A report on

Detection of Price Scraping using Behavioral Analysis

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Aakash Bali (151378)

Under the supervision of

Prof. Rizwan Ur Rehman

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology,

Waknaghat, Solan

Himachal Pradesh, 173234

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitled "**Detection of Price scraping using Behavioral Analysis**" in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from **July 2018** to **May 2019** under the supervision of **Prof. Rizwan Ur Rehman**, Assistant Professor (Grade-II), Jaypee Univeristy of Information Technology, Waknaghat, Solan.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Aakash Bali- 151378

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Prof. Rizwan Ur Rehmann,

Assistant Professor (Grade-II),

Department of Computer Science & Engineering and Information Technology.

Dated:

ACKNOWLEDGMENT

Any serious and lasting achievement cannot be achieved without the help, guidance and co-operation of numerous people involved in the work.

First and foremost, we would like to express my gratefulness to Prof. Dr. Satya Prakash Ghrera, FBCS, SMIEEE Professor, Brig (Retd.) and Head Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology for providing us the opportunity to carry out this project as our final year project. It gives us immense pleasure to express my deepest gratitude and thanks to Prof. Rizwan Ur Rehman, Assistant Professor (Grade-II), Department of Computer Science & Engineering and Information Technology, for not only imparting his knowledge but also his constant supervision, advice and guidance throughout the project, without which this project wouldn't have been possible.

We would also like to thank all other department faculty at Jaypee University of Information Technology. Not only did they taught us and made us capable enough to undertake this project but were always there at the need of the hour and provided with all the help, facilities and co-operation, which was required towards the completion of our project.

A special mention to Ravi Raina Sir and Sanjeev Kumar Sir who assisted our project lab and guided us towards all the minor issues.

Last but not the least, We would like to express our thanks to our parents and family members for their support at every step of my life.

TABLE OF CONTENT

List of Figures

List of Tables

ABSTRACT

Web scraping is also known by some other names like web harvesting and web data extraction basically it is used for extraction of data from the websites on the WORLD WIDE WEB. Or in another way, it can be defined as the process consisting of the extraction and combination of content gathered from the web in a systematic manner.

Price scraping is a sub part of web scraping technique which is used to scrape the prices of various items from e-commerce website. To carry out this work bots are used to scrape the site. These bots can be any software or any piece of code. Price scraping can be considered illegal if any competitor is using the scraped prices to reduce his own price to get benefit in his sales. So in order to detect these bots we can use various techniques to differentiate between bots and humans.

Chapter-1 INTRODUCTION

1.1 Introduction

Nowadays the internet is at it speak everything is available online or is going to be available soon. So the Internet has provided us with some facilities like online Shopping, Bookings of trains and buses, education and many more. But there is always another side to the coin, with the facilities comes the cyber attacks of various types like DDOS, Man in the Middle, SQL injection etc. One of them is Web scraping which is a very serious issue nowadays it's affecting the market of online E-commerce very much. So we are going to talk about how it's done and what are the ways we can stop it from happening.

1.1.1 Web Scraping

Web scraping is also known by some other names like **web harvesting and web data extraction** basically it is used for extraction of data from the websites on the **WORLD WIDE WEB**. Or in another way, it can be defined as the process consisting of the extraction and combination of content gathered from the web in a systematic manner.

There is software present for doing the web scrapping which may do their work of accessing the World Wide Web using Hypertext Transfer protocol or web browser.

Web scraping can also be done manually by the user but is preferably done in an automated fashion implemented using a **bot or web crawler.** In this, some software aka Web robot is mimicking the browsing between the Web and the human in a conventional web traversal.

This robot may gather the data from as many websites as needed and the parsing of the contents is done to easily find and fetch the data required and stores them in the structures as desired.

Generally, this task of web scraping is somewhat similar to copying in this particular data is collected and copied from the internet into some manageable and readable storage structure like some spreadsheets or databases.

In this process, the web page is downloaded or fetched (it happens whenever the browser opens up some pages) first and saved for later use and then the data is extracted from it. Hence we can say that web crawling is an important component of the process.

At the second step of the process the content present in the page is parsed, searched or some type of reformatting is done to understand the content for the data to get it inserted into the spreadsheets or database by copying. Generally, the web scrapping software may sometime take a part of the page which can be useful for the authority for some other purpose.

Web Scrapping is being used in various things in today's life like in advertisements and marketing generally by contact scraping and also an important part of the application made for data mining and web mining, and sometimes used to do some price comparisons, for online price change monitoring, weather data monitoring, research and for providing a service to the user where the content comprises of more than one source also known as web mashup ex. like **trivago** and **mybestprice** applications.

Basically, these web scrapers are APIs which are used to extract data from a web page or a website present on the internet. Also, some big companies like Amazon Web Services and Google provide web scrapping tools free of cost to end users.

Nowadays a new form has been also used for web scrapping which consists of listening or monitoring the data feed from the web servers. And also some web scraping systems are also using DOM parsing techniques, computer vision and NLP to simulate human browsing as to pass the checks for bots that some websites are using to prevent web scrapping.

Now the question arises **IS WEB SCRAPING ILLEGAL?**

Web scraping started in a legal grey area where the use of bots to scrape a website was simply a nuisance. Not much could be done about the practice until in 2000 eBay filed a preliminary injunction against Bidder's Edge. In the injunction eBay claimed that the use of bots on the site, against the will of the company violated Trespass to Chattels law. The court granted the injunction because users had to opt in and agree to the terms of service on the site and that a large number of bots could be disruptive to eBay's computer systems. The lawsuit was settled out of court so it all never came to a head but the legal precedent was set.

In 2016, Congress passed its first legislation specifically to target bad bots -- the Better Online Ticket Sales (BOTS) Act, which bans the use of software that circumvents security measures on ticket seller websites. Automated ticket scalping bots use several techniques to do their dirty work including web scraping that incorporates advanced business logic to identify scalping opportunities, input purchase details into shopping carts, and even resell inventory on secondary markets.

To counteract this type of activity, the BOTS Act:

- Prohibits the circumvention of a security measure used to enforce ticket purchasing limits for an event with an attendance capacity of greater than 200 persons.
- Prohibits the sale of an event ticket obtained through such a circumvention violation if the seller participated in, had the ability to control, or should have known about it.
- Treats violations as unfair or deceptive acts under the Federal Trade Commission Act. The bill provides authority to the FTC and states to enforce against such violations.

1.1.2 Types Web Scraping

There are various types of web scraping. Some of them are:

1.1.2.1 Data Scraping

is a process in which computer program gathers data from some humanly understandable output that is coming from another computer program. Generally, data transfer between programs is done using various data structures best for processing by computers, not humans. These interchangeable formats and protocols are rigid in structure, mostly well documented and easily parsable.

