy person
7. I am confident in my knowledge about computer science
8. I am confident in my knowledge about artificial intelligence
9. There are many beneficial applications of artificial intelligence
10. I support the increasing prevalence of AI in society.
11. AI is used too much in daily life
12. It is important to educate the public on how AI systems work.
13. It is important to focus on the potential harms and limitations of AI systems.
14. We should be more cautious than excited about AI
15. I have taken a formal class on AI/ML
16. If so, where did you take the class?
17. I am confident in my knowledge about TinyML

A.2 Post-Test

A.2.1 Explainables Quiz

The following text was included at the top of the test:

This survey is to be taken after using the interactive web application about TinyML. The answers to this survey will be completely anonymous and is for collecting information about user knowledge and explainable effectiveness (we're evaluating the explainable, not you!).

1. List the 5 main stages of the TinyML pipeline
2. Why is TinyML becoming popular?
3. What is the benefit of using decision trees over neural networks
4. We can trade accuracy for...
5. We can trade accuracy for...
6. Why or why not?

A.2.2 Explainable Evaluation

The following text was included at the top of the test:

Please rate how you agree/disagree with the following statements:

1. I found the various sections in this explainable were well integrated.
2. I found the explainable to be well-designed
3. I found this explainable engaging and interesting.
4. I enjoyed completing this explainable.
5. I thought this explainable provided an effective explanation of TinyML.
6. I could generally explain the TinyML pipeline to another person.
7. I could generally explain the TinyML pipeline to another person.
8. I am interested in learning more about TinyML.
9. I am confident in my knowledge about artificial intelligence
10. There are many beneficial applications of artificial intelligence
11. I support the increasing prevalence of AI in society.
12. AI is used too much in daily life
13. It is important to educate the public on how AI systems work.
14. It is important to focus on the potential harms and limitations of AI systems.
15. We should be more cautious than excited about AI
16. I am confident in my knowledge about TinyML

Bibliography

[1] Abras, C., Maloney-Krichmar, D., Preece, J., et al. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications 37*, 4 (2004), 445–456.

[2] Adar, E., and Lee, E. Communicative visualizations as a learning problem. *CoRR abs/2009.07095* (2020).

[3] Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., and Taha, K. Efficient machine learning for big data: A review. *Big Data Research 2*, 3 (2015), 87–93. Big Data, Analytics, and High-Performance Computing.

[4] Anik, A. I., and Bunt, A. Data-centric explanations: Explaining training data of machine learning systems to promote transparency. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2021), CHI '21, Association for Computing Machinery.

[5] Banbury, C. R., Reddi, V. J., Lam, M., Fu, W., Fazel, A., Holleman, J., Huang, X., Hurtado, R., Kanter, D., Lokhmotov, A., Patterson, D. A., Pau, D., Seo, J., Sieracki, J., Thakker, U., Verhelst, M., and Yadav, P. Benchmarking tinyml systems: Challenges and direction. *CoRR abs/2003.04821* (2020).

[6] Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion 58* (2020), 82–115.

[7] Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA, 2021), FAccT '21, Association for Computing Machinery, p. 610–623.

[8] Bifarin, O., and Fernandez, F. Automated machine learning and explainable ai (automl-xai) for metabolomics: improving cancer diagnostics. *bioRxiv : the preprint server for biology* (10 2023).

[9] Cai, C. J., Jongejan, J., and Holbrook, J. The effects of example-based explanations in a machine learning interface. In *Proceedings of the 24th international conference on intelligent user interfaces* (2019), pp. 258–262.

[10] Cao, K., Liu, Y., Meng, G., and Sun, Q. An overview on edge computing research. *IEEE access 8* (2020), 85714–85728.

[11] Chaoji, V., Rastogi, R., and Roy, G. Machine learning in the real world. *Proc. VLDB Endow. 9*, 13 (sep 2016), 1597–1600.

