Configuration Manual

MSc Research Project
MSc Cybersecurity

Donnel Shinto
Student ID: X23154748

School of Computing
National College of Ireland

Supervisor: Dr Arghir Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Donnel Shinto		
Student ID:	X23154748		
Programme:	MSc Cybersecurity	**Year:**	2024
Module:	MSc Research Project		
Lecturer:	Dr Arghir Nicolae Moldovan		
Submission Due Date:	December 12, 2024		
Project Title:	Enhancing Security in Electric Vehicle Charging Stations Through Advanced Anomaly Detection Systems		
Word Count:	590	**Page Count:**	8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Donnel Shinto

Date: December 12, 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Donnel Shinto
X23154748

1 Introduction

This configuration manual is a report that explains how to reproduce the work 'Enhancing Security in Electric Vehicle Charging Stations Through Advanced Anomaly Detection Systems'. This manual will outline all the steps and procedures involved in reproducing this research. The document also contains the details about the software, hardware and configurations used in the building of this research project. The dataset used for this research is CICEVSE2024(*EVSE Dataset 2024 | Datasets | Research | Canadian Institute for Cybersecurity | UNB*, n.d.).

2 Hardware and Software Specifications

Specification details of the PC

- Processor - 12th Gen Intel(R) Core (TM) i5-12500H 2.50 GHz
- Installed RAM - 16.0 GB (15.6 GB usable)
- System type - 64-bit operating system, x64-based processor
- GPU – Nvidia GeForce RTX 3050
- Operating System – Windows 11
- Storage – 512 GB SSD

Software Used

- Anaconda Navigator 2.5.2
- Jupyter Notebook 7.0.8
- Python 3.11.7
- Java 1.8.0_202
- Weka 3.8.6

3 Preprocessing Using Python

The network traffic dataset was divided into EVSE-A and EVSE-B. The files were classified as charging, idle and malicious. To perform machine learning on network traffic dataset it is essential to merge these into one single analysable file. Two separate notebooks were created for EVSE-A and EVSE-B and another notebook was used to combine the outputs of EVSE-A and EVSE-B.

The Host Events dataset was cleaned using a python code from another project. The link to the project is given in the reference section. The file after cleaning has an additional column named isDOS which is not required for our research. It can be removed in Weka during model building(*GitHub - CrashedBboy/ML-NetworkAttack-Detection*, n.d.).

The power consumption dataset was already cleaned and combined by the original authors of the dataset and no further preprocessing using python was required.

```
[1]: import warnings
     warnings.filterwarnings("ignore")

     import os
     import pandas as pd
```

```
[6]: folder_path = r'Network Traffic/EVSE-B/csv'

     # List all CSV files in the folder
     csv_files = os.listdir(folder_path)   # https://www.geeksforgeeks.org/python-list-files-in-a-directory/
```

```
[7]: df1 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-aggressive-scan.csv") # https://www.geeksforgeeks.org/read-multiple-csv-files-into-separate
     df2 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-icmp-flood.csv")
     df3 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-os-fingerprinting.csv")
     df4 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-port-scan.csv")
     df5 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-push-ack-flood.csv")

     df6 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-service-detection-scan.csv")
     df7 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-syn-flood.csv")
     df8 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-syn-stealth.csv")
     df9 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-synonymous-ip-flood.csv")
     df10 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-tcp-flood.csv")

     df11 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-udp-flood.csv")
     df12 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-charging-vulnerability-scan.csv")
     df13 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-idle-aggressive-scan.csv")
     df14 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-idle-icmp-flood.csv")
     df15 = pd.read_csv("Network Traffic\EVSE-B\csv\EVSE-B-idle-icmp-fragmentation.csv")
```