The main difference between data scraping and parsing is just that it scrapes the data which was for display for humans and was not used as input to another program. It ignores data in binary format. It is mostly used as an output to a legacy system which has no other way by which it can display the data or it is used for the systems for which the APIs are not good enough to provide the data. It is not the most preferable technique to be used to extract the data it is usually the last resort when no other technique is available

1.1.2.2 Content Scraping

It is generally the technique of lifting off the displayed content from various websites and using it somewhere else or displaying it somewhere else. The technique is illegal asit; generally done without the permission of the original source. The content scrapers mostly copy the whole content being displayed and share it as their own content.

It is an illegal method to steal the content from a trusted website and publishing to another website without the knowledge of the content;' owner.

Content scraping is being done at the expense of the website that has invested time, resources and money as it will also affect their SEO ranking. It can be done by manually copying and pasting or by using special software or by HTTP programming etc.

1.1.2.3 Price Scraping

The technique of extracting or collecting the prices of various sellable things available over the internet on various websites without the consent of the corresponding authorities and slowing down their network by frequently requesting for the data being displayed by them.It is generally used to beat the competition in the market or to make profits out someone's loss by planning according to the scrapped data and lowering the prices of things being searched by users to attract them to their own website and increase their profits through ads being shown on the website.

Price scarping can be done for various uses in order to predict some share of the market being occupied by the biggie of e-commerce or for the newcomers to know about the various new trends being prevailed in the market in order to enter the competition of selling things online and also to gather the information about the supply and demand of various products being sold on the website.

It's illegal to scrape the data for commercial use but still, it is being done by many companies for various purposes and there are no % efficient methods present to have a check on it.

1.1.2.4 Database Scraping

The technique of directly extracting data from the database is known as the Database scraping. Some time s the data is being stored in the database file or .csv file in which the data is being stored in the structured format for some special use or classification to do some calculation etc.

The data is being stored in columns and rows so there may not be difficulty in scraping the data as the data has some pattern that is stored. It can be used for various purposes but is basically used for research purposes as a large database are being scraped off data to classify the data into some important information which can help in predicting the results of some work or to provide automation.

Sometimes it is also used for commercial purposes in order to scrape the database of persons personal details without his/her consent. So it can be used for illegal as legal purposes.

1.1.2.5 News Scraping

The techniques of scraping used for scraping the news from the newspapers websites is known as news Scraping. The articles of news are being scraped to some blogs or some database in order to do some discussion or to have the database of the news which can be used to predict the type of acts of crimes can happen or to calculate the results of the distribution of the crimes happening.

It can be used by some educational purposes in order to be updated on the various things happening or we can say enhance students general knowledge in preparation of various exams. It generally doesn't affect much but it lowers the ranking of the website ISEO rankings and the Web ratings and the traffic on the website will be affected accordingly.

1.1.2.6 Article Scraping

The scraping of the data being written over blogs or various websites on the internet is known as Article scraping. Generally done for the purpose if the collection of data that can be used for various purposes like educational, research etc.

This practice can be used for some type of security purposes sometimes in order to find some potential threats to the nation or some specific words being used against some things of political importance or It can be used to ease the process of collecting data on some specific topic which is new and evolving and to improve the related information on that specific topic in order to provide reliable information which can be used further for various process developments and experiments.

1.1.2.7 Email Harvesting

The mechanism to obtain a large number of e-mail addresses using different methods or techniques. The main purpose of email harvesting is for spamming or advertising purpose.

Mostly done by some specialized harvesting software or programs also known as the harvester. The email can be scraped using various ways like from mailing lists, stored data of a web browser, from yellow pages through social engineering etc. Some spammer use the dictionary attack to extract an email address, in this the valid emails addresses are found by means of guessing the most used usernames . or by offering a product or service to users absolutely free till they provide their genuine email address and then from them collect email addresses for spamming or bulk email.

1.2 Problem Statement

At the time of shopping, online users prefer to compare the things and buy those things which are available at a lesser price, which has given rise to the practice of price scraping. In this practice, various e-commerce websites take help of the price scraping through which they get the data of the prices of various things being sold by their competitors and at which price and the extra services they are providing. So as to increase their share of business they collect all this data and start taking steps accordingly and lower down their prices of the same things being offered by them as to attract users and provide various facilities and services at the lower rate than others. This practice hampers the market of the online e-commerce websites. The other effects of price scraping are that most of the traffic nowadays on the internet is due to these bots which does the scraping and other things which sometimes hinders the user experience of some websites for the human user

and lowers their ratings and sometimes the sites crash due to these effects of scrapers. Also due to scraping of the data the ranking of the website gets lower in the SEO which reduces the number of users visiting the website. So in order to prevent the loss of various types done to these e-commerce website We are trying to devise a detection mechanism to identify between the legitimate and illegitimate request being done to web servers and hence protect these websites from being scraped off their data and to maintain their market share and profits and lower down the loss done due to these bots by the help of behavioural analysis of the users active on the website.

1.3 Objectives

The objective of the project is to understand the working of the price scrapers , how the price scrappers work and what areas they attack in order to do the scraping of the prices on the e-commerce website and to devise an efficient way to detect their presence on the website by classifying the various attributes of the behavior of surfing of these bots on the websites and suggesting various measures to stop them from doing any damage.

1.4 Methodologies

Various techniques are there to do automated web scraping which can be used by various web scrapers to scrape off the prices from the websites and collecting the data in some formatted manner to be used in the future and to prevent that from being done in an efficient way.

1.4.1 Automated Web Scraping Techniques

1.4.1.1 Text Pattern matching

The method of doing the extraction of data from the parsed web page in the process of web scraping by using the Regular Expression matching in the collected data.

Regular Expression can be defined as a sequence of characters that define a search pattern. Generally, this can be used by various algorithms of string searching to be used on strings and for validating some inputs taken by the user.

A regex processor is the breaks down the regular expression into an internal representation which is executed and matched for a string that represents the text that is searched in.

An easy way to specify a finite set of strings is to list its members. For example

There is a set containing two strings "Handel", "Haendel" can be matched by the pattern ***H(a|ae?)ndel;***

The regular expression is widely used in UNIX. There is a module **re**present in python which provides full support for PERL-like regexes.

The whole process of doing web scraping can or can't have a regex in your implementation of the scraper. It depends on the type of website being scraped and is the last resort to do the scraping when no other methods can be used in order to scrape the useful data from the website. It is generally used when there is no pattern defined in a website or the data is randomly given an there is only some textual pattern present in the website.

