National College of Ireland

Log-Based Intrusion Detection System Using Machine Learning Algorithms

MSc Research Project

Artificial Intelligence

Sonia Francis Javior

Student ID: x22175903

School of Computing

National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland Project Submission Sheet School of Computing

Student Name:	Sonia Francis Javior
Student ID:	x22175903
Programme:	MSc. in Artificial Intelligence
Year:	2023
Module:	MSc Research Project
Supervisor:	Rejwanul Haque
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	750
Page Count:	12

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:	Sonia Francis Javior
Date:	14th December 2023

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):	

Log Based Intrusion Detection System Using Machine Learning Algorithms Configuration Manual

Sonia Francis Javior

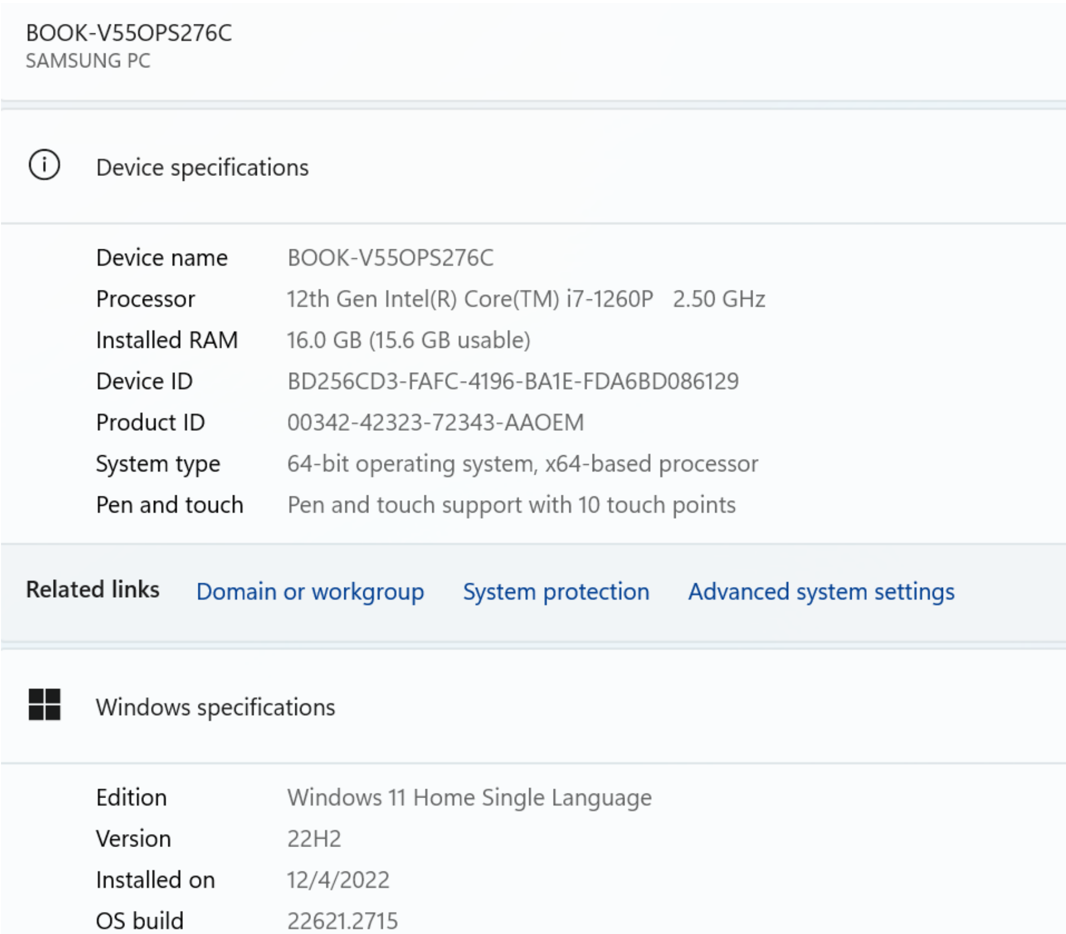
x22175903

1 Introduction

Configuration manuals generally provide an extended analysis of the system or device system required to implement the research method. The main objective of the manual is to set up the technical configuration. This also describes the setting procedure of Jupyter Notebook and other Python configurations.