Figure 1: Loading CSV files into dataframe

```
import pandas as pd
import os

# Define paths to the folders containing the labeled files
charging_folder = "Network Traffic/EVSE-B/csv_label/EVSE-B-charging"
idle_folder = "Network Traffic/EVSE-B/csv_label/EVSE-B-idle"
malicious_folder = "Network Traffic/EVSE-B/csv_label/EVSE-B-malicious"

# Define output file paths for merged datasets
charging_output_path = "Network Traffic/EVSE-B/merged/EVSE-B-charging-merged.csv"
idle_output_path = "Network Traffic/EVSE-B/merged/EVSE-B-idle-merged.csv"
malicious_output_path = "Network Traffic/EVSE-B/merged/EVSE-B-malicious-merged.csv"

# Ensure the output directory exists
os.makedirs(os.path.dirname(charging_output_path), exist_ok=True) # https://stackoverflow.com/questions/273192/how-do-i-create-a-directory-and-any-missi
os.makedirs(os.path.dirname(idle_output_path), exist_ok=True)
os.makedirs(os.path.dirname(malicious_output_path), exist_ok=True)

# Function to merge CSV files in a folder                    # https://www.geeksforgeeks.org/how-to-read-multiple-data-files-into-pandas/
def merge_csv_files(folder_path, class_group):               # https://saturncloud.io/blog/how-to-import-multiple-csv-files-into-pandas-and-concatenate
    merged_df = pd.DataFrame()
    for file_name in os.listdir(folder_path):
        if file_name.endswith(".csv"):
            file_path = os.path.join(folder_path, file_name)
            df = pd.read_csv(file_path)
            merged_df = pd.concat([merged_df, df], ignore_index=True)
            merged_df['Class_Group'] = class_group
    return merged_df


# Merge labeled files for charging and idle datasets
charging_merged_df = merge_csv_files(charging_folder, 'charging')
idle_merged_df = merge_csv_files(idle_folder, 'idle')
```

Figure 2: Merging files into charging, idle and malicious

```
[3]: import pandas as pd
     import warnings
     warnings.filterwarnings("ignore")

     # Specify the file paths directly
     file1 = "Network Traffic/EVSE-A/merged/EVSE-A-charging-merged.csv"
     file2 = "Network Traffic/EVSE-A/merged/EVSE-A-idle-merged.csv"
     file3 = "Network Traffic/EVSE-B/merged/EVSE-B-charging-merged.csv"
     file4 = "Network Traffic/EVSE-B/merged/EVSE-B-idle-merged.csv"
     file5 = "Network Traffic/EVSE-B/merged/EVSE-B-malicious-merged.csv"

     # Read and concatenate all files
     merged_data = pd.concat([pd.read_csv(file1), pd.read_csv(file2), pd.read_csv(file3), pd.read_csv(file4), pd.read_csv(file5)], ignore_index=True) # https:

     # Save the merged data into a new CSV file
     output_file = "Network Traffic/EVSE.csv"
     merged_data.to_csv(output_file, index=False)

     print(f"Merged data saved to {output_file}")

     Merged data saved to Network Traffic/EVSE.csv

[ ]:
```

Figure 3: Merging all files into a single CSV

```
[ ] corrupted_df = df[df['State'] == '0']
    corrupted_df[:5]
```

	time	alarmtimer_alarmtimer_cancel	alarmtimer_alarmtimer_fired	alarmtimer_alarmtimer_start	alarmtimer_alarmtimer_suspend	alignme fau
6171	5.004938602	0	0	0	0	
6172	10.06490129	0	0	0	0	
6173	15.12084978	0	0	0	0	
6174	20.18085179	0	0	0	0	
6175	25.24072949	0	0	0	0	

5 rows × 915 columns

```
[ ] fixed_df = corrupted_df.copy(deep=True)

    fixed_df['State'] = fixed_df['interface']
    fixed_df['Attack'] = fixed_df['Unnamed: 911']
    fixed_df['Scenario'] = fixed_df['Unnamed: 912']
    fixed_df['Label'] = fixed_df['Unnamed: 913']
    fixed_df['interface'] = fixed_df['Unnamed: 914']
    fixed_df[:5]
```

Figure 4: Cleaning host events data

4 Model Building Using Weka

Steps to recreate the models with Weka

- Opening Weka through terminal using the command ‘java -Xmx8192m -jar weka.jar’ to increase the memory of Weka to 8GB. This is essential so that the software does not run of memory while processing resource intensive dataset like network traffic.

