

AN EFFICIENT TRAFFIC MONITORING AND IDENTIFICATION BASED ON CATEGORIZATION IN INTERNET ACCESS



Submitted in partial fulfillment of the requirements for the degree of

**BACHELOR OF TECHNOLOGY
in
Computer Science and Engineering**

by

**SAACHI JAISWAL
18K91A05J5**

**Under the guidance of
DR.S.A KALAISELVAN
Professor**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
TKR COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
(ACCREDITED BY NBA AND NAAC WITH 'A' GRADE)
Medbowli, Meerpeta, Saroornagar, Hyderabad-500097

DECLARATION BY THE CANDIDATE

I, **SAACHI JAISWAL** bearing Hall Ticket Number **18K91A05J5** , hereby declare that the main project report titled ”**AN EFFICIENT TRAFFIC MONITORING AND IDENTIFICATION BASED ON CATEGORISATION IN INTERNET ACCESS**” under the guidance of **Dr.S.A.KALAISELVAN, Professor** in Department of Computer Science and Engineering is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

Signature By Candidates

18K91A05J5

Place: Meerpet

Date:

CERTIFICATE

This is to certify that literature survey report entitled **AN EFFICIENT TRAFFIC MONITORING AND IDENTIFICATION BASED ON CATEGORIZATION IN INTERNET ACCESS.**, being submitted by Ms.SAACHI JAISWAL bearing Hall Ticket Number: **18K91A0-5J5**, in partial fulfillment of requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering, to the TKR College of Engineering and Technology is a record of bonafide work carried out by him/her under my guidance and supervision.

Signature of the Guide

Dr.S.A.Kalaiselvan

Professor

Signature of the HoD

Dr.A.Suresh Rao

Professor

Place: Meerpet

Date:

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGEMENTS	ii
LIST OF FIGURES	iii
1 INTRODUCTION	1
1.1 Problem definition	1
1.2 Limitations of existing system	1
1.3 Proposed system	2
2 LITERATURE REVIEW	3
2.1 Review of Literature	3
2.2 Network Traffic Classification Techniques	3
2.2.1 A. Port-Based Technique	3
2.2.2 B. Payload-Based Technique	3
2.2.3 C. Machine Learning (ML) Technique	4
2.3 Network Traffic Classification Model	4
2.3.1 A. Network Traffic Capture	4
2.3.2 B. Defining classes of websites to be classified	4
2.3.3 D. Classification	4
2.4 2.3 Packages required	4
3 REQUIREMENTS ANALYSIS	7
3.1 Software Requirements Specification	7
3.2 Functional Requirements	7
3.3 Non-Functional Requirements	7
3.3.1 i. Security	7
3.3.2 ii. Usability	7
3.3.3 iii. Reliability	7
3.3.4 iv. Performance	7
3.4 3.4 Feasibility Study	7
3.4.1 Technical feasibility	8

3.4.2	Economic feasibility	8
3.4.3	Operational feasibility	8
3.5	Software Requirements	8
3.6	Hardware Requirements	8
4	DESIGN	9
4.1	SYSTEM ARCHITECTURE	9
4.2	UML diagrams	9
4.3	Usecase diagram	10
4.4	Class Diagram	10
4.5	Sequence Diagram	11
4.6	Activity Diagram	12
4.7	Deployment Diagram	13
4.7.1	Description	14
5	CODING	15
5.1	Sample Code	15
6	IMPLEMENTATION and RESULTS	18
6.1	Explanation of Key functions	18
6.2	Method of Implementation	18
6.2.1	UDP	18
6.2.2	TCP	18
6.2.3	Result Analysis	18
6.3	Method of Implementation	18
6.3.1	Packet Inspection	18
6.4	OUTPUT GRAPHS	22
6.4.1	GENERAL CONTENT	22
6.4.2	OTT CONTENT	23
6.4.3	NEWS	24
6.4.4	GAMES	25
7	TESTING and VALIDATION	26
7.1	Design of Test Cases and Scenarios	26
7.2	Validation	26

7.3	Testing	26
7.4	TESTING STRATEGIES	26
7.4.1	White Box Testing	26
7.4.2	Black Box Testing	26
7.5	LEVELS OF TESTING	27
7.5.1	Integration Testing	27
7.5.2	Acceptance Testing	27
7.6	Test Cases	27
8	CONCLUSION	28
	REFERENCES	28

ABSTRACT

Due to the progress in network advances, the number of organization clients is developing rapidly, which prompts age of enormous traffic data. This huge traffic information inclined to attacks and intrusions. Traffic observing and examination is fundamental inorder to all the more successfully investigate and determine issues when they occur. This paper talk about packet based strategy where internet traffic can without much of a stretch ordered into various sections. The information can be classified into movies, education, news, explicit content. The benefit of this methods is review of parcels can undoubtedly finished by utilizing UDP and TCP protocols. The traffic can classify quicker and less time is required.

Keywords: *Internet Traffic, Intrusions, Tcp, Udp.*

ACKNOWLEDGEMENTS

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance have crowned my efforts with success.

I am indebted to the Internal Guide, **Dr.S.A KALAISELVAN**, Professor, Dept. of Computer Science and Engineering, TKR College of Engineering and Technology, for his/her support and guidance throughout my Thesis/Dissertation.

I am also indebted to the **Head of the Department, Dr.A.Suresh Rao**, Professor, Computer Science and Engineering, TKR College of Engineering and Technology, for his support and guidance throughout my Thesis/Dissertation.

I extend my deep sense of gratitude to the **Principal, Dr.D.V.Ravi Shankar**, TKR College of Engineering and Technology, for permitting me to undertake this Thesis/Dissertation.

Finally, I express my thanks to one and all that have helped me in successfully completing this Thesis/Dissertation. Furthermore, I would like to thank my family and friends for their moral support and encouragement

SAACHI JAISWAL 18K91A05J5

Place: Meerpet

Date:

LIST OF FIGURES

4.1	System Architecture	9
4.2	Usecase diagram	10
4.3	Class Diagram	11
4.4	Sequence Diagram	12
4.5	Activity Diagram	13
4.6	Deployment Diagram	14
6.1	GENERAL	22
6.2	OTT	23
6.3	NEWS	24
6.4	GAMES	25

Chapter 1

INTRODUCTION

Network Traffic Classification is a critical subject nowadays in the field of Computer Science. It is incredibly central for Internet Service Providers (ISPs) to manage the overall show of an network. Traffic arrangement is the initial step to character and classify obscure organization classes. Network Traffic Classification assumes an exceptionally essential part in network security and the executives, like Intrusion Detection, Quality of Service (QoS)[2]. Through this procedure, network administrators can make a few moves, for example, to obstruct a few streams and oversee assets. They can likewise track down the development of organization applications. Over the most recent twenty years, various organization traffic order procedures [2] [3] have been proposed to group obscure classes. The first is Port Based Technique. It is a unimaginable method for network traffic classification/ID.

Dynamic port number means unregistered number with Internet Assign Number Authority (IANA). Then, second one is Payload Based strategy. This strategy gives exact results in network traffic request. This system is in like manner called Deep Packet Inspection (DPI) technique. However, there is an issue in this technique. The issue is that it can't be utilized for encoded information network applications as various organization applications utilize scrambled methods to safeguard information from identification.

1.1 Problem definition

Description: Institution of Large scale have difficulty in monitoring the internet traffic that is going to and from routers round the clock. An mechanism where all the traffic can intercepted and categorised into the categorise such as news, education, sexually explicit content, movies and so on.

1.2 Limitations of existing system

In this section, we analyze relatively few systems for network traffic request. Two sorts of techniques are accessible for network traffic classification: port-based and pay load based.[7]

Port-based procedures: In Port-based methods network traffic distinguishing proof should be possible in view of the port numbers. In these methods, numerous applications utilize the port number relegated by (IANA) Internet Assigned Numbers Authority on a local host in the network, which will be utilized as a gathering point and different hosts might start correspondence utilizing that nearby host.[7] Tragically, these methods experiences the accompanying drawbacks: As the applications are developing, they might utilize flighty port numbers.[7] These procedures may not be appropriate when a few applications didn't enlisted their port numbers with IANA.[7] Thus, more refined characterization procedures ought to be utilized for traffic arrangement. Accordingly, payload-based strategies are introduced.(2) Payload based techniques:In payload-based strategies, which is otherwise called Deep Packet Inspection (DPI), the items in the bundles are seen by alluding trademark marks of organization applications in the traffic.[7] The greater part of the payload based methods reviews the items in the parcel and attempts to coordinate them with a bunch of marks put away in the data set. These methods give exact outcomes contrasted with Port based techniques.[7]

1.3 Proposed system

We propose a solution that is based on network packet inspection. We propose a lightweight model that works with any load and across different protocols.

Chapter 2

LITERATURE REVIEW

2.1 Review of Literature

2.2 Network Traffic Classification Techniques

Network Traffic Classification is the cycle to character the association applications or show that exists in an organization [1]. Network traffic plan has uncommon significance over the latest twenty years. Researchers have proposed various procedures to arrange network applications. In this part, we analyze Port-based Technique, Payload Based Technique and Machine Learning (ML) strategies.