2 System Specification

System specification provides the technical features and variants of the device required to implement the model. Generally, contains various specs of the system such as storage, operating system and processor, Figure 1. To implement the research a basic device system of 8GB more sufficient.

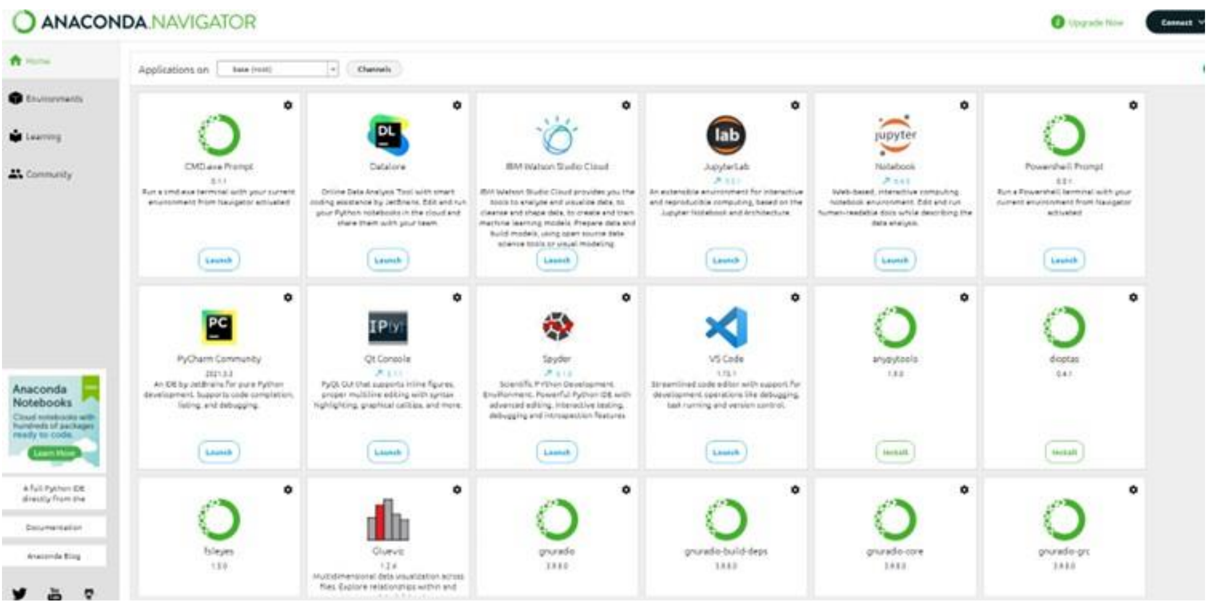


3 Software Used

- Microsoft excel: Used for initial exploration. MS-Excel For the preliminary Analysis of data.
- Jupyter Notebook: For implementing the model and analyzing the evaluation metrics

Installation and Environment Setup

Depending on the Operating System, the latest version of the software is suggested. Python 3.10 for Windows 11 has been installed. After installing Python, we need an Integrating Development environment to implement the model. Jupyter Notebook is installed from Anaconda which is a combination of several applications. The outline of the anaconda navigator is shown below.



4 Project Development

After the installation, host the Jupyter Notebook and open a Python file with .py extension to start the implementation. All the necessary libraries are installed using pip command.