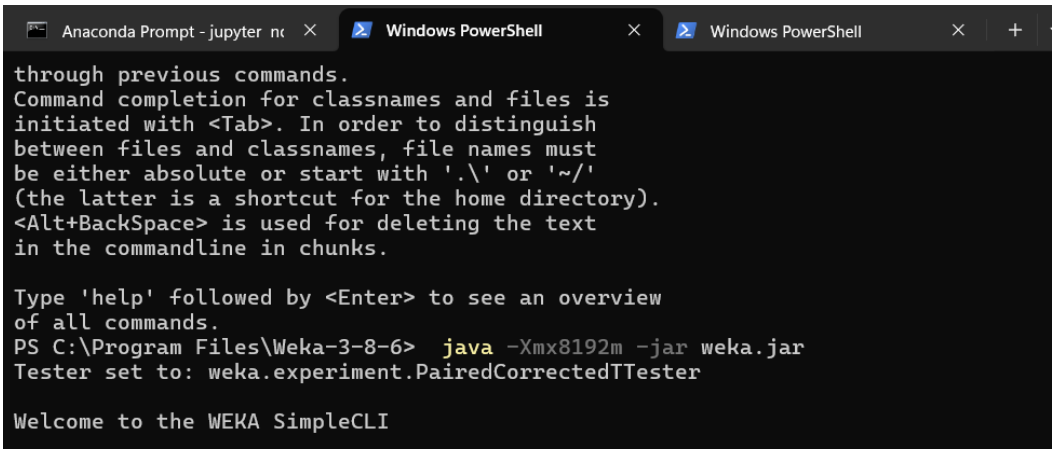

Figure 5: Opening Weka

- Chose Weka workbench from the list of options. After opening workbench open the desired dataset to perform machine learning model.

Figure 6: Weka workbench

- Perform preprocessing steps like normalisation through Weka filters.

