Configuration Manual

MSc Research Project
Cloud Computing

Danni Zhang
Student ID: x22174435

School of Computing
National College of Ireland

Supervisor: Aqeel Kazmi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Danni Zhang
Student ID:	x22174435
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Aqeel Kazmi
Submission Due Date:	14/12/2023
Project Title:	Configuration Manual
Word Count:	587
Page Count:	6

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:	
Date:	28th January 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

Danni Zhang
x22174435

1 Introduction

This configuration manual details the process of developing an anomaly detection system using machine learning and deep learning techniques applied to the IoT-23 dataset in a serverless cloud environment. It provides instructions for recreating the development environment and running the code associated with the research project. The tools utilized throughout the project include Google Colab, Python programming language, and various libraries such as scikit-learn, TensorFlow, Pandas, and Matplotlib.

2 System configuration requirements

The configuration process involves accessing Google Colab through a web browser, allowing users to develop and run Python code in a collaborative environment.

2.1 Hardware Requirements

For this project, a standard configuration with sufficient RAM and GPU capabilities is used to facilitate the training of machine learning and deep learning models efficiently.

- Colab Runtime type: Python 3
- Colab Hardware accelerator: T4 GPU

2.2 Software Requirements

A web browser.

3 Project Implementation

3.1 Environment Setup

1. Accessing Google Colab:

   Open a web browser and navigate to Google Colab at the following website: https://colab.research.google.com/.

   Sign in using Google account.

2. Setting up the Environment:

- Create a new Colab notebook by selecting "File" - "New Notebook".
- Choose the appropriate runtime type and hardware acceleration (GPU/TPU) through "Runtime" - "Change Runtime Type".

3. Installing Required Libraries:

   Use pip commands to install necessary libraries not pre-installed in Colab (Radovanovic; 2022):

```
!pip install pandas scikit-learn tensorflow keras matplotlib plotly numpy
```

4. Upload files:

   - Upload the IoT-23 dataset and code file to the Google Drive to access the dataset seamlessly.

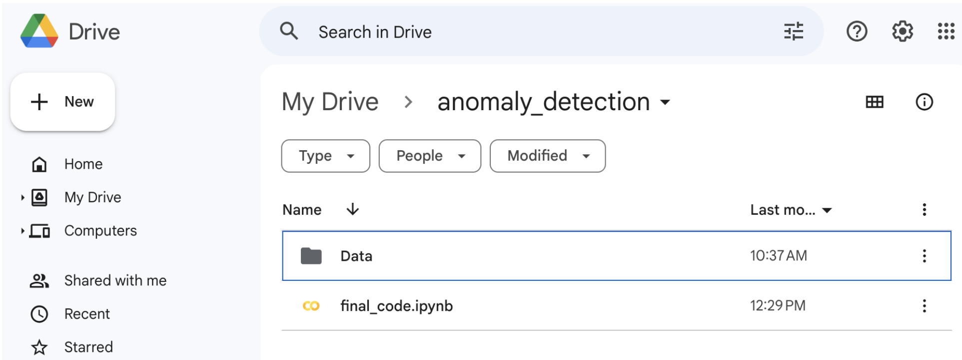


Figure 1: Upload to Google Drive

   - Navigate to the dataset directory by using the appropriate path.