Importing Library

The packages used in the project are displayed in Figure 4. The cloud platform comes with several necessary libraries already installed. If necessary, additional libraries should be imported.

```
import pandas as pd
from sklearn import preprocessing
from pandas import DataFrame
#installing the pyspark packages for the further analysis from pyspark.sql import SparkSession, Window
from pyspark.sql import functions as F
from pyspark.sql.functions import regexp_extract
from pyspark.sql.types import IntegerType
from pyspark.sql.types import StringType
from matplotlib import pyplot as plt
import matplotlib.patches as mpatches
from pyspark.sql.functions import col, isnan, when, count
from pyspark.sql.functions import *
from collections import Counter
from sklearn.datasets import make_classification
from matplotlib import pyplot
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from time import time
from packaging import version
from matplotlib import pyplot
from matplotlib import pyplot as plt
import seaborn as sns
import missingno as msno
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from scipy.special import boxcoxip
from statsmodels.stats.outliers influence import variance_inflation_factor
from sklearn.impute import KNNImputer
from sklearn.feature_selection import VarianceThreshold
from scipy import stats
from pylab import *
from sklearn.inspection import permutation_importance
from sklearn.feature_selection import RFE
import matplotlib.patches as mpatches
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, plot_confusion_matrix, plot_roc_curve fr
from keras import models, layers
from sklearn.model_selection import train_test_split from sklearn.ensemble import RandonForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier from sklearn.neighbors import Nearest Neighbors from sklearn.linear_model
from sklearn.model_selection import GridSearchCV from sklearn.datasets import make classification from imblearn.over_sampling
```

Importing Files: Importing all the datasets in csv format to dataframes using pandas library.

In [107]:

```
#Load the ecml.csv file into dataframe d1 from the https://github.com/msudol/Web-Application-Attack-Datasets
d1 = pd.read_csv(r"C:\Users\sonia\Desktop\Final\Dataset\ecml.csv")
```

```
C:\Users\sonia\AppData\Local\Temp\ipykernel_51288\4180735838.py:2: DtypeWarning: Columns (5) have mixed types. Specify dtype
option on import or set low_memory=False.
  d1 = pd.read_csv(r"C:\Users\sonia\Desktop\Final\Dataset\ecml.csv")
```

In [108]:

```
c1 = pd.read_csv(r"C:\Users\sonia\Desktop\Final\Dataset\allAnomalies1.csv")
c2 = pd.read_csv(r"C:\Users\sonia\Desktop\Final\Dataset\allAnomalies2.csv")
c3 = pd.read_csv(r"C:\Users\sonia\Desktop\Final\Dataset\allAttacks1.csv")
c4 = pd.read_csv(r"C:\Users\sonia\Desktop\Final\Dataset\allAttacks2.csv")
c5 = pd.read_csv(r"C:\Users\sonia\Desktop\Final\Dataset\allAttacks3.csv")
c6 = pd.read_csv(r"C:\Users\sonia\Desktop\Final\Dataset\allAttacks4.csv")
c7 = pd.read_csv(r"C:\Users\sonia\Desktop\Final\Dataset\allAttacks5.csv")
c8 = pd.read_csv(r"C:\Users\sonia\Desktop\Final\Dataset\allNormals1.csv")
```

To Fetch data from dataframe to csv to the exact location

```
r_df.to_csv(r"C:\Users\sonia\Desktop\Final\Generated\final_dataset.csv")
```

Description of Dataset: The datasets are collected from the open-source platform. The ecml and the zip file data from CSIC dataset. The dataset contains the details of the log file entries which are captured from the web servers.

Processing

Null values are removed and missing values are replaced by some data handling techniques.

```
c7['result'] = c7['query'].fillna('') + c7['body'].fillna('')
c7 = c7[['attack','path','result']]
c7.rename(columns = {'attack':'type','result':'query'}, inplace = True)
```

Distribution Analysis of type of attacks

```
r_df.groupby(['type']).count()
```

6]:

type	path	query
CRLFi	327	327
LdapInjection	2353	2353
OsCommanding	3420	3420
PathTransversal	3029	3029
SSI	2349	2349
SqlInjection	45573	45573
Valid	43369	43369
XPathInjection	2454	2454
XSS	6643	6643
anomalous	16459	16459

Feature Selection

By defining the keywords for each type of attack and then classifying them to features. Features are selected for each type of attacks.

```
df
```