For example:

The website's name is **exotic gifts.co.in** and it looks like :

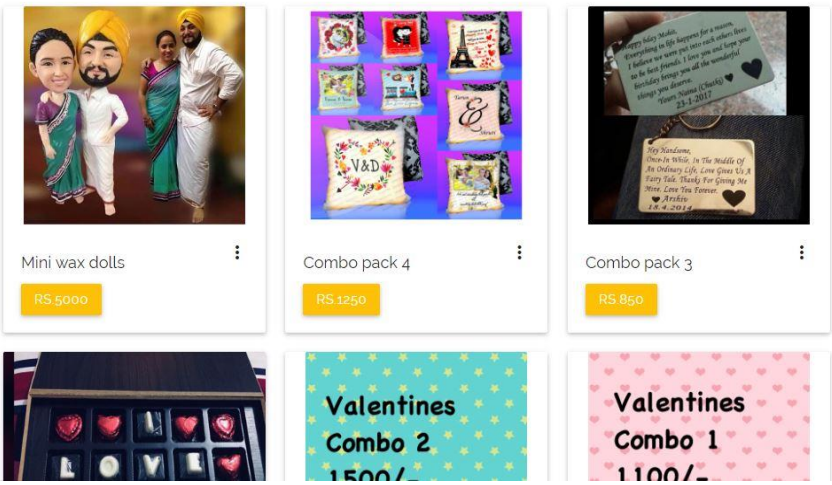


Fig.1 Dummy Website

The HTML Document of this website is:

```
    <div class="card">
  <div class="card-image waves-effect waves-block waves-light">
    <a href="product.php?pid=EG0051"><img class="activator responsive-img imgres" src="saksham/products/EG-7-FE7EE507-CDB0-4052-955E-40C17D6A5642.jpeg"></a>
  </div>
  <div class="card-content">
    <span class="card-title activator grey-text text-darken-4">Mini wax dolls<i class="material-icons right">more_vert</i></span>
    <p><a href="product.php?pid=EG0051" class="btn amber waves-light btn">Rs.5000</a>
    </p>
  </div>
  <div class="card-reveal">
    <span class="card-title grey-text text-darken-4">EG0051<span class="pnmsize">(Mini wax dolls)</span><i class="material-icons right">close</i></span>
    <p>Wax miniatures- can be done for any picture.</p>
    <a href="product.php?pid=EG0051" class="btn btn-large amber waves-light btn" style="font-size:22px; font-weight: bold;
><i class="material-icons left">shopping_cart</i>Rs.5000</a>
  </div>
</div>
    <div></div>
```

For the website having price as a text on the buttons of the products along with the “Rs.”

The code is:

```
# import libraries

import bs4
import requests
import re
res = requests.get("http://exoticgifts.co.in/")
soup = bs4.BeautifulSoup(res.text, 'html.parser')

print(re.findall(r'Rs\.[0-9]+', res.text))
```

Output is:

```
[Running] python -u "c:\Users\Danish\Desktop\rizwan\New folder\New folder\webscrapper.py"
['Rs.5000', 'Rs.5000', 'Rs.1250', 'Rs.1250', 'Rs.850', 'Rs.850', 'Rs.600', 'Rs.600', 'Rs.1500', 'Rs.1500', 'Rs.1100', 'Rs.1100', 'Rs.2000', 'Rs.2000', 'Rs.1200', 'Rs.1200', 'Rs.450', 'Rs.450', 'Rs.1399', 'Rs.1399', 'Rs.2000', 'Rs.2000', 'Rs.450', 'Rs.450']

[Done] exited with code=0 in 1.951 seconds
```

1.4.1.2 HTML Parsing

As we know every website present on the internet are written using HTML (Hypertext Mark-up Language).As it is a structured language so that implies that every web page is a structured document. And sometimes we might need data from the websites and web pages and preserve their structure as well.

In general, parsing is basically to break (a sentence) into its components and specify their syntactic roles. Parsing or syntactic analysis is the process of analyzing a collection or string of symbols can be in the natural language or in a computer language agreeing to the rules of a formal grammar.

In this case of HTML parsing comprise of:

Taking in HTML code and extracting needed information like the title of the page, paragraphs in the page, headings in the page, links etc.

The request will be made to get the web page using request module get function and then the extracted HTML script will be converted into HTML tree which will have the structured data from that we will make use the various ways to go over that tree like XPath or CSS Select and then the extracted t=data will be added to the various lists.

The *lxml* is a pretty extensive library written for parsing XML and HTML documents in no time and can also handle messed up tags in the process of parsing.

There are two ways to extract the data from the web page after being converted into HTML tree by using XPath or CSS Select. This example will make the use of XPath.

XPath is a way of locating information in various structured documents like HTML and XML.

For example

The code is:

```
from lxml import html
import requests
page = requests.get('http://exoticgifts.co.in/')
tree = html.fromstring(page.content)
```

```
# This will create a list of names of items:
names = tree.xpath(
'//span[@class="card-title activator grey-text text-darken-4"]/text()')
# This will create a list of prices
prices = tree.xpath('//a[@class="btn amber waves-light btn"]/text()')
print(names)
print(prices)
```

Output is:

```
[Running] python -u "c:\Users\Danish\Desktop\rizwan\New folder\New folder\browser.py"
['Mini wax dolls', 'Combo pack 4', 'Combo pack 3', 'Name Chocolates ', 'Combo 2', 'Combo pack 1', 'Miniature Bouquet', 'Picture Bouquet', 'Personalised notebooks', 'Magic Mirror', 'Decanter', 'Wallnut Box']

['Rs.5000', 'Rs.1250', 'Rs.850', 'Rs.600', 'Rs.1500', 'Rs.1100', 'Rs.2000', 'Rs.1200', 'Rs.450', 'Rs.1399', 'Rs.2000', 'Rs.450']

[Done] exited with code=0 in 1.723 seconds
```

1.4.1.3 DOM Parsing

As we have explained parsing in the previous techniques. There are two types of parsers present

1.4.1.3.1Top-down parser:

Top-down parsing can be seen as a dry run to find the left-most derivations of an input stream by looking for parse trees using a top-down opening out of the given approved set of grammar rules. Tokens are generally put to use from left to right.

1.4.1.3.2 Bottom-up parser:

A parser can begin with the input and tryout to rework it to the start symbol. The parser attempts to detect the most primary elements than the elements comprising of these and so on.

DOM stands for the Document Object Model is a programming API for HTML and XML documents or web pages. It specifies the logical structure as well as the way to access the document and manipulate it.

With DOM developers can create and build documents, navigate through their structure and can add, modify or delete content or elements.

In Document Object Model, documents consist of a tree-like logical structure or a forest comprising of more than one tree. In this, the documents are modelled using objects and the model contains not only the structure of the document but also the behaviour of a document and the objects which it comprises of.

