University of Toronto

Faculty of Applied Science and Engineering

Summer 2021 Research Project

Enhancing First Responders' Safety in Fire Incidents using an Interactive Hazardous Material Detection Platform

Team Fire Safety

Authors: Zijian Wang, Huifeng Wu, Cathy Feng, Sophie Tian

Supervisor: Prof. Chi-Guhn Lee

Aug 23rd, 2021

Table of Content

1. Background

During a freight fire emergency, proper detection and respondence to unknown chemicals can help control the spread of a fire and prevent potential explosions. Failing to identify the hazardous materials in a timely manner might obstruct or even mislead the use of the correct fire extinguishers for the first responders, resulting in heavy casualties and economic losses. Thus, in fire incidents, it is crucial to shorten the decision-making time required to identify the emergency with an interactive hazardous-material detection platform, and that the overall damage could be minimized.

2. Objective

In collaboration with the National Research Council of Canada (NRC), we look forward to developing a multifunctional interface powered by a decision-making model. The designed interface is considered as an aid for first responders in freight fire incidents. Functional requirements include showing the detected materials precisely and recommending first responders with the corresponding emergency response guides and strategies.

Given combustion signatures collected in a device named Fourier Transform Infrared Spectroscopy (FTIR), the designed, the user interface detects hazards such as corrosiveness and toxicity of the burning material and shows the appropriate response strategy from (Emergency Response) ERG handbook

In terms of metric units, the platform should be user-friendly and easy-to-understand for first responders who are not familiar with combustion processes. The final product should fulfil aspects of satisfaction, usability, and efficiency. The average time to navigate through the end-to-end process may take up to only seconds by users.

3. Workflow Overview

The scope of the summer project is the last step which includes the design of the platform.

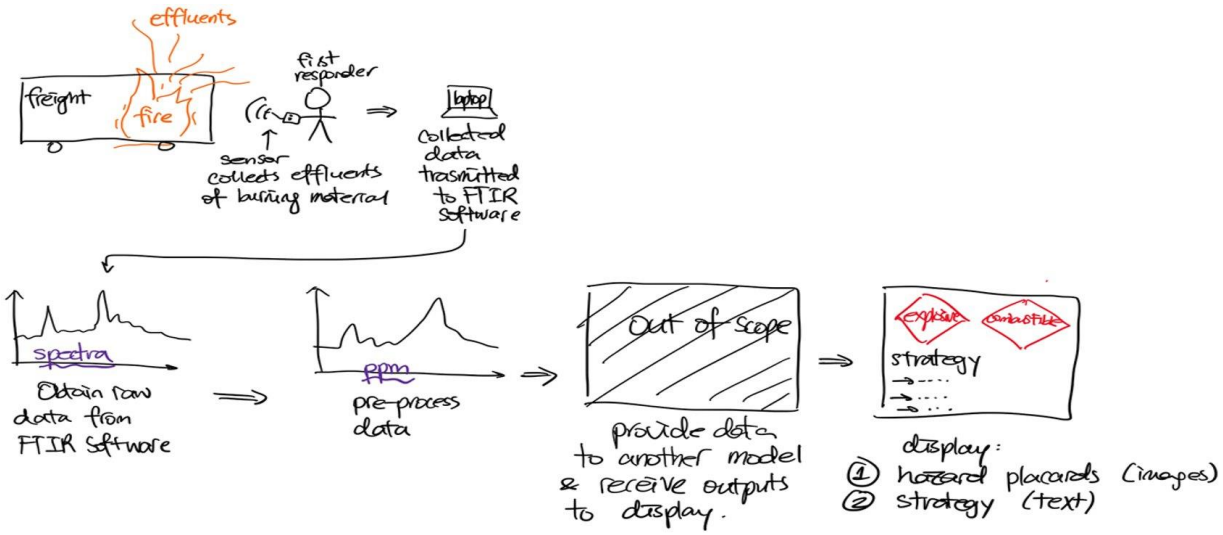

Figure 3.1 Workflow Simulation of Data

4. Technical Summary

One of the software development platforms for this project is Qt Creator. It was created to be a Qt Widgets Application which contains a main window and a second dialog, both written as QWidget class.