2.2.1 A. Port-Based Technique

As we referred to in fragment I that for the most part, in this methodology, a request for association applications is executed using the striking ports number. Also, we furthermore discussed that first organization applications are enrolled in their ports in the Internet Assigned Number Authority (IANA). In along these lines, the traffic is distinguished relating to the enlisted ports number enlisted in IANA. Table shows various kinds of uses and its ports numbers allocated by IANA. For example, E-mail applications utilize 25 (SMTP) port number to send messages and to get email 110 (POP3) port is utilized. [n along these lines, web applications utilize 80 ports number.

2.2.2 B. Payload-Based Technique

This strategy is likewise called Deep Packet Inspection method (DPI). [n this procedure, the items in the bundles are analyzed looking attributes marks of the organization applications in the rush hour gridlock. This is the essential choice rather than ports-based procedure. This methodology is extraordinarily proposed for Peer to Peer (P2P) applications. it implies applications which utilize dynamic port number to character traffic in an organization. The following is the table, which represents models, utilized by Karagiannis et al. in [4].

2.2.3 C. Machine Learning (ML) Technique

AI (ML) strategy [5],[6] depends on informational index (Labeled Data Set) . In this strategy, an AI classifier is prepared as information and afterward utilizing the prepared example forecast, obscure classes are characterized. There are two fundamental regions in AI procedure: the regulated and solo learning strategy..

2.3 Network Traffic Classification Model

In this segment, we make sense of the organization traffic grouping structure model, which incorporates bit by bit process. This bit by bit process strategy will tell you the best way to utilize network traffic order method to personality/group obscure organization traffic classes.

2.3.1 A. Network Traffic Capture

This is the first and most significant stage, which incorporates information assortment. In this progression, the ongoing organization traffic is caught. It is otherwise called information assortment step. There are many apparatuses for network traffic catching, yet Tcp dump device can be utilized to catch the continuous organization traffic. To catch network traffic, we use Wire Shark instrument [IS] for parcel catching and breaking down. We got the traffic the range of one snapshot of WWW, DNS, FTP, P2P and Telnet application.

2.3.2 B. Defining classes of websites to be classified

In our python code we need to specify the classes and data into a list which can later be used for pattern matching. C. Logging In this step, we will log the UDP traffic into a file which can be used as database for future classification.

2.3.3 D. Classification

This will be the last step. In this step we will add a daemon job that runs constantly and checks for any new traffic. Once the traffic is logged it immediately consumes the data prints the classification on the console.

2.4 2.3 Packages required

2.4.0.1 a. SciPy

SciPy (explained "Mumble Pie") is open-source programming for math, science, and planning. The SciPy library depends upon NumPy, which gives invaluable and fast N-layered group

control. The SciPy library is attempted to work with NumPy displays, and gives various simple to utilize and capable numerical timetables like timetables for numerical blend and progression. Together, they run on all popular working structures, hurry to present, and are in vain. NumPy and SciPy are easy to use, but sufficiently able to be depended on by a piece of the world's driving scientists and subject matter experts. If you truly have any desire to control numbers on a PC and show or convey the results, look at SciPy!.

2.4.0.2 b. OS

This module gives a compact approach to utilizing working framework subordinate usefulness. To peruse or compose a record see `open()`, to control ways, see the `os.path` module, and to peruse every one of the lines in every one of the documents on the order line see the `fileinput` module. For making impermanent records and catalogs see the `tempfile` module, and for undeniable level document and registry dealing with see the `shutil` module.

2.4.0.3 c. Pandas

pandas is a Python bundle that gives fast, versatile, and expressive information structures expected to make working with "social" or "stamped" data both straightforward and instinctual. It plans to be the crucial undeniable level structure block for doing commonsense, genuine information examination in Python. Also, it has the more extensive objective of turning into the most impressive and adaptable open source information examination/control apparatus accessible in any language. It is as of now well on its way towards this objective.

2.4.0.4 d. base64

This module gives capacities to encoding double information to printable ASCII characters and disentangling such encodings back to twofold information. It gives encoding and translating capacities to the encodings indicated in RFC 4648, which characterizes the Base16, Base32, and Base64 calculations, and for the accepted standard Ascii85 and Base85 encodings.

2.4.0.5 e. Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical number related development NumPy. It gives an article arranged API to introducing plots into applications using all around valuable GUI device stash like Tkinter, wxPython, Qt, or GTK.

There is moreover a procedural "pylab" interface considering a state machine (like OpenGL), planned to eagerly seem to be that of MATLAB, but its use is deflected. [2] SciPy uses Matplotlib.

Chapter 3

REQUIREMENTS ANALYSIS

3.1 Software Requirements Specification

3.2 Functional Requirements

- pandas
- scipy
- Matplotlib
- Base64
- os

3.3 Non-Functional Requirements

3.3.1 i. Security

a. We are providing security to our application means that there should be no hacking of information.

3.3.2 ii. Usability

a. We should get response within seconds. b. The application must have a simple, user friendly interface to save time and confusion.

3.3.3 iii. Reliability

The application is reliable because of the qualities that are inherited by the reliable Web application standards.

3.3.4 iv. Performance

The application is high performing as the user's interaction with the online pharmacy management system is responded within seconds which describe the performance of the application.

3.4 Feasibility Study

Feasibility study is conducted once the problem is clearly understood. The possibility concentrate on which is a significant level container rendition of the whole framework examination and configuration process. The objective is to determine whether the proposed system is fea-

sible or not and it helps us to the minimum expense of how to solve the problem and to determine, if the problem is worth solving. The following are the three important tests that have been carried out for feasibility study.

3.4.1 Technical feasibility

In the technical feasibility study, one has to test whether the proposed system can be developed using existing technology or not. It is planned to implement the proposed system in Python. The project entitled is technically feasible because of the following reasons. (a) All necessary technology exists to develop the system. (b) The existing system is so flexible that it can be developed further.

3.4.2 Economic feasibility

As a part of this, the costs and benefits associated with the proposed systems are to be compared. The project is economically feasible only if tangible and intangible benefits outweigh the cost. We can say the proposed system is feasible based on the following grounds. (a) The cost of developing the full system is reasonable. (b) The cost of hardware and software for the application is less.

3.4.3 Operational feasibility

The project is operationally feasible because there is sufficient support from the project management and the users of the proposed system. Proposed system definitely does not harm and will not produce the bad results and no problem will arise after implementation of the system.

3.5 Software Requirements

Operating System : Minimum Windows 8/10

Coding Language : Python.

3.6 Hardware Requirements

Processor : Minimum Intel i3

Clock Speed : 1.7 GHz

Hard Disk : Minimum 250 GB

RAM : Minimum 4 GB

Chapter 4

DESIGN

4.1 SYSTEM ARCHITECTURE

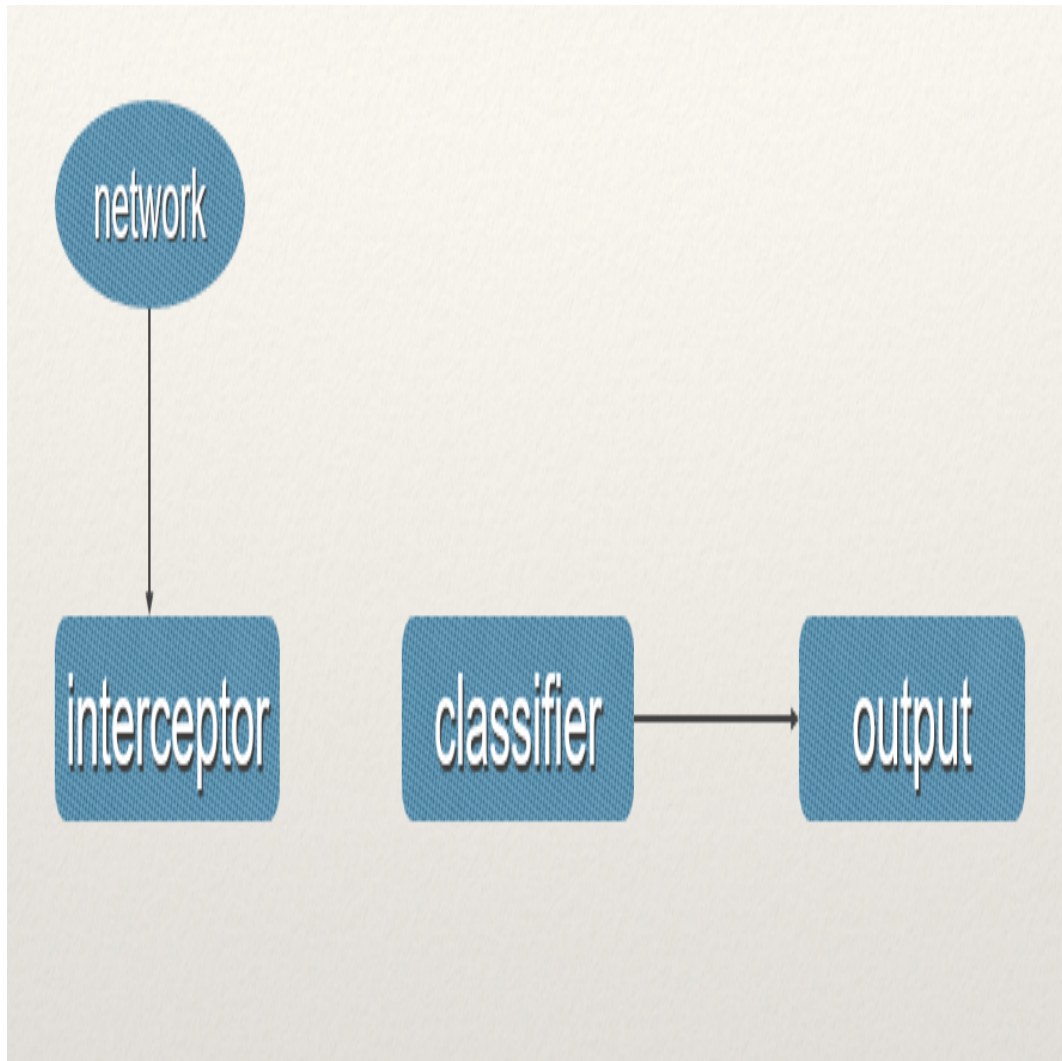


Fig. 4.1 System Architecture

As mentioned in literature review all the steps are perfectly designed to follow standard architecture depicted by the picture above to extract the results in most efficient way.

