# Realising the Power of Edge Intelligence: Addressing the Challenges in AI and tinyML Applications for Edge Computing

**Michael Gibbs & Eiman Kanjo**

**Smart Sensing Lab, Department of Computer Science, Nottingham Trent University, United Kingdom**

## Introduction

Over the past few years, embedded systems and machine learning communities have come together to make AI ubiquitous and available near the data source, unlocking many untapped application areas that await development. As a result, hardware, software, and research have changed extremely rapidly with many recent releases of ML-enabled microcontrollers. Consequently, many frameworks have been developed for different platforms to facilitate the deployment of ML models and standardise the process. With the Artificial Intelligence of Things (AIoT) expected to grow exponentially over the next few years, more researchers and companies are expected to enter the research space. Although certain challenges of tinyML deployment can be overlooked, which makes entering the field challenging. For tinyML applications to flourish, it is important to consider how to solve entry-level challenges. The challenges below were experienced when deploying simple deep learning models to a variety of microcontrollers. These include, but are not limited to Syntiant TinyML Development Board, Sony Spresense Main Board, and Raspberry Pi Pico4ML.

This poster reveals the often-overlooked challenges of tinyML and emphasizes the importance of raising awareness within the community. By addressing these issues, not only will the existing tinyML community benefit, but it will also attract a broader range of people. This will accelerate research in the field, pushing the boundaries of edge AI further and faster than ever before.

## The Challenges Summarised

### Programming Language Choice

Programming languages are always not optional when developing on microcontrollers. This influences the decision of which board to invest time into. Moreover, different programming languages yield varying benefits such as compile time and library availability.

### Lack of Support on Development Boards

Development boards are often released with limited tutorials and library compatibility, resulting in low community engagement. Therefore, the learning becomes steep, potentially stagnating the user base.

### Neglect of Preprocessing

When tinyML algorithms are deployed, it can be common to neglect the state of the input data to the model. Feature extraction can be a resource-consuming (and therefore power-consuming) task requiring many mathematical computations.

### Choice of Sensors

Edge AI / tinyML systems use sensors to collect data for inference, but selecting the right sensor and integrating it into the algorithm is not always straightforward. Choosing an efficient sensor for a specific task is a complex decision in the deployment of tinyML systems.

### Insufficient Labelled Data

Data labelling is required for supervised learning approaches, and obtaining such data can be a prolonged task. This is especially prevalent for multimodal sensor data.

## Programming Language Choice

When developing on microcontrollers, programming language choice becomes limited to typically C++ or python.


- **Memory and Resources**: C++ optimizes memory usage better for constrained microcontrollers, while Python may have higher overhead.
- **Performance**: C++ is faster due to direct hardware access and compilation, vital for real-time tasks on microcontrollers.
- **Language Features**: C++ offers low-level control, while Python prioritizes faster development and ease of use.
- **Libraries and Ecosystem**: C++ has extensive microcontroller-specific libraries, while Python's offerings are comparatively smaller but growing.
- **File Size**: C++ generates smaller binaries, advantageous for microcontrollers with limited storage.
- **Learning Curve**: C++ is more complex, Python is beginner-friendly.

The choice between Python and C++ for programming on a microcontroller depends on performance needs, memory constraints, available libraries, and developer experience. C++ provides control and efficiency, while Python offers faster development but may use more resources.

## Lack of Support on Development Boards

Community engagement is crucial for microcontrollers, small devices used in IoT and automation systems. It offers significant benefits:

1) **Learning and education**: Online forums and workshops help users understand microcontrollers, encouraging their adoption and expanding their application potential.
2) **Collaboration**: Open-source projects enable developers to collaborate, resulting in faster software development and more reliable applications.
3) **Support**: Community platforms provide troubleshooting assistance, benefiting new users and complex projects.
4) **Innovation**: Sharing knowledge and ideas drives innovation in microcontroller usage, advancing the field of IoT and automation.

However, limited support for new boards can make development challenging for new users and potentially impractical.


## Neglect of Preprocessing

Preprocessing plays an important role in AI/ML on microcontrollers, but it is often overlooked.

Feature engineering involves extracting key information from sensor data using mathematical techniques. Traditional methods like the Fourier transform (FT) can be resource-intensive due to complex calculus. However, variations such as the Fast Fourier Transform (FFT) have been developed, trading off precision for low latency.

Emphasizing the importance of preprocessing can lead to more effective and efficient deployments on resource-constrained microcontroller platforms.