We had the main window to introduce the name of the software as well as prompting the user to execute the prediction model. The prediction model is written in python with a machine learning library called sktime. To successfully run the model, users should download all dependencies in the folder of their python interpreter. A virtual environment is allowed and recommended. Once finished, the user should key in the path to their python interpreter manually into the function called void MainWindow::on_pushButton_clicked() of mainwindow.cpp. When clicking the button “Run Model” on the main window interface, the python script starts to execute and eventually creates two images, “dataframe.png” and “predictionResult.png” as outputs. This process usually takes around one minute.

Prompts will pop out to inform the user when the data is ready. From there, the user can go to the second dialog through the button at the bottom, and they will see the outputs loaded on the interface.

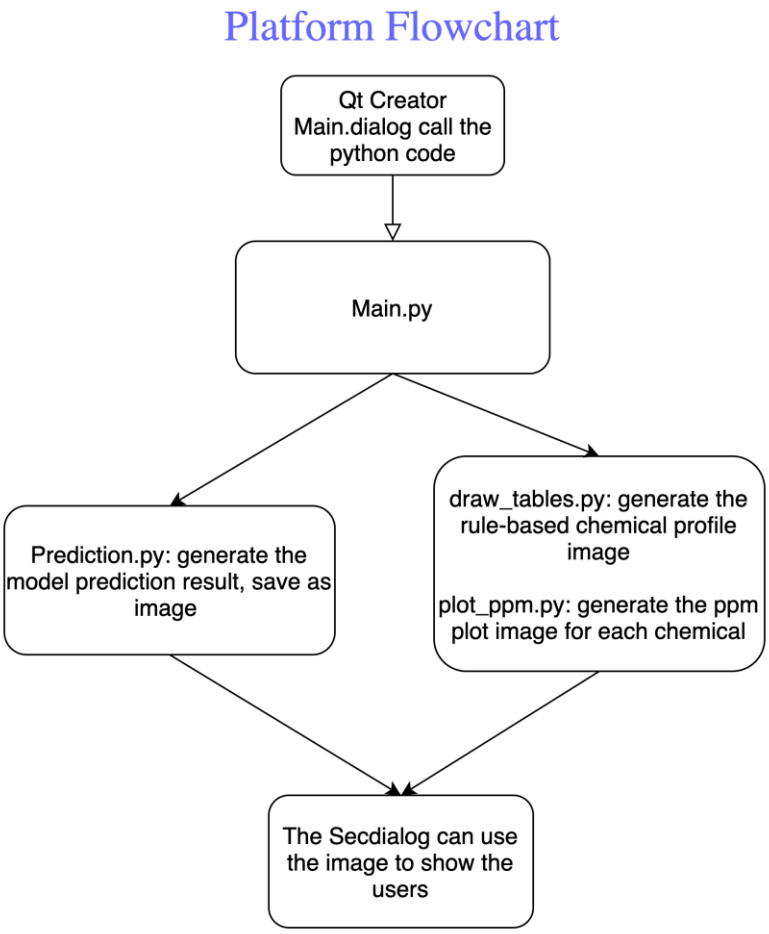

Figure 4.0 Platform Structure and Data Flowchart

4.1 Plot functions (draw_tables.py and plot_ppm.py)

Located at: https://github.com/CathyF9600/Fire_Safety/blob/main/scripts/plot_ppm.py

Located at: https://github.com/CathyF9600/Fire_Safety/blob/main/scripts/draw_tables.py

plot_ppm.py is a function that firstly filters negative values in detected chemical measurements, and it reshapes the time axis with 4 seconds intervals. Selecting desired chemical categories, we then plot its time series with matplotlib. Figure X is an example of the ppm plot for acetylene.

