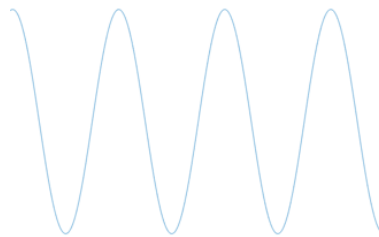


User Manual

Robin Mannberg, Martin Andersson, Emma Beskow, Ella Grundin,
Joel Nilsson, Gabriel Suihko and Jianxin Qu

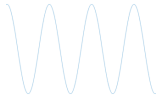
December 16, 2021

Version 1.0



Status

Reviewed	The project group	2021-12-23
Approved		



Project Identity

Group E-mail: tsks23group1@gmail.com

Orderer: Danyo Danev
Phone: +46 (0)13 28 13 35
E-mail: danyo.danev@liu.se

Customer: Danyo Danev
Phone: +46 (0)13-28 13 35
E-mail: danyo.danev@liu.se

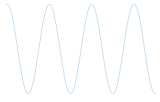
Supervisor: Ema Becirovic
Phone: +46 (0)13-28 19 11
E-mail: ema.becirovic@liu.se

Supervisor: Jianan Bai
Phone: +46 (0)13-28 26 13
E-mail: jianan.bai@liu.se

Course Responsible: Danyo Danev
Phone: +46 (0)13-28 13 35
E-mail: danyo.danev@liu.se

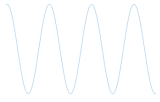
Participants of the group

Name	Role	E-mail
Robin Mannberg	Project Manager	robma370@student.liu.se
Martin Andersson	Test Manager	maran594@student.liu.se
Emma Beskow	Document Manager	emmbe571@student.liu.se
Ella Grundin	Hardware Manager	ellgr825@student.liu.se
Joel Nilsson	Chief of Design	joeni078@student.liu.se
Gabriel Suihko	Graphics Manager	gabsu290@student.liu.se
Jianxin Qu	Software Manager	jiaqu952@student.liu.se



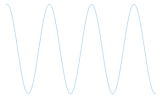
CONTENTS

1	Introduction	1
2	Getting Started	2
3	Setting Up the Environment	6
4	Collecting Data	7
5	Model Creation	10
6	Live Classification	15



DOCUMENT HISTORY

Version	Date	Changes made	Sign	Reviewer
1.0	2021-12-16	First version.	The project group	Supervisors
0.2	2021-12-16	Cosmetically changes and some spelling errors.	The project group	Supervisors
0.1	2021-12-16	First draft.	The project group	Supervisors



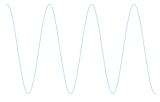
1 INTRODUCTION

In this manual for our detection system, you'll get easy-to-understand instructions on getting started with our product. Here we will describe what you as a user need to do to install the software needed, on how to set up a proper environment, and how to use and navigate the Graphical User Interface (GUI).

In addition to providing instructions on how to use the system, there are also some recommendation and examples for how to do your first experiment.

The system and product is designed to be trained in the environment it will later detect in. Some functionality and detection can be done in other environments, but it is in the trained environment the performance is guaranteed.

Let's get started!



2 GETTING STARTED

Before we can use the Pluto SDR devices, we first need to download and install some drivers. Depending on your operating system, there are different actions you need to take. We will describe the procedure for Windows-users here, but a full guide for Linux-, and Mac OS-users can be found here:

https://wiki.analog.com/university/tools/pluto/users/quick_start.

Step 1 - Downloading Python 3.9.X

The first step we need to do is to make sure that we have Python 3.9 installed. If you already have Python on your computer, you can open a command window and type `python --version` to check which version you have.

If you have an older version, you can download Python 3.9 here: <https://www.python.org/downloads/windows/>

Step 2 - Installing Drivers

The next thing we need to do is download and install the drivers for our Pluto-SDR device.

The next step is to install the IIO-scope on our host-computer. Now you need to download **libiio-0.23.gc14a0f8-Windows-setup.exe** from <https://github.com/analogdevicesinc/libiio/releases>, and run it from the terminal using `pip install pylibiio==0.23.1` followed by `pip install pyadi-iio` to finish the installation.

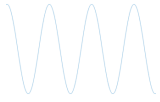
Step 3 - Required Python-packages

Now that we have downloaded the drivers for the Pluto SDR devices, we need to download the Python packages for the GUI and the software. Beside the standard packages in Python, you need to add the following packages needed from the list below, and the simple way to install these are to use: `pip install name` to add them to your library.

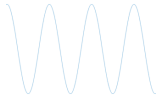
1. sklearn
2. pandas
3. numpy
4. scipy
5. hmmlearn
6. tkinter
7. matplotlib
8. seaborn

Step 4 - Connecting the Devices

In this step we are going to connect our four devices to our host-computer by doing the following steps:



1. **Step A:** Connect the devices to your computers USB-ports. If your computer does not have four (4) USB-ports, you need to use a USB docking station. Attach the cable to the device using the socket marked with a “3” in the Figure 1. Each device has a unique IP-adress, and for the transmitter to work correctly, it has to have the IP-adress: 192.168.1.1. The receivers needs to have the IP: 192.168.2.1, 192.168.3.1, 192.168.4.1 for each device respectively.
2. **Step B:** Attach appropriate antennas to each device according to their role. For the Transmitter, you attach the antenna to the spot marked “Tx” (number 1 in Figure 1, and the remaining antennas on “Rx” for the receivers (marked with a “2” in Figure 1). The antennas can be of various length and it is encouraged to try different antennas with different cable length to reach desired result and antenna placement. More about this in Section 3.
3. **Step C:** Connect the devices (or the USB docking station) to a power source using the socket marked with a 4 in Figure 1. If you’ve done everything correctly up until now, the lights “LED1” and “Ready” should be flashing and you should be ready to go.