While scraping a website the first step is to request the web page and receive the page HTML DOM tree. After this, the step is to parse the DOM tree to extract the data we want. Usually, it is done with dstring operations and using regular expressions. The other methods are by using DOM parser library either CSS Select or XPath to extract the DOM elements which contain the required information.

For Example:

The code:

```
from htmldom import htmldom
dom = htmldom.HtmlDom("http://exoticgifts.co.in/").createDom()
# Find all the links present on a page and prints its "href" value
```

```
a = dom.find("a")
for link in a[:5]:
print(link.attr("href"))
```

Output is :

```
[Running]    python    -u    "c:\Users\Danish\Desktop\rizwan\New
folder\New folder\browser.py"
category.php?cat=51
category.php?cat=52
category.php?cat=53
category.php?cat=54
category.php?cat=55

[Done] exited with code=0 in 1.043 seconds
```

1.4.1.4 Optical Character Recognition (OCR) Techniques

Optical Character Recognition techniques can be used for images or documents which cannot be parsed into text directly which contain some information required.

It is the mechanical or electronic conversion of images of typed handwritten or printed text into machine-encoded text. It is widely used for solving CAPTCHA's (Completely Automated Public Turing test to tell Computers and Humans Apart) which is being used as the preventive measure for the web scraping done on the web pages on various websites. The OCR technique is useful in some web pages where the required information is in some images or PDFs so to scrape that required information this technique is used. It is not much efficient but is constantly worked upon to increase its efficiency to help in scraping and various other streams.

There are four types of **Character Recognition Techniques**:

1. **Optical Character Recognition (OCR)-** generally targets typewritten text, one glyph (an elemental symbol within an agreed set of symbols, intended to represent a readable character for the purposes of writing) or character at a time.
2. **Optical Word Recognition-** targets one word at a time (generally for languages use a " " as a word splitter)
3. **Intelligent Character Recognition (ICR)** – targets handwritten print scripts or cursive text one glyph or character at a time using Machine Learning.
4. **Intelligent Word Recognition**- targets one word at a time. Mostly used for languages where glyphs are not separated in cursive script.

1.5 Organization

Chapter 1: Includes a brief introduction to the project. A basic idea of what we are doing and what we are trying to accomplish with this Project has also been provided and technologies we are using in this project have also been listed.

Chapter 2: Includes literature survey. We have studied various papers and journal from reputed sources on machine learning and web scraping and have mentioned those in this chapter.

Chapter 3: Includes details on system development. Explanation about the project design, models implemented and formulas applied have been mentioned.

Chapter 4: Includes result and result from analysis. This section provides the results are implemented models are yielding and accuracy of those results have been scrutinized in this section.

Chapter 5: Conclusion. Outcomes and the future scope of the project have been discussed briefly.

Chapter-2 LITERATURE SURVEY

2.1 Introduction to Literature Review

There has been a substantial increase in the research associated with the prediction of stock market fluctuations. This area of research walks alongside with other businesses in the world thus rightly naming it as the “Mother of all Businesses”. Before going in the details of the research associated with the stock market, there is a need to understand the terms that are associated with it.

2.1.1 Key Terms

i. **UserAgent**

User Agent is the description of the web browser and OS which sends the request to the web server. User Agent field is included in the HTTP header and it vary from browser to browser. User Agent field tells about the Operating system, Windows version, and information about web browser. Below is the example of User Agent field:-

Mozilla/5.0(compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)

ii. **Clickstream**

Clickstream is a kind of data which contains the information about the user’s clicks on a website. It helps to keep a track of user activity on the website.

2.2 Summary of Papers

2.2.1

Title	**Discovery of Web Robot Sessions based on their Navigational Patterns**
Authors	Pang-Ning Tan and Vipin Kumar Department of Computer Science, University of Minnesota
Year of Publications	December 2002
Publishing Details	Intelligent technologies for Information Analysis, Springer, Berlin, Heidelberg
Summary	Web robots or Web Scrapers are programs that as a matter of course cut across the hyperlink structure of the WWW (World Wide Web) to discover and extract information. Many reasons are there to recognize visits by Web Scrapers and differentiate them from human users.The e-commerce website owners are especially troubled about the unapproved deployment of bots for collecting business intel from their websites. Also, these robots have a tendency to consume a large amount of network bandwidth at the price of other human users. Sessions due to bots make it challenging to do conduct clickstream analysis on the website data. Commonly used techniques used for detecting Web scrapers rely on identifying the IP and user agent of requests. These techniques can be used to detect many bots but are not satisfactory to catch hiding and unknown bots. In this, they have come up with a different approach that uses navigational patterns of the user's clickstream data to check whether it's a robot or not. The experimental results of their research implied that very solid classification models can be fabricated using their way. They have also shown that their classification models were able to detect many hiding and unknown bots.

2.2.2

Title	**Detection of Web Scraping Using Machine Learning**
Authors	Kaushal Parikh, Dilip Singh2, Dinesh Yadav, Mansingh Rathod Department of Information Technology, KJSIEIT, Mumbai
Year of Publications	March 2018
Publishing Details	Open Access International Journal of Science and Engineering
Summary	Web Scraping is a technique of taking data from a website with or without the permission of the owner of that site. As we know it is done through various tools like Visual Web Ripper, ImportIO, Scrappy etc. This paper focuses on the use of Machine Learning to remove the scraping of data from a website. Now-a-days AI is used to detect our daily problems and provide a solution for it. Similarily in this case Machine Learning is used to get rid of this problem The steps taken by these authors are:- Firstly they have collected the logs of the site which has been under attack and parse the data, then they saved it to the elastic search database and visualise it using Kibana. Data is imported with the help of some script and then features are extracted from this data. After feature extraction, these features are fed to machine in order to train it and attack pattern is detected.

2.2.3

Title	**Using Diverse Detectors for Detecting Malicious Web Scraping**

	Activity
Authors	Pedro Marques, Zayani Dabbabi, Miruna-Mihaela Mironescu, Olivier Thonnard, Alysson Bessani, Frances Buontempo , Ilir Gashi
Year of Publications	June 2018
Publishing Details	IEEE/IFIP International Conference on Dependable Systems and Networks
Summary	The paper focuses on the use of tools which can detect the malicious web scraping activity. These tools were used on the HTTP Apache access log provided by Amadeus. A commercial as well as an in-house tool called Arcane are used to detect the malicious activity. The data set consisted of the activity for 8 days and these two tools gave the information about the total HTTP request and then the diversity given by these tools were examined. The HTTP alerts generated by these tools were also considered.

