MLP301: Machine Learning Principles

Assessment 2: Machine Learning Application Proposal

David Lawler: A00075945

Submission Date: 06/11/2022

# Executive Summary

The summary covers the key sections of the report and details the learner’s motivation and inspiration for selecting the ML application.

This brief addresses the problem of optimising crop irrigation, and how it can be solved using tinyML. An AI managed crop irrigation system is proposed and its inputs, outputs and storage telemetry are explained. This system is a modified version of the Edge Impulse case study, and we will use the pre existing dataset supplied initially, and get additional data by letting the system run and then importing the data to Edge Impulse Studio as a CSV file from the SD card data.

The process of importing, training, testing and optimising data using a Kera Neural Network on Edge Impulse is explained in detail, as is improving the models performance and accuracy. A project plan, including costs are laid out for the continuation of this project in Assessment 3 of Machine Learning Practices.

The risks, challenges and benefits of embedded AI are explored, as well as potential licensing solutions to promote the responsible development and use of Artificial Intelligence for social good.

# ML Application and Architecture

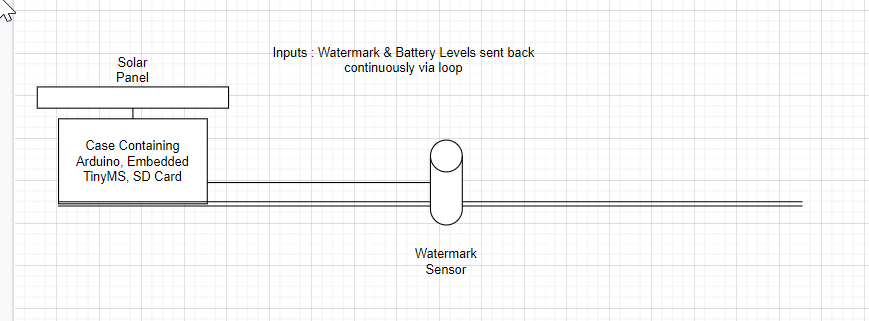
It is estimated that the world population will increase to approximately 9 billion by 2050. The Food and Agriculture Organisation estimates that by then food production must increase by around 60% if we want to ensure global food security. To answer this rising concern, Smart Farming technologies(also called Precision Agriculture) is reshaping agricultural practices(Chollet, 2022).

Through the use of pesticides, fertilisers, mechanisation, and lots or water, modern man has developed the ability to grow large quantities of food. However, these practices are often unsustainable, ruining the environment and depleting valuable resources.

The problem this report addresses is designing a complete AI managed system to automate and optimise crop irrigation using tinyML. This project is meant to tackle water usage, being able to pinpoint exactly when water is needed while preventing overwatering(Edge Impulse, 2022).

The AI managed crop irrigation system architecture consists of(Edge Impulse, 2022):

* An Arduino edge controller
* An Arm Cortex-M4F Processor (nRF52840) clocked at 64MHz with 1MB flash and 256KB RAM
* 16 hydrostatic watermark sensor inputs
* 8 latching relay command outputs with drivers and 8x without drivers
* 4 60V/2.5A solid state relays
* Dual MKR Sockets
* A 12V SLA battery that can be recharged directly from solar panels
* DHT22 temperature/ humidity sensor
* A Micro SD card



The AI managed crop irrigation system’s inputs, outputs and storage telemetry(Edge Impulse, 2022):

* On Arduino Edge Control boot up, peripherals such as the LCD, watermark sensor, and MKR slot 2 are initialised
* Then within the loop function, both the watermark and battery level values are read repeatedly
* Over on the MKR board there’s a handler for the I2C bus that gets called whenever new data is received. Once parsed back into a SensorValue object, this data is formed into a serialised JSON string which is stored on the Micro SD card.
* Micro SD card can be removed, data extracted and then stored within a CSV file for analysis and training.

