# Predictive Model/Deep Neural Network Training

[Predictive Model/Deep Neural Network Training 1](#_Toc946463623)

[Data Collection 1](#_Toc1405136763)

[Python server manual collection 1](#_Toc2118854012)

[Python server and labelling Script date collating 2](#_Toc426283845)

[Environment Setup 4](#_Toc924507949)

[Recommended Setup - Anaconda Environment setup, Use Spyder to run Scripts 5](#_Toc2011558185)

[Step 1 – Download Anaconda 5](#_Toc1160288129)

[Step 2 – Open Anaconda Navigator 5](#_Toc217104729)

[Step 3 – Setup Environment 5](#_Toc609193962)

[Step 4 – Install Spyder 5](#_Toc269145390)

[Step 5 – Install Dependency's 6](#_Toc825678728)

[Step 6 – Test Run Script 7](#_Toc756431762)

[Model Training / Data formatting 7](#_Toc1050241058)

[Model integration into Python Server 8](#_Toc588649380)

## Data Collection

To Collect Data for use in training the predictive model or neural network there are 2 methods that can be used.

* Python server manual collection
* Python server and labelling Script date collating.

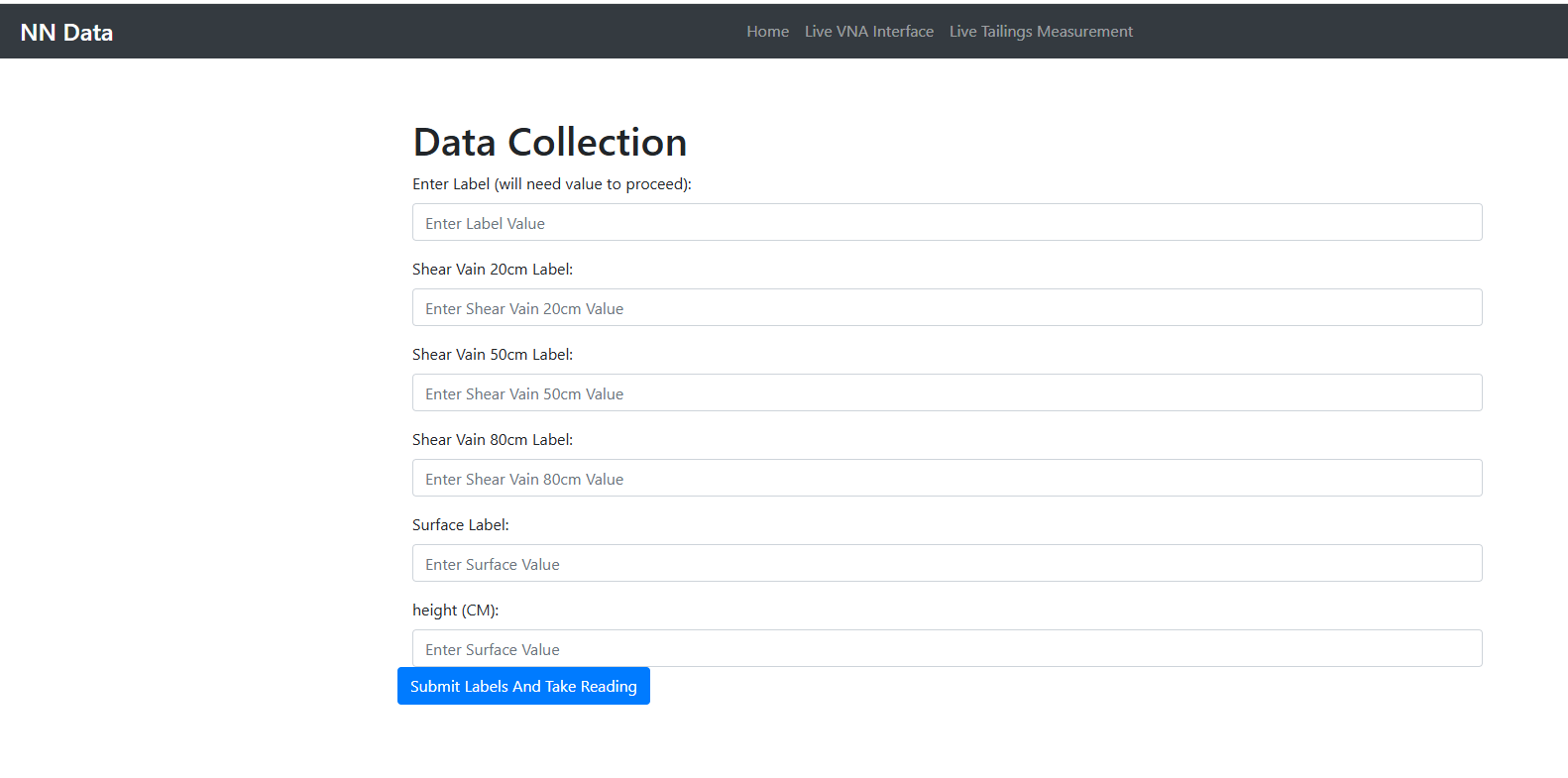
## Python server manual collection

The python Server or Sensor UI has been setup to collect the data in the correct format for the training script to use (should be in same folder as this document - Train\_DNN.py).