Chapter-3 SYSTEM DEVELOPMENT

The system which we have developed mimics the actual scenario in which the scraper attack any website and what mechanism can be used to detect the presence of the bot on the website. The System comprises two Subsystems :

3.1 Attack System

The attacking scenario contains many steps on the type of scraping is done. Here to do the price scrapping we have to build the scrapper in three parts

i. **GUI Graphical User Interface**
ii. **Web Crawler or Spider**
iii. **Web Scraper**

1.Graphical User Interface

The GUI refers to some type of visible program output, with which the user can interact to give the input to the scraper to do price scrapping. In this**,** we have used the library Tkinter in python 3. In this, we have made a simple Browser window consisting of one input field which takes the input from the user of the website to be scraped off the price and then there is a button named label “go” is present which starts the spider and sends the URL of the website to the spider.

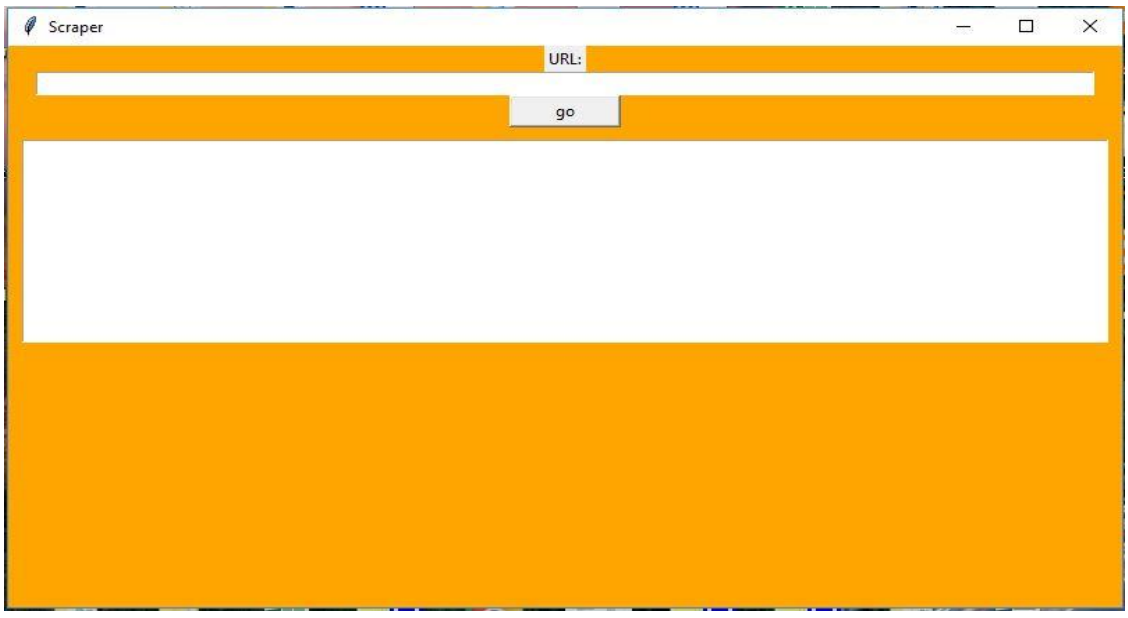


Fig.2 GUI Interface

2. Spider or Web crawler

In this spider, the basic work that is done is to collect all the links present on the websites that belong to the website only. The process of crawling all website is a very hectic process as nowadays we know that the single website contains a great number of links present in it or have a very high amount of pages present in the website. So we have to make a web crawler which is efficient and speedy too that increases the difficulty of the making of spider very much.

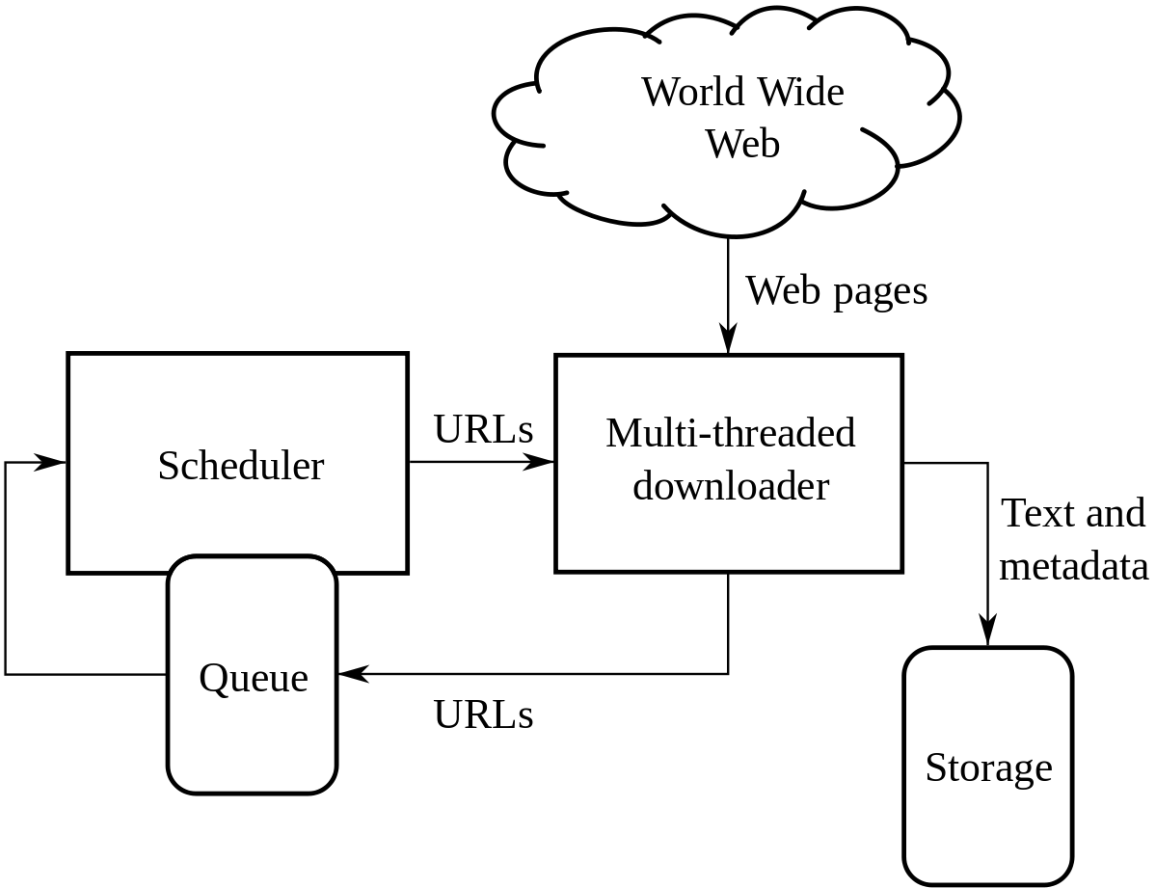
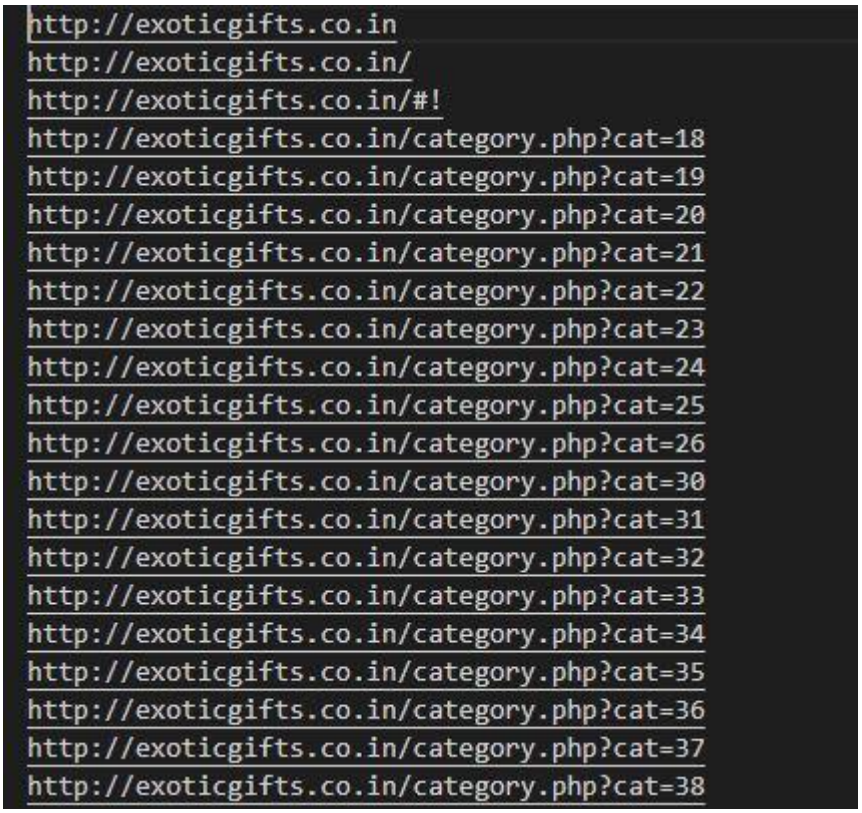


