tinyML Summit
Enabling Ultra-low Power Machine Learning at the Edge

Products and applications enabled by tinyML

March 28 – 29, 2023

www.tinyML.org
Why TinyML Applications Fail

An examination of common challenges and issues encountered in real-world projects

Chris Knorowski
CTO SensiML
chris.knorowski@sensiml.com
Qualifying if TinyML is the right tool for an application
Does machine learning make this solution scalable

Security/Privacy

Latency

Remote Devices or Limited Connectivity

Power

Economical
Steps to build a TinyML Model

1. Raw Signal Capture
2. Data Insight Labeling
3. Algorithm Generation
4. Firmware Code Generation
5. Test, Validation, and Support

Data Capture Lab, Analytics Studio, Test App
microcontrollers  sensors  accelerators  data curation  model training  connectivity

deployment  model monitoring  updating models  dashboard  model compression/quantization
Steps to build an application that uses TinyML
1. Customer wants feature X
2. Do initial demo of tools/model with internal PoC similar to their use case convince yourselves this can be done and convince them externally that the feature is possible.
3. Come up with an SoW to work on a PoC for their specific application
4. Come up with data collection protocol/plan for PoC
5. Carry out data collection plan with customer device
6. Build model
7. Integrate model into PoC for test and validation
8. Customer happy with PoC, but now wants it in the product
Take a holistic to application development to build a robust model that can make it to production
What are the keys to the success of a TinyML application?

- Machine Learning
- Data Science
- Analytics
- Embedded Firmware
- Hardware
- Connectivity
- Domain and Business Expertise
Setting priorities: Horizontal vs Vertical Scaling
“I need this feature in my application. Can you build it?”

“I need this feature in your application. Can you add it?”

Key Partner Platforms  
Simplifying Model APIs  
Easy to Integrate Data API  
Documentation and PoCs
Data collection and annotation effort
Models and Datasets as IP
Defining the Model Scope
Slow test and validation iterations
Condemn technology based on a misunderstanding of limits
Where does TinyML fit into Predictive maintenance

Rotating Machinery Faults
• Vibration data sometimes temperature
• Compute FFT – send up sample once every X hours (probably not going to beat a historical FFT data set analyzed in the cloud)
• Set threshold - set alert above threshold (can we improve over a threshold method?)

Anomaly detection in repeated gestures of industrial machines
• Need to know context of the machine motion. This can involve being in the control path. Which means TinyML may not be necessary.

High bandwidth sensors in difficult to reach places (audio, image) are good fits for TinyML applications
Sensor calibration vs. real-world deployment
Optimized modules targeting specific applications

• IMU + Mic + Wifi + LCD Module + Enclosure
• IMU mounted enclosure for picking up vibration signatures
• Dual Mic + Enclosure with different dampening characteristics
Why TinyML Applications Fail

An examination of common challenges and issues encountered in real-world projects

Chris Knorowski
CTO SensiML
chris.knorowski@sensiml.com
A pipeline is a container for a series of data processing steps. The pipeline object allows you to get an existing pipeline or create a new one with a given name. With this object, you can set input data sources, add transforms, feature generators, feature selectors, feature transforms and classifiers. An example pipeline is shown below.

Examples:

```python
# Feature decay
client.pipeline

# Scale to unit
client.pipeline

# Perform feature selection
with high variance
# (Note: Random forests are often used to select large number of features, but it is recommended to use other feature selection algorithms to first reduce the number of features)
client.pipeline.add_feature_selector()
```

This pipeline uses frequency based feature extractors combined with supervised machine learning classification algorithms to classify different states of machines based on their vibration signatures.

Expected Sensors: AccelerometerX, AccelerometerY, AccelerometerZ

Configuring Template:

1. Select pipeline that best fits your project.
2. Select template information.
3. Select parameters.

Next.
**Supported Devices**
- Arduino Nano 33 BLE Sense
- Arduino Nica Sense ME
- Infineon PSoC™ 6
- M5Stack M5StickC PLUS ESP32-PICO
- Mini IoT Dev Kit
- Microchip Technology AVR128DA48 Curiosity Nano Evaluation Kit
- Microchip Technology PIC-IoT WG Development Board
- Microchip Technology SAMD21 ML Eval Kit (SAM-IoT WG)
- Nordic Thingy
- onsemi RSL10 Sense
- QuickLogic Chilkat
- QuickLogic QuickAI
- QuickLogic QuickFeather
- Raspberry Pi
- Silicon Labs Thunderboard Sense 2
- Silicon Labs xG24 Dev Kit
- SparkFun QuickLogic Thing Plus - EOS S3
- ST SensorTile
- ST SensorTile.box

**Select a Target**

<table>
<thead>
<tr>
<th>Compilers</th>
<th>Development Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>arm</td>
<td>Arduino Nano 33 BLE Sense</td>
</tr>
<tr>
<td>Android NDK</td>
<td>Arduino Nica Sense ME</td>
</tr>
<tr>
<td>ESPRESSIF ESP-IDF</td>
<td>Infineon CY8CKIT-062S2-43012</td>
</tr>
<tr>
<td>MPLAB XIC16</td>
<td>M5Stack M5StickC PLUS ESP32-PICO Development Kit</td>
</tr>
<tr>
<td>Mobile</td>
<td>Microchip AV128DA48 Curiosity Nano Evaluation Kit</td>
</tr>
<tr>
<td>MPLAB XC12</td>
<td>Microchip PIC-IoT WG Development Board</td>
</tr>
<tr>
<td>MPLAB XC8</td>
<td>Microchip SAMD21 ML Evaluation Kit</td>
</tr>
<tr>
<td>Windows x64_64</td>
<td>Nordic NRF52 SDK 13</td>
</tr>
<tr>
<td>x86 GCC Generic</td>
<td>onsemi RSL10 Sense</td>
</tr>
<tr>
<td></td>
<td>QuickLogic QuickFeather</td>
</tr>
</tbody>
</table>

**Supported Compilers**
- Android Native Development Kit
- Arm GCC Cortex M4/M7/A53
- Espressif ESP-IDF ESP32
- Microchip XC8/XC16/XC32
- x86_64 (GCC 9.2/mingw-64 9.3)

**Application Examples**
- Keyword Spotting
- Boxing Punch Gesture Recognition
- Robot Arm Gesture Recognition
- Fan State Condition Monitoring
- Audio Anomaly Detection
- Vibration Anomaly Detection
- Audio Cough Detection
- Guitar Note Audio Recognition
- Smart Lock Audio Recognition
- Wizard Magic Wand Gesture Game
Meet the customer where they are at
We Sell: TinyML toolkit for Smart Sensors

To: Application developers and system integrators

To enable: rapid development of novel applications for edge devices

SensiML Analytics Toolkit - Market Leading AutoML Technology for IoT Endpoint Algorithms
Customer Profile: Hotbeds of Activity
Consumer
Health and Wellness
Agricultural
Industrial
- DSCNN is not the write solution for audio, baggage
- Adam NXP
- Matheiu STM
- Does it scale to a real product – Person detect endless amount of data conditions
- Play the audio backwards
Copyright Notice

This presentation in this publication was presented at the tinyML® Summit (March 28 - 29, 2023). The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org