Configuration Manual

MSc Research Project
Cybersecurity

Chandhiya Ramasamy
Student ID: X22105042

School of Computing
National College of Ireland

Supervisor: Jawad Salahuddin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Chandhiya Ramasamy
Student ID:	X22105042
Programme:	MSc. Cybersecurity **Year:** 2023-2024
Module:	Research Project
Supervisor:	Jawad Salahuddin
Submission Due Date:	14th December 2023
Project Title:	Strengthening Proactive Cyber Defence: Innovative Approaches for Effective Cyber Threat Intelligence Gathering, Analysis and Application
Word Count:	293 **Page Count:** 3

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: .....

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

Chandhiya Ramasamy
Student ID: X22105042

1 Introduction

This manual outlines a step-by-step configuration guide for implementing the Isolation Forest algorithm, Logistic Regression, and Support Vector Machines (SVM) for the detection of fake data in the Common Vulnerabilities and Exposures (CVE) dataset.

2 Software Requirements

1. Operating System: The code provided should work on any operating system (Windows, macOS, Linux).
2. Install Python from python.org or use a package manager like Anaconda.
3. Development Environment: Use an integrated development environment (IDE) such as Jupyter Notebook.
4. Machine Learning Libraries: Ensure you have machine learning libraries like scikit-learn installed.
5. Libraries: Import necessary Python libraries: 'pandas', 'numpy', 'scikit-learn', 'matplotlib', 'seaborn', 'plotly.graph_objects' , 'plotly.express' , IsolationForest', 'RandomForestClassifier', 'confusion_matrix', 'accuracy_score', 'precision_score', 'recall_score', 'f1_score', 'roc_curve', 'auc', 'precision_recall_curve', 'LabelEncoder', 'SVC' , 'LogisticRegression' in Jupyter notebook.

Import necessary libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.ensemble import IsolationForest, RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
import warnings
```

Figure 1. Library Import

3 Load the CVE Dataset

Download the CVE dataset from Kaggle which is Available at: https://www.kaggle.com/datasets/andrewkronser/cve-common- vulnerabilities-and-exposures?datasetId=500243. (Random 10000 records have been chosen for processing)

Load the augmented dataset with original and bogus data into a Pandas DataFrame.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Unnamed: 0	mod_date	pub_date	cvss	cwe_code	cwe_name	summary	access_authentication	access_complexity	access_vector	impact_availability	impact_confidentiality
2	DUMMY-1252	06/06/2023 0:00	06/06/2023 0:00	2.055773700615504	373	Authentication Bypass	DUMMY Summary 125	SINGLE	MEDIUM	ADJACENT_NETWORK	COMPLETE	PARTIAL
3	CVE-2006-1018	18/10/2018 16:30	07/03/2006 0:02	7.5	89	Improper Neutralizatic	SQL injection vulnerabi	NONE	LOW	NETWORK	PARTIAL	PARTIAL
4	CVE-2018-18774	29/11/2018 14:21	20/11/2018 19:29	4.3	79	Improper Neutralizatic	CentOS-WebPanel.com	NONE	MEDIUM	NETWORK	NONE	NONE
5	CVE-2014-100006	08/09/2017 1:29	13/01/2015 11:59	4.3	79	Improper Neutralizatic	Multiple cross-site scri	NONE	MEDIUM	NETWORK	NONE	NONE
6	CVE-2010-4785	21/04/2011 10:55	21/04/2011 10:55	4	399	Resource Managemen	The do_extendedOp fu	SINGLE	LOW	NETWORK	PARTIAL	NONE
7	DUMMY-1340	02/09/2023 0:00	02/09/2023 0:00	3.3191459966009234	488	Incorrect Type Convers	DUMMY Summary 134	SINGLE	MEDIUM	ADJACENT_NETWORK	PARTIAL	COMPLETE
8	CVE-2019-10361	17/09/2019 23:15	31/07/2019 13:15	2.1	255	Credentials Managem	Jenkins Maven Release	NONE	LOW	LOCAL	NONE	PARTIAL
9	DUMMY-202	21/07/2020 0:00	21/07/2020 0:00	2.454658426851524	402	Inconsistent Interpreta	DUMMY Summary 202	SINGLE	MEDIUM	NETWORK	NONE	NONE
10	DUMMY-1363	25/09/2023 0:00	25/09/2023 0:00	6.684041980832685	466	Improper Limitation of	DUMMY Summary 136	SINGLE	MEDIUM	NETWORK	NONE	NONE
11	CVE-2011-2001	26/02/2019 14:04	12/10/2011 2:52	9.3	20	Improper Input Validat	Microsoft Internet Expl	NONE	MEDIUM	NETWORK	COMPLETE	COMPLETE
12	CVE-2017-6755	28/07/2017 17:36	25/07/2017 19:29	4.3	79	Improper Neutralizatic	A vulnerability in the w	NONE	MEDIUM	NETWORK	NONE	NONE
13	DUMMY-2487	23/10/2026 0:00	23/10/2026 0:00	8.771927702630906	225	Improper Resource Sh	DUMMY Summary 248	SINGLE	HIGH	ADJACENT_NETWORK	NONE	COMPLETE
14	DUMMY-272	29/09/2020 0:00	29/09/2020 0:00	8.803650867220934	485	Resource Managemen	DUMMY Summary 272	NONE	HIGH	ADJACENT_NETWORK	PARTIAL	COMPLETE

Figure 2. Augmented CVE Dataset

4 Data Pre-Processing

Convert categorical values into numerical values using label encoding.

```
# List of categorical columns for label encoding

label_cols = ['access_authentication', 'summary',  'access_complexity', 'cwe_name','access_vector',

# Apply label encoding to the specified categorical columns

final_df[label_cols] = final_df[label_cols].apply(LabelEncoder().fit_transform)
```

Figure 3. Label Encoder

5 Data Visualization

Create Correlation Matrix among variables and plot it as a heatmap.

```
# Identify non-numeric columns, drop them, calculate correlation matrix, and visualize it as a heatmap.

non_numeric_columns = final_df.select_dtypes(exclude=['float64', 'int64']).columns

# Drop non-numeric columns
numeric_df = final_df.drop(columns=non_numeric_columns)

# Create a correlation matrix
correlation_matrix = numeric_df.corr()

# Plot the correlation matrix as a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix')
plt.show()
```

Figure 4. Correlation Matrix

6 Split Dataset

Using train_test_split , divide the data into training and testing sets.

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)
```

```
# Standardize (mean=0, std=1) the features in the training set (X_train) and apply the same transformation
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Figure 5. Training and Testing Dataset

7 Classifiers

Import and initialize all the models one by one.

Train the models using the trained dataset (real + augmented).

Plot confusion matrix, roc curve, auc for all the models

```
# Dictionary of classifiers with their corresponding instances: Support Vector Machine (SVC) and Logistic Regression


classifiers = {

    'Support Vector Machine': SVC(),
    'Logistic Regression': LogisticRegression()

}
```

```
#This code iterates through a dictionary of classifiers, trains each classifier on the training data, evaluates its


for name, classifier in classifiers.items():
    # Train the model
    classifier.fit(X_train, y_train)
```

Figure 6. Import Classifiers

8 Performance Evaluation

Use test dataset to assess the model's performance using relevant metrics , such as accuracy, precision, recall, and F1-score.

```
# Calculate and evaluate Isolation Forest model performance metrics: Accuracy, Precision, Recall, and F1 Score.
iso_accuracy = accuracy_score(y_test, iso_preds)  # Measures the overall correctness of the model predictions.
precision = precision_score(y_test, iso_preds)  # Quantifies the accuracy of positive predictions among all predicte
recall = recall_score(y_test, iso_preds)  # Captures the proportion of actual positives correctly predicted by the m
f1 = f1_score(y_test, iso_preds)  # Balances precision and recall, providing a harmonic mean of the two metrics.
```

Figure 7. Performance Evaluation