[12] Clark, R. E., Feldon, D. F., Van Merrienboer, J. J., Yates, K. A., and Early, S. Cognitive task analysis. In *Handbook of research on educational communications and technology.* Routledge, 2008, pp. 577–593.

[13] Cooper, A. *The Inmates are Running the Asylum.* Vieweg+Teubner Verlag, Wiesbaden, 1999, pp. 17–17.

[14] Costa, J., Silva, C., Santos, M., Fernandes, T., and Faria, S. Framework for intelligent swimming analytics with wearable sensors for stroke classification. *Sensors 21*, 15 (2021), 5162.

[15] D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., Hormozdiari, F., Houlsby, N., Hou, S., Jerfel, G., Karthikesalingam, A., Lucic, M., Ma, Y., McLean, C., Mincu, D., Mitani, A., Montanari, A., Nado, Z., Natarajan, V., Nielson, C., Osborne, T. F., Raman, R., Ramasamy, K., Sayres, R., Schrouff, J., Seneviratne, M., Sequeira, S., Suresh, H., Veitch, V., Vladymyrov, M., Wang, X., Webster, K., Yadlowsky, S., Yun, T., Zhai, X., and Sculley, D. Underspecification presents challenges for credibility in modern machine learning. *J. Mach. Learn. Res. 23*, 1 (jan 2022).

[16] David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Wang, T., Warden, P., and Rhodes, R. Tensorflow lite micro: Embedded machine learning for tinyml systems. In *Proceedings of Machine Learning and Systems* (2021), A. Smola, A. Dimakis, and I. Stoica, Eds., vol. 3, pp. 800–811.

[17] de Kock, E., van Biljon, J., and Pretorius, M. Usability evaluation methods: mind the gaps. In *Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists* (New York, NY, USA, 2009), SAICSIT '09, Association for Computing Machinery, p. 122–131.

[18] Deisenroth, M. P., Faisal, A. A., and Ong, C. S. *Mathematics for machine learning.* Cambridge University Press, 2020.

[19] Doorley, S., Holcomb, S., Klebahn, P., Segovia, K., , and Utley, J. Design thinking bootleg, 2018.

[20] Eiband, M., Schneider, H., Bilandzic, M., Fazekas-Con, J., Haug, M., and Hussmann, H. Bringing transparency design into practice. In *23rd International Conference on Intelligent User Interfaces* (New York, NY, USA, 2018), IUI '18, Association for Computing Machinery, p. 211–223.

[21] Eisinga, R., Grotenhuis, M. t., and Pelzer, B. The reliability of a two-item scale: Pearson, cronbach, or spearman-brown? *International journal of public health 58* (2013), 637–642.

[22] Ervas, F., Gunia, A., Lorini, G., Stojanov, G., and Indurkhya, B. Fostering safe behaviors via metaphor-based nudging technologies. In *International Conference on Software Engineering and Formal Methods* (2021), Springer, pp. 53–63.

[23] Felzmann, H., Fosch-Villaronga, E., Lutz, C., and Tamò-Larrieux, A. Towards transparency by design for artificial intelligence. *Science and Engineering Ethics 26*, 6 (2020), 3333–3361.

[24] Gade, K., Geyik, S. C., Kenthapadi, K., Mithal, V., and Taly, A. Explainable ai in industry. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (New York, NY, USA, 2019), KDD '19, Association for Computing Machinery, p. 3203–3204.

[25] Ghai, B., Liao, Q. V., Zhang, Y., Bellamy, R., and Mueller, K. Explainable active learning (xal): An empirical study of how local explanations impact annotator experience. *arXiv preprint arXiv:2001.09219* (2020).

[26] Gunning, D. Darpa's explainable artificial intelligence (xai) program. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2019), IUI '19, Association for Computing Machinery, p. ii.

[27] Hall, R. R. Prototyping for usability of new technology. *International Journal of Human-Computer Studies 55*, 4 (2001), 485–501.