Note

* The data will be outputted to the location specified in
  + Megiq VNA Server\Python Server\mudmasterui\MudMasterUI\NN\_Data\_Collection\routes.py
  + At line 68   
    
  + You may need to change this file path.

Once that file path has been changed run the python server and navigate to

* <http://localhost:8080/NN_Data_Collect>
* 

Currently the Train\_DNN.py script only trains on the data in Enter Label (will need value to proceed): field. However the other labels are present for future development (recommend collecting this data as well).

## Python server and labelling Script data collating

The Python server and labelling Script date collating works in a similar way to the previous method however the labelling and recording of sensor readings occurs separately and is later combined using the time stamps on the data and label data.

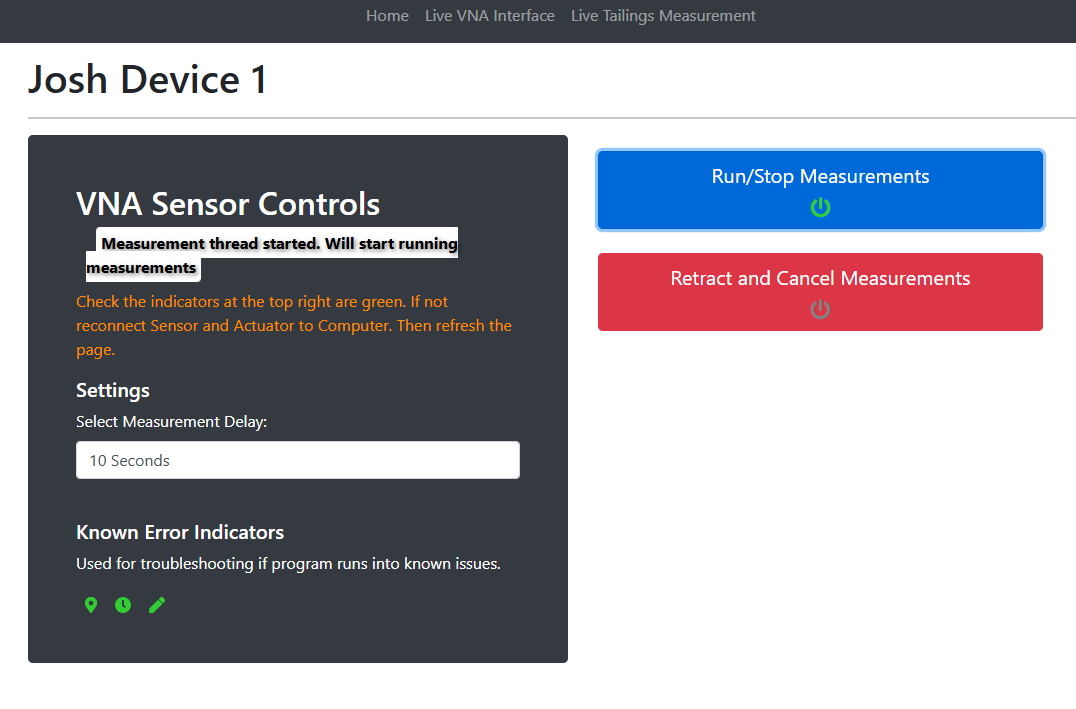
To use this method the first thing that will need to be done is to start the python server located. Refer to the transfer guide located in the ig88 documentation located

* \IG88 - General\03 Development\Dielectric Antenna\

Then once the server is running you should be able to access a web UI located at the web address

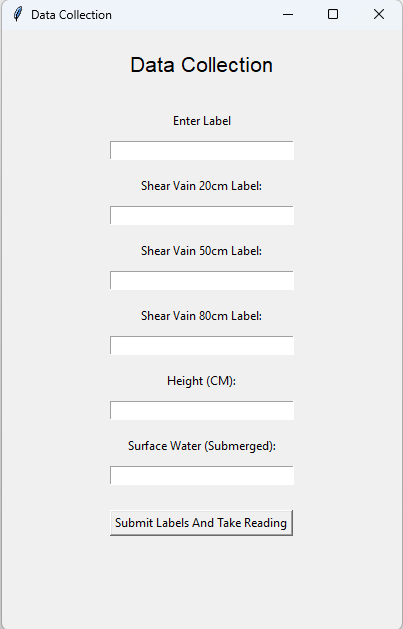
* <http://localhost:8080>

Once there press the Run/Stop Measurement. This will make it so the system will start to take measurements every 10 seconds.