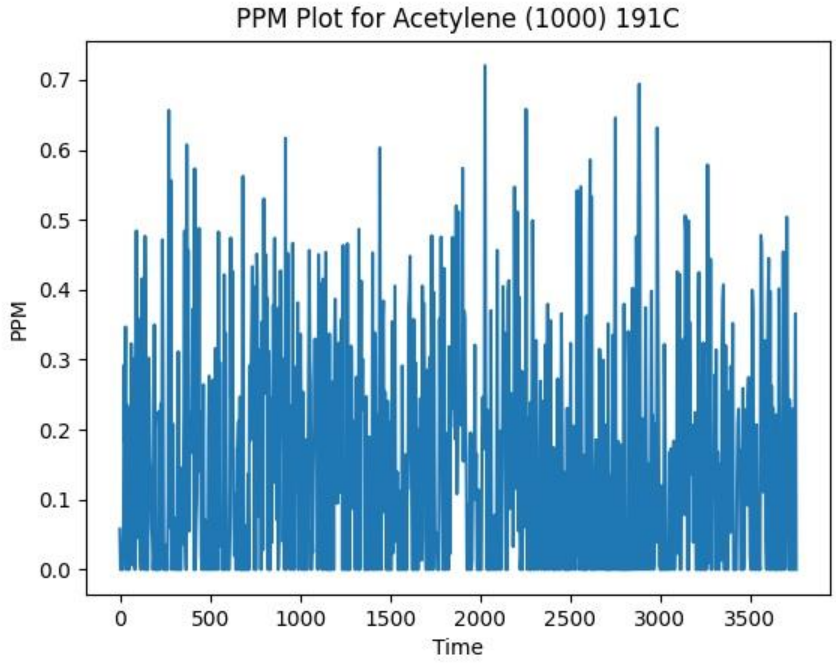

Figure 4.1.2 Chemical detected PPM Time-Series Plot

draw_table.py is a function for generating an image of rule-based knowledge to our UI. It considers the chemical varieties that appeared in a .prn file, solely preserving the existing ones in this experiment to a table. Each gas will respond “yes” or “no” to labels, such as flammability,

toxicity /health hazard, explosivity, water reactivity (solubility), corrosive and oxidizing properties.

	Chemicals	flammibality	toxicity /health hazard	explosivity	water reactivity (solubility)	corrosive	oxidizing
0	CO2	no	no	no	no	no	no
1	CO	yes	yes	no	no	no	no
2	benzene	yes	yes	no	no	no	no
5	ethylene	yes	no	yes	no	no	no
6	ethane	yes	no	yes	no	yes	yes
11	NO	no	yes	no	no	yes	yes
12	N2O	no	no	no	no	no	yes
13	NO2	no	yes	no	no	yes	yes
19	acetylene	yes	no	no	no	no	no
20	propane	yes	no	yes	no	no	no

Figure 4.1.2 Chemicals Profile

4.2 Prediction Model (Prediction.py)

Located at:

https://github.com/CathyF9600/Fire_Safety/blob/main/PredictionModel/Prediction.py

This function contains the core prediction model that is used. The input is the .prn file, the first step is to transform the data to a .csv file. Then for each prediction category (flammability', 'toxicity', 'explosivity', 'water extinguishable', 'corrosive', 'oxidizing'), the model predicts Yes/No. Finally, the model saves the result as an image (See Figure 4.2).

	flammability	toxicity	explosivity	water extinguishable	corrosive	oxidizing
Prediction Result	no	no	no	no	no	no

Figure 4.2 Model Prediction Result

4.2.1 Result Explanation

The chemical profile image contains the properties of the chemicals detected in the environment. However, the prediction model predicts the overall effect that includes the mixture of chemicals in burning situations. That's the reason why the properties are different.

4.3 UI (Main Window, Second Window)

To start the project view, the developer should open the .pro file through Qt Creator (Figure 4.3.1). From the project view, the files related to Main Window are mainwindow.h, mainwindow.cpp, and mainwindow.ui. In particular, mainwindow.h is the header file that contains all the variables and functions in mainwindow.cpp. And mainwindow.ui is the Qt Designer Form that serves as a control panel for the developers to design the user interface. Similarly, the files related to Second Window are secdialog.h, secdialog.cpp, and secdialog.ui. And main.cpp in Sources folder is the overall file for the user to start the application and have the file shown.