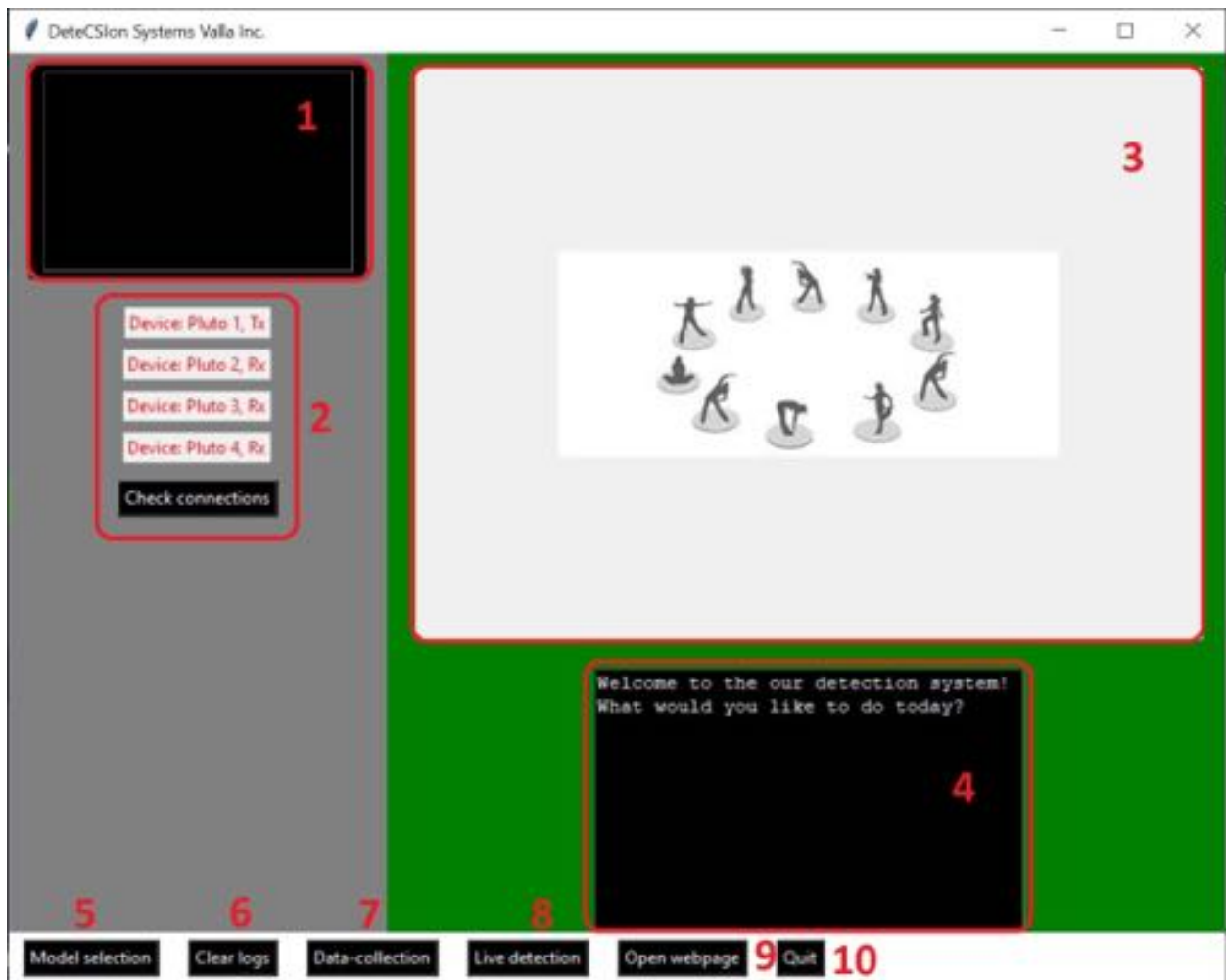
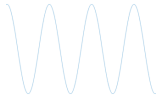
The Pluto SDR device.

Figure 1:

1. The connection for the transmitting antenna.
2. The connection for the receiving antenna.
3. USB-connection socket
4. The power socket
5. LED-lights indicating if the device is on and ready.

Step 5 - Booting up the User Interface

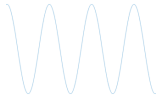
If you have done all the steps up until now, we are ready to boot up the GUI. To start the interface, in the command prompt, move to the product directory by entering `cd tsks23_project/src/gui` and then enter `py GUI3.py`. From this interface, we will control the program and start our detection journey. We can see the front page of the GUI in Figure 2. We can familiarise ourselves with the GUI now, but before we start doing any work, we first need to prepare our environment.



The front-page of the GUI.

Figure 2:

- 1. Device status log.
- 2. Status of Pluto devices.
- 3. Graphs are presented here.
- 4. System message log.
- 5. Model selection button.
- 6. Button to clear the logs.
- 7. Button to start a data-collection.
- 8. Button to start live classification.
- 9. Opens our website.
- 10. Turns off the program.

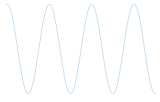


3 SETTING UP THE ENVIRONMENT

When we have installed our drivers, we need to setup our environment. The system consists of one (1) transmitter device (marked as Tx), and three (3) receiver devices (marked as Rx). It is recommended to space the receivers out in the wanted area, with the transmitter being placed in a somewhat open space (like you would a Wi-Fi router). An example of this can be seen in Figure 3.



Figure 3: An example of an environment using extra long cables for better antenna placement.