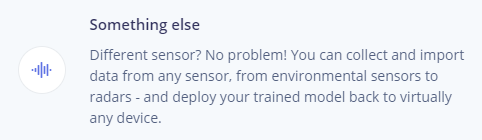
# Data Set and Acquisition Strategy

This system is a modified version of the Edge Impulse case study reference. I have removed the wifi signal and server in preference to local storage, and removed the DHT22 temperature/ humidity sensor, to reduce cost and simplify the project. This means that I can use the dataset supplied with the case study, but I will have to modify the dataset by removing two variables; temperature and humidity. The project is cloned here <https://studio.edgeimpulse.com/studio/152734>

Variables are:

* Timestamp (time offset in milliseconds)
* Watermark sensor reading
* Battery level (can be used to determine night or day)

(Sensor Data Link on Edge Impulse)



To get additional data will be a matter of letting the system run and then importing the data to Edge Impulse Studio as a CSV file from the SD card data.

# Model Training and Features

Consider how you intend to use ML. Some pre-trained models may already exist that

you can reuse (e.g., keyword spotting, running faucet or continuous gestures). You

are free to use whichever ML techniques are justified for your project, but you must

use Edge Impulse for the model training. Edge Impulse uses TensorFlow and Keras to

train your model in the cloud and deploy it on your smartphone or microcontroller.

Feature extraction from sensor data, including time series data (e.g., audio, accX, accY

and accZ) is a critical success factor to generate reliable ML models on a device.

A number of learning techniques can be used to classify new data learning patterns

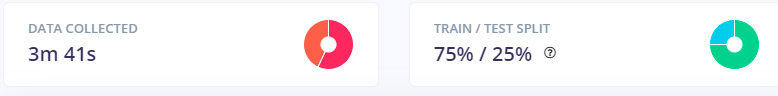
from data (neural networks), find outliers (K-means anomaly detection) and fine tune

pre-trained image models (transfer learning).

I intend on using the pre-trained model, using the three inputs; time, watermark level and battery level, the output will be simply needing water and not needing water. Time series data split into 1 second chunks will be used to train a Keras neural network. The AI-Managed Crops Irrigation study achieved a 100% accuracy in both model training and separate model validation. I doubt this will be the case here as we are using fewer inputs, but we will see.

The basic steps in this process are:

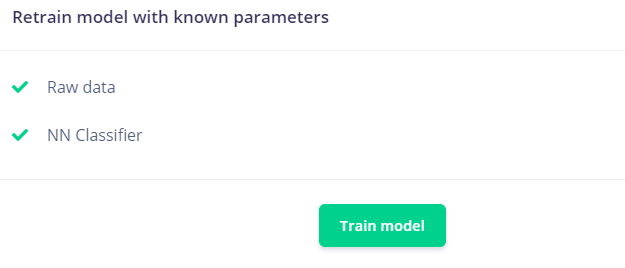
1. Modify case study time series data, removing humidity and temperature data.
2. Split Data set into 75/25% train/test split.