4.2 UML diagrams

The list of UML diagrams is as follows:

- Use case diagram

- Class diagram
- Sequence diagram
- Activity diagram
- Deployment diagram

4.3 Usecase diagram

A use case portrays succession of activities that give something of quantifiable worth to an entertainer and is drawn as a flat oval. An entertainer is an individual, association or broadened framework that assumes a part in at least one cooperations with the framework.

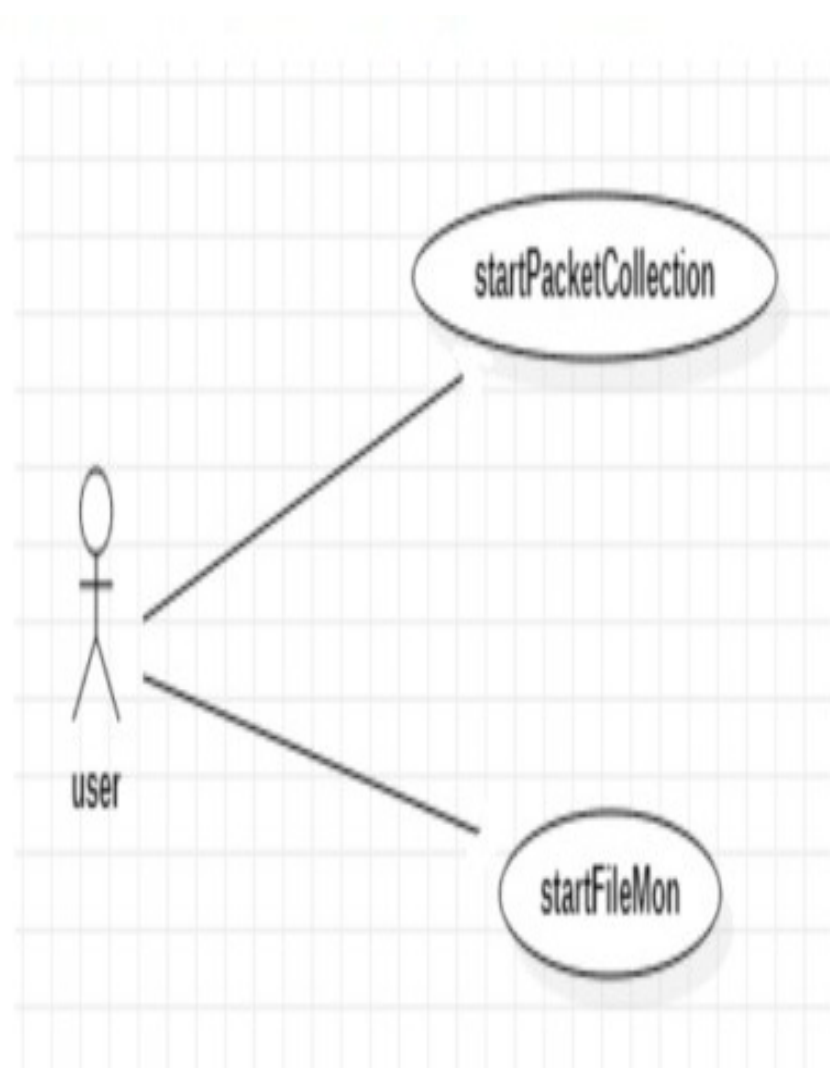


Fig. 4.2 Usecase diagram

4.4 Class Diagram

Classes are the main structure blocks of any item - situated language. A class is a bunch of items that share similar traits, tasks and connections.

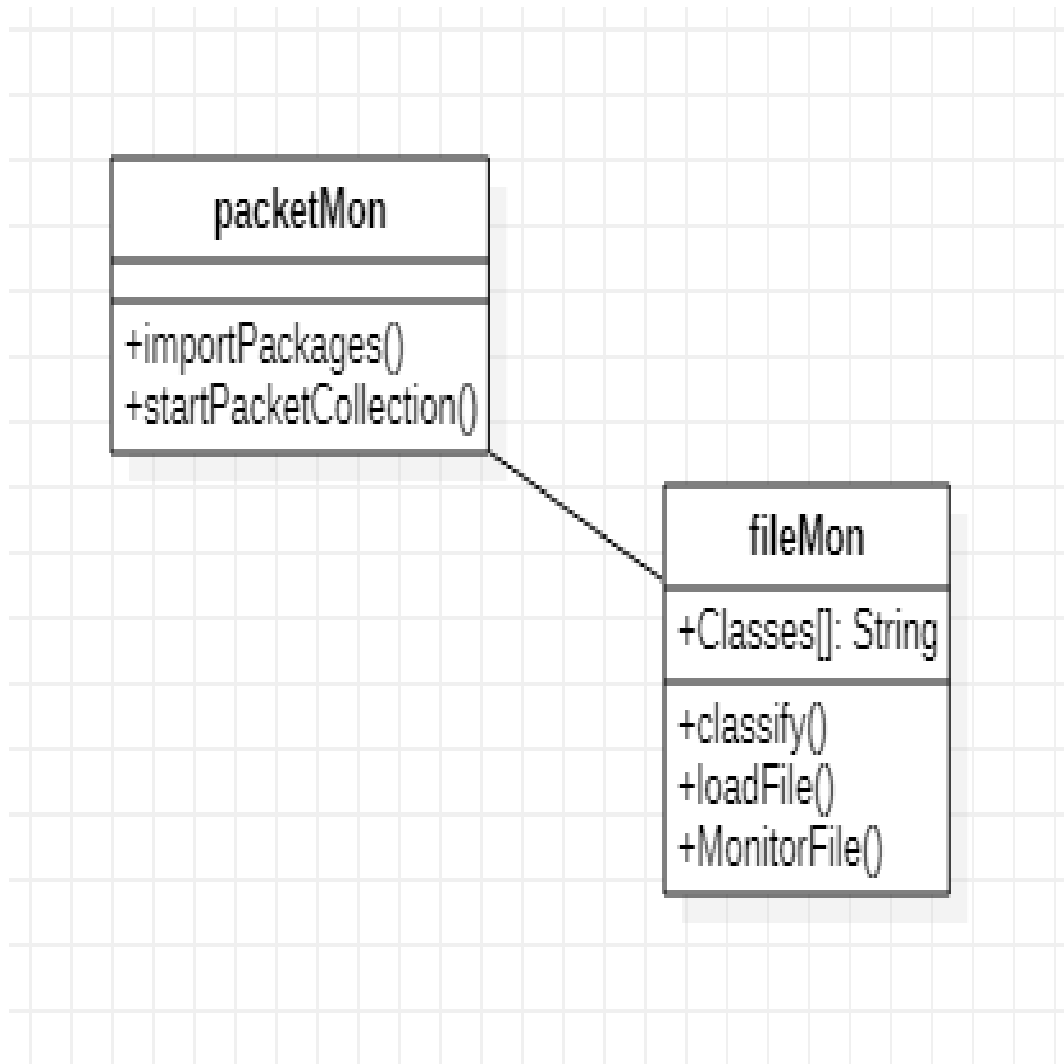


Fig. 4.3 Class Diagram

4.5 Sequence Diagram

Sequence diagram model the progression of rationale inside your framework in a visual way, empowering you both to report and approve your rationale, and are ordinarily utilized for both examination and configuration purposes.