4 COLLECTING DATA

Now that we have installed the libraries needed for our Pluto SDR devices, and set up a proper environment, we can start collecting data.

In the GUI, click the button “Data-collection”, and a new window will open. Now you’ll have a choice, if this is your first time using the system, you need to click the button called “Create Dataset” (button 1 in Figure 4) and the system will automatically create a new data set folder. If you already have a data set that you wish to continue collecting data for, use the “Select dataset”, select your desired data set which is located in `tsks23_project/test_data/DS#` and press the “Next” button.

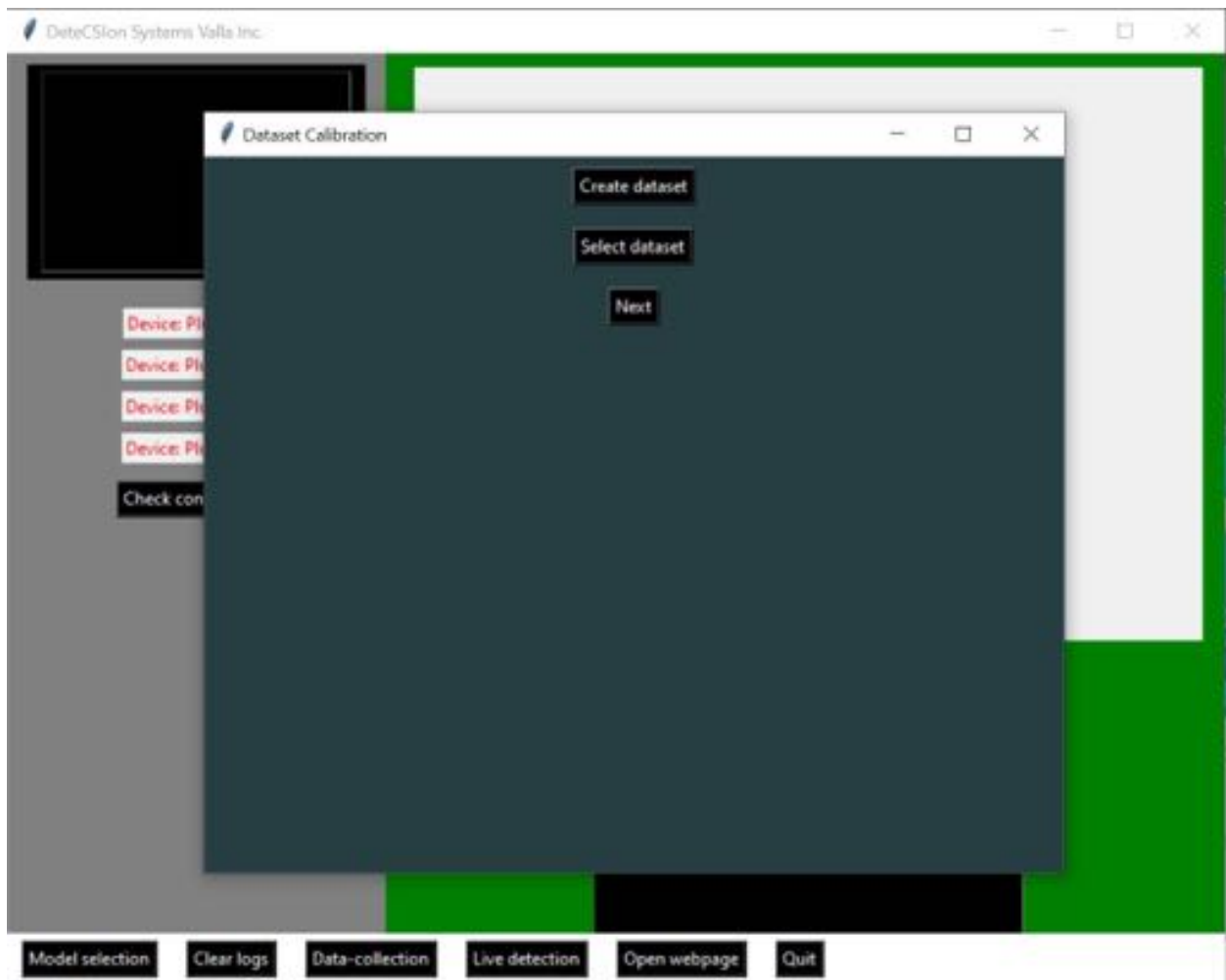
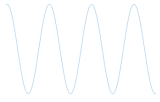


Figure 4: Here we can either create a new data set or select an already existing one.



Now you'll be able to select and customize the receivers for your data-collection. In Figure 5, the default parameter values are the recommended settings, but you are free to customize to your hearts content.