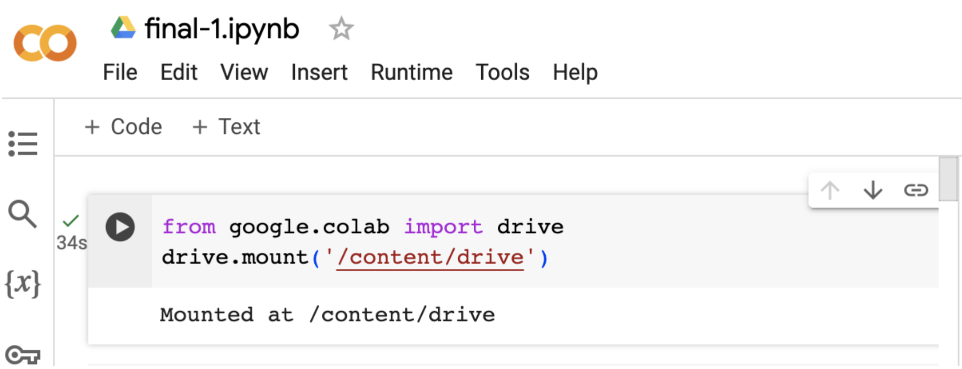


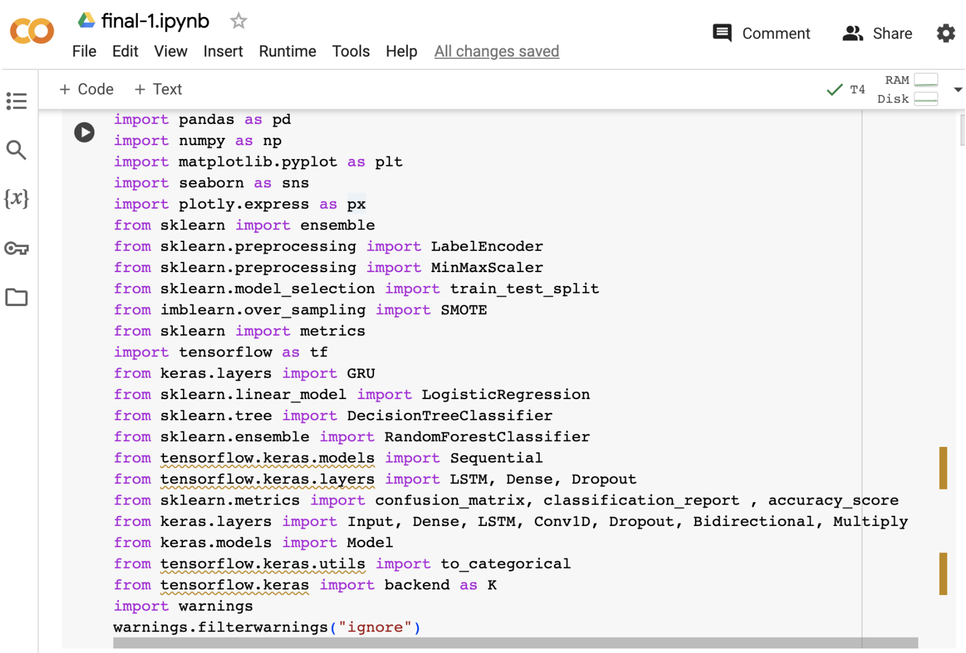
Figure 2: Navigate to Google Drive

5. Import Python libraries:

   Before implementing the code, make sure the following libraries have been imported after the dataset has been uploaded.

   Data Manipulation and Visualization:

   - `pandas`: For data manipulation and analysis.




```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from sklearn import ensemble
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE
from sklearn import metrics
import tensorflow as tf
from keras.layers import GRU
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from sklearn.metrics import confusion_matrix, classification_report , accuracy_score
from keras.layers import Input, Dense, LSTM, Conv1D, Dropout, Bidirectional, Multiply
from keras.models import Model
from tensorflow.keras.utils import to_categorical
from tensorflow.keras import backend as K
import warnings
warnings.filterwarnings("ignore")
```

Figure 3: Import Python libraries

- `numpy`: For numerical computations and array manipulation.
- `matplotlib.pyplot`: For creating visualizations like plots and charts.
- `seaborn`: For statistical data visualization.
- `plotly.express`: For interactive and complex visualizations.

Machine Learning Algorithms and Evaluation:

- `sklearn.ensemble`: For ensemble learning methods.
- `sklearn.preprocessing`: For data preprocessing tasks like label encoding and feature scaling.
- `sklearn.model_selection`: For splitting data into training and testing sets.
- `imblearn.over_sampling.SMOTE`: For oversampling to address class imbalance.
- `sklearn.metrics`: For evaluating model performance (confusion matrix, classification report, accuracy).