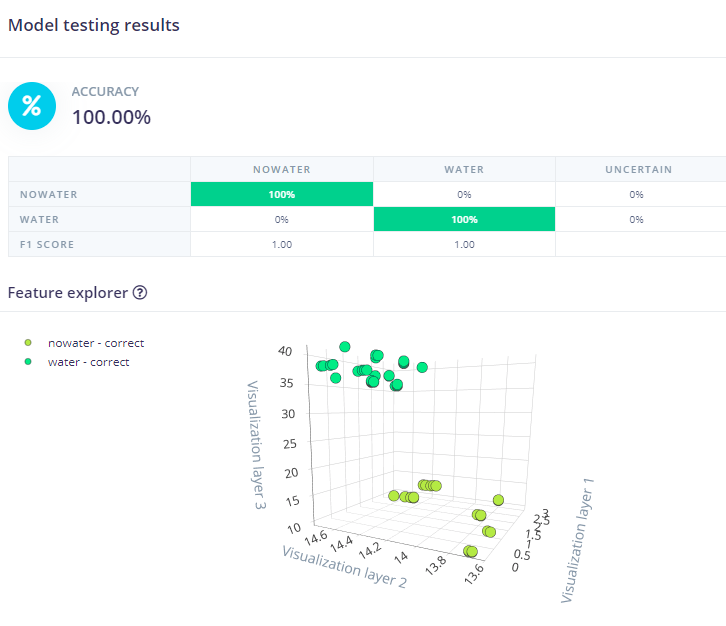
1. Design the Impulse

| We use Time series data with a 1000ms window. |  |
| --- | --- |
| Raw data includes the same variables(we remove temperature and humidity) |  |
| Keras Neural Network is selected as our classifier |  |
| Desired Output features |  |

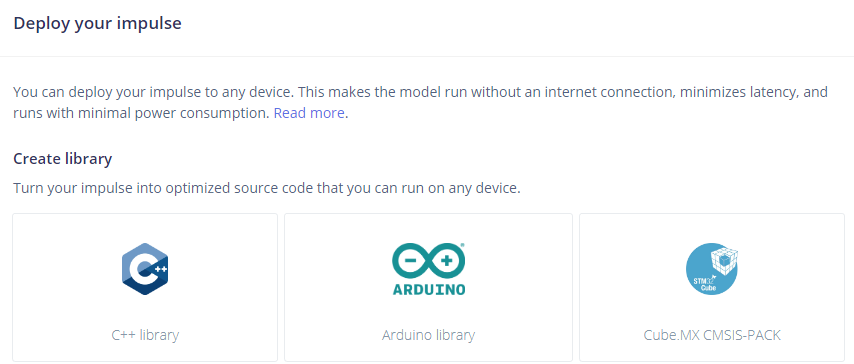
1. Train the Model. With edge impulse training the model is a one click process.



1. Test the Model. Next we classify the data, also a one click step with Edge Impulse. Note the Accuracy and F1 score



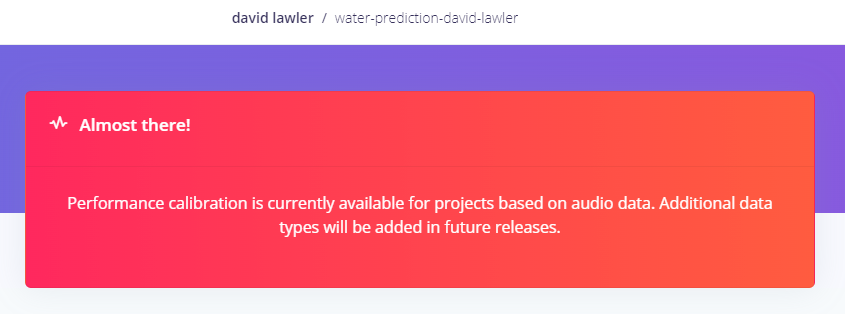
1. Deploy. Easy to do using export as an Arduino Library in Edge Impulse



1. Gather Data. Gathering Data is explained previously in Dataset and Acquisition
2. Repeat steps 4-7. This enables the AI to optimise as the dataset expands.

# Evaluation of Model Performance and Improving Accuracy

While there are currently no options for automatic performance calibration on Edge Impulse using sensor data, they will be added in the future according to the website.



Ways to increase performance of the model include:

* Acquiring more data. Neural networks need to learn patterns in data sets, and the more data the better.
* There may be a problem with dropping the humidity and temperature data from the dataset. If the data does not look like other data the network has seen before it can cause poor model performance. If this occurs in the test set or during live classification we can push the sample to the training set.
* Train the model more. Up the number of training cycles and see if performance increases. If there's no difference then you probably don't have enough data, or the data does not separate well enough.
* Overfitting. If you have a high accuracy on your neural network, but the model performs poorly on new data, then your model might be overfitting. Try adding more data, or reduce the learning rate.
* The neural network architecture may not be a great fit for your data. Playing with the number of layers and neurons may increase performance.
* Hardware issues. Processing power of the microcontroller or computer, and storage capacity of the SD card might need to be increased.