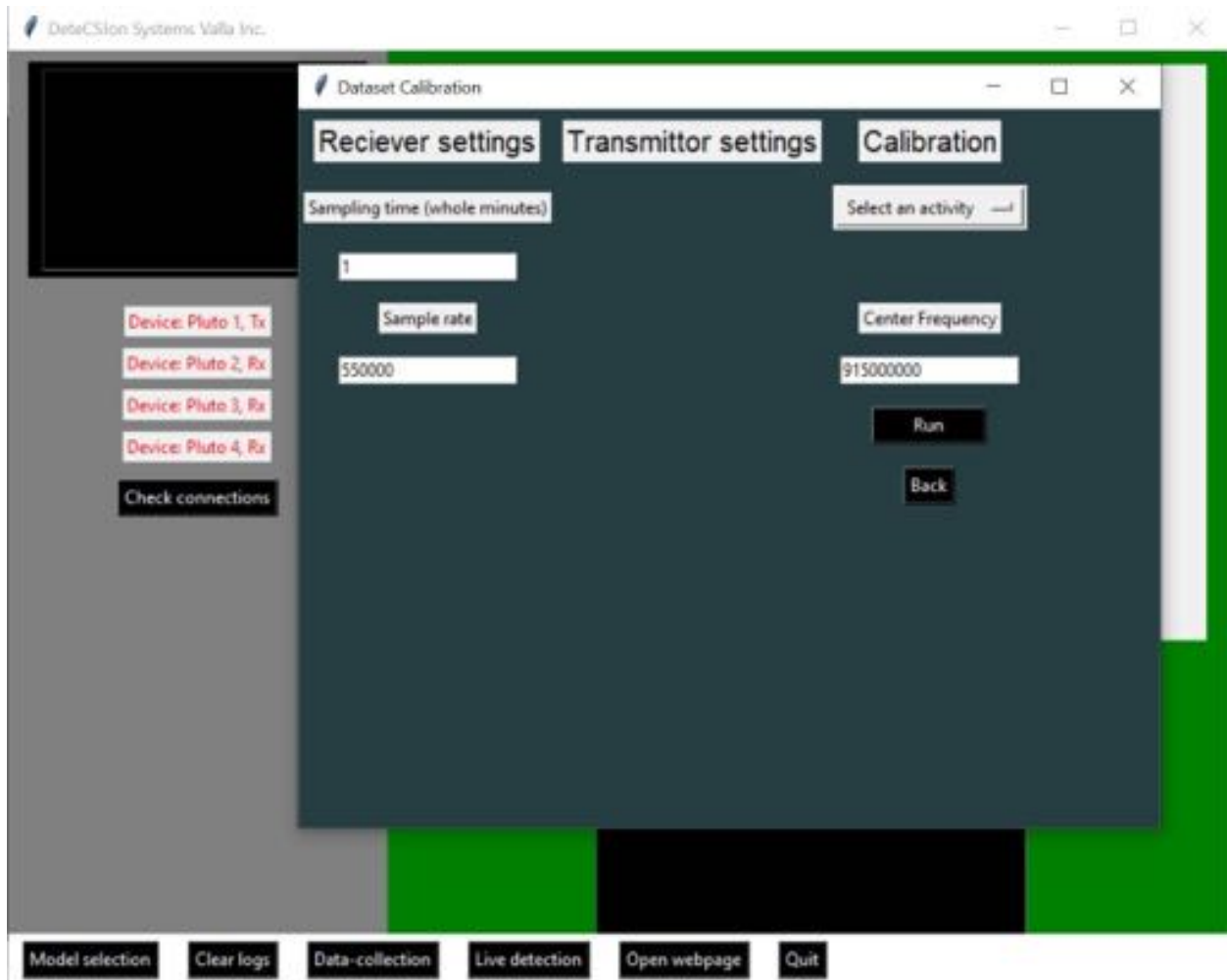
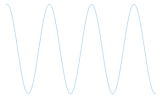
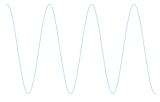


Figure 5: Settings window for data-collection

1. **Sampling time:** How many minutes you want to run the data collection.
2. **Sample rate:** How many samples per second, can be between 200 kHz - 20 MHz.
3. **Select an activity:** Select which type of activity you want to collect data for: static, balloon, walking, jumping, and dancing.
4. **Center Frequency:** Set the transmitted waves frequency. The Pluto devices allows this center frequency to be between 300 MHz - 3.8 GHz.



Once an activity has been selected, you can start collecting data by pressing the “Run” button. Once the collection has been completed, you can change the activity (or keep your choice) and collect again.



5 MODEL CREATION

After you have created and collected one (or several) data sets, you can now create a model. To create your first model, you first need to select the button “Model selection” on the front page and then in the consequent window, press the button called “Create new model” (see Figure 6). If you’ve already created a model and want to re-use one, you select the button “Choose existing model” and select the model in the path `tsks23_project/test_data/features_#`.

