Configuration Manual

MSc Research Project
Programme Name

Venkata Nithin Krishna Singamsetty
Student ID: X23263326

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Venkata Nithin Krishna Singamsetty

Student ID: X23263326

Programme: MSc Cybersecurity **Year:** 2025

Module: MSc Practicum/Internship part 2

Lecturer: Michael Prior

Submission Due Date: 11-12-2024

Project Title: Detection of Flood and Brute force Attacks on IOT devices using hybrid model approach

Word Count: 801 **Page Count:** 5

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: Venkata Nithin Krishna

Date: 12-12-2024

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

Venkata Nithin Krishna Singamsetty
Student ID: X23263326

1 System Requirements:

Hardware Requirements:

- 8 GB RAM (16 GB recommended)
- 20 GB storage
- Multi-core CPU for faster computation.

Software or Libraries:

- Python 3.8
- Pandas
- Numpy
- Scikit-learn imblearn
- Matplotlib

Environment:

- Jupyter Notebook
- Google Colab (optional)

```
pip install pandas numpy scikit-learn imblearn matplotlib
```

This command installs all the necessary python libraries using pip which is a python package manager. It includes the tools which helps in analysis of data and manipulate it, performing mathematical operation on datasets, implementing ML algorithms, handling class imbalances in datasets, creating graphs to visualize the data.

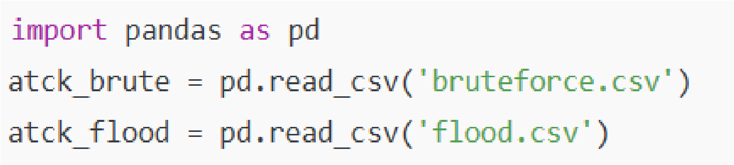
2 Data Preparation:

Datasets used in the research are bruteforce.csv and flood.csv.

- **bruteforce.csv**: Size of the dataset is 14,501 entries and 60 columns. Its class is labelled as **bruteforce.** It contains packet level details like frame.time_delta, tcp.srcport, tcp.dstport, etc
- **flood.csv:** Size of the datset is 613 entries and 60 columns and its class is labelled as **flood.** IT also contains similar data as bruteforce.

3 Data Preprocessing:

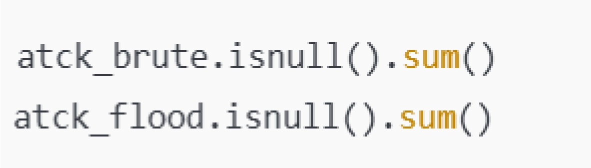
1. Loading the data



```
import pandas as pd
atck_brute = pd.read_csv('bruteforce.csv')
atck_flood = pd.read_csv('flood.csv')
```

First step is to load the data into pandas dataframes

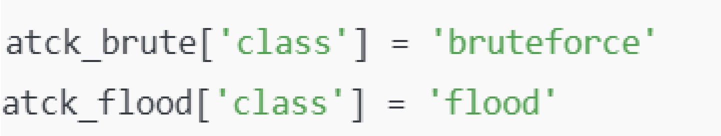
2. Null Value Analysis:



```
atck_brute.isnull().sum()
atck_flood.isnull().sum()
```

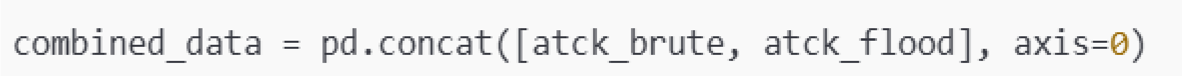
Next step is to ensure that there is no null value in the datframe. Check for the null value in the entire datframe and remove it.

3. Class Lable Assignment:



```
atck_brute['class'] = 'bruteforce'
atck_flood['class'] = 'flood'
```

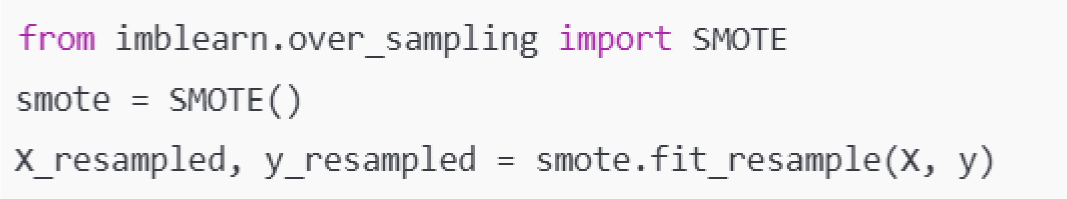
4. Merge the datasets:



```
combined_data = pd.concat([atck_brute, atck_flood], axis=0)
```

The two separate dataframes is combined as a single dataframe to analyse the entire data and perform operations on it.

5. Handling Class Imbalance :



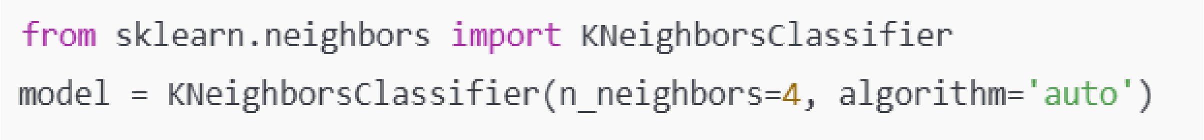
```
from imblearn.over_sampling import SMOTE
smote = SMOTE()
X_resampled, y_resampled = smote.fit_resample(X, y)
```

As there is a huge imbalance between bruteforce and flood data, imblearn library is used to deal this. SMOTE is used to balance the datasets.(‘SMOTE — Version 0.12.4’, no date)

4 Machine Learning Models:

Single ML Models:

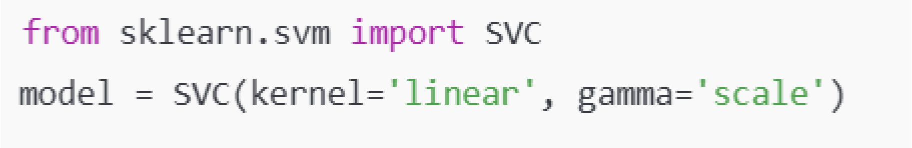
KNN:



```
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors=4, algorithm='auto')
```

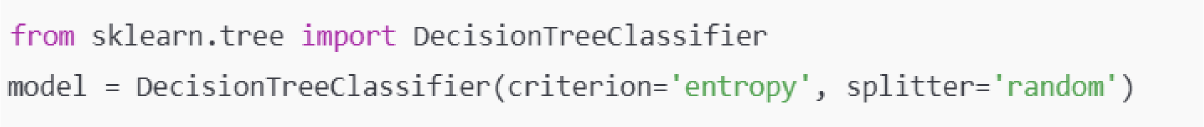
(*KNeighborsClassifier*, no date)

Support Vector Machine:



```
from sklearn.svm import SVC
model = SVC(kernel='linear', gamma='scale')
```

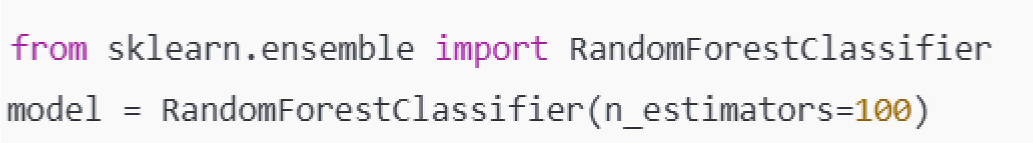
Decision Tree:



```
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(criterion='entropy', splitter='random')
```

(‘Decision Tree Classifier - an overview | ScienceDirect Topics’, no date)

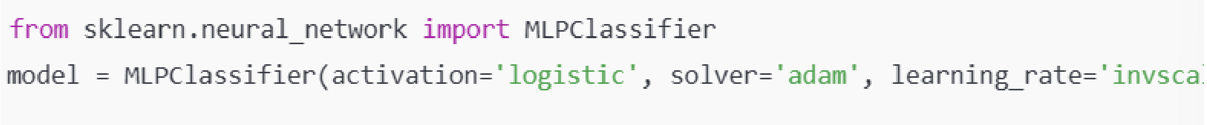
Random Forest:



```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100)
```

(*RandomForest Classifier*, no date)

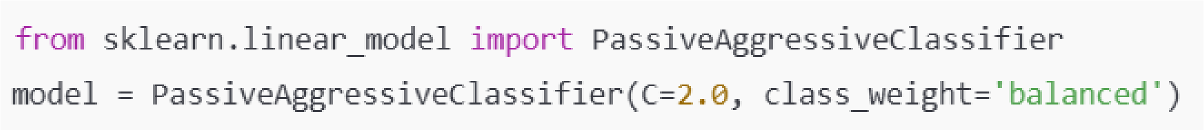
MLP:



```
from sklearn.neural_network import MLPClassifier
model = MLPClassifier(activation='logistic', solver='adam', learning_rate='invsca
```

(*MLPClassifier*, no date)

Passive Aggressive Classifier:

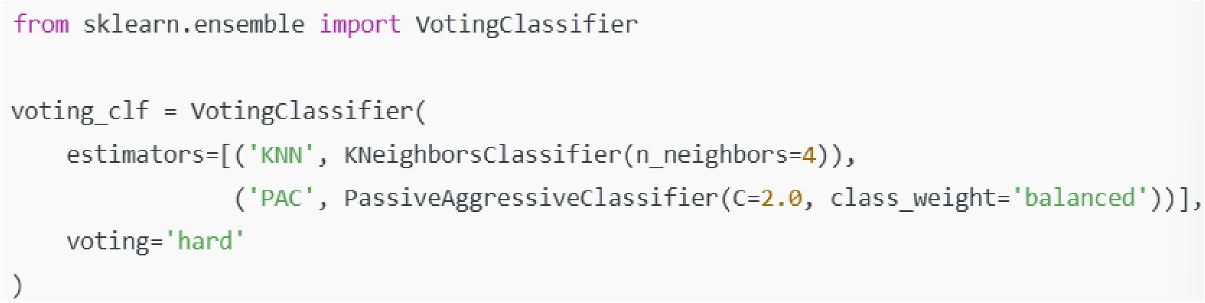


```
from sklearn.linear_model import PassiveAggressiveClassifier
model = PassiveAggressiveClassifier(C=2.0, class_weight='balanced')
```

Hybrid Model Configuration:

Hybrid Model is the combination of two or more single ML models for better performance. This approach used the Voting Classifier .(*VotingClassifier*, no date)

Configuration of KNN and Passive Classifier:



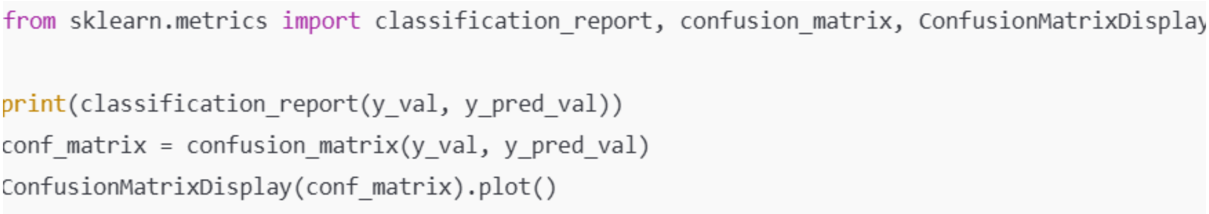
```
from sklearn.ensemble import VotingClassifier

voting_clf = VotingClassifier(
    estimators=[('KNN', KNeighborsClassifier(n_neighbors=4)),
                ('PAC', PassiveAggressiveClassifier(C=2.0, class_weight='balanced'))],
    voting='hard'
)
```

5 Model Training and Testing

Data Splitting: Combined dataset is spitted as follows:
Training set : 60 %
Validation set: 20%
Testing set: 20%

6 Performance Metrics

Each model is evaluated using the metrics like accuracy, precision, F1-score, confusion matrix. Example code for confusion matrix:(*confusion_matrix*, no date)



```
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay

print(classification_report(y_val, y_pred_val))
conf_matrix = confusion_matrix(y_val, y_pred_val)
ConfusionMatrixDisplay(conf_matrix).plot()
```

7 Results Analysis:

Single Model Results:

- KNN, SVM, DT, RF has acjeived highest accuracy up to 100%
- PC and MLP has showed lower accuracy equivalent to 50%

Hybrid Model Results:

- DT+SVM+RF has achieved the accuracy of 100%
- PC+KNN has achieved 50% accuracy

References

confusion_matrix (no date) *scikit-learn*. Available at: https://scikit-learn/stable/modules/generated/sklearn.metrics.confusion_matrix.html (Accessed: 12 December 2024).

'Decision Tree Classifier - an overview | ScienceDirect Topics' (no date). Available at: https://www.sciencedirect.com/topics/computer-science/decision-tree-classifier (Accessed: 11 December 2024).

KNeighborsClassifier (no date) *scikit-learn*. Available at: https://scikit-learn/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (Accessed: 12 December 2024).

MLPClassifier (no date) *scikit-learn*. Available at: https://scikit-learn/stable/modules/generated/sklearn.neural_network.MLPClassifier.html (Accessed: 12 December 2024).

RandomForestClassifier (no date) *scikit-learn*. Available at: https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (Accessed: 12 December 2024).

‘SMOTE — Version 0.12.4’ (no date). Available at: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html (Accessed: 12 December 2024).

VotingClassifier (no date) *scikit-learn*. Available at: https://scikit-learn/stable/modules/generated/sklearn.ensemble.VotingClassifier.html (Accessed: 12 December 2024).