[28] Han, H., and Siebert, J. Tinyml: A systematic review and synthesis of existing research. In *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)* (2022), pp. 269–274.

[29] Harackiewicz, J. M., Barron, K. E., Pintrich, P. R., Elliot, A. J., and Thrash, T. M. Revision of achievement goal theory: Necessary and illuminating. *Journal of Educational Psychology* (2002).

[30] Heim, L., Biri, A., Qu, Z., and Thiele, L. Measuring what really matters: Optimizing neural networks for tinyml. *CoRR abs/2104.10645* (2021).

[31] Hoffman, R. R., Mueller, S. T., Klein, G., and Litman, J. Metrics for explainable AI: challenges and prospects. *CoRR abs/1812.04608* (2018).

[32] Hohman, F., Conlen, M., Heer, J., and Chau, D. H. P. Communicating with interactive articles. *Distill 5*, 9 (2020), e28.

[33] Holzinger, A., Carrington, A., and Müller, H. Measuring the quality of explanations: the system causability scale (scs) comparing human and machine explanations. *KI-Künstliche Intelligenz 34*, 2 (2020), 193–198.

[34] Ibrahim, L., Mesinovic, M., Yang, K.-W., and Eid, M. A. Explainable prediction of acute myocardial infarction using machine learning and shapley values. *IEEE Access 8* (2020), 210410–210417.

[35] Ivory, M. Y., and Hearst, M. A. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv. 33*, 4 (dec 2001), 470–516.

[36] Jacqueline, A. A., and Chapman, A. Putting ai ethics to work: are the tools fit for purpose? *AI and Ethics 2* (September 2021), 405?429.

[37] Janapa Reddi, V., Plancher, B., Kennedy, S., Moroney, L., Warden, P., Agarwal, A., Banbury, C., Banzi, M., Bennett, M., Brown, B., Chitlangia, S., Ghosal, R., Grafman, S., Jaeger, R., Krishnan, S., Lam, M., Leiker, D., Mann, C., Mazumder, M., Pajak, D., Ramaprasad, D., Smith, J. E., Stewart, M., and Tingley, D. Widening Access to Applied Machine Learning with TinyML. *arXiv e-prints* (June 2021), arXiv:2106.04008.

[38] Jentner, W., Sevastjanova, R., Stoffel, F., Keim, D. A., Bernard, J., and El-Assady, M. Minions, sheep, and fruits: Metaphorical narratives to explain artificial intelligence and build trust, 2018.

[39] Jordan, M. I., and Mitchell, T. M. Machine learning: Trends, perspectives, and prospects. *Science 349*, 6245 (2015), 255–260.

[40] Kallimani, R., Pai, K., Raghuwanshi, P., Iyer, S., and López, O. L. A. Tinyml: Tools, applications, challenges, and future research directions. *Multimedia Tools and Applications* (Sep 2023).

[41] Kenny, E. M., Ford, C., Quinn, M., and Keane, M. T. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies. *Artificial Intelligence 294* (2021), 103459.

[42] Khan, L. U., Yaqoob, I., Tran, N. H., Kazmi, S. M. A., Dang, T. N., and Hong, C. S. Edge-computing-enabled smart cities: A comprehensive survey. *IEEE Internet of Things Journal 7*, 10 (2020), 10200–10232.

[43] Koçak, B. Key concepts, common pitfalls, and best practices in artificial intelligence and machine learning: focus on radiomics. *Diagnostic and Interventional Radiology 28*, 5 (2022), 450.

[44] Kujala, S., and Kauppinen, M. Identifying and selecting users for user-centered design. In *Proceedings of the Third Nordic Conference on Human-Computer Interaction* (New York, NY, USA, 2004), NordiCHI '04, Association for Computing Machinery, p. 297–303.

[45] Lacoste, A., Luccioni, A., Schmidt, V., and Dandres, T. Quantifying the Carbon Emissions of Machine Learning. *arXiv e-prints* (Oct. 2019), arXiv:1910.09700.

