

# **NETWORK INTRUSION DETECTION FOR IOT SECURITY BASED ON LEARNING TECHNIQUES**

**PRESENTED BY**

Presented by

- Jeevidesh.O.J (712217205026)
- JeyaShalini.S (712217205027)
- Lokeshwara kalaivani.G (712217205031)
- Shoba.A (712217205063)

Supervised by

-MR.S.Raja  
Assistant Professor  
Information Technology  
Park College Of Engineering  
And Technology

# INTRODUCTION

- Network Intrusion Detection system plays a essential part in defending the malicious attacks.
- An intrusion detection system is a device or software application that monitors a network or system for policy violations and suspicious activities.
- Intrusion Detection is efficient for analysing different type of attacks, identifies patterns of malicious content and implement effective control.
- Network Intrusion Detection is based on various Machine learning techniques that have been carried out for finding the cause of problems.

# ABSTRACT

- Intrusion detection is one of the important security problems in today's cyber world.
- A significant number of techniques have been developed which are based on machine learning approaches. So for identifying the intrusion we have designed the machine learning algorithms.
- By using the algorithm we find out intrusion and we can identify the attacker's details also.
- In our project we are proposing two types of algorithm (Decision tree algorithm ), (KNN classification algorithm).Using these algorithm we are going to find out which algorithm gives the best result to detect intrusion attack.

# LITERATURE SURVEY

Year	Title	Concept	Author	Drawback
2018	Evolutionary computation for feature selection in classification problems	Data analyzed by data mining algorithms can involve a large number of redundant or irrelevant features or simply too many features for a learning algorithm to handle them efficiently. Feature selection is becoming essential as databases grow in size and complexity.	B. de la Iglesias	It does not used in real time applications. It has high latency.

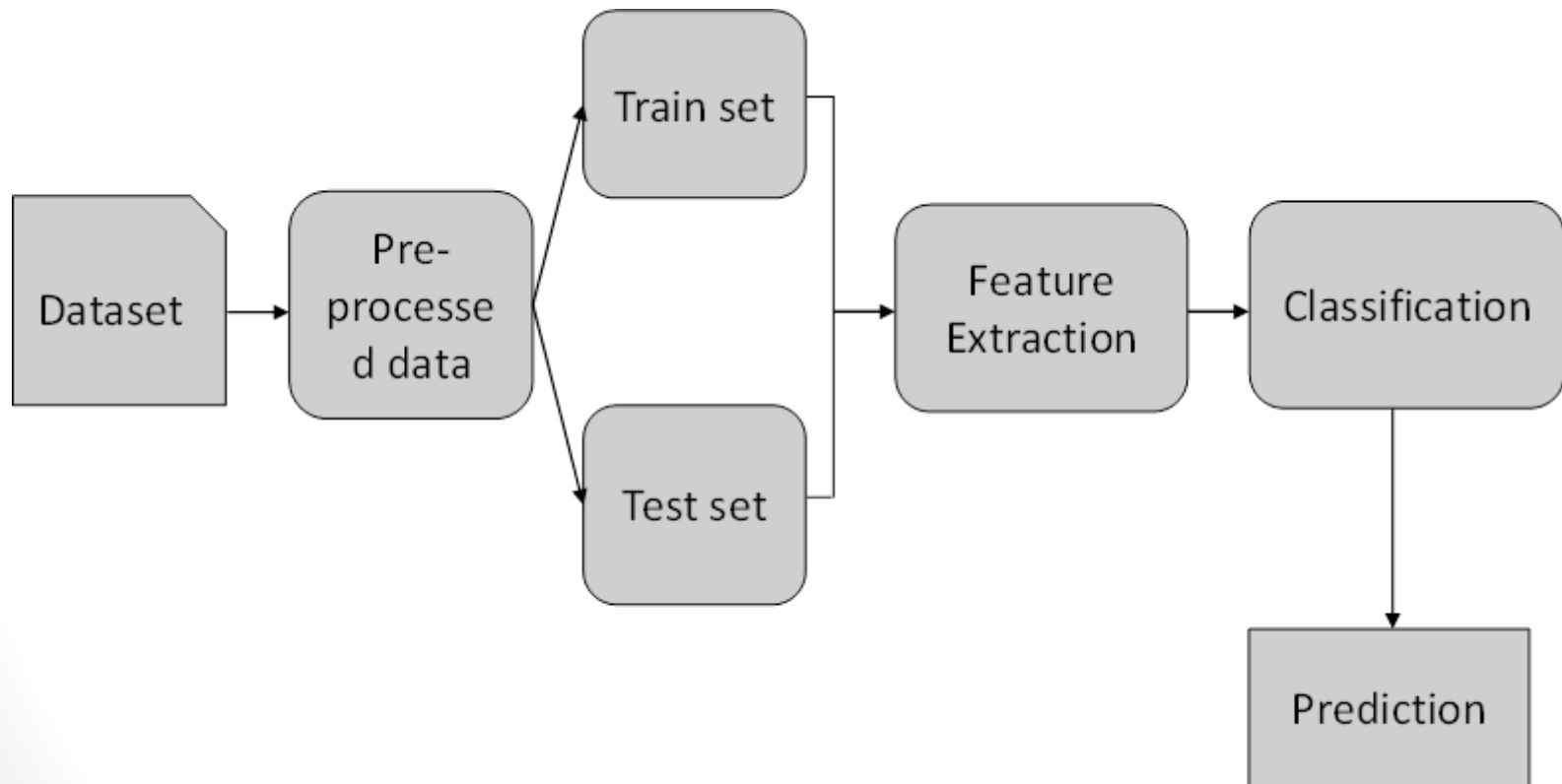
Year	Title	Concept	Author	Drawback
2019	On the use of MapReduce for imbalanced big data using Random Forest	In this work, we analyse the performance of several techniques used to deal with imbalanced datasets in the big data scenario using the Random Forest classifier. The Random Forest classifier provides a solid basis for the comparison because of its performance, robustness and versatility.	S. del R'io, V. L'opez, J. M. Ben'itez, and F. Herrera	Doesn't provide the accurate result.