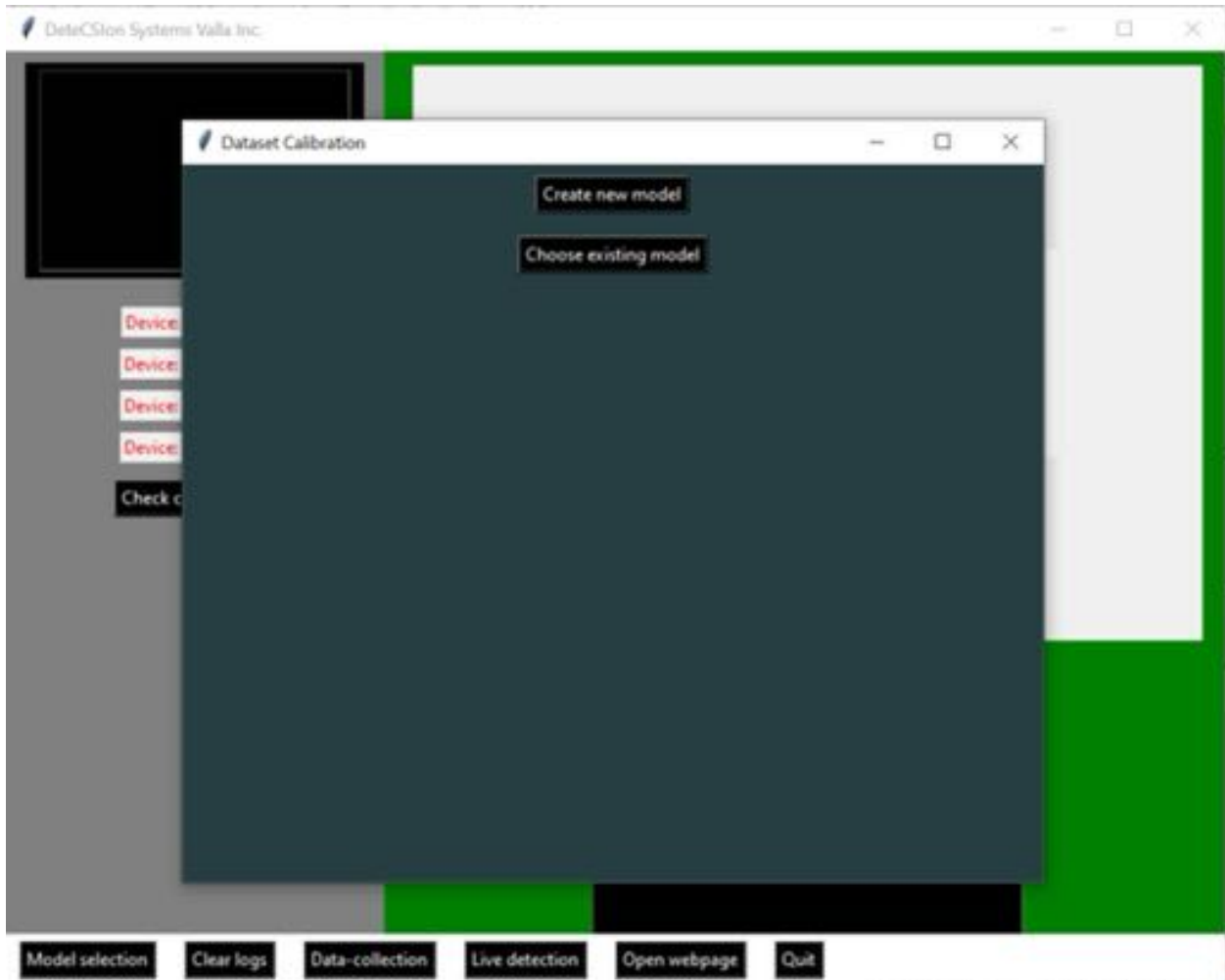


Figure 6: Here you can choose between creating a new model or choose and existing one.

The next step now is to select which one (1) or several data sets you want to use for the model. By clicking the button “Select data sets” in Figure 7, you will be given an option to select which existing data sets to use.

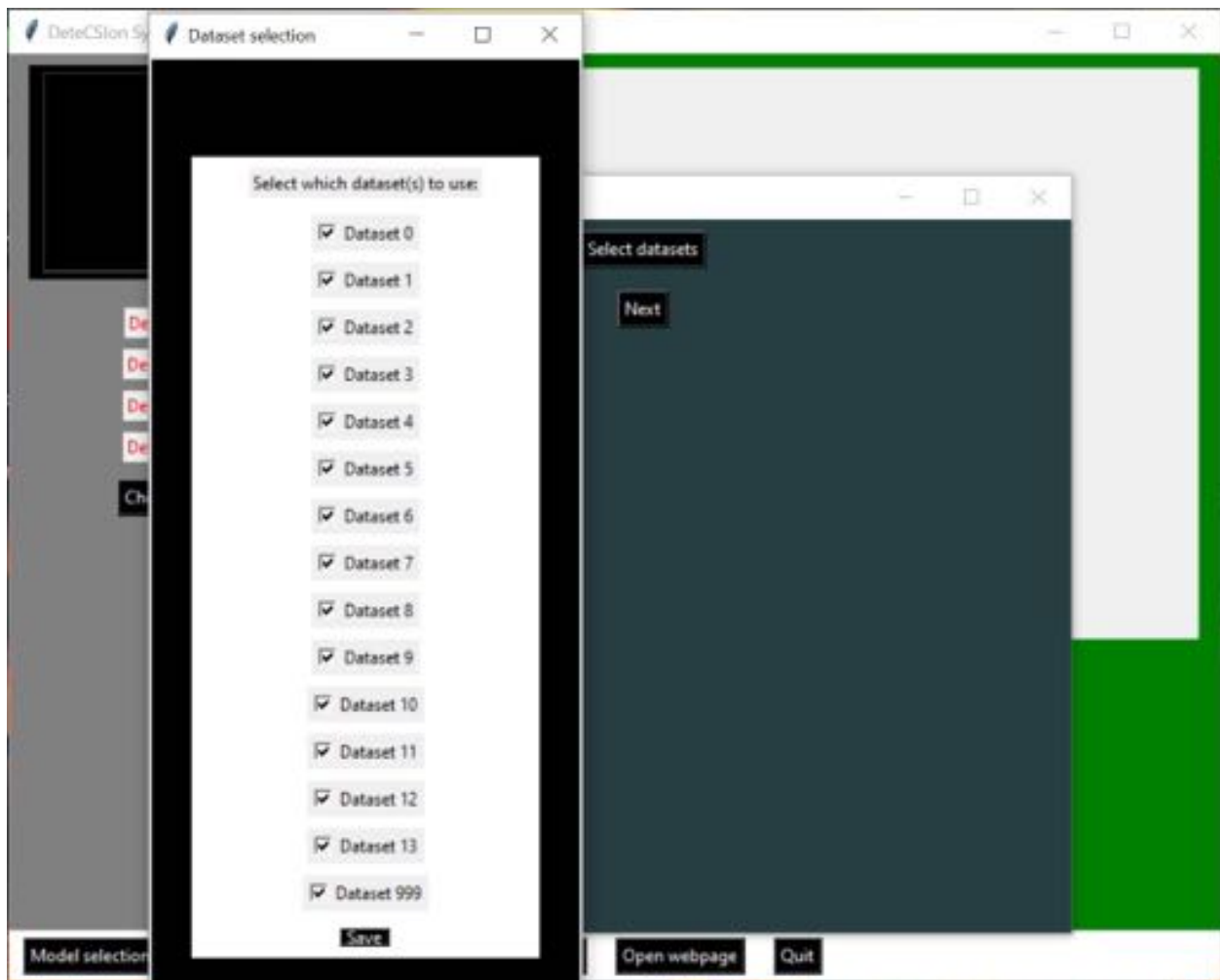
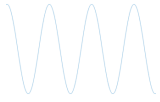


Figure 7: Here you can choose which (or all) data sets you want to use.