[46] Lakkaraju, H., and Bastani, O. "how do i fool you?": Manipulating user trust via misleading black box explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (New York, NY, USA, 2020), AIES '20, Association for Computing Machinery, p. 79–85.

[47] Lakkaraju, H., Kamar, E., Caruana, R., and Leskovec, J. Interpretable & explorable approximations of black box models. *CoRR abs/1707.01154* (2017).

[48] Lipton, Z. C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue 16*, 3 (2018), 31–57.

[49] Mahbooba, B., Timilsina, M., Sahal, R., Serrano, M., and Khalil, A. M. Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model. *Complex. 2021* (jan 2021).

[50] Markus, A. F., Kors, J. A., and Rijnbeek, P. R. The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics 113* (2021), 103655.

[51] Mueller, J. P., and Massaron, L. *Machine learning for dummies*. John Wiley & Sons, 2021.

[52] Mukhtar, S. Decoding the black box: Explainable ai strategies in data engineering pipelines. *Journal Environmental Sciences And Technology 3*, 1 (2024), 207–221.

[53] Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., Schlötterer, J., van Keulen, M., and Seifert, C. From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai. *ACM Computing Surveys 55*, 13s (2023), 1–42.

[54] Nielsen, J., and Molich, R. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1990), CHI '90, Association for Computing Machinery, p. 249–256.

[55] Noyes, J., and Baber, C. *User-centred design of systems*. Springer Science & Business Media, 1999.

[56] Osman, A., Abid, U., Gemma, L., Perotto, M., and Brunelli, D. Tinyml platforms benchmarking. In *Applications in Electronics Pervading Industry, Environment and Society* (Cham, 2022), S. Saponara and A. De Gloria, Eds., Springer International Publishing, pp. 139–148.

[57] Pan, M.-S., Huang, K.-C., Lu, T.-H., and Lin, Z.-Y. Using accelerometer for counting and identifying swimming strokes. *Pervasive and Mobile Computing 31* (2016), 37–49.

[58] Perer, A., Strobelt, H., El-Assady, M., Bäuerle, A., Boggust, A., Hohman, F., Johnson, I., and Wang, Z. J. Ieee vis workshop on visualization for explainable ai.

[59] Pinto dos Santos, D., Giese, D., Brodehl, S., Chon, S.-H., Staab, W., Kleinert, R., Maintz, D., and Baessler, B. Medical students' attitude towards artificial intelligence: a multicentre survey. *European radiology 29* (2019), 1640–1646.

[60] Poursabzi-Sangdeh, F., Goldstein, D. G., Hofman, J. M., Wortman Vaughan, J. W., and Wallach, H. Manipulating and measuring model interpretability. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2021), CHI '21, Association for Computing Machinery.

[61] Prakash, S., Stewart, M., Banbury, C., Mazumder, M., Warden, P., Plancher, B., and Reddi, V. J. Is tinyml sustainable? *Commun. ACM 66*, 11 (oct 2023), 68–77.

[62] Rajapakse, V., Karunanayake, I., and Ahmed, N. Intelligence at the extreme edge: A survey on reformable tinyml. *ACM Computing Surveys 55*, 13s (2023), 1–30.

[63] Ray, P. P. A review on tinyml: State-of-the-art and prospects. *Journal of King Saud University - Computer and Information Sciences 34*, 4 (2022), 1595–1623.

[64] Rebala, G., Ravi, A., and Churiwala, S. *An introduction to machine learning*. Springer, 2019.

[65] Reed, J., DeVito, Z., He, H., Ussery, A., and Ansel, J. torch.fx: Practical program capture and transformation for deep learning in python. In *Proceedings of Machine Learning and Systems* (2022), D. Marculescu, Y. Chi, and C. Wu, Eds., vol. 4, pp. 638–651.

[66] Ribera, M., and Lapedriza, A. Can we do better explanations? a proposal of user-centered explainable ai, 2019.