Once the system is taking measurement you can run a separate program located at

* C:\Users\JoshuaPaterson\Phibion Pty Ltd\IG88 - General\03 Development\Dielectric Antenna\Predictive Model Training Guide and FIles\Data Labeling Scripts\NN\_Collect\_V2.py

This python script will open a UI like in the image below.

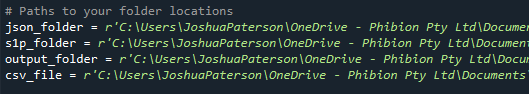


To use enter information from in the proximity (or directly below) the sensor unit into the fields and press the Submit Labels button. This will create a file with the label information with a timestamp to later correlate. This information should be stored in the Data folder within the same folder as the python script.

To then combine the data in the way the Provided Training Script (refer to Model Training / Data formatting Section) can use run the program S1P\_File\_Finder.py to merge the data.

Note that you will need to modify this script to point at the required data. The image below shows the variables that will need to be changed.

* Json\_folder
  + This variable is to be file path of the folder containing the data.json files outputted by the labelling script (NN\_Collect\_V2.py)
* S1p\_folder
  + This variable is to be the file path of the folder containing the S1P Files from the Python Server
    - Find the logs folder for the python server ->found in the VNA TouchStones folder -> Then look for the folder that has a timestamp of when you took the labels and sensor recordings -> then the vnaData folder is the folder with the S1P files.
* Output\_folder
  + This variable is to be the file path of the folder you wish to produce the new combined data.
* Csv\_file
  + This variable is to be the file path of the log file for the python server.
  + Find the logs folder for the python server
    - Should be at the highest level and will be a csv file.



## Environment Setup

Due to the VTC’s CPU not being AVX compatible a model must be trained on version 1.12 version of TensorFlow which will require Version 3.7 of python to function. There are also a few other library’s that must be a certain version which will be explained further on.

To set up an environment that can be used to train a model for the VTC’s I recommend downloading Anaconda and setting up a virtual environment with the required dependencies.

However you may use any method to set up an environment, what’s important is you have the correct dependency’s/ library's

Being

* numpy==1.21.6
* scipy==1.7.3
* scikit-rf==1.1.0
* flask==2.2.5
* Pyserial
  + – version doesn't matter
* psutil==6.1.0
* requests==2.31.0
* matplotlib==3.5.3
* tensorflow-1.12.0-cp37-cp37m-win\_amd64.whl
* protobuf==3.19.6
* h5py==2.10.0
* pip install scikit-learn
  + – version doesn't matter

### Recommended Setup - Anaconda Environment setup, Use Spyder to run Scripts

#### Step 1 – Download Anaconda

Download anaconda and run the installer (should be in the same folder as this document or go to <https://www.anaconda.com/download>) and download the installer there).

Use the default settings.

#### Step 2 – Open Anaconda Navigator

Open anaconda navigator.

Then open anaconda\_prompt

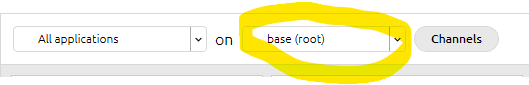
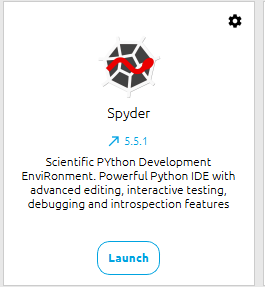
#### Step 3 – Setup Environment