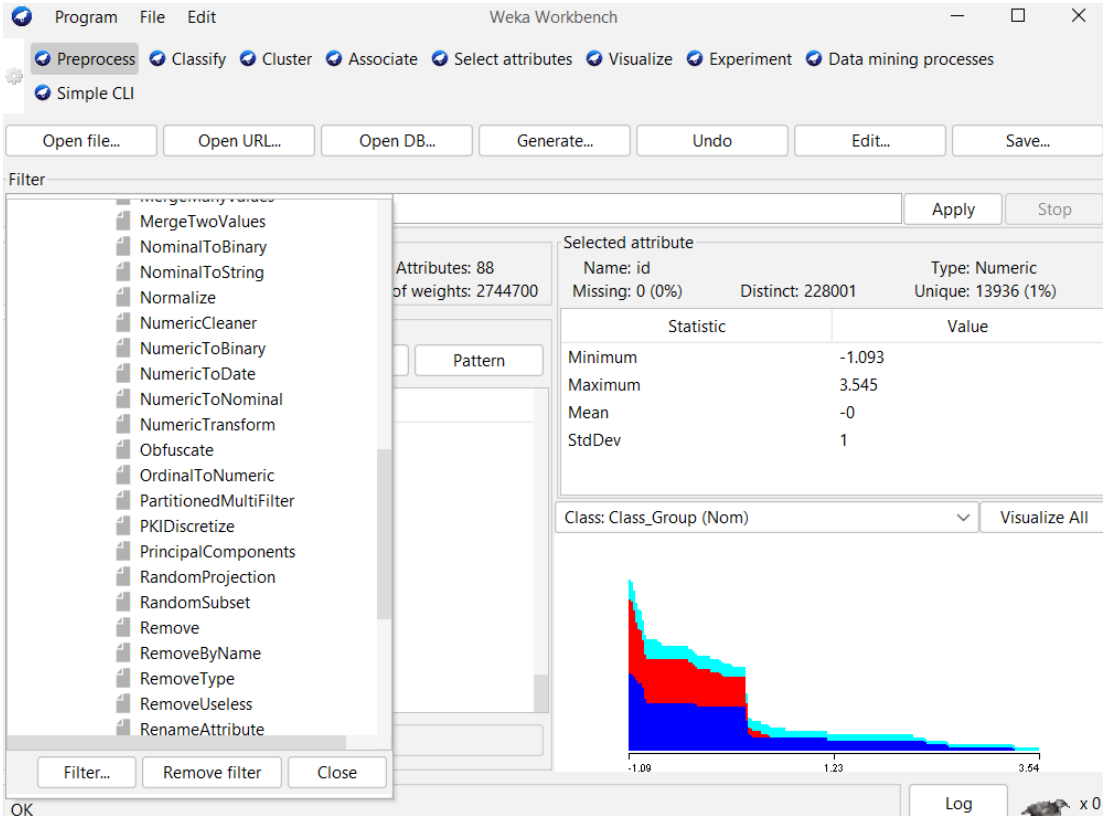

Figure 7: Preprocessing using Weka

- For Network Traffic dataset calculate information gain using the select attribute's function. Remove columns according to the information gain. Calculate information gain again for the preprocessed data and remove columns accordingly. Continue this step until you get attributes with information gain>1.

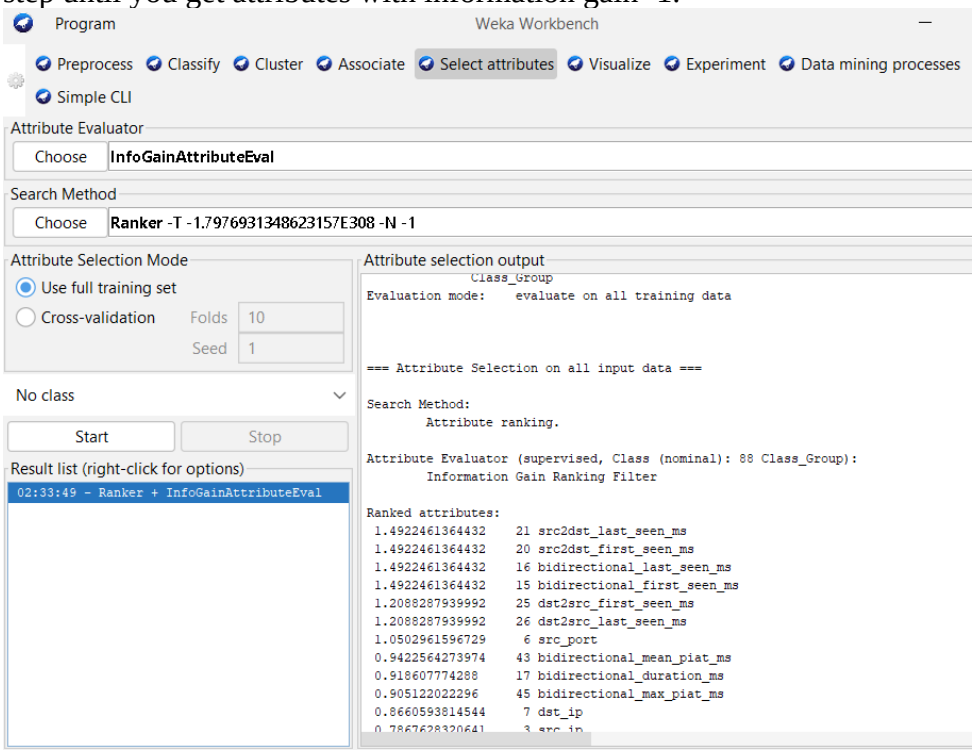

Figure 8: Calculating Information Gain

- For power consumption dataset the time column is removed. Columns like label, attack and attack group are removed according to the model configuration.