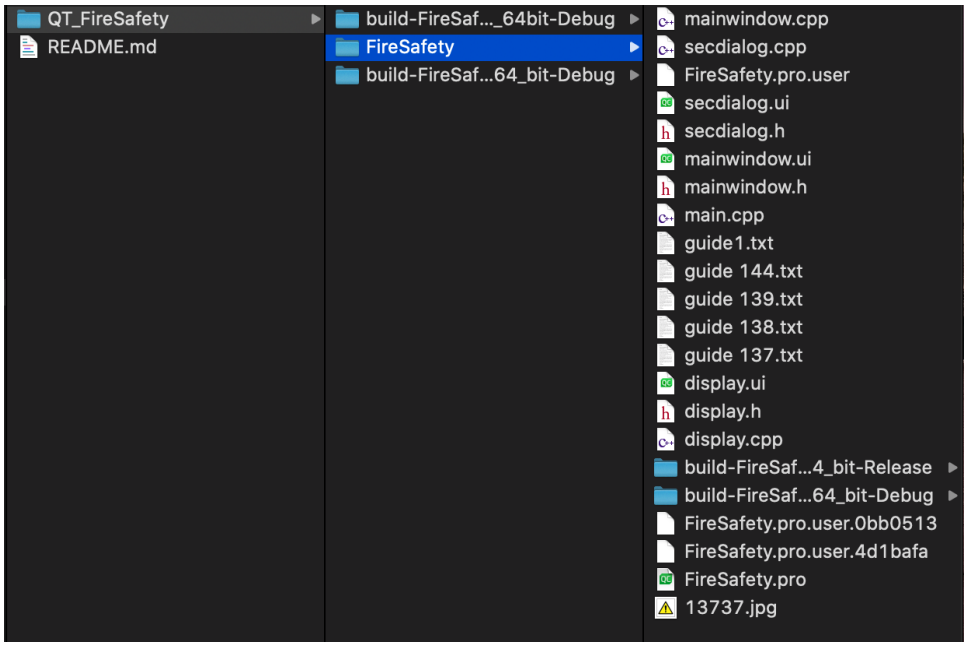

Figure 4.3.1 The Qt Program in File Explorer

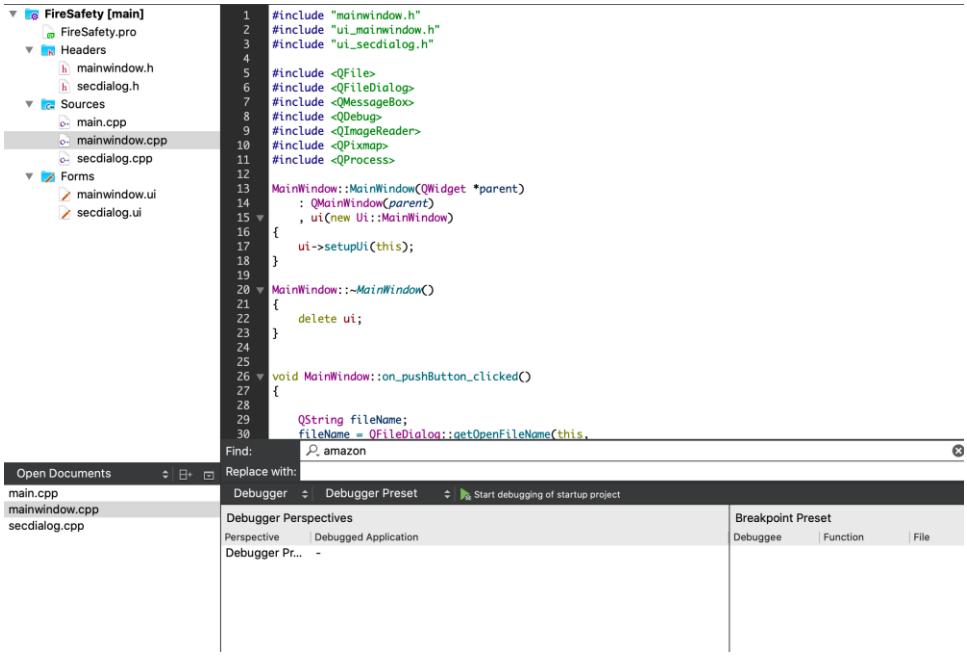

Figure 4.3.2 Qt Creator Project Composition Screenshot

5. User Manual

5.1 MainWindow

Step 1:

Once click the .exe file to start the program, the Main Window will pop up (Figure 5.1.1). Click the run model button, then a “Have Started” dialog (Figure 5.1.2) indicates that the model has started. After around 1 minute, “Data Ready” and “Have Finished” (Figure 5.1.3 & Figure 5.1.4) dialogs indicate that the model has finished prediction. In the meantime, the code running process is also shown in the main window (Figure 5.1.5).