Fig 3 Flow Diagram of Working of Spider

So to overcome this problem, we have to divide the work into different workers or spiders. This can be done by using threads in the making of spider which can divide the work in themselves and can specify the work. This is an efficient process to specify the process but it creates one more problem of inconsistency and redundancy as well as various Spider will try to access the same link and will do the scraping or crawling of the web pages multiple time which is a waste of resources and time as well. So there is one

solution to this problem that we are facing that if we can make only one instance of the links queued and one instance of links to be crawled so that every spider knows which link to pick out of the queued ones. When the process of the spider comes to an end the process of the scraping begins. All those crawled links of the website are stored in some temporary memory or in a file which can be accessed by the scrapper after the crawling is being done.



Web Scraper

The scraper takes the input of the links which has been crawled by the spider and starts the scraping of data in the following ways:

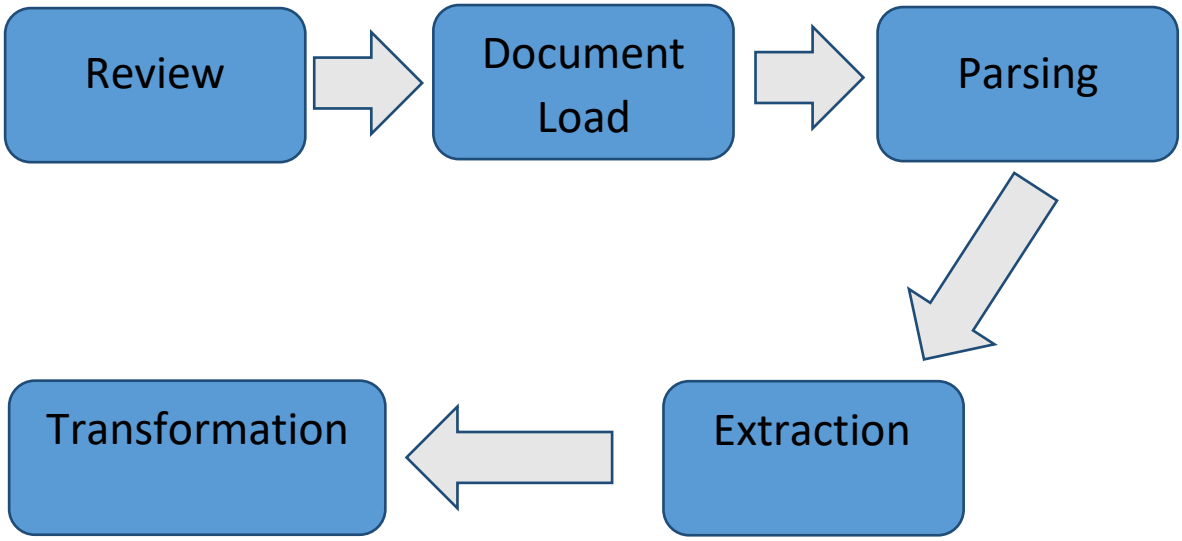


Fig 4 Flow Diagram of Working of WEB Scraper

Step1: Review

It is the most important step of the web scraping as all of the further process depends on this step. Because it is the step in which the structure of the web page is analyzed and the most effective techniques of scraping are decided as to how to scrape the data or in which way we can filter out the data from the web page.

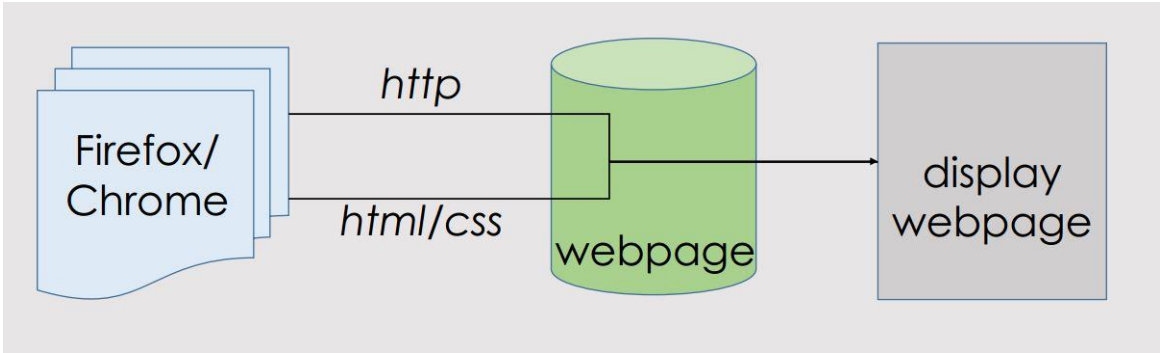


Fig 5 How web scraper review the site

Step 2: Access or Document Load

In this step, the page is requested from the server and the page downloaded from the internet in order to parse it into a string of characters. Then there comes the HTML parser which converts the downloaded page into Document Object Model so the scraper can scrape out the data from the structured document easily by accessing various techniques HTML tags and Dom objects.