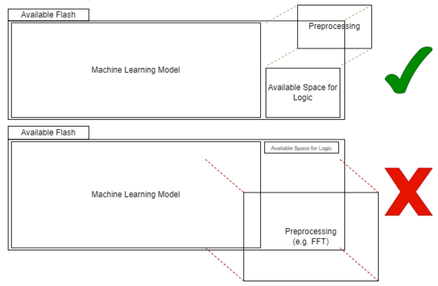


**Figure demonstrating how preprocessing can over encumber a typical microcontroller.**

## Choice of Sensors

Edge systems rely on sensors to collect data, with some capable of processing inputs from multiple sources. However, the availability of sensors depends on the hardware used.

The choice of sensors varies based on the problem at hand. For instance, monitoring mental wellbeing may require numerous sensors, while human activity recognition can be accomplished with just an accelerometer. Many new edge devices come with built-in sensors. The Arduino Nano 33 Sense, a popular microcontroller, includes a variety of sensors in a compact form factor (see table 1).

Opting for a board with built-in sensors ensures compatibility and reduces footprint by eliminating the need for external sensors. However, these built-in sensors are generally designed for multiple applications. Specialized purposes like air quality monitoring may require external sensors as boards rarely have specific sensors pre-installed [1].

| | Development Board | Himax WE-I Plus EVB† | Sony Spresense Main Board* | Syntiant TinyML Development Board | SparkFun MicroMod (Artemis) | SparkFun MicroMod (SAMD51) | Raspberry Pi Pico4ML | Arduino Nano 33 Sense BLE† | Arduino Nicla Sense ME | Arduino Nicla Vision | Arduino Nicla Voice |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Sensors | Camera | ✓ | ✓* | ✗ | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | |
| | Accelerometer | ✓ | ✗ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | |
| | Microphone | ✓ | ✓* | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| | Bluetooth | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | |
| | Light/Proximity | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | |
| | Pressure | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | |
| | Temperature/Humidity | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | |
| | Gas | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | |
| | Time of Flight (ToF) | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | |

** Extendable Sony Spresense hardware available*
*† Supported boards for TensorFlow Lite for Microcontrollers (as of 04/2023)*

**Available built-in sensors for development boards while highlighting the boards fully supported by TFLite.**

## Insufficient Labelled Data

Labelled data is essential for training supervised machine learning models. Deep learning, a gradient-based learning method, relies on labelled data to ensure accurate generalization and adjust the model accordingly. However, obtaining cleaned and labelled datasets is challenging.

Data labelling remains a labour-intensive task, usually performed by humans. Although methods like semi-supervised learning can leverage a subset of labelled data, they are most effective in big data applications. Limited research exists on labelling data directly on microcontrollers, with only one known attempt [2]. While not fully automated, this approach simplifies data collection on target devices.

## Discussion

All these challenges must be considered and/or overcome before even designing an AI/ML model, which is considered the *main* challenge. This poses the question;

**How do we work to solve these challenges?**

By highlighting these areas that literature typically misses or at least, fails to mention will lead to a wider community understanding of how to design a tinyML system. Therefore, choosing the appropriate board, sensors, and input features for an optimised intelligent system, before considering the model itself. Expressing justification for the choices made when developing tinyML systems can influence others on which boards to use and create an inviting research space for others to join. Overall, this will increase community engagement across the tinyML research space, thus nurturing its exponential growth.

## Conclusion

In conclusion, the fusion of edge computing and AI has the potential to unlock numerous application areas, particularly in the Internet of Things (IoT). However, deploying AI to microcontrollers presents challenges discussed in this paper. These challenges include programming language selection, limited development board support, preprocessing neglect, sensor choices, and insufficient labeled data. Addressing these challenges enables researchers and developers to design more efficient and effective tinyML systems for research and commercial purposes. Solving these challenges benefits the current edge intelligence and tiny machine learning (tinyML) community, making edge AI research more accessible and driving faster progress in pushing its boundaries.

## References

[1] Johnson, T., & Kanjo, E. (2023). Urban Wellbeing: A Portable Sensing Approach to Unravel the Link Between Environment and Mental Wellbeing. *IEEE Sensors Letters*, *7*(3). https://doi.org/10.1109/LSENS.2023.3243790

[2] Woodward, K., Kanjo, E., Oikonomou, A., & Chamberlain, A. (2020). LabelSens: enabling real-time sensor data labelling at the point of collection using an artificial intelligence-based approach. *Personal and Ubiquitous Computing*, *24*(5), 709–722. https://doi.org/10.1007/S00779-020-01427-X/FIGURES/10

Read More Here

SCAN ME