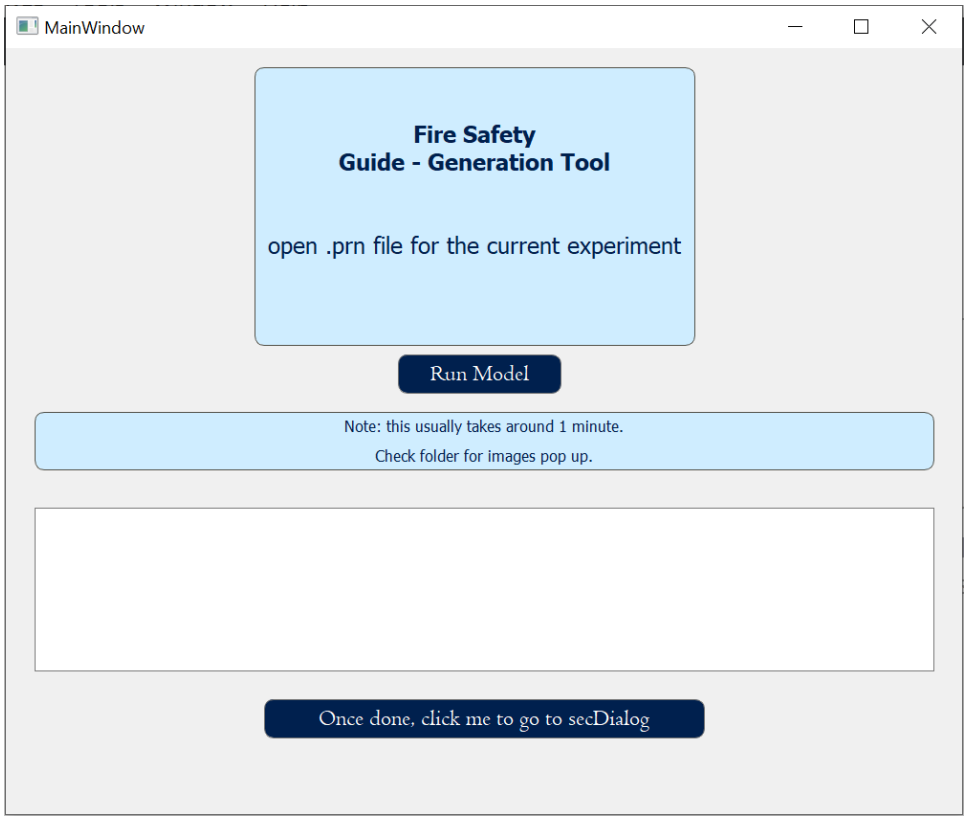

Figure 5.1.1 MainWindow

Note: the “Not Responding” error is a harmless indication that the program is running.

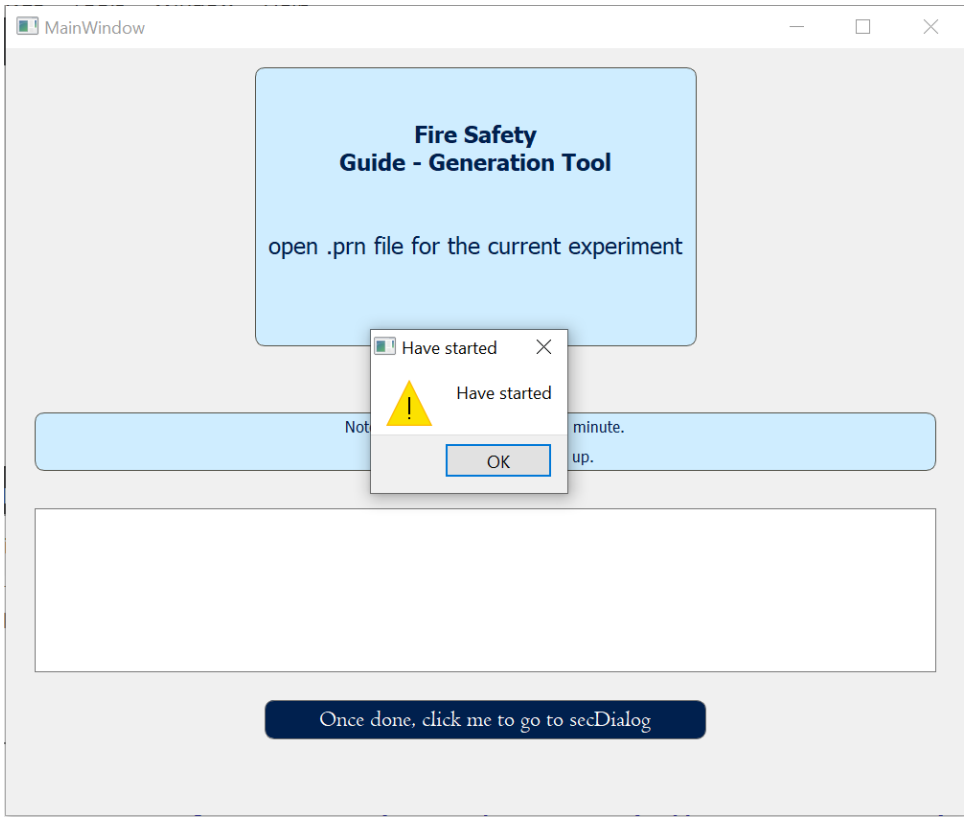

Figure 5.1.2 MainWindow after clicking “Run Model”