Year	Title	Concept	Author	Drawback
2018	Fast, scalable and cloud-ready tool for the interactive genomic data analysis with nucleotide precision	Many time-consuming analyses of next generation sequencing data can be addressed with modern cloud computing. The Apache Hadoop based solutions have become popular in genomics because of their scalability in a cloud infrastructure.	M. S.Wiewiorka , A. Messina, A. Pacholewska , S. Maffioletti, P.Gawrysiak, and M. J. Okoniewski	It is not recommended with small quantities of data.

# PROPOSED SYSTEM

- This system effectively classify and predict the attack which is presented in the network.
- This system will increase the accuracy of the classification results by classifying the data based on the intrusion detection dataset and others using KNN algorithm and Decision Tree.
- It enhances the performance of the overall classification results.

# SYSTEM DESIGN



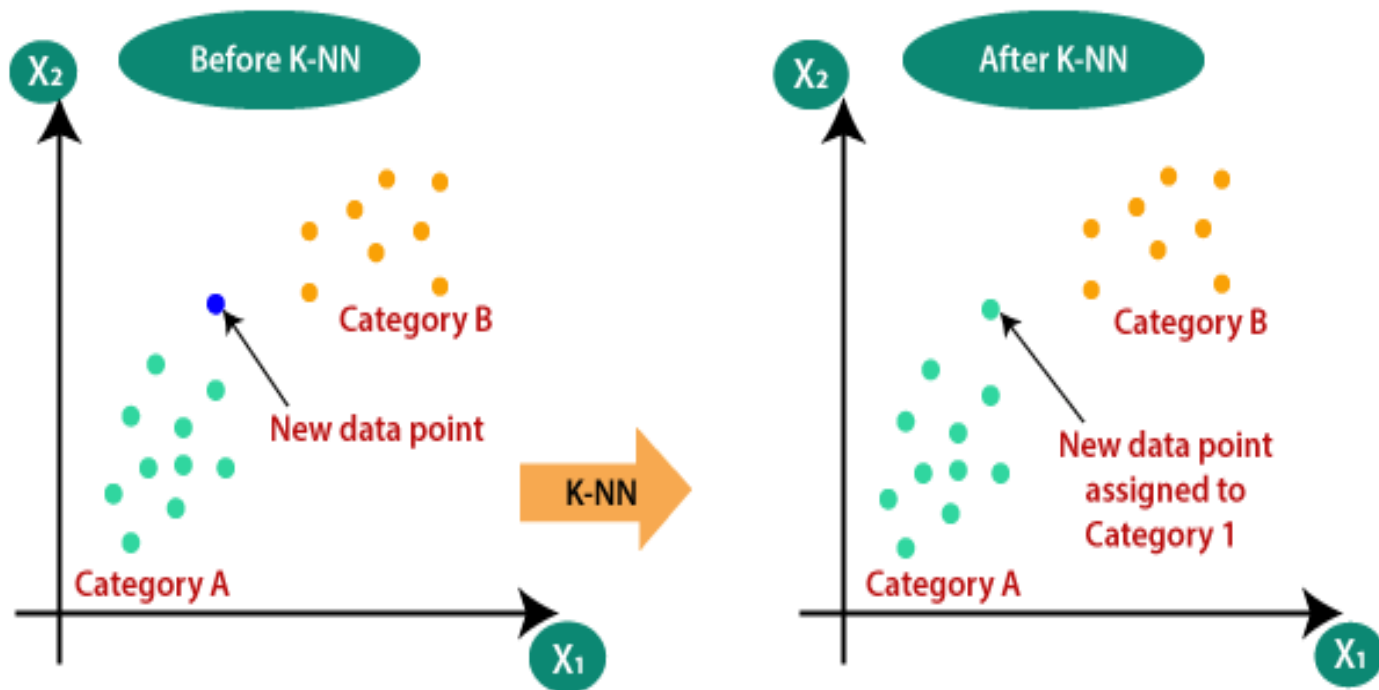


# DECISION TREE ALGORITHM

- Decision Tree Algorithm belongs to the family of supervised learning algorithms
- Unlike other supervised learning algorithms , the decision tree algorithm can be used for solving regression and classification problems.
- The goal of using a Decision Tree is to create a Training model that can use to predict the class or values of the target variable by learning simple decision rules inferred from prior data(Training Data).
- Decision trees doesn't need much energy for information training during pre-processing.