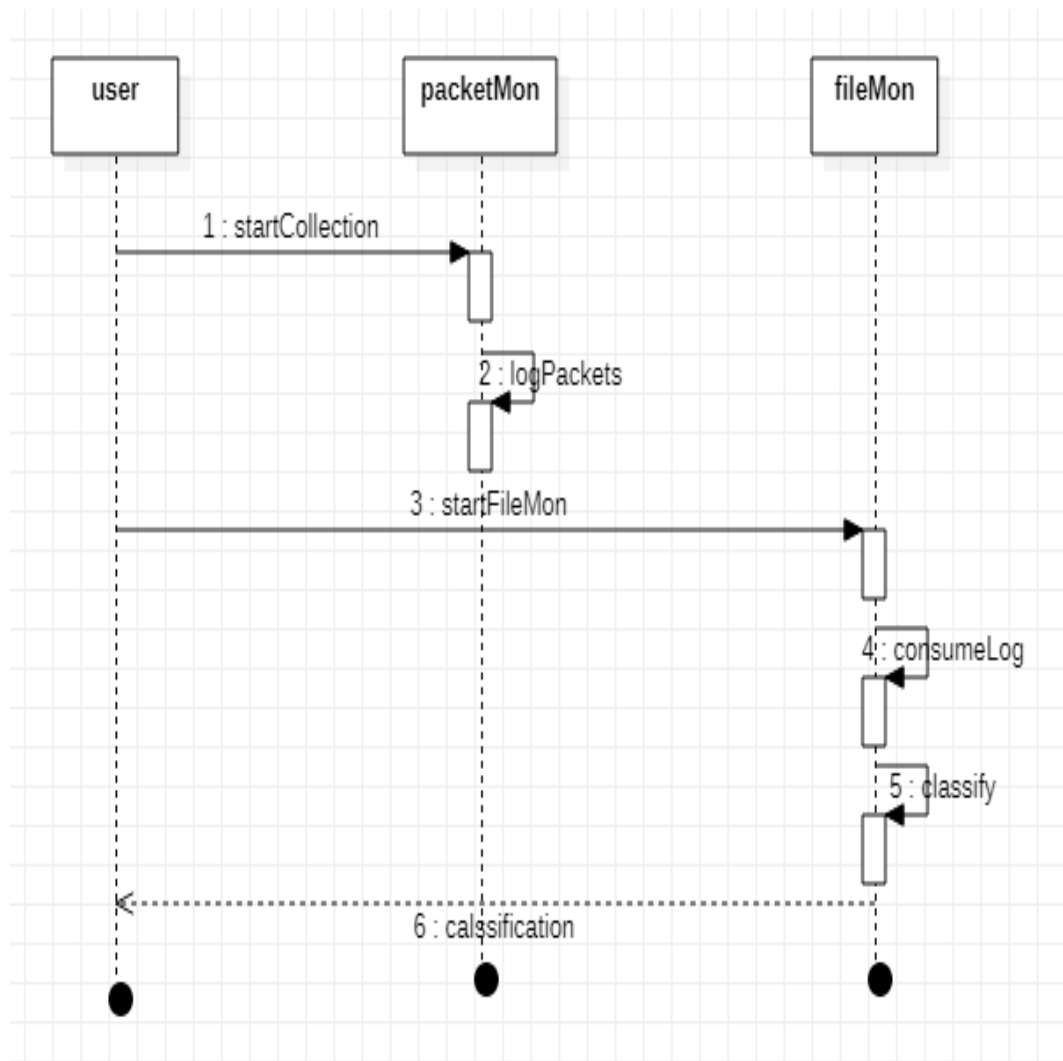


Fig. 4.4 Sequence Diagram

4.6 Activity Diagram

An activity diagram kind of graph that address a work process or a cycle . A flowchart can likewise be characterized as a diagrammatic portrayal of a calculation.

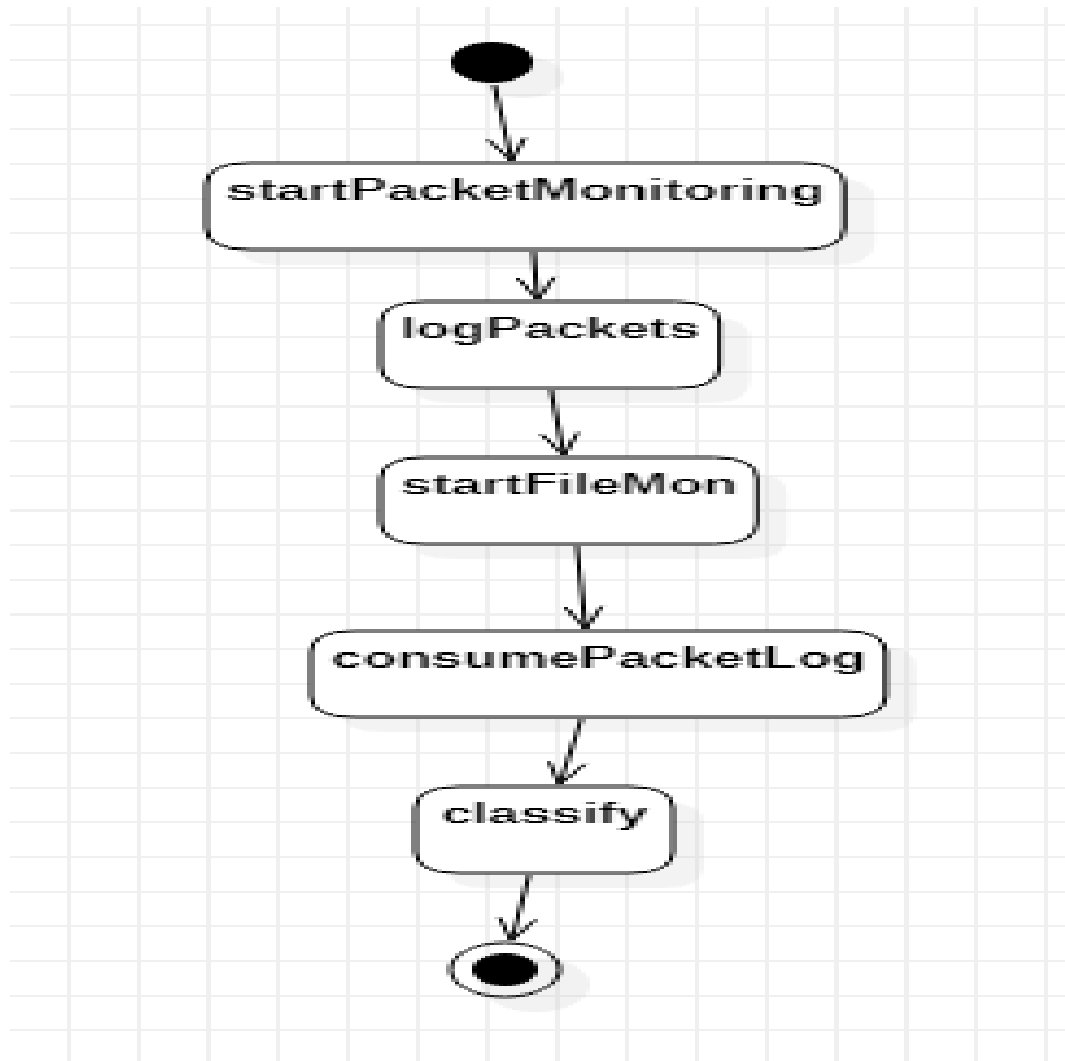


Fig. 4.5 Activity Diagram

4.7 Deployment Diagram

Deployment diagram is a construction outline which shows design of the framework as sending (dispersion) of programming relics to arrangement targets. Relics address substantial components in the actual world that are the consequence of an improvement cycle.

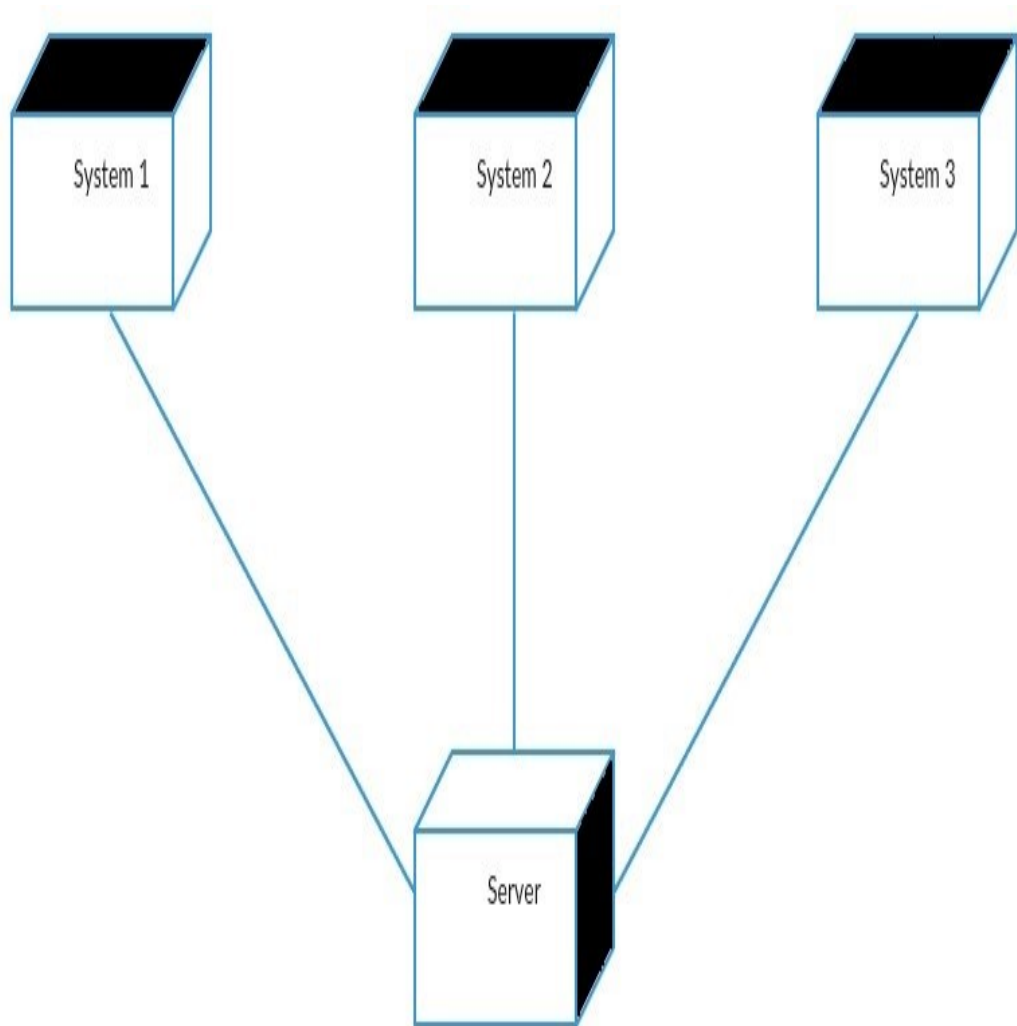


Fig. 4.6 Deployment Diagram

4.7.1 Description

The deployment diagram pictures the actual parts of a framework where the product parts are sent. This application is created utilizing Android Programming Language. These parts are addressed by hubs. The concealed region of the hub portrays the security of the application. The security is given to the data set to guarantee that there is no spillage of information