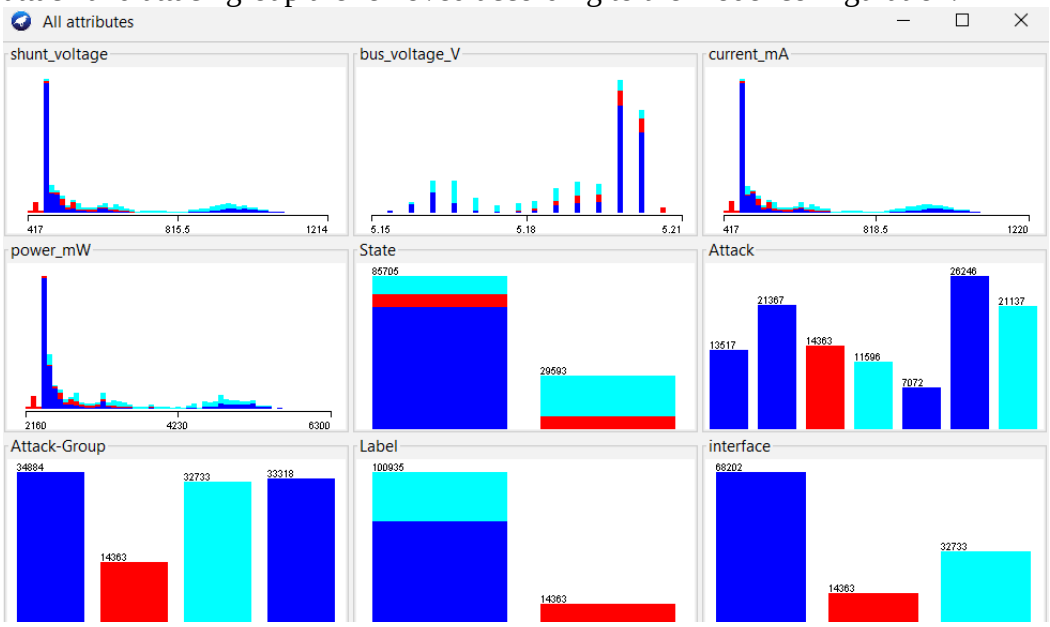

Figure 9: Visualisation of power consumption data

- Choosing algorithm from the list of algorithms Weka provides. Here I have chosen Random Forest.