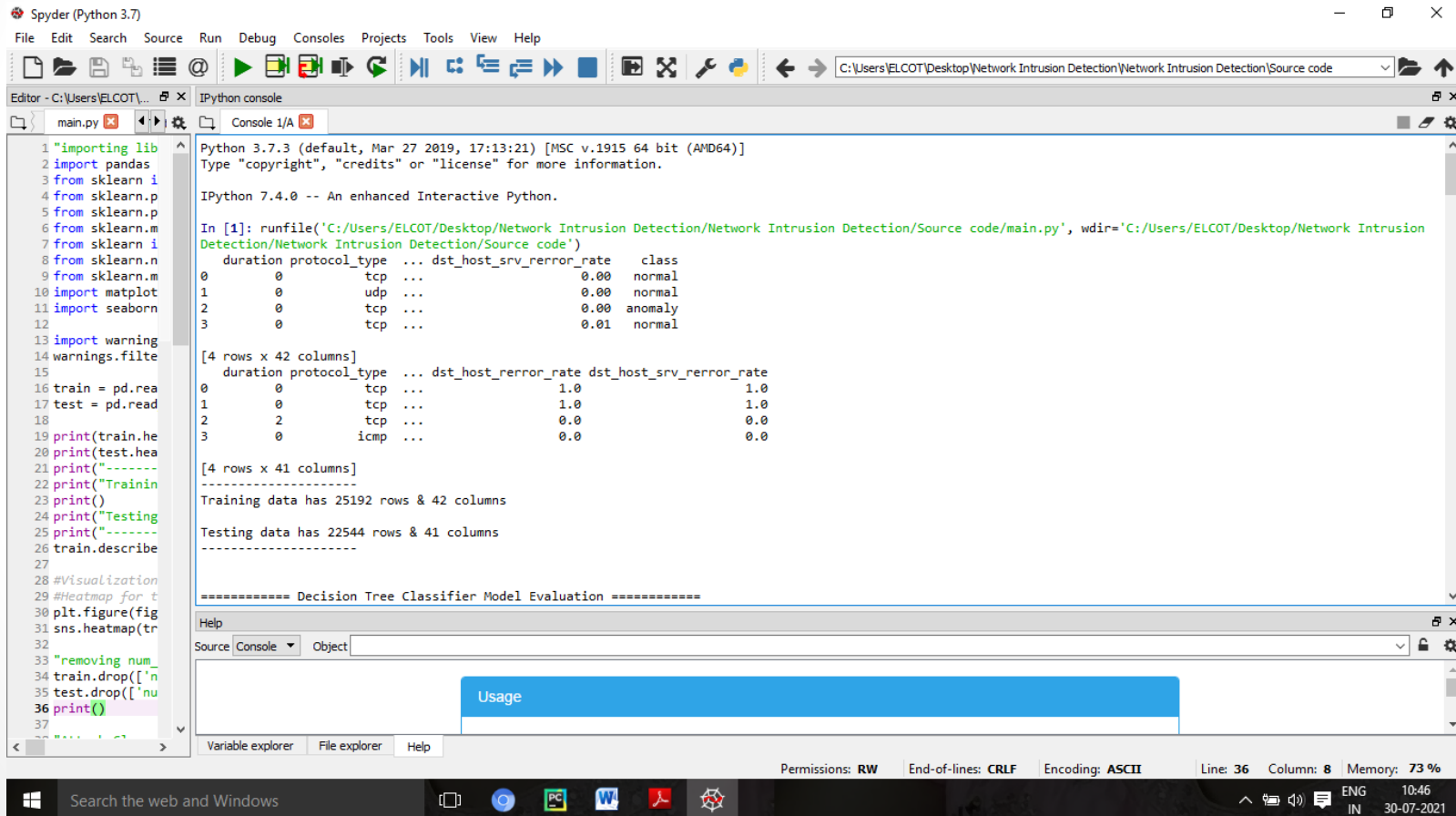
# K NEAREST NEIGHBOUR ALGORITHM

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- KNN is simple to implement . It can be more effective if the training data is large.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.



# OUTPUT:

## Training and Testing data set rows and columns



```
1 importing lib
2 import pandas
3 from sklearn.i
4 from sklearn.p
5 from sklearn.p
6 from sklearn.m
7 from sklearn.i
8 from sklearn.n
9 from sklearn.m
10 import matplotlib
11 import seaborn
12
13 import warnings
14 warnings.filte
15
16 train = pd.rea
17 test = pd.read
18
19 print(train.he
20 print(test.he
21 print("-----
22 print("Trainin
23 print()
24 print("Testing
25 print("-----
26 train.describe
27
28 #Visualization
29 #Heatmap for t
30 plt.figure(fig
31 sns.heatmap(tr
32
33 "removing num_
34 train.drop(['n
35 test.drop(['nu
36 print()
37
```

Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.

IPython 7.4.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/ELCOT/Desktop/Network Intrusion Detection/Network Intrusion Detection/Source code/main.py', wdir='C:/Users/ELCOT/Desktop/Network Intrusion Detection/Network Intrusion Detection/Source code')

duration	protocol_type	...	dst_host_srv_error_rate	class
0	0	tcp	...	0.00 normal
1	0	udp	...	0.00 normal
2	0	tcp	...	0.00 anomaly
3	0	tcp	...	0.01 normal

[4 rows x 42 columns]

duration	protocol_type	...	dst_host_error_rate	dst_host_srv_error_rate
0	0	tcp	...	1.0 1.0
1	0	tcp	...	1.0 1.0
2	2	tcp	...	0.0 0.0
3	0	icmp	...	0.0 0.0

[4 rows x 41 columns]