[67] Rozemberczki, B., Watson, L., Bayer, P., Yang, H.-T., Kiss, O., Nilsson, S., and Sarkar, R. The shapley value in machine learning, 2022.

[68] Samek, W., Wiegand, T., and Üller, K. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *CoRR abs/1708.08296* (2017).

[69] Schepman, A., and Rodway, P. Initial validation of the general attitudes towards artificial intelligence scale. *Computers in Human Behavior Reports 1* (2020), 100014.

[70] Shen, H., and Huang, T.-H. How useful are the machine-generated interpretations to general users? a human evaluation on guessing the incorrectly predicted labels. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* (2020), vol. 8, pp. 168–172.

[71] Sitzmann, T., Ely, K., Brown, K. G., and Bauer, K. N. Self-assessment of knowledge: A cognitive learning or affective measure? *Academy of Management Learning & Education 9*, 2 (2010), 169–191.

[72] Snyder, C. *Paper prototyping: The fast and easy way to design and refine user interfaces.* Morgan Kaufmann, 2003.

[73] Song, C., Ristenpart, T., and Shmatikov, V. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2017), CCS '17, Association for Computing Machinery, p. 587–601.

[74] Speith, T. A review of taxonomies of explainable artificial intelligence (xai) methods. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA, 2022), FAccT '22, Association for Computing Machinery, p. 2239–2250.

[75] Spinner, T., Schlegel, U., Schäfer, H., and El-Assady, M. explainer: A visual analytics framework for interactive and explainable machine learning. *IEEE transactions on visualization and computer graphics 26*, 1 (2019), 1064–1074.

[76] Sun, C., Shrivastava, A., Singh, S., and Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. *CoRR abs/1707.02968* (2017).

[77] Sundararajan, M., and Najmi, A. The many shapley values for model explanation. In *International conference on machine learning* (2020), PMLR, pp. 9269–9278.

[78] Tsoukas, V., Boumpa, E., Giannakas, G., and Kakarountas, A. A review of machine learning and tinyml in healthcare. In *Proceedings of the 25th Pan-Hellenic Conference on Informatics* (New York, NY, USA, 2022), PCI '21, Association for Computing Machinery, p. 69–73.

[79] Van Veen, R., Biehl, M., and De Vries, G.-J. Sklvq: Scikit learning vector quantization. *J. Mach. Learn. Res. 22*, 1 (jan 2021).

[80] Vellido, A. The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural computing and applications 32*, 24 (2020), 18069–18083.

[81] Villalobos, P., Sevilla, J., Besiroglu, T., Heim, L., Ho, A., and Hobbhahn, M. Machine Learning Model Sizes and the Parameter Gap. *arXiv e-prints* (July 2022), arXiv:2207.02852.

[82] Wang, D., Yang, Q., Abdul, A., and Lim, B. Y. Designing theory-driven user-centric explainable ai. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2019), CHI '19, Association for Computing Machinery, p. 1–15.

[83] Warden, P., and Situnayake, D. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media, 2019.

[84] Wiggins, G., and McTighe, J. *Understanding by Design*. Gale Reference. Association for Supervision and Curriculum Development, 2005.

[85] Wolcott, M. D., and Lobczowski, N. G. Using cognitive interviews and think-aloud protocols to understand thought processes. *Currents in Pharmacy Teaching and Learning 13*, 2 (2021), 181–188.

[86] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., and Zhu, J. Explainable ai: A brief survey on history, research areas, approaches and challenges. In *Natural Language Processing and Chinese Computing* (Cham, 2019), J. Tang, M.-Y. Kan, D. Zhao, S. Li, and H. Zan, Eds., Springer International Publishing, pp. 563–574.

[87] Yu, C.-h. Reliability of self-report data. *Retrieved August 13* (2010), 2011.

[88] Zhou, Y., Yu, Y., and Ding, B. Towards mlops: A case study of ml pipeline platform. In *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)* (2020), pp. 494–500.