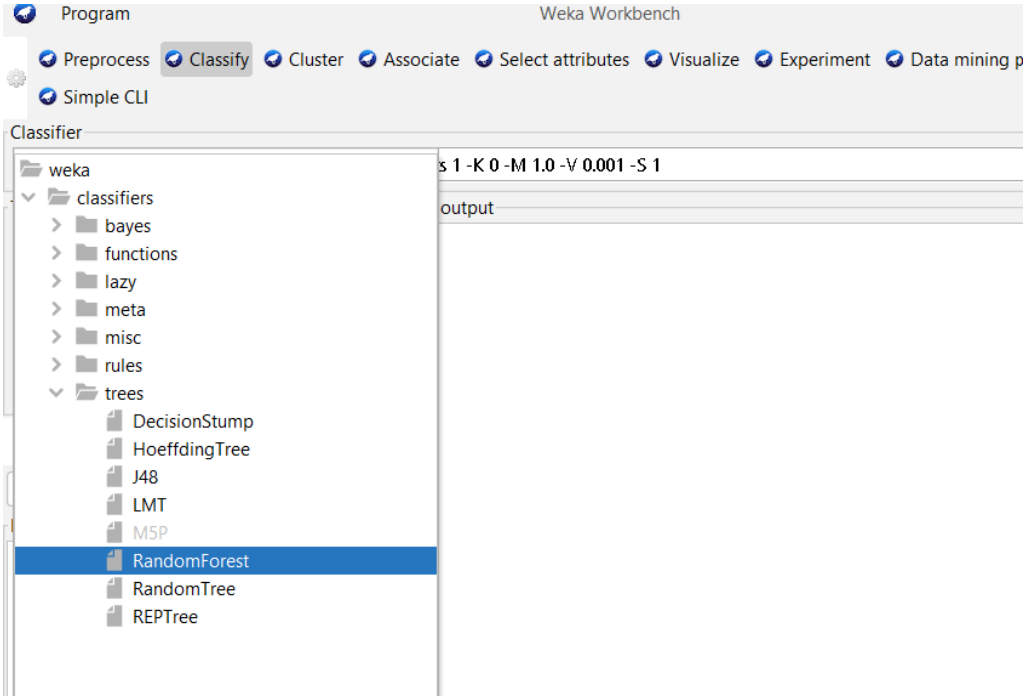

Figure 10: Chosing Algorithm

- Choosing a 70-30 test split to effectively evaluate the mode.