Training data has 25192 rows & 42 columns

Testing data has 22544 rows & 41 columns

-----

===== Decision Tree Classifier Model Evaluation =====

Help

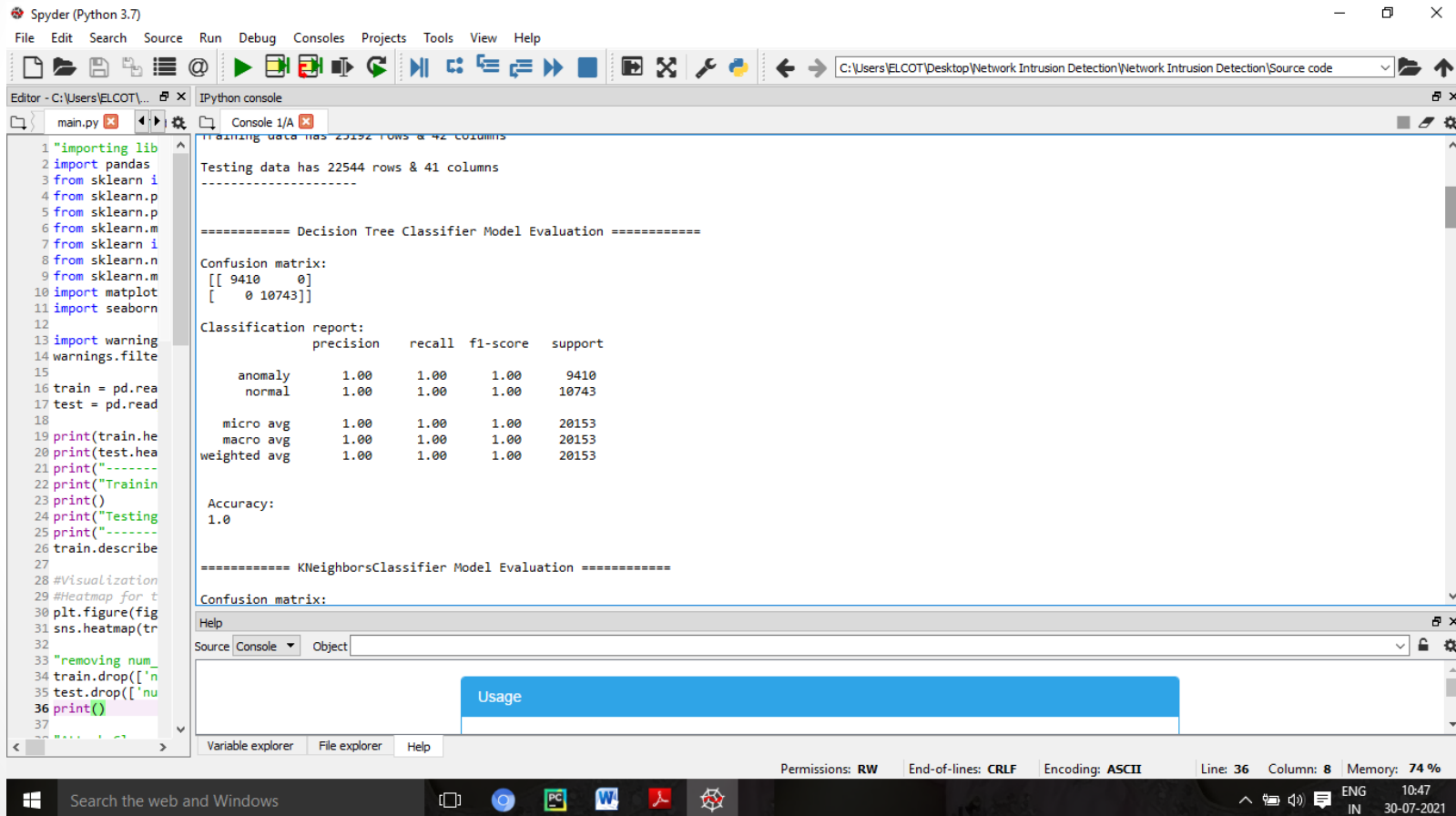
Source Console Object

Usage

Variable explorer File explorer Help

Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 36 Column: 8 Memory: 73 %

# Decision tree classifier accuracy



```
1 "importing lib
2 import pandas
3 from sklearn.i
4 from sklearn.p
5 from sklearn.p
6 from sklearn.m
7 from sklearn.i
8 from sklearn.n
9 from sklearn.m
10 import matplotlib
11 import seaborn
12
13 import warnings
14 warnings.filter
15
16 train = pd.read
17 test = pd.read
18
19 print(train.head
20 print(test.head
21 print("-----
22 print("Training
23 print()
24 print("Testing
25 print("-----
26 train.describe
27
28 #Visualization
29 #Heatmap for t
30 plt.figure(figsize
31 sns.heatmap(tr
32
33 "removing num
34 train.drop(['n
35 test.drop(['nu
36 print()
37
```

Training data has 23192 rows & 42 columns

Testing data has 22544 rows & 41 columns

-----

===== Decision Tree Classifier Model Evaluation =====

Confusion matrix:

```
[[ 9410   0]
 [   0 10743]]
```

Classification report:

	precision	recall	f1-score	support
anomaly	1.00	1.00	1.00	9410
normal	1.00	1.00	1.00	10743
micro avg	1.00	1.00	1.00	20153
macro avg	1.00	1.00	1.00	20153
weighted avg	1.00	1.00	1.00	20153

Accuracy:

```
1.0
```

===== KNeighborsClassifier Model Evaluation =====

Confusion matrix:

Help

Source: Console Object

Usage

Variable explorer File explorer Help

Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 36 Column: 8 Memory: 74 %

Search the web and Windows

ENG 10:47 IN 30-07-2021

# Kneighbour classifier accuracy

The screenshot displays the Spyder Python IDE interface. The left pane shows the code for training and evaluating a KNeighborsClassifier. The right pane shows the output of the code execution.

**Code (main.py):**

```
1 "importing lib
2 import pandas
3 from sklearn i
4 from sklearn.p
5 from sklearn.p
6 from sklearn.m
7 from sklearn i
8 from sklearn.n
9 from sklearn.m
10 import matplot
11 import seaborn
12
13 import warning
14 warnings.filte
15
16 train = pd.rea
17 test = pd.read
18
19 print(train.he
20 print(test.he
21 print("-----
22 print("Trainin
23 print()
24 print("Testing
25 print("-----
26 train.describe
27
28 #Visualization
29 #Heatmap for t
30 plt.figure(fig
31 sns.heatmap(tr
32
33 "removing num
34 train.drop(['n
35 test.drop(['nu
36 print()
37
```

**Output (Console 1/A):**

```
micro avg      1.00      1.00      1.00      20153
weighted avg    1.00      1.00      1.00      20153

Accuracy:
1.0

===== KNeighborsClassifier Model Evaluation =====

Confusion matrix:
[[ 9335   75]
 [   35 10708]]

Classification report:
      precision    recall  f1-score   support

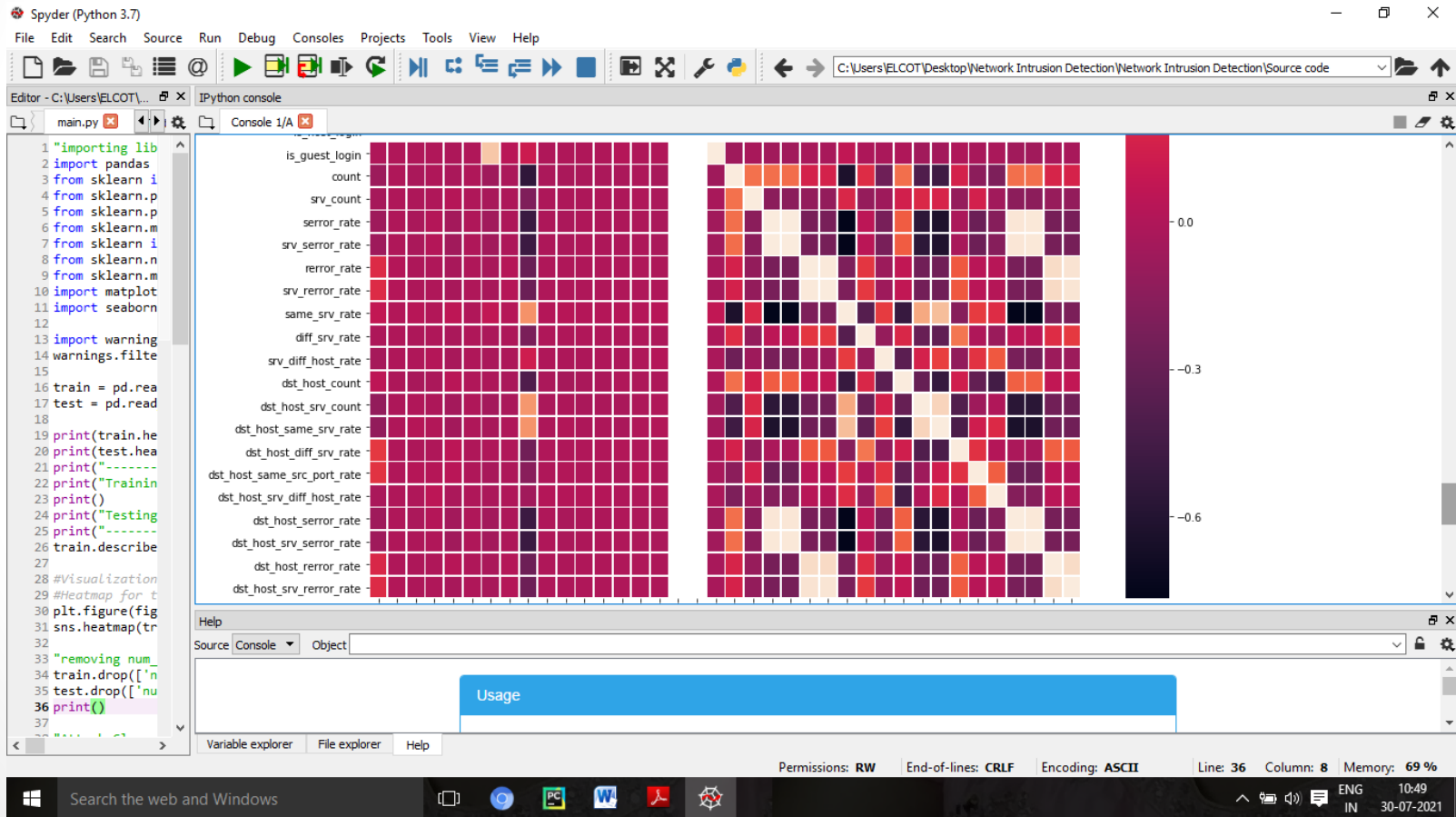
 anomaly      1.00      0.99      0.99      9410
  normal      0.99      1.00      0.99     10743

 micro avg      0.99      0.99      0.99     20153
 macro avg      0.99      0.99      0.99     20153
weighted avg      0.99      0.99      0.99     20153

Accuracy:
0.9945417555698903
```

The bottom status bar shows: Permissions: RW | End-of-lines: CRLF | Encoding: ASCII | Line: 36 | Column: 8 | Memory: 74 %

# Heat Map



# SYSTEM REQUIREMENTS

## Software Requirements

- Operating System : Windows 7,8,9,10
- Language : Python
- IDE : Anaconda –Spyder

## Hardware Requirements

- Hard disk : 1000GB
- RAM : 4GB



# CONCLUSION

- We reviewed several influential algorithms for intrusion detection based on various machine learning techniques.
- The Machine Learning Algorithm such as KNN Algorithm and Decision Tree algorithm gives the correct accuracy rate and its helps to predict.
- These two algorithms share many similarities such as adaptation, fault tolerance, high computational speed and error resilience in the face of noisy information, conform the requirement of building efficient intrusion detection systems.

# FUTURE WORK

- In future, it is possible to provide extensions or modifications to the proposed clustering and classification algorithms using intelligent agents to achieve further increased performance.
- Apart from the experimented combination of data mining techniques, further combinations such as artificial intelligence, soft computing and other clustering algorithms can be used to improve the detection accuracy.
- Finally, the intrusion detection system can be extended as an intrusion prevention system to enhance the performance of the system.

THANK YOU