The **requests** library can be used to and the **get** method:

```
from requests import get



response=get(url)
```

Step 3: Parsing

Once the page is parsed into the required document or data then it is analyzed for any type of pattern present in the document where the required data is being stored in the document as finding some pattern will ease the process of filtering the information out of the data or the document.

The libraries it can make use of are **BeautifulSoup and HTML Parser**

```
import bs4


soup = bs4.BeautifulSoup(response.text, 'html.parser')
```

Step 4: Extraction

Then there comes the process of filtering the data out or scraping as we know there are various steps of filtering the data is through Regular expression, accessing the DOM

object or by accessing the Html tags etc. But it depends on the placement of the data inside the HTML page just loaded on to the buffer. So different methods are applied to filter out the data sometime the methods are combined to do the scraping.

```
req = requests.get('http://exoticgifts.co.in/')
soup = bs4.BeautifulSoup(req.text, 'HTML.parser')
elements = soup.find_all('div', {'class': 'card-content'})

record = []
for element in elements:
    name = element.span.contents[0]
    price = element.p.a.text
    price = price[price.find('Rs.')+3:-1]
    pid = element.p.a['href']
    start = pid.find('pid')+4
    prodId = pid[start:-1]
    record.append((prodId, name, price))
```

Step 5: Transformation

After the data is filtered out the structured or meaningful information is present on hand which has to be stored somewhere in order to use that information in near future. So the data can be stored in various places like in CSV files, database, cloud and many more. the data is stored in forms of tables mostly as it is easier to classify that data for further use in various things.

The **pandas** module can be used to store the **DataFrame** of the data:

```
import pandas
import os

        records.append((ip, date, method, byte, name))

df = pandas.DataFrame(
```

```
    records, columns=['ip', 'date', 'method', 'byte', 'name'])
df.to_csv('logdata.csv', index=False, encoding='utf-8')
```

This is the whole process of scraping the data off the website we are generally taking the case of price so the main websites will be consisting of the various e-commerce website present on the internet. There are thousands of e-commerce websites present on the internet which can be scraped for their data this data can be used to enhance one's business by getting the lowest price for the things being searched or posted on his/her website. And increasing the hits on the website and also increasing the customer market and the profit by selling these things on the fewer prices than the competitor websites.

3.2 Defense System

Scraping is very useful when we want to scrape large amount of data from any website as it saves our time and resources but sometimes people make the wrong use of this technique. For ex:- we have two online e-commerce website which are competitors. Owner of first website take the prices of another website using scraping techniques and show his prices less than other which is illegal. In order to get rid of this problem, some defence mechanisms are designed. These defense mechanisms are of two types:-

1. Preventive Approach
2. Detective Approach

3.2.1 Preventive Approach

This kind of approach is used to prevent the bots from entering into the websites and stealing the important data. Various sites including Google, Amazon, Coursera etc. uses this approach for their data safety. Various techniques of preventive approach are:-

1. Manually
2. CAPTCHAs
3. Geo-Fencing
4. Flow Enforcement

3.2.1.1 Manually

During web scraping attacker HTTP request your server which further sends back the web page to the program. The attacker parses this HTML and extracts the required information. This process is repeated over and over again so that all the required information can be extracted from the website.

In order to get rid of this problem the webmaster can follow some steps:-

1. **Take a legal stand:**-A person can clearly mention that web scraping is not allowed on this website. For instance Medium's terms of services contains following line:-

   Crawling the Services is allowed if done in accordance with the provisions of our robots.txt file, but scraping the Services is prohibited.

   The owner of website can take legal action against the potential attacker.
2. **Prevent Denial of Service(DoS) attack:-** After putting the legal notice there is a chance that attacker will again attack on your website. He/She can cause disrupt the daily services of your website causing DoS attack on your servers. To avoid this kind of situation you can detect the IP address of attacker and can block the requests coming from this IP address.
3. **Using** *.htaccess* **file:-** .htaccess is the configuration file of Apache web server and it can help to prevent the scraping of your data. First you have to detect the attacker using Google Webmasters and then you can stop the attacker by doing some changes in configuration file.

3.2.1.2 CAPTCHAs

CAPTCHA stands for **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part. This technique is used to detect whether the person entering the website is a bot or human with the help of images or text. We can see when we enter any website it asks to write the text written in distorted form, this type of captcha was invented in 1997

by two parallel groups. CAPTCHA is efficient technique to prevent web scraping but sometimes it irritates the user also.

- You can design your own CAPTCHA by writing few lines of code but it is easier to use something like Google's reCAPTCHA in which user have to only tick a radio button and he can enter the website easily.
- If you are making your own CAPTCHA, try to hide its solution. If you provide the solution of CAPTCHA in its code it will become easier for attacker to decode it.
- Image CAPTCHAs can irritate the user easily as if there is some fault detected it will appear again and again.

3.2.1.3 Geo Fencing

Geo fencing is a technique in which websites are exposed only in some specific geographical regions where they do their business. This will not stop the web scraping but the attacker has to make extra effort because they have to run their web scraper over a specific geographic location. The attacker have to use the VPN link to a local Point of Presence(PoP-: Point at which two or more network build a connection between them).

3.2.2 Detective Approach

The Detective Approach is used to detect the bots entering into the websites to steal data and take defensive steps according to the behaviour of these bots. Various techniques can be followed to stop the scraping of data.

1. Machine Learning Approach
2. Robots.txt Access Approach
3. User Agent Check
4. IP Address Approach

3.2.2.1 Machine Leaning Approach

In machine learning approach we detect the behaviour in which the bots are stealing our data. We take a dataset which includes the information about uses to differentiate between bots and humans. Database includes fields like time spent on a site, files scraped by any

user etc. This database is passed to the supervised ML phase in which there are two phases like training set and testing set.

In training set the machine will be trained using above explained database and various training models like K-means clustering, PAM etc. and then testing will be done on other data.

3.2.2.2 Robot.txt Access

The Robot Exclusion Standard allows the Web Administrator to specify the part of site which is off-limits. Robot.txt includes the files specified by Web Administrator which are off-limits for robots which are going to attack the site. So, whenever robot visits any site, it should examine this file first. For ex:- when a bot visit **www.abc.com ,** the robot.txt file is accessed using the URL:- http://www. abc.com/robots.txt.