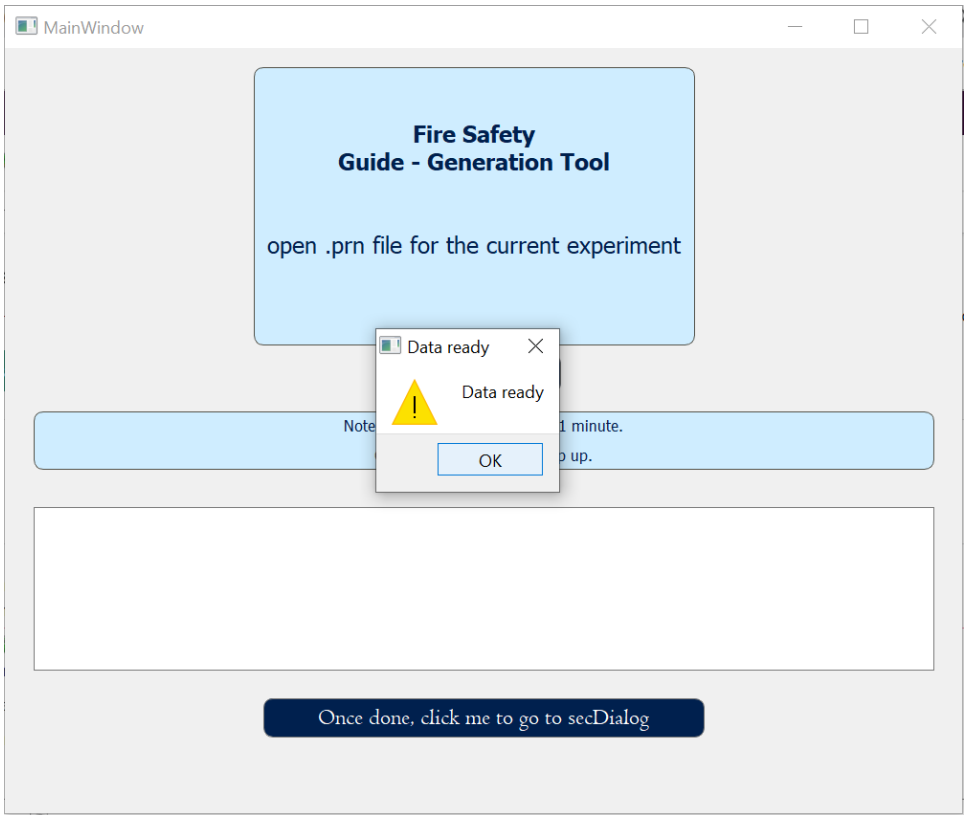

Figure 5.1.3 Data Ready Dialog

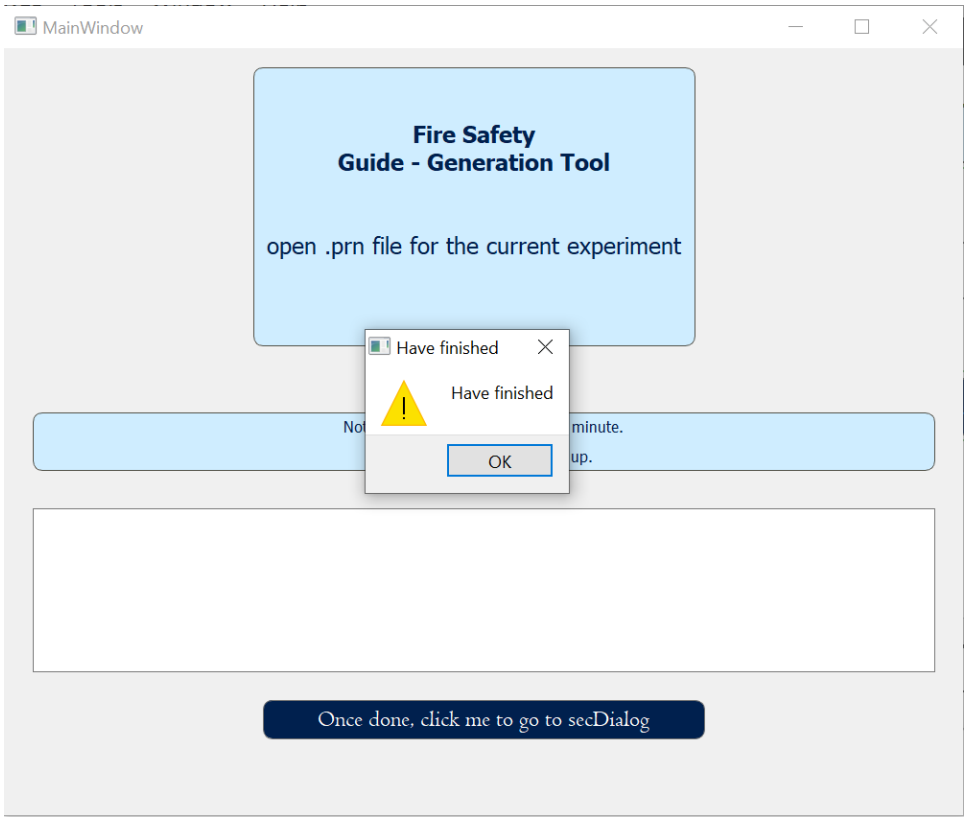

Figure 5.1.4 Have Finished Dialog

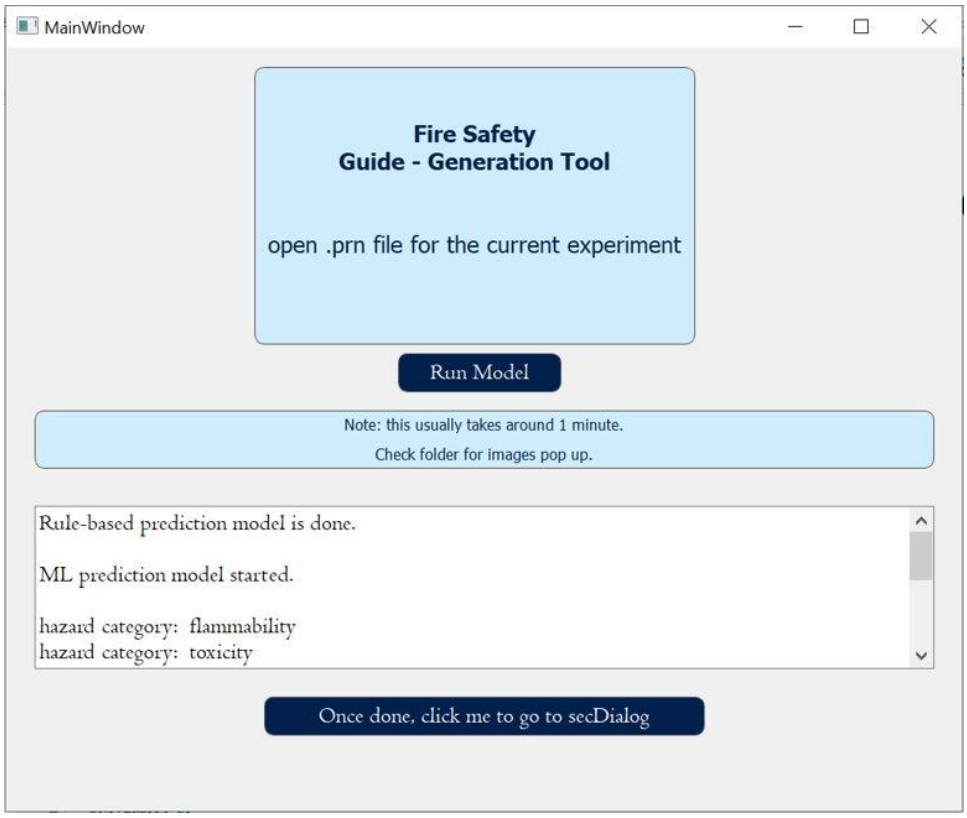

Figure 5.1.5 Dialog showing the code runing process

5.2 SecDialog Window

Step 2:

Navigation to the SecDialog, the user can reload Rule-Based prediction and machine learning prediction by clicking the dark blue bars on the left page.