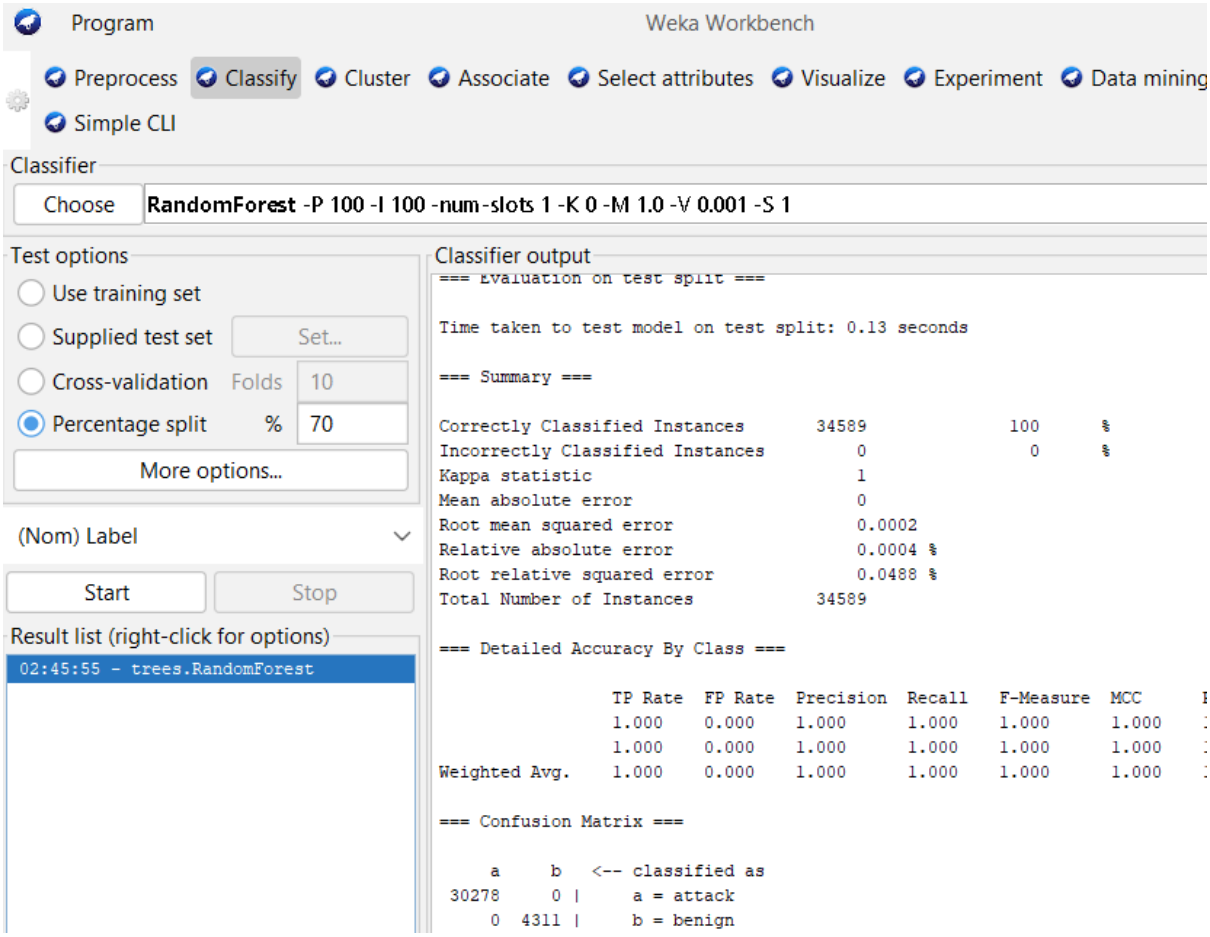

Figure 11: Specifying train test split

- Saving the result as a text file using the save result buffer option.

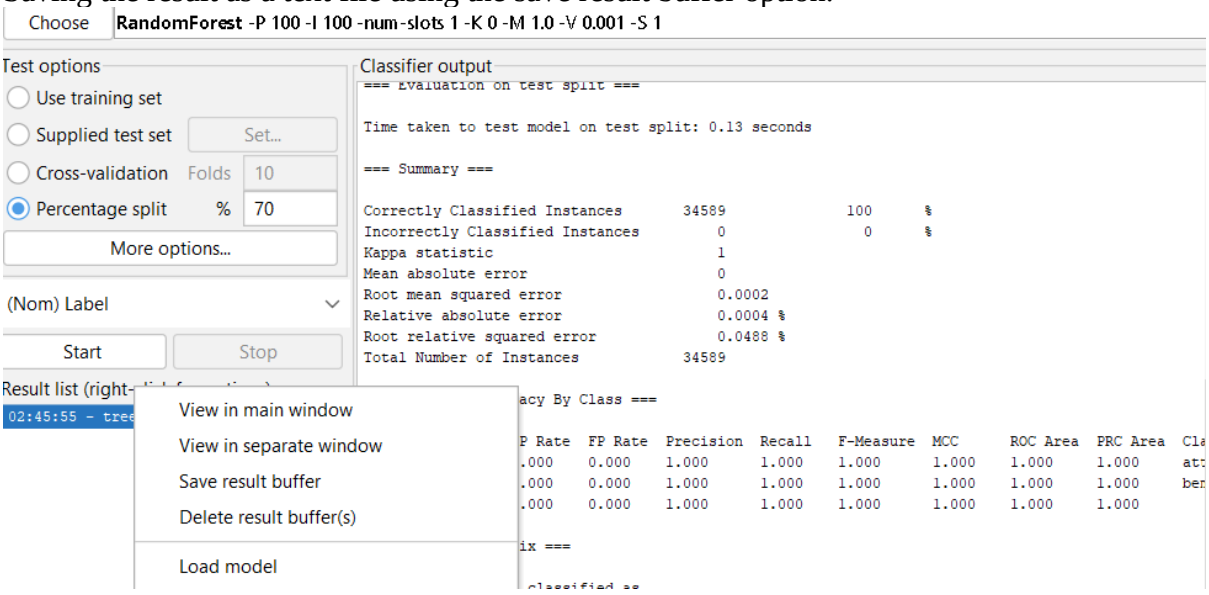

Figure 12: Saving the result as a text file

References

EVSE Dataset 2024 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d.). Retrieved December 11, 2024, from https://www.unb.ca/cic/datasets/evse-dataset-2024.html

GitHub - CrashedBboy/ML-NetworkAttack-Detection: Machine Learning-Based Attack Detection For Electric Vehicle Charging Infrastructure Security. (n.d.). GitHub. Retrieved December 12, 2024, from https://github.com/CrashedBboy/ML-NetworkAttack-Detection