Let us consider that the robot.txt file contains file named **X.html** which is not allowed to get accessed by the robot. But when bot comes to the site is scans each and every file related to the website. It will also go to **X.html** file and the bots get detected easily when they try to access robot.txt file. Any website does not provide direct hyperlink to this file from any HTML page, so many users are unaware of these files.

3.2.2.3 User Agent Check

When the bots do not work fine while scraping data it can cause the DoS(Denial of Service) and server will not be able to fine and there will be overloading on server. It is required to have a beneficial relationship between Web servers and bots. To establish a good relation between server and bot, the robot must declare its identity to the server. The User Agent field plays an important role which contains the identity of bot including its name. Sometimes bot designers' uses user agent fields as that of Web browsers and detection of bot becomes very difficult.

3.2.2.4 IP Address Check

Another way to detect a bot is to match the IP address of Web client with web bots. This approach is not much useful as compared to other because the World Wide Web expands, it has become difficult to keep the record of all the bots coming onto the site. Also sometimes the web client uses the same IP address as that of the web bot. This approach

is applicable only if the robot has been detected earlier. This doesn't mean that new robots could not get detected, new robots can be detected by examining the top visited IP address and manually checking the origin of all the IP addresses. This approach is very time consuming and require more labour than any other approach.

3.3 Test Plan

We are going to use various algorithms to detect the difference between bots and humans visiting the site. For this we have made a dummy e-commerce website in which we want scrape the prices of different items. As mentioned in chapter 3 we have designed web crawlers and scrapers to perform this function. To differentiate we will make a dataset in which there will be name of bot or human that are visiting our site along with the date, time, and IP Address of the visiting device. It is stored in the log file of the website and we have to extract it in a structured well defined document or data set which can help us is recognising differences between users and bots.

```
220.181.7.76 - - [20/May/2010:07:26:23 +0100] "GET / HTTP/1.1"20029460"-
""Baiduspider+(+http://www.baidu.com/search/spider.htm)"
220.181.7.116 - - [20/May/2010:07:26:43 +0100] "GET / HTTP/1.1"20029460"-
""Baiduspider+(+http://www.baidu.com/search/spider.htm)"
209.85.228.85 - - [20/May/2010:07:26:49 +0100] "GET /feeds/latest/
HTTP/1.1"20045088"-""FeedBurner/1.0 (http://www.FeedBurner.com)"
209.85.228.84 - - [20/May/2010:07:26:57 +0100] "GET /feeds/latest/
HTTP/1.1"20045088"-""FeedBurner/1.0 (http://www.FeedBurner.com)"
125.22.2.42 - - [20/May/2010:07:32:04 +0100] "GET /feeds/latest/
HTTP/1.1"20045088"-""Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1;
.NET CLR 1.1.4322; .NET CLR 2.0.50727; MS-RTC LM 8; .NET CLR 3.0.04506.648;
.NET CLR 3.5.21022; MSOffice 12)"
98.155.184.203 - - [20/May/2010:07:32:43 +0100] "GET /feeds/latest/
HTTP/1.1"20045088"-""Apple-PubSub/65.12.1"
61.135.216.104 - - [20/May/2010:07:32:56 +0100] "GET /feeds/latest/
HTTP/1.1"20045088"-""Mozilla/5.0
(compatible;YoudaoFeedFetcher/1.0;http://www.youdao.com/help/reader/faq/top
ic006/;1 subscribers;)"
99.168.127.53 - - [20/May/2010:07:34:11 +0100] "GET / HTTP/1.1"20029460"-
""Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_3 like Mac OS X; en-us)
```

```
AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7E18
Safari/528.16"
```

The data is now passed through a python script to convert it into a .csv structured format.

The Code:

```
import re
import pandas
import os

ipDate = []
records = []
withopen("New folder\\access.log", 'rt') as f:
for line in f:
        ipDate.append(line.split('\"')[0])
        method = line.split('\"')[1]
        byte = line.split('\"')[2]
        name = line.split('\"')[5]
        ip = (ipDate[0].split("- -")[0]).strip()
        date = ((ipDate[0].split("- -")[1]).strip().split(' ')[0])[1:]
# print(ip+"\t"+date+"\t"+method+"\t"+byte+"\t"+name)
        records.append((ip, date, method, byte, name))

df = pandas.DataFrame(
    records, columns=['ip', 'date', 'method', 'byte', 'name'])
df.to_csv('logdata.csv', index=False, encoding='utf-8')
```

Output is:

ip	date	method	byte	name
220.181.7.76	20/May/2010:07:26:23	GET / HTTP/1.1	200 29460	Baiduspider+(+http://www.baidu.com/search/spider.htm)
220.181.7.76	20/May/2010:07:26:23	GET / HTTP/1.1	200 29460	Baiduspider+(+http://www.baidu.com/search/spider.htm)
220.181.7.76	20/May/2010:07:26:23	GET /feeds/latest/ HTTP/1.1	200 45088	FeedBurner/1.0 (http://www.FeedBurner.com)
220.181.7.76	20/May/2010:07:26:23	GET /feeds/latest/ HTTP/1.1	200 45088	FeedBurner/1.0 (http://www.FeedBurner.com)
220.181.7.76	20/May/2010:07:26:23	GET /feeds/latest/ HTTP/1.1	200 45088	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR
220.181.7.76	20/May/2010:07:26:23	GET /feeds/latest/ HTTP/1.1	200 45088	Apple-PubSub/65.12.1
220.181.7.76	20/May/2010:07:26:23	GET /feeds/latest/ HTTP/1.1	200 45088	Mozilla/5.0 (compatible;YoudaoFeedFetcher/1.0;http://ww
220.181.7.76	20/May/2010:07:26:23	GET / HTTP/1.1	200 29460	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_3 like Mac OS X;
220.181.7.76	20/May/2010:07:26:23	GET /media/style.css HTTP/1.1	200 4847	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_3 like Mac OS X;
220.181.7.76	20/May/2010:07:26:23	GET /media/exmpl.png HTTP/1.1	200 28479	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_3 like Mac OS X;
220.181.7.76	20/May/2010:07:26:23	GET /media/img/m-act.gif HTTP/1	200 143	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_3 like Mac OS X;
220.181.7.76	20/May/2010:07:26:23	GET /media/img/m-inact.gif HTTF	200 2571	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_3 like Mac OS X;
220.181.7.76	20/May/2010:07:26:23	GET /media/img/side-container.g	200 1415	Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_3 like Mac OS X;

The proposed approach to rectify this problem is given by this flowchart:-