Now you'll be given several choices for your model, as seen in Figure 8, there are several parameters for you to choose from and customise. These are:

1. **Activities:** Which activities did you collect data for. These activities have to be within the selected data sets. **OBS!** You need to select at least two (2) activities here, one of which has to be static.
2. **Features:** From this list you can select a single or several features to train your model on. Each combination will yield different results.
3. **Training data ratio:** How much of the collected data you want to use for training. Recommended is 70%.

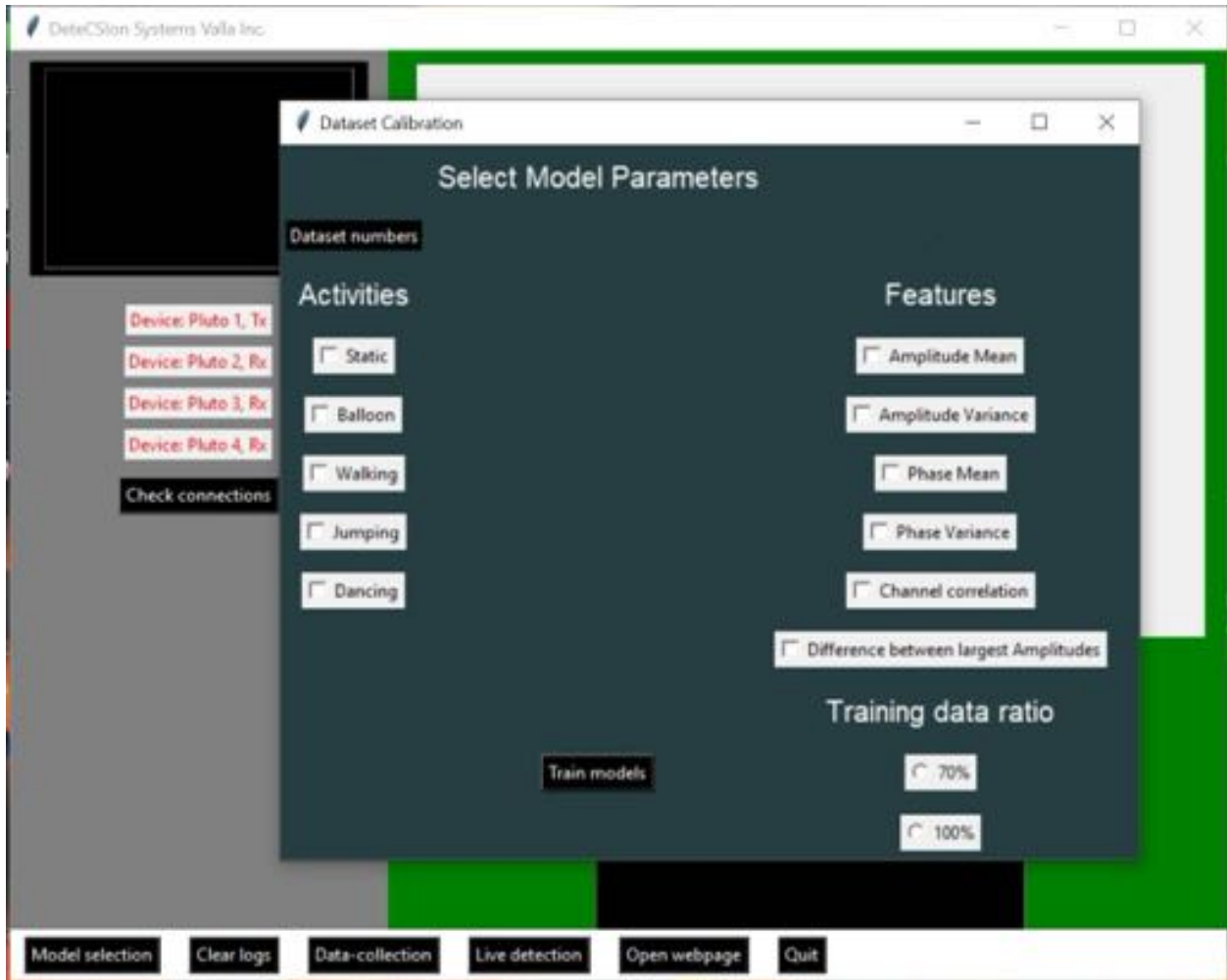
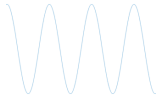


Figure 8: Settings window for your brand new model.