# Risks, Limitations and Advantages

Risks of embedded ML include:

* Wrong predictions or classifications being made: Models misbehaving could lead to devastating consequences without a safeguard(Like my crops dying!, or if implemented on a large scale; national food shortages).
* Security: Data that is embedded can be physically stolen from a device that may not be secure(a delivery drone getting its SD card with personal data on it stolen for example).

Limitations of embedded ML include:

* Limited memory: TinyML devices have kilobytes or megabytes of memory. This puts restrictions on the size and the runtime of the machine learning models deployed on these devices. Currently, there is a limited number of ML frameworks which can meet the requirements of TinyML devices. TensorFlow Lite is one such framework.
* Troubleshooting: Since the ML model trains on the data that the device collects and runs on the device itself, it is harder to determine and fix the performance issues than in a cloud setting where troubleshooting can be done remotely.

Advantages of embedded ML include:

* Fast inference with low latency: Since TinyML enables on-device analytics without the necessity of sending data to a server, edge devices can process data and provide inference with low latency.
* Data privacy: Keeping the data on the edge device reduces the risk of sensitive data being compromised.
* Doesn’t depend on connectivity: With TinyML, smart edge devices can make inferences without an internet connection.

# Project Plan

| Task ID | Description | Week |
| --- | --- | --- |
| 1 | Set Up Edge Impulse Developer Account | 9 |
| 2 | Order Parts for Device | 9 |
| 3 | Build Device | 10 |
| 4 | Export preexisting data onto device | 10 |
| 5 | Deploy device in garden | 10 |
| 6 | Sit in rocking chair on the back porch, glaring at device while it collects data. | 11 |
| 7 | Collect data, retrain, retest, analyse | 11 |
| 8 | Complete Assignment 3 | 11 |

# Investment

Prices from Amazon.com.au:

* Arduino Edge Control Board - $239
* Seed M4F Processor(nRF52840) - $25
* Gikfun DS18B20 Temperature Sensor Waterproof Digital Thermal Probe Sensor for Arduino (Pack of 5pcs) EK1083 - $12
* 4x 60V/2.5A solid state relays - $20
* 8x latching relay command outputs with drivers and 8x without drivers - $20
* 12V SLA battery - $35
* Small Solar Panel - $100

Total: $451

This has become quite expensive and I may look at alternative used purchases or altering the device to not require the Arduino Control Board or Solar Panel for the completion of assignment 3.

# Responsible Artificial Intelligence

Current best practices in responsible AI involve having an end user licence making the user responsible for their actions, and a responsible AI source code licence referenced at each block of source code. This is important to align the ML application to social good.

(Responsible AI licenses, 2022)

# 

# References

Bagnoli, M. (2021, January 5). TinyML: Making Smart Devices Tinier than Ever

Source. <https://www.plugandplaytechcenter.com/resources/tinyml-making-smart-devices-tinier-ever/>

Ihoume, I. & Tadili, R. & Arbaoui, N. & Benchrifa, M. & Idrissi, A. & Daoudi, M. (2022). Developing a multi-label tinyML machine learning model for an active and optimized greenhouse microclimate control from multivariate sensed data

Artificial Intelligence in Agriculture, Volume 6, Pages 129-137. <https://www.sciencedirect.com/science/article/pii/S2589721722000101>

Edge Impulse Case Study. (2022). AI-Managed Crops Irrigation

Source. <https://assets-global.website-files.com/618cdeef45d18e4ef2fd85f3/621cef64de63616986408c70_AI-Managed-Crops-Irrigation.pdf>

Project. <https://studio.edgeimpulse.com/public/34631/latest>

Chollet, N. & Bouchemal, N. & Ramdane-Cherif, A. (2022). TinyML Smart Sensor for Energy Saving in Internet of Things Precision Agriculture platform

ICUFN. <https://manuscriptlink-society-file.s3.amazonaws.com/kics/conference/icufn2022/abs/3B-4.pdf>

Responsible AI Licenses (2022).

Source: <https://www.licenses.ai/ai-licenses/>