Deep Learning (TensorFlow and Keras):

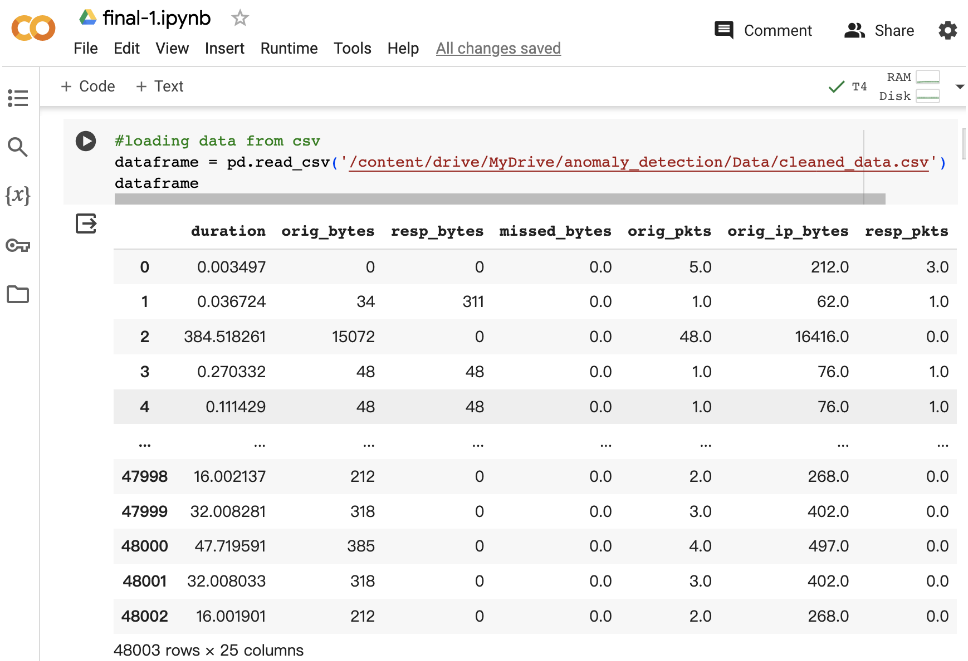
- `tensorflow`: Deep learning framework for building neural networks.
- `keras.layers`: Provides various types of neural network layers (GRU, LSTM, Conv1D, Dense, Dropout, Bidirectional).
- `tensorflow.keras.models.Sequential`: For creating sequential models in Keras.
- `tensorflow.keras.utils`: For categorical data handling.
- `tensorflow.keras.backend`: Backend operations for Keras.

Miscellaneous:

- `warnings`: For managing warnings in the code execution.

4 Phases

4.1 Data Loading



```
#loading data from csv
dataframe = pd.read_csv('/content/drive/MyDrive/anomaly_detection/Data/cleaned_data.csv')
dataframe
```

	duration	orig_bytes	resp_bytes	missed_bytes	orig_pkts	orig_ip_bytes	resp_pkts
0	0.003497	0	0	0.0	5.0	212.0	3.0
1	0.036724	34	311	0.0	1.0	62.0	1.0
2	384.518261	15072	0	0.0	48.0	16416.0	0.0
3	0.270332	48	48	0.0	1.0	76.0	1.0
4	0.111429	48	48	0.0	1.0	76.0	1.0
...	...	...	...	...	...	...	...
47998	16.002137	212	0	0.0	2.0	268.0	0.0
47999	32.008281	318	0	0.0	3.0	402.0	0.0
48000	47.719591	385	0	0.0	4.0	497.0	0.0
48001	32.008033	318	0	0.0	3.0	402.0	0.0
48002	16.001901	212	0	0.0	2.0	268.0	0.0

48003 rows × 25 columns

Figure 4: Data Loading

4.2 Data Cleaning

This step include checking null values, dropping null values, checking statistics of data and removing outlier from the dataset.

```
[ ] dataframe['label'].value_counts()

    Benign                      26001
    PartOfAHorizontalPortScan   12369
    C&C                          5618
    Attack                       3814
    Name: label, dtype: int64
```

Figure 5: After Data Cleaning

4.3 Data Visualization

- Class distribution

  Displays a bar chart with the count of each target class.

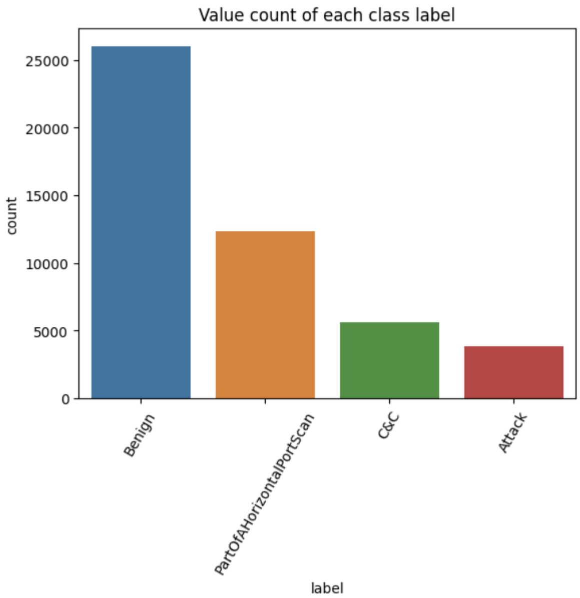


Figure 6: Count Plot of Target Class

- Duration analysis

  The top two maximum durations for each label are shown in a funnel chart.

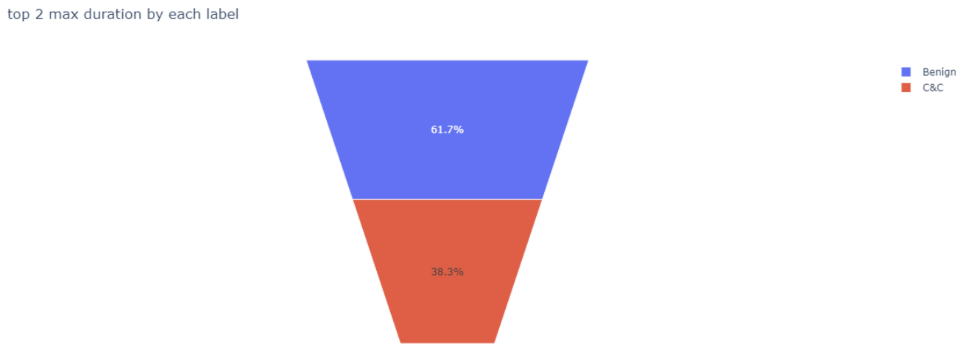


Figure 7: Top Two Maximum Durations by Label in Funnel Chart

- Feature interdependencies

5 Running Models

Utilize the implemented machine learning and deep learning models to train, validate, and test on the IoT-23 dataset.

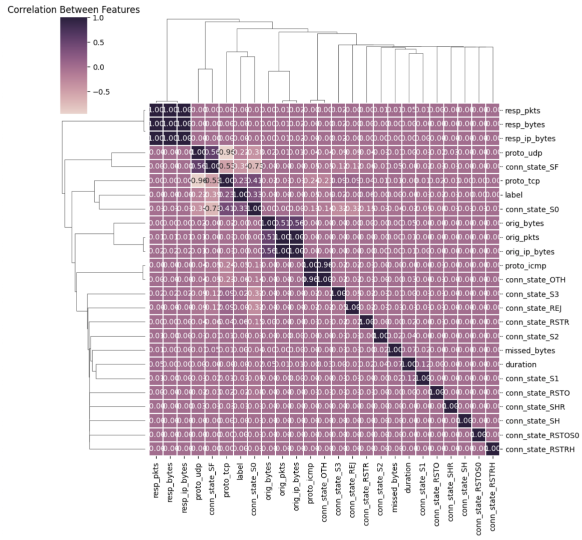


Figure 8: Correlation Matrix of Features

6 Results

Table 1: Machine learning models

Model	**Accuracy**	**Precision (Class 1)**	**Recall (Class 1)**	**F1-Score (Class 1)**
Logistic Regression	0.53	0.98	0.08	0.15
Decision Tree Classifier	0.69	0.00	0.00	0.00
Random Forest Classifier	0.61	1.00	0.23	0.38

Table 2: Deep learning models

Model	**Accuracy**	**Precision (Class 1)**	**Recall (Class 1)**	**F1-Score (Class 1)**
CNN with LSTM	0.66	0.96	0.33	0.50
CNN with BiLSTM	0.83	0.86	0.37	0.52

References

Radovanovic, I. (2022). Google colab - a step-by-step guide - algotrading101 blog. **URL:** *https://algotrading101.com/learn/google-colab-guide/*