3]:

	type	url	label	RL	AL	NA	PL	FU	FL	FD	FS	NK
0	Valid	/l_t@/_Feu1wvhtpass2/1nieQnnrvnzktuasain/tg1AR...	0	213	76	3	136	47	114	21	31	27
1	Valid	/inssgtz7ieltdSstbw7e/neQhmsdwu7imdb0etet/eT/h...	0	360	250	13	109	58	213	44	45	41
2	Valid	/fonRnt/7N.D-4BRSsXb@TU/qgheW.cfm	0	33	0	0	33	10	14	2	7	5
3	Valid	/dyylkL.XD9cPu/4Ot0ta/ts6xNrP1/hssh/a2cuerht/s...	0	285	190	11	94	44	165	35	41	36
4	Valid	/2m6VLb1r37jSPC/cWVv/Mbar/oqrd0/msc/etceebwgi/...	0	360	284	14	75	47	228	41	44	41
...	...	...	...	...	...	...	...	...	...	...	...	...
125971	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	184	154	13	29	4	118	32	30	29
125972	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	184	154	13	29	2	126	26	30	29
125973	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	188	158	13	29	9	123	26	30	29
125974	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	189	159	13	29	2	131	26	30	29
125975	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	186	156	13	29	2	123	31	30	29

125976 rows × 12 columns

```
df
```

	type	url	label	./	.bat	passwd	log	exec	boot	:/	...	../	%00	.conf	ini	windows	\\.	..\\	:	etc
0	Valid	/l_t@/_feu1wvhtpass2/1nieqnnrvnzktuasain/tg1ar...	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	Valid	/inssgtz7ieltdsstbw7e/neqhmsdwu7imdb0etet/et/h...	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	Valid	/fonrnt/7n.d-4brssxb@tu/qghew.cfm	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	Valid	/dyylkl.xd9cpu/4ot0ta/ts6xnrp1/hssh/a2cuerht/s...	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	Valid	/2m6vlb1r37jspc/cwvv/mbar/oqrd0/msc/etceebwgi/...	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
125971	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0
125972	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0
125973	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0
125974	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0
125975	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0

125976 rows × 29 columns

```
df
```

	type	url	label		'	|	update	)	>=	or	...	+	count	<>	into	()	<	%	exec	betwwen	sp
0	Valid	/l_t@/_feu1wvhtpass2/1nieqnnrvnzktuasain/tg1ar...	0	1	0	0	0	0	0	1	...	1	0	0	0	0	0	0	0	0	0
1	Valid	/inssgtz7ieltdsstbw7e/neqhmsdwu7imdb0etet/et/h...	0	1	1	0	0	0	0	0	...	1	0	0	0	0	0	0	0	0	0
2	Valid	/fonrnt/7n.d-4brssxb@tu/qghew.cfm	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	Valid	/dyylkl.xd9cpu/4ot0ta/ts6xnrp1/hssh/a2cuerht/s...	0	1	0	1	0	0	0	0	...	1	0	0	0	0	0	1	0	0	0
4	Valid	/2m6vlb1r37jspc/cwvv/mbar/oqrd0/msc/etceebwgi/...	0	1	0	0	0	0	0	1	...	1	0	0	0	0	0	1	1	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
125971	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	1	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	1
125972	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	1	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	1
125973	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	1	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	1
125974	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	1	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	1
125975	Valid	/tienda1/publico/registro.jsp?modo=registro&lo...	0	1	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	1

125976 rows × 60 columns

Splitting of Data:

Spliting the training data and testing set. The training data is 80% of original data and 20 % is testing data.

Model Building:

After preprocessing and appropriate feature extraction, the next important step is the model building. The supervised machine learning algorithms are used. SVM, Logistic Regression, Random Forest, Gradient Boost, KNearest Neighbor and Decision Tree.

```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, accuracy_score, log_loss, classification_report
from matplotlib import pyplot as plt
from seaborn import heatmap

# Instantiate the Logistic Regression model
LR = LogisticRegression()
LR.fit(X_train, Y_train)

# Calculate and print accuracy on training set
print('Accuracy of Logistic Regression classifier on training set: {:.2f}'.format(LR.score(X_train, Y_train)))

# Calculate and print accuracy on test set
print('Accuracy of Logistic Regression classifier on test set: {:.2f}'.format(LR.score(X_test, Y_test)))

# Make predictions on the test set
Y_pred = LR.predict(X_test)

# Calculate and print confusion matrix
print('\nConfusion Matrix:')
cm = confusion_matrix(Y_test, Y_pred)
fig, ax = plt.subplots(figsize=(8, 6))

# Use seaborn to create the heatmap
heatmap(cm, annot=True, cmap='Blues')

plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.tight_layout()
plt.show()

# Calculate and print accuracy using metrics.accuracy_score
accuracy_lr = metrics.accuracy_score(Y_test, Y_pred)
print('\nAccuracy:', accuracy_lr)

# Calculate and print log loss using log_loss
Y_pred_proba = LR.predict_proba(X_test)
loss_lr= log_loss(Y_test, Y_pred_proba)
print('\nLog Loss:', loss_lr)
classification_lr = classification_report(Y_test, Y_pred)

# Print the classification report
print('\nClassification Report:')
print(classification_lr)
```

```python
from sklearn import svm
from sklearn.metrics import confusion_matrix, accuracy_score, hinge_loss,classification_report

# Instantiate the Linear SVM model
SVM = svm.LinearSVC()
SVM.fit(X_train, Y_train)

# Calculate and print accuracy on training set
print('Accuracy of Linear SVM classifier on training set: {:.2f}'.format(SVM.score(X_train, Y_train)))

# Calculate and print accuracy on test set
print('Accuracy of Linear SVM classifier on test set: {:.2f}'.format(SVM.score(X_test, Y_test)))

# Make predictions on the test set
Y_pred_svm = SVM.predict(X_test)

# Calculate and print confusion matrix
print('\nConfusion Matrix:')
print(confusion_matrix(Y_test, Y_pred_svm))

# Calculate and print accuracy using metrics.accuracy_score
accuracy_svm = accuracy_score(Y_test, Y_pred_svm)
print('\nAccuracy:', accuracy_svm)

# Calculate and print hinge loss using hinge_loss
loss_svm = hinge_loss(Y_test, SVM.decision_function(X_test))
print('\nHinge Loss:', loss_svm)
classification_svm = classification_report(Y_test, Y_pred)

# Print the classification report
print('\nClassification Report:')
print(classification_svm)
```

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, log_loss, classification_report

# Instantiate the Random Forest model
RF = RandomForestClassifier()
RF.fit(X_train, Y_train)

# Calculate and print accuracy on training set
print('Accuracy of Random Forest classifier on training set: {:.2f}'.format(RF.score(X_train, Y_train)))

# Calculate and print accuracy on test set
print('Accuracy of Random Forest classifier on test set: {:.2f}'.format(RF.score(X_test, Y_test)))

# Make predictions on the test set
Y_pred_rf = RF.predict(X_test)

# Calculate and print confusion matrix
print('\nConfusion Matrix:')
print(confusion_matrix(Y_test, Y_pred_rf))

# Calculate and print accuracy using metrics.accuracy_score
accuracy_rf = accuracy_score(Y_test, Y_pred_rf)
print('\nAccuracy:', accuracy_rf)

# Calculate and print log loss using log_loss
Y_pred_proba_rf = RF.predict_proba(X_test)
loss_rf = log_loss(Y_test, Y_pred_proba_rf)
print('\nLog Loss:', loss_rf)
classification_rf = classification_report(Y_test, Y_pred)

# Print the classification report
print('\nClassification Report:')
print(classification_rf)
```

Evaluation Metrics:

Accuracy, Precision, Recall, Fi Score in the classification report. Most of them achieved maximum accuracy, so loss is calculated.

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	23866
1	0.99	0.93	0.96	1330
accuracy			1.00	25196
macro avg	0.99	0.97	0.98	25196
weighted avg	1.00	1.00	1.00	25196

Classification Report:

	precision	recall	f1-score	support
0	0.99	1.00	1.00	23866
1	0.96	0.89	0.93	1330
accuracy			0.99	25196
macro avg	0.98	0.95	0.96	25196
weighted avg	0.99	0.99	0.99	25196