Chapter 5

CODING

5.1 Sample Code

```
#!/usr/bin/python

from scapy.all import *

capture = sniff()

sniff(lfilter=lambda x: x.haslayer(UDP) and x.haslayer(DNS),prn=lambda x:x.summary())

from base64 import decode

from logging import exception

import pandas as pd

import time

import matplotlib.pyplot as plt try:

filePath = 'G:/Python/packetclassification/req.txt'

explicit=['porn','sex','fuck','adult']

games=['zapak','game','play']

ott=['netflix','sony','prime','hotstar','youtube']

edu=['edu','ac.in']

news=['sakshi','times','hindu','eenadu']

exp=0

gam=0

ott1=0

edu1=0

news1=0

gen=0

def check(i):

global exp,gam,ott1,edu1,news1,gen

for j in explicit:
```



```

if(j in i):
exp=exp+1
print(i+" is explicit")
return
for j in games:
if(j in i):
gam=gam+1
print(i+" is games")
return
for j in ott:
if(j in i):
ott1=ott1+1
print(i+" is ott")
return
for j in edu:
if(j in i):
edu1=edu1+1
print(i+" is Educational")
return
for j in news:
if(j in i):
news1=news1+1
print(i+" is News")
return
gen=gen+1
print(i+" is General")
strt=0
while(True):
with open(filePath, 'r', encoding="utf8") as file:
data = file.read()

```

```

strings = data.split("")
data=[]
for i in range(strt, len(strings)):
try:
check(strings[i].split("
")[8].split("")[1].replace("b","").replace(
"", ""))

pd.DataFrame([exp,gam,ott1,edu1,news1,gen],columns=['e
xplicit','games','ott','edu','news','general']).tocsv
("G:/Python/packetclassification/stats.csv")
except BaseException as ex:
print(ex)
k=1
strt=len(strings)
time.sleep(1)
except KeyboardInterrupt:
plt.bar(['explicit','games','ott','education','news','g
eneral'],[exp,gam,ott1,edu1,news1,gen])
plt.show()
print([exp,gam,ott1,edu1,news1,gen])

```

Chapter 6

IMPLEMENTATION and RESULTS

6.1 Explanation of Key functions

6.2 Method of Implementation

6.2.1 UDP

6.2.2 TCP

6.2.3 Result Analysis

6.3 Method of Implementation

6.3.1 Packet Inspection

With TCP and UDP review, bundles entering the organization should precisely match a current meeting: the entering parcels should have a similar source or objective locations and source or objective port numbers as the leaving parcel (however switched). Any other way, the entering parcels will be impeded at the connection point. Likewise, all TCP bundles with a grouping number beyond the window are dropped.

With UDP review designed, answers might be allowed back in through the firewall in the event that they are gotten inside a configurable time after the last solicitation was conveyed. (This time is designed with the ip review udp inactive time order.)

Ether / IP / UDP / DNS Ans "b'cadcc4ca2cd08579.spotify - connect.tcp.local.'"
Ether/IP/UDP/DNSAns"b'cadcc4ca2cd08579.spotify - connect.tcp.local.'"
Ether/IPv6/UDP/DNSQry"b'logfiles.zoom.us.'"
Ether/IPv6/UDP/DNSQry"b'logfiles.zoom.us.'"
Ether/IPv6/UDP/DNSAns"b'us01 - logfiles - va.zoom.us.'"
Ether/IPv6/UDP/DNSAns"b'us01 - logfiles - va.zoom.us.'"
Ether/IPv6/UDP/DNSQry"b'ocsp.digicert.com.'"
Ether/IPv6/UDP/DNSQry"b'ocsp.digicert.com.'"
Ether/IPv6/UDP/DNSAns"b'cs9.wac.phicdn.net.'"
Ether/IPv6/UDP/DNSAns"b'cs9.wac.phicdn.net.'"
Ether/IPv6/UDP/DNSQry"b'default.exp - tas.com.'"

Ether/IPv6/UDP/DNSQry"b'default.exp - tas.com."
Ether/IPv6/UDP/DNSAns"b'deault - exp - tas - com.e - 0014.e - msedge.net."
Ether/IPv6/UDP/DNSAns"b'deault - exp - tas - com.e - 0014.e - msedge.net."
Ether/IPv6/UDP/DNSQry"b'update.code.visualstudio.com."
Ether/IPv6/UDP/DNSQry"b'update.code.visualstudio.com."
Ether/IPv6/UDP/DNSAns"b'vscode - update.trafficmanager.net."
Ether/IPv6/UDP/DNSAns"b'vscode - update.trafficmanager.net."
Ether/IPv6/UDP/DNSQry"b'dc.services.visualstudio.com."
Ether/IPv6/UDP/DNSQry"b'dc.services.visualstudio.com."
Ether/IPv6/UDP/DNSAns"b'dc.applicationinsights.microsoft.com."
Ether/IPv6/UDP/DNSAns"b'dc.applicationinsights.microsoft.com."
Ether/IPv6/UDP/DNSQry"b'vscodeexperiments.azureedge.net."
Ether/IPv6/UDP/DNSAns"b'vscodeexperiments.ec.azureedge.net."
Ether/IPv6/UDP/DNSQry"b'marketplace.visualstudio.com."
Ether/IPv6/UDP/DNSQry"b'marketplace.visualstudio.com."
Ether/IPv6/UDP/DNSAns"13.107.42.18"
Ether/IPv6/UDP/DNSAns"2620 : 1ec : a92 :: 175"
Ether/IPv6/UDP/DNSQry"b'default.exp - tas.com."
Ether/IPv6/UDP/DNSAns"b'deault - exp - tas - com.e - 0014.e - msedge.net."
Ether/IPv6/UDP/DNSQry"b'default.exp - tas.com."
Ether/IPv6/UDP/DNSAns"b'deault - exp - tas - com.e - 0014.e - msedge.net."
Ether/IPv6/UDP/DNSQry"b'default.exp - tas.com."
Ether/IPv6/UDP/DNSAns"b'deault - exp - tas - com.e - 0014.e - msedge.net."
Ether/IPv6/UDP/DNSQry"b'web.whatsapp.com."

Ether/IPv6/UDP/DNSQry"b'web.whatsapp.com."
Ether/IPv6/UDP/DNSAns"b'mmx - ds.cdn.whatsapp.net."
Ether/IPv6/UDP/DNSAns"b'mmx - ds.cdn.whatsapp.net."
Ether/IPv6/UDP/DNSQry"b'capi.grammarly.com."
Ether/IPv6/UDP/DNSQry"b'capi.grammarly.com."

Ether/IPv6/UDP/DNSAns"b'capi.core – prod.grammarlyaws.com."
Ether/IPv6/UDP/DNSAns"b'capi.core – prod.grammarlyaws.com."
Ether/IPv6/UDP/DNSQry"b'signaler – pa.clients6.google.com."
Ether/IPv6/UDP/DNSAns"2404 : 6800 : 4009 : 827 :: 200a"
Ether/IPv6/UDP/DNSQry"b'f – log – win – extension.grammarly.io."
Ether/IPv6/UDP/DNSQry"b'f – log – win – extension.grammarly.io."
Ether/IPv6/UDP/DNSAns"3.227.227.145"
Ether/IPv6/UDP/DNSAns
Ether/IPv6/UDP/DNSQry"b'education.github.com."
Ether/IPv6/UDP/DNSQry"b'education.github.com."
Ether/IPv6/UDP/DNSAns"b'glb – db52c2cf8be544.github.com."
Ether/IPv6/UDP/DNSAns"b'glb – db52c2cf8be544.github.com."
Ether/IPv6/UDP/DNSQry"b'default.exp – tas.com."
Ether/IPv6/UDP/DNSAns"b'deault – exp – tas – com.e – 0014.e – msedge.net."
Ether/IPv6/UDP/DNSQry"b'mobile.events.data.microsoft.com."
Ether/IPv6/UDP/DNSQry"b'mobile.events.data.microsoft.com."
Ether/IPv6/UDP/DNSAns"b'mobile.events.data.trafficmanager.net."
Ether/IPv6/UDP/DNSAns"b'mobile.events.data.trafficmanager.net."
Ether/IPv6/UDP/DNSQry"b'femetrics.grammarly.io."
Ether/IPv6/UDP/DNSQry"b'femetrics.grammarly.io."
Ether/IPv6/UDP/DNSAns
Ether/IPv6/UDP/DNSAns"54.158.97.24"
Ether/IPv6/UDP/DNSQry"b'vortex.data.microsoft.com."
Ether/IPv6/UDP/DNSQry"b'vortex.data.microsoft.com."
Ether/IPv6/UDP/DNSAns"b'asimov.vortex.data.trafficmanager.net."
Ether/IPv6/UDP/DNSAns"b'asimov.vortex.data.trafficmanager.net."
Ether/IPv6/UDP/DNSQry"b'dit.whatsapp.net."
Ether/IPv6/UDP/DNSQry"b'dit.whatsapp.net."
Ether/IPv6/UDP/DNSAns"b'scontent.whatsapp.net."
Ether/IPv6/UDP/DNSAns"b'scontent.whatsapp.net."

Ether/IPv6/UDP/DNSQry"b'www.google.com."
Ether/IPv6/UDP/DNSQry"b'www.google.com."
Ether/IPv6/UDP/DNSAns"142.250.182.4"
Ether/IPv6/UDP/DNSAns"2404 : 6800 : 4009 : 80b :: 2004"
Ether/IP/UDP/DNSAns"b'cadcc4ca2cd08579.spotify-connect.cp.local."
Ether/IP/UDP/DNSAns"b'cadcc4ca2cd08579.spotify-connect.cp.local."
Ether/IPv6/UDP/DNSQry"b'vortex.data.microsoft.com."
Ether/IPv6/UDP/DNSAns"b'asimov.vortex.data.trafficmanager.net."
Ether/IPv6/UDP/DNSQry"b'gnar.grammarly.com."
Ether/IPv6/UDP/DNSQry"b'gnar.grammarly.com."
Ether/IPv6/UDP/DNSAns"b'gnar-writer.data.grammarlyaws.com."
Ether/IPv6/UDP/DNSAns"b'gnar-writer.data.grammarlyaws.com."
Ether/IPv6/UDP/DNSQry"b'settings-win.data.microsoft.com."
Ether/IPv6/UDP/DNSQry"b'settings-win.data.microsoft.com."
Ether/IPv6/UDP/DNSAns"b'atm-settingsfe-prod-geo.trafficmanager.net."
Ether/IPv6/UDP/DNSAns"b'atm-settingsfe-prod-geo.trafficmanager.net."
Ether/IPv6/UDP/DNSQry"b'mobile.events.data.microsoft.com."
Ether/IPv6/UDP/DNSQry"b'mobile.events.data.microsoft.com."
Ether/IPv6/UDP/DNSAns"b'mobile.events.data.trafficmanager.net."
Ether/IPv6/UDP/DNSAns"b'mobile.events.data.trafficmanager.net."
Ether/IPv6/UDP/DNSQry"b'settings-win.data.microsoft.com."
Ether/IPv6/UDP/DNSAns"b'atm-settingsfe-prod-geo.trafficmanager.net."
Ether/IP/UDP/DNSQry"b'Ether/IP/UDP/DNSQry"b'Ether/IP/UDP/DNSAns"b'BRAVIA-VU1-a43a255de080af767523f32

6.4 OUTPUT GRAPHS

6.4.1 GENERAL CONTENT

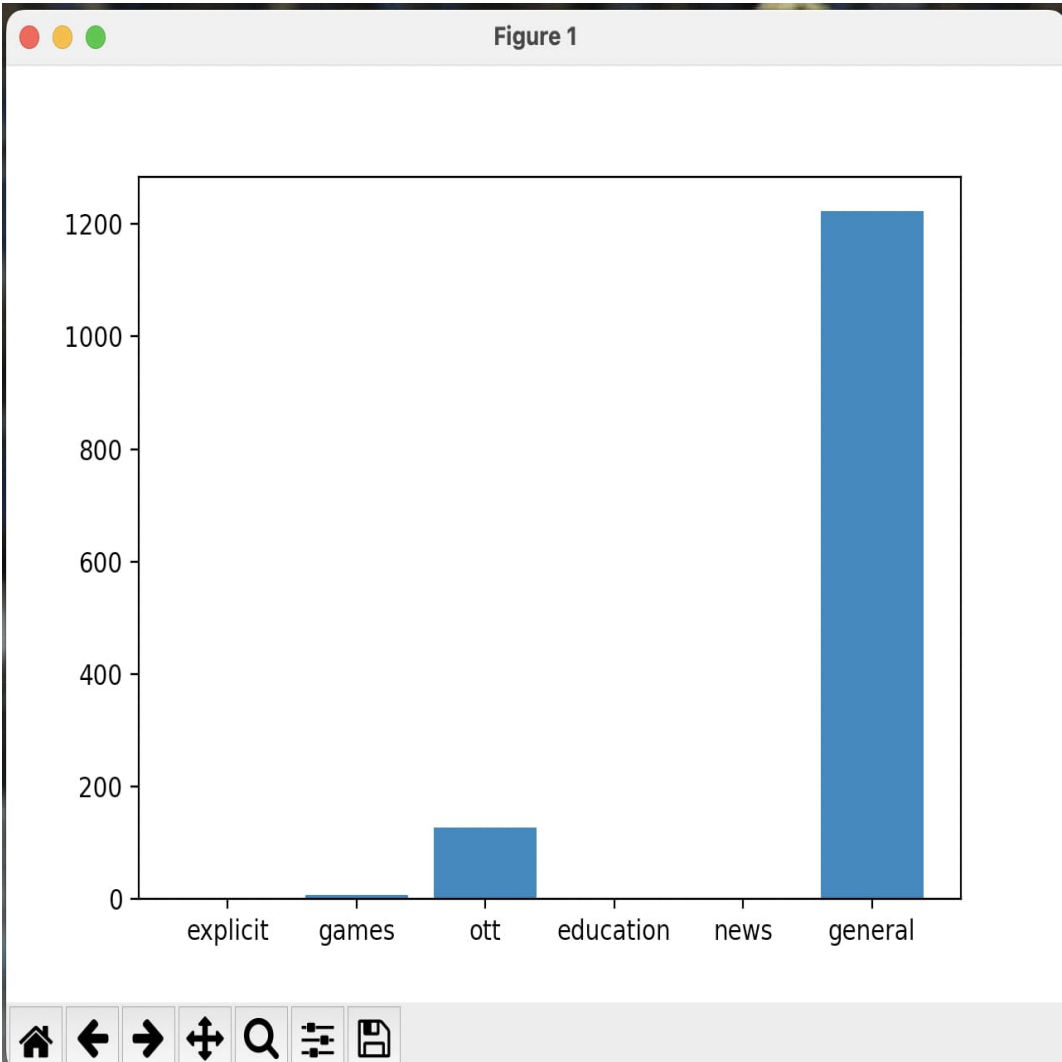


Fig. 6.1 GENERAL

6.4.2 OTT CONTENT

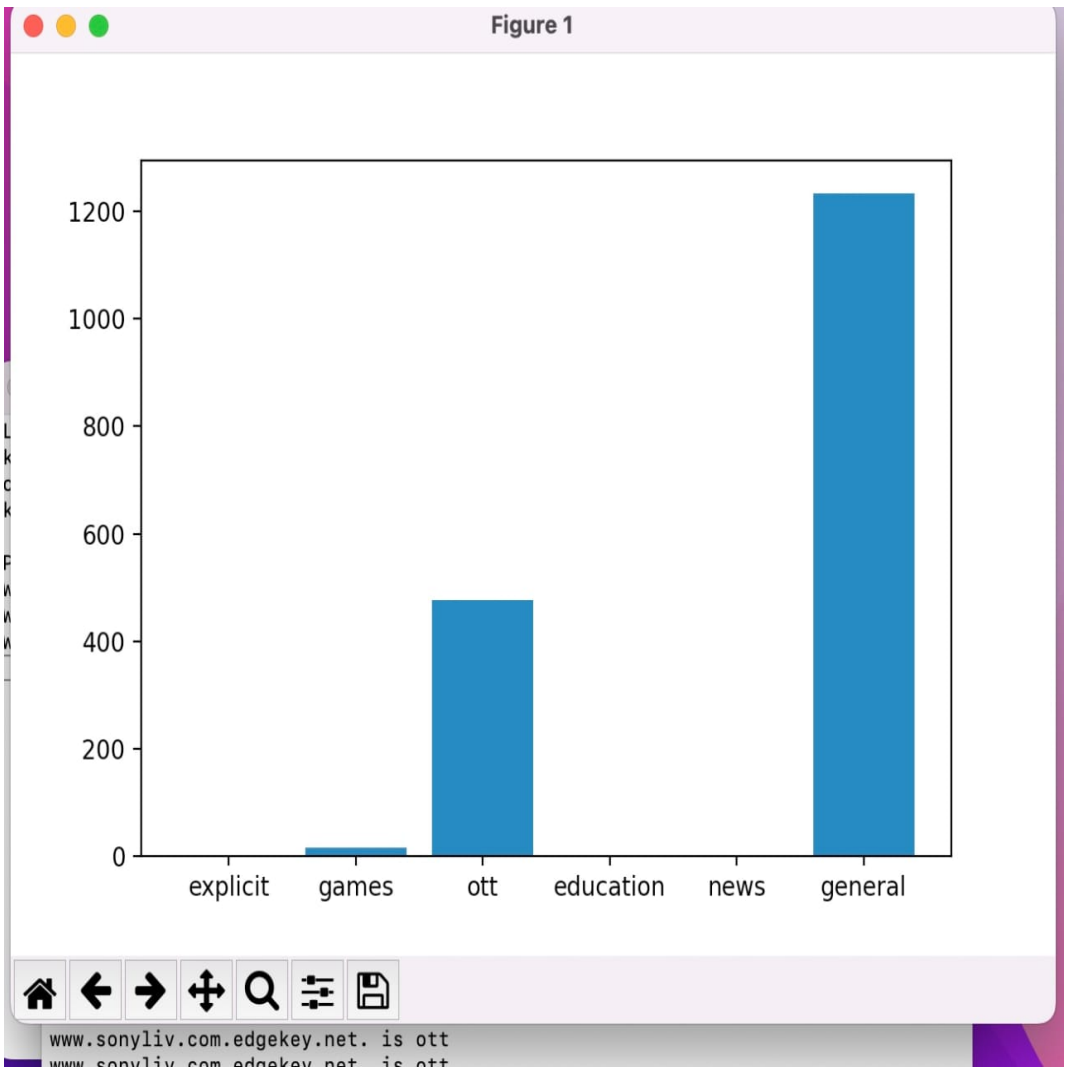


Fig. 6.2 OTT

6.4.3 NEWS

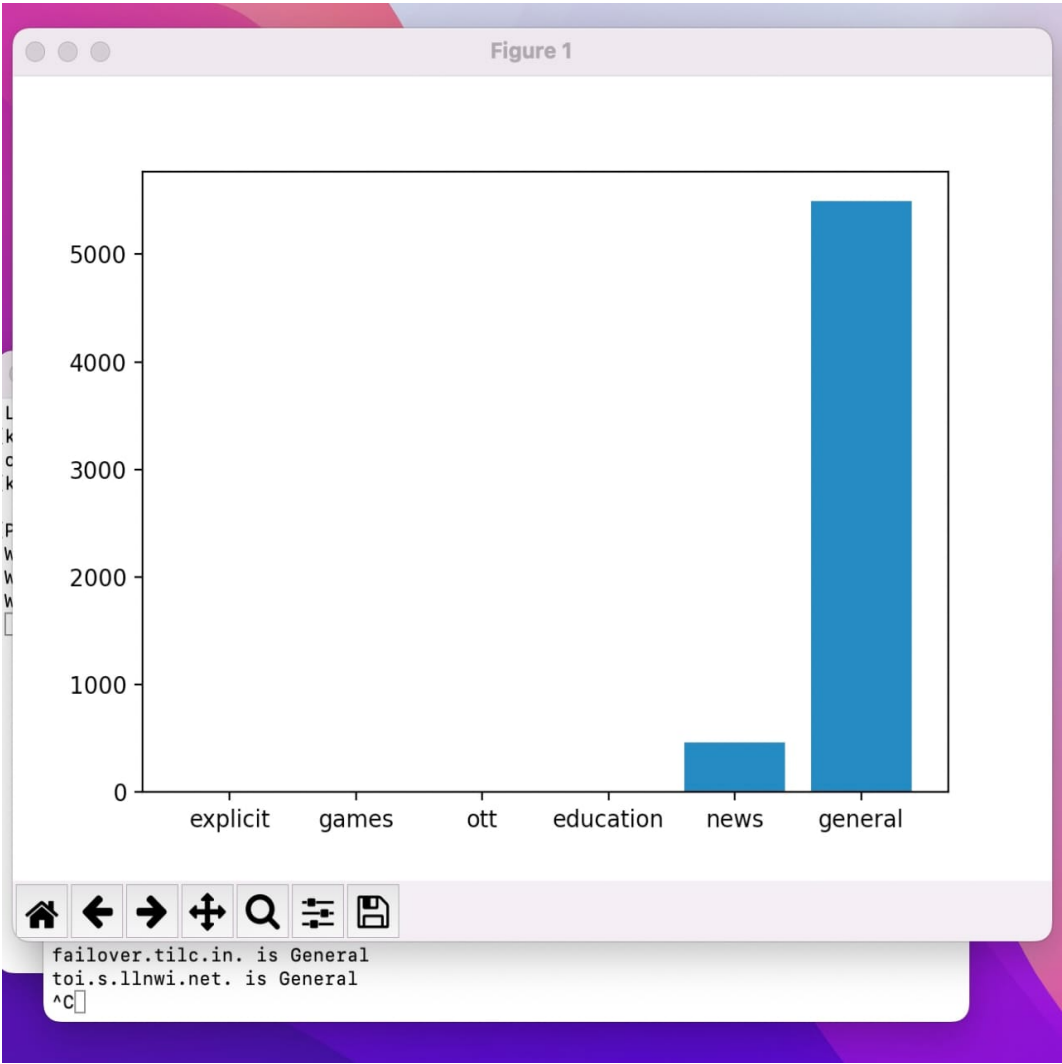


Fig. 6.3 NEWS

6.4.4 GAMES

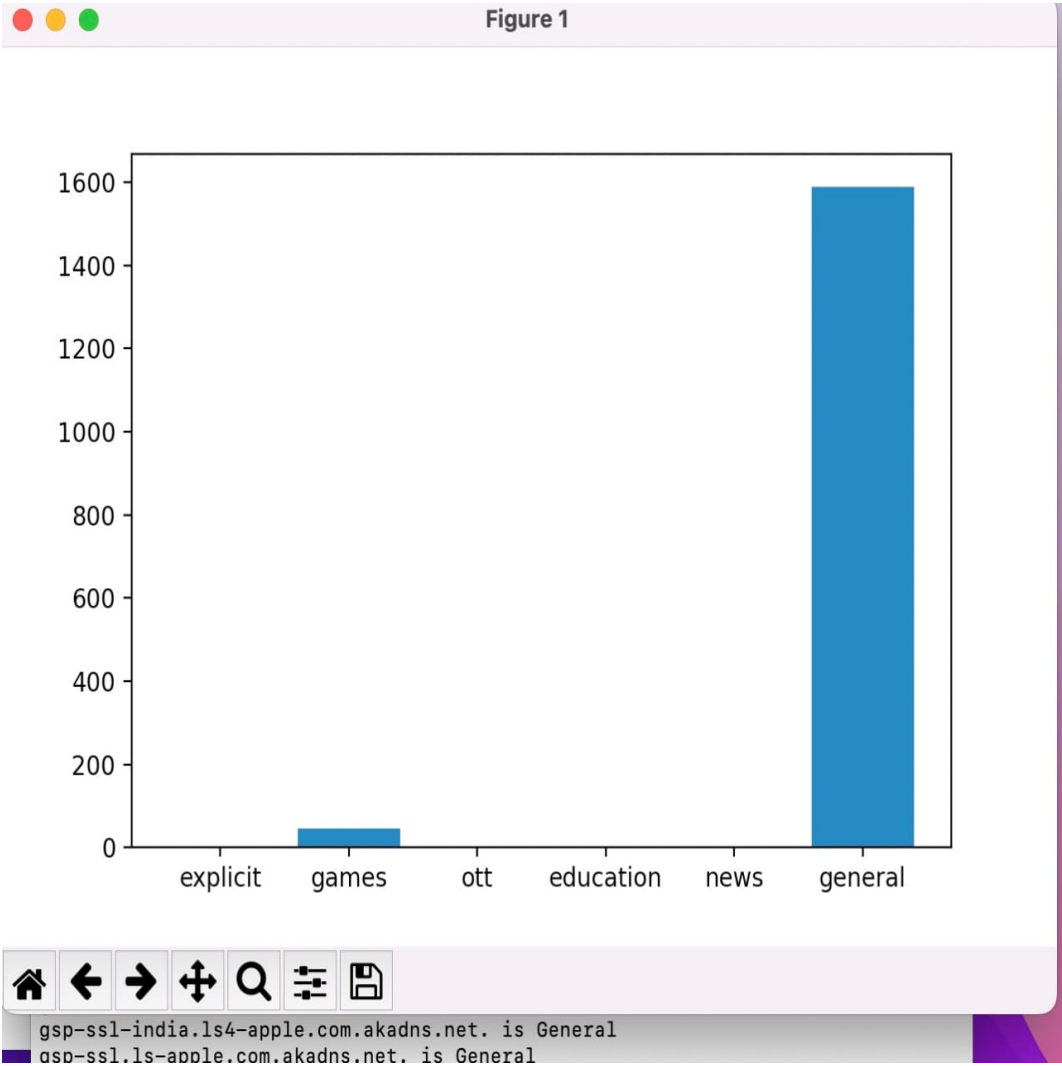


Fig. 6.4 GAMES

Chapter 7

TESTING and VALIDATION

7.1 Design of Test Cases and Scenarios

7.2 Validation

7.3 Testing

Testing involves the execution of a software component or system component to evaluate one or more properties of interest with respect to our project, Online pharmacy management system. In general, these properties indicate the extent to which the project is tested:

- Meets the prerequisites that directed its plan and improvement.
- Responds correctly to all kinds of inputs.
- Performs its functions within an acceptable time.
- Sufficiently usable.
- Can be opened and run in intended environments
- Achieves the general result

7.4 TESTING STRATEGIES

In order to perform testing to generate a test case that should be applied while testing, there have been developed various methods for designing the test cases. There are generally two categories of test case design techniques. They are,

7.4.1 White Box Testing

White box testing is performed by checking the inward code of the application. Syntactic blunders, choices, control stream and explanations of the code are checked line by line for every one of the modules engaged with our framework like Admin and User. We could distinguish control stream mistakes, for example, wrong redirection utilizing white box testing.

7.4.2 Black Box Testing

In this framework black box testing is performed by giving arbitrary contributions to the functionalities and checked whether expected results are acquired. For instance, arbitrary sources

of info are given to login certifications and checked whether it's diverting to landing page regardless of whether the qualifications are off-base. At the point when these sorts of bugs are distinguished then these are braced. In this testing we checked whether our application is as per the necessity determination. Any missing functionalities can be identified in this stage.

7.5 LEVELS OF TESTING

7.5.1 Integration Testing

Integration testing is any kind of programming testing that tries to confirm the points of interaction between parts against a product plan. Programming parts might be coordinated in an iterative manner or through and through. System Testing System Testing is a level of the product testing where a total and coordinated programming is tried. The motivation behind this test is to assess the framework's consistence with the predetermined prerequisites. The most common way of testing an incorporated framework is to confirm that it meets indicated necessities. Regularly, free analyzers perform System Testing.

7.5.2 Acceptance Testing

Acceptance Testing is a level of the software testing where a framework is tried for worthiness. The reason for this test is to assess the framework's consistence with the business necessities and evaluate whether it is adequate for conveyance.

7.6 Test Cases

An experiment in programming regularly comprises of a remarkable identifier, prerequisite references from a plan detail, preconditions, occasions, a progression of steps (otherwise called activities) to follow, information, result and it approves at least one framework necessities and produces a pass or fizzle.

Chapter 8

CONCLUSION

Conclusion will need to have several elements, including:

Network traffic categorization has been effectively researched/analyzed by utilizing network packet inspection. In the end we developed a free and effort less web packet classification application using python and it's libraries. As per the tests the application seems to work perfectly with any load.

- we can sort the traffic into different classifications Movies,Education,Games,explicit content.
- With TCP and UDP protocols, packets entering the organization should precisely match a current meeting: the entering packets should have a similar source or objective locations and source or objective port numbers as the leaving bundle (however switched). Any other way, the entering packets will be hindered at the point of interaction. Likewise, all TCP data with a succession number beyond the window are dropped.
- By using packet based technique we can sort data especially fast speed of 1.5kb with ipv6 64bits thusly this approach less expense and quick accessible.

REFERENCES

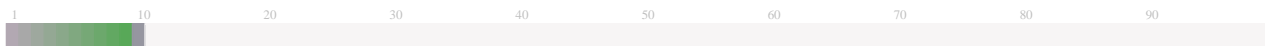
- [1] Evaluating visualization approaches to detect abnormal activities in network traffic data. *International Journal of Computer Applications*, 10(2):1–5, 2010.
- [2] Taimur Bakhshi and Bogdan Ghita. On internet traffic classification: A two-phased machine learning approach. *Canadian Journal of Electrical and Computer Engineering*, 39(2):141–149, 2022.
- [3] Alisha Cecil. A summary of network traffic monitoring and analysis techniques. *Journal of Scientific and Industrial Research*, 76(3):149–153, 2020.
- [4] Conference papers. Network traffic classification techniques and comparative analysis using machine learning. *Journal of Scientific and Industrial Research*, 76(3):149–153, 2020.
- [5] Conference1 papers. Internet traffic classification using machine learning. *Journal of Scientific and Industrial Research*, 76(3):149–153, 2020.
- [6] HENRY FRIDAY NWEKE² SYLVESTER I. ELE¹ 3 UZOMA RITA ALO¹, *. Machine learning-based framework for automatic malware detection using android traffic data. *Journal of Scientific and Industrial Research*, 76(3):149–153, 2020.
- [7] Dr. Ambika R Yoga Durgadevi Goli. Network traffic classification techniques-a review. *Computers and Electrical Engineering*, 71:309–320, 2022.

Submission Information

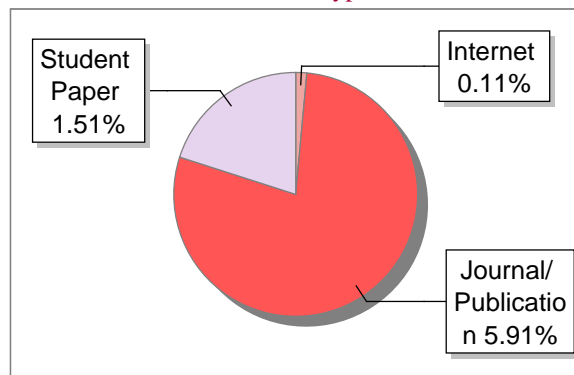
Author Name	D11
Title	D11
Paper/Submission ID	539607
Submission Date	2022-06-09 14:32:07
Total Pages	32
Document type	Thesis

Result Information

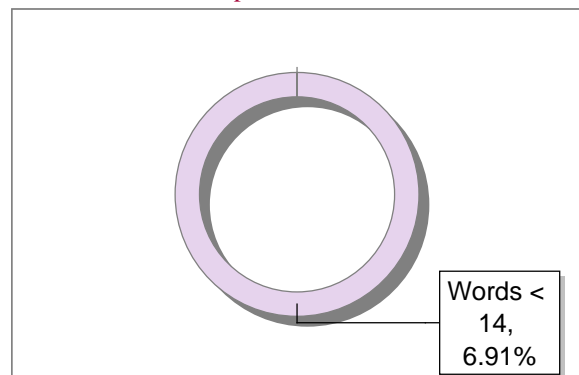
Similarity **11 %**



Sources Type



Report Content



Alert...! Improper usage of 'Quotes' work in a report

Exclude Information

Quotes	Not Excluded
References/Bibliography	Excluded
Sources: Less than 14 Words Similarity	Not Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

A Unique QR Code use to View/Download/Share Pdf File



DrillBit Similarity Report

11

SIMILARITY %

28

MATCHED SOURCES

B

GRADE

A-Satisfactory (0-10%)

B-Upgrade (11-40%)

C-Poor (41-60%)

D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	www.ijitee.org	1	Publication
2	INTERNET BANKING -Submitted to JNTUH,TELANGANA By '15U01E0095'	1	Student Paper
3	www.ijitee.org	1	Publication
4	A STUDY ON PAYROLL MANAGEMENT BY 19C71E0044 Yr-2021 SUBMITTED TO JNTU	1	Student Paper
5	IEEE 2019 IEEE 19th International Conference on Software Quality, Re	1	Publication
6	Paper published at International Journal of Computer Science & Mobile Computing - www.ijcsmc.com	<1	Publication
7	Fine-grained traffic classification based on functional separation by Park-2013	<1	Publication
8	DETECTING DECEPTIVE REVIEW SPAMMERS USING DATA MINING TECHNIQUES BY 18L51D5813 YR 2020, JNTUH	<1	Student Paper
9	Article Published by International Journal Of Engineering & Computer Science - www.ijecs.in	<1	Publication
10	The approach in Scotland to child contact disputes involving allegations of dome by Mackay-2018	<1	Publication
11	idosi.org	<1	Publication

12	Development of cooperative learning based electric circuit kit trainer for basic by Hamid-2020	<1	Publication
13	www.dx.doi.org	<1	Publication
14	Innovative Head-Mounted System Based on Inertial Sensors and Magnetometer for De by Lin-2020	<1	Publication
15	Design space exploration of thermal-aware many-core systems by Ra-2013	<1	Publication
16	Moderately naturalistic metaphysics by Morganti-2016	<1	Publication
17	Environments for Information Systems Implementation in Small Businesses by Thong-1997	<1	Publication
18	An Adaptive Spiking Neural P System for Solving Vehicle Routing Problems by RamachandranPillai-2019	<1	Publication
19	A Chair-Based UnconstrainedNonintrusive Cuffless Blood Pressure Monitoring Syst by Lee-2019	<1	Publication
20	Text classification using Nave Bayes classifier by Kolluri-2020	<1	Publication
21	Sound propagation over soft ground without and with crops and potential for surf by Bashir-2015	<1	Publication
22	openjicareport.jica.go.jp	<1	Publication
23	Muariban mannan A Case of Phraseology and Evolving Motifs of Lit, by Shalev, Donna- 2014	<1	Publication
24	Managing the value network by -1998	<1	Publication
25	Improving Classification Accuracy of Hand Gesture Recognition Based on 60 GHz FM by Lee-2020	<1	Publication
26	Employer and Employed by -1884	<1	Publication

27	docs.microsoft.com	<1	Internet Data
28	A predictive analytics framework as a countermeasure for attrition of students by Gontzis-2019	<1	Publication