Run the commands

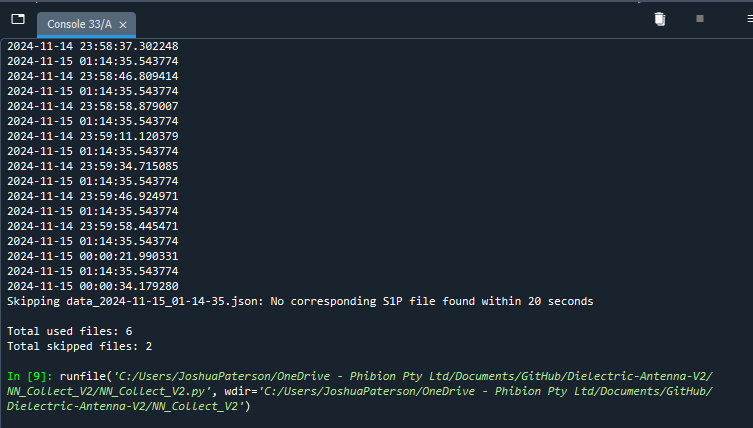
* conda update conda
* conda create --name TrainingEnv python=3.7.9
* conda activate TrainingEnv

Close prompt

#### Step 4 – Install Spyder

* Return to anaconda Navigator
* Check to make sure you're in the correct environment
* 
* Install then launch spyder
* 

#### Step 5 – Install Dependency's

* Once in spyder Find the Terminal (example in the below image)
* 
* Then run the following commands into it
  + pip install “/path/ tensorflow-1.12.0-cp37-cp37m-win\_amd64.whl”
    - The /path/ is the path to the tensorflow-1.12.0-cp37-cp37m-win\_amd64.whl file. One can be found at IG88 - General\03 Development\Dielectric Antenna\Predictive Model Training Guide and FIles
  + pip install numpy==1.21.6
  + pip install scipy==1.7.3
  + pip install scikit-rf==1.1.0
  + pip install flask==2.2.5
  + pip install pyserial
  + pip install psutil==6.1.0
  + pip install requests==2.31.0
  + pip install matplotlib==3.5.3
  + pip install tensorflow-1.12.0-cp37-cp37m-win\_amd64.whl
  + pip install protobuf==3.19.6
  + pip install h5py==2.10.0
  + pip install scikit-learn

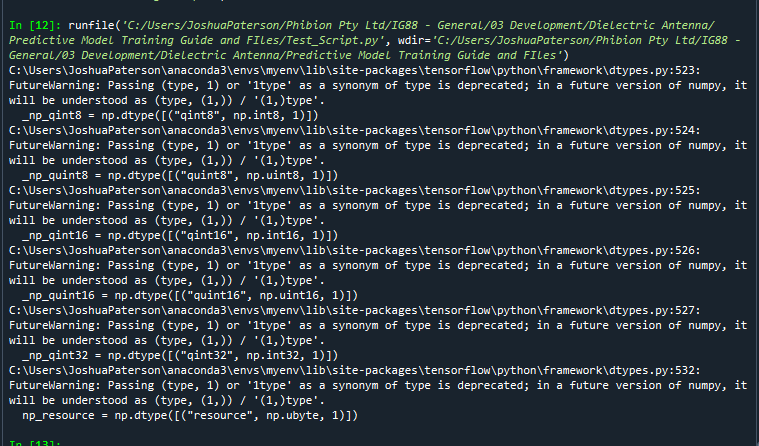
Note that you won't need all the library's installed to run the scripts for training how above are the library's that the VTC’s will have. This is don't to more accurately simulate the same environment as the VTC’s

#### Step 6 – Test Run Script

Find the test script which should be in the same folder as this document.

It's called Test\_Script.py.

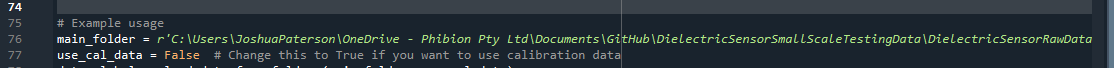
Open this file in spyder and run the script. (the desired output can be seen in the image below)



## Data Processing

## Model Training / Data formatting

Open or edit the Train\_DNN.py file. You will need to modify the lines

* Line 76
* Change the main\_folder variable to the file path of where the data taken from the NN\_Collector page is.
  + 
* Line 143 and 144
  + Change these lines to what you wish to name the model
    - Keep the .keras extension at the end of the name
  + A computer code with white text

    Description automatically generated

Run the python script

* + Note an example of how the data should look can be found in the folder Example\_Training\_Data which should be found in the same folder as this guide
* After running the script, you should see you file in the same folder as the Train\_DNN script.

## Model integration into Python Server

From the previous step or heading you should have produced a file with the .keras extension this is your trained model.

To integrated into the python server put the .keras file into the folder

* \Megiq VNA Server\Python Server\mudmasterui\MudMasterUI\nnData

Then go to the config.py file found at

* \Megiq VNA Server\Python Server\mudmasterui\MudMasterUI

Then alter line 12 of config.py or the DNN\_MODEL\_LOCATION to the location of the .keras folder

* 