5.2.1 The Chemicals Detected

This part includes all the chemicals detected from the .prn file. It is created by the file called plot_ppm.py in scripts.

5.2.2 Select a Chemical for the PPM plot

Users can scroll down the menu to select chemical, and the PPM plot will show below.

5.2.3 Response Strategy Guide

The bottom right corner is the strategy guide. The user can scroll up and down to inspect the given emergency response corresponding to the current prediction.

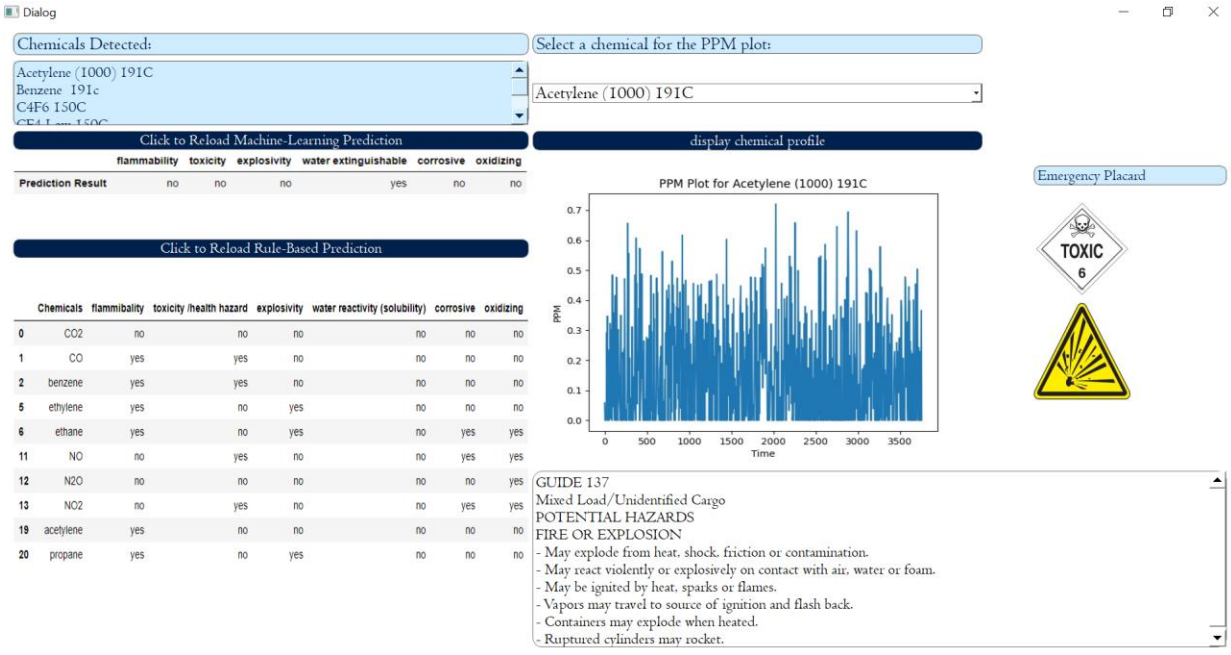

Figure 5.2.3 The SecDialog Window

6. Future Development

Existing improvements that need to be addressed in the future are the following:

1. The model gives different prediction results even though the development team saved the model parameters and fix the random seed of the model (Random_State of Ridge Regression), which may be caused by the data transformation process.
2. Current model prediction time is around 60 seconds even though the team have saved the parameters of training result and import directly which decreases the training time. The extra time was more resulted from the prediction process instead of the training process.
3. Future developers can integrate the file plot_ppm.py into the software so that users can directly run this code from the interface and dynamically generate chemical concentration profile graphs.
4. Once the emergency handbook are compiled in the form of a collection of emergency guides corresponding to the six hazardous categories, the future developers can add a navigation bar for the users to browse through several emergency guides on the interface. Same ideas also work for the emergency placard since they are all currently just placeholders.

7. Appendix

1. Download Qt Creator:

https://www.qt.io/download-open-source?hsCtaTracking=9f6a2170-a938-42df-a8e2-a9f0b1d6cdce%7C6cb0de4f-9bb5-4778-ab02-bfb62735f3e5

2. All the related code could be found in the following GitHub repository:

https://github.com/CathyF9600/Fire_Safety