After we have now trained our model, we can finally see the result on how the model performed on the test data! The result will be shown in a plotted figure which can be found in the path `tsks23_project/test_data/plots`. The result could look like this (Figure 9):

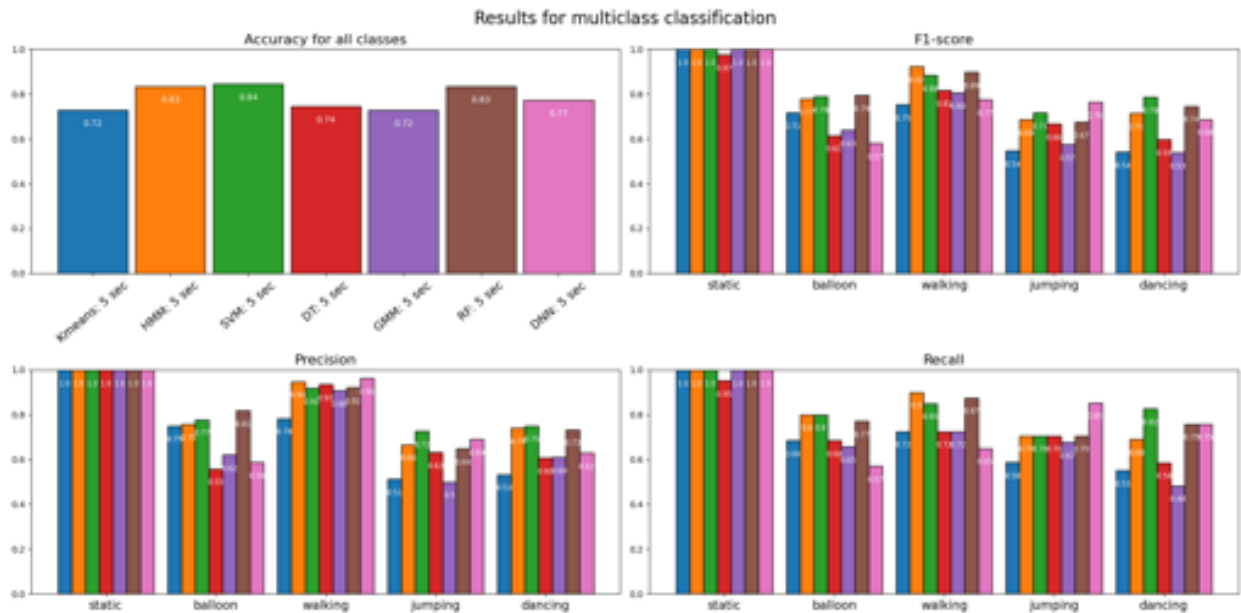
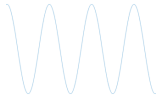


Figure 9: An example of how the result could look like.

The four (4) sub-graphs represented each show:

1. **Accuracy for all classes:** Shows how accurate each Machine-learning model was on the test data. Takes the total amount of correct prediction divided by the amount of total predictions.
2. **Precision:** Can be interpreted as the rate of how many classification that as correct for one class in relationship to the number of samples for this class.
3. **Recall:** Recall is the rate of how many classifications that was correct for one class in relationship to how many samples that got classified as this class.
4. **F1-score:** The harmonic mean between the precision and recall is called F1-score.

A detailed explanation of these measurements can be found in the Technical Report and/or the Design Specification.

Once this has been completed, you'll be seeing the final window in the model creation path, which can be seen in Figure 10. Here you can select you newly created model again or select an existing one, and then click the button "predict" to test the model again.

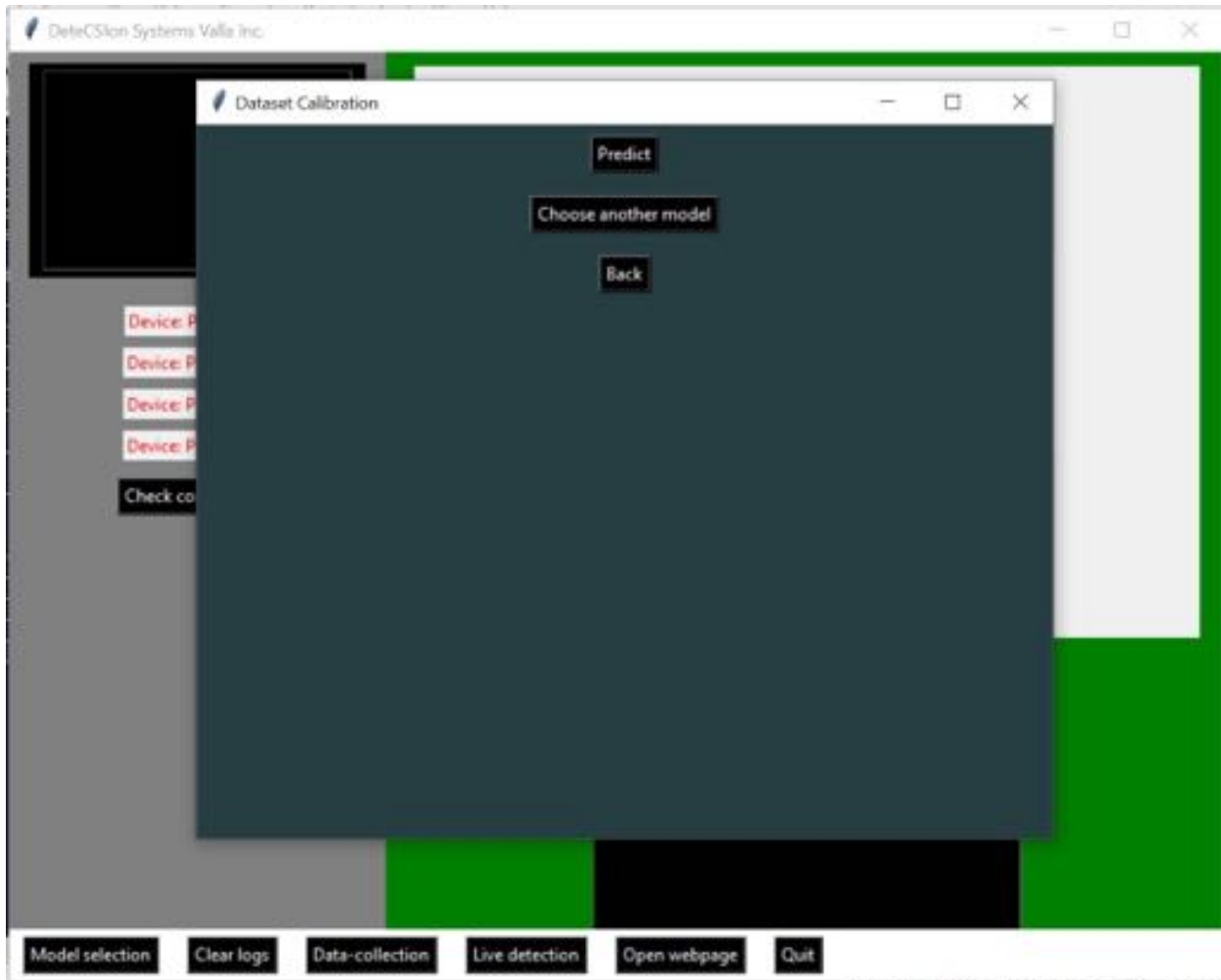
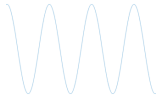
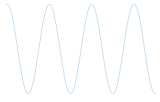


Figure 10: Here we can select our model and predict again, or select an already created model and predict that one.



6 LIVE CLASSIFICATION

At last, now that we have collected data, from which we have created our brand new detection model, we can finally start our live classification. In Figure 11, we can select an existing model, or create a new model using our recommendation by clicking the button “Create model using dataset 999”, and then selecting the newly created model. We recommend using the default settings, but you can alter the amount of iterations the program runs freely. The program will collect data for five (5) seconds, and then give a prediction on the activity in the room. The live prediction will tell you which of the available classes in the data set that most likely occurred in the room. Performance can not be guaranteed if many different activities are used in the data set, but discrimination between static and dynamics event will be good if used in the training environment.

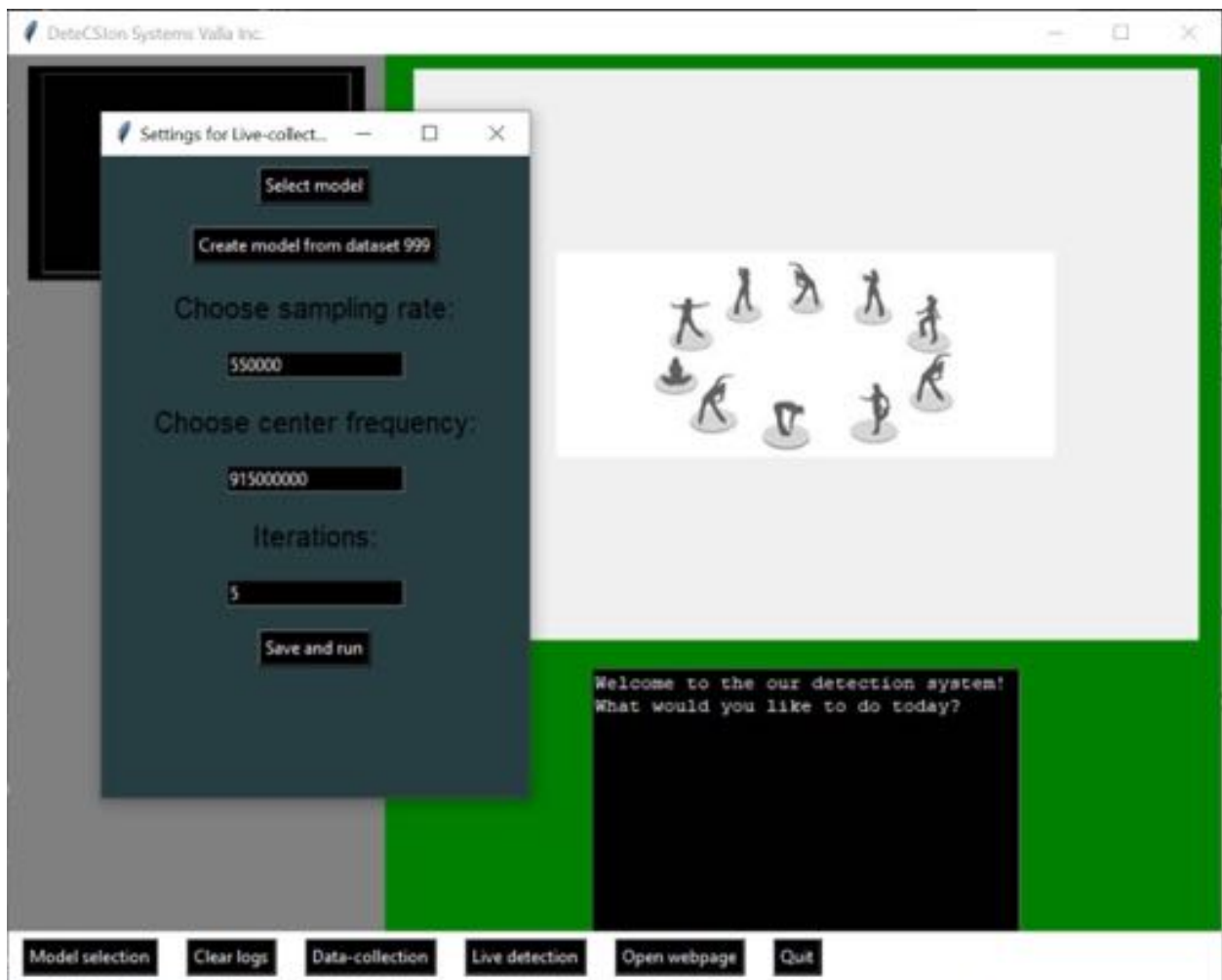


Figure 11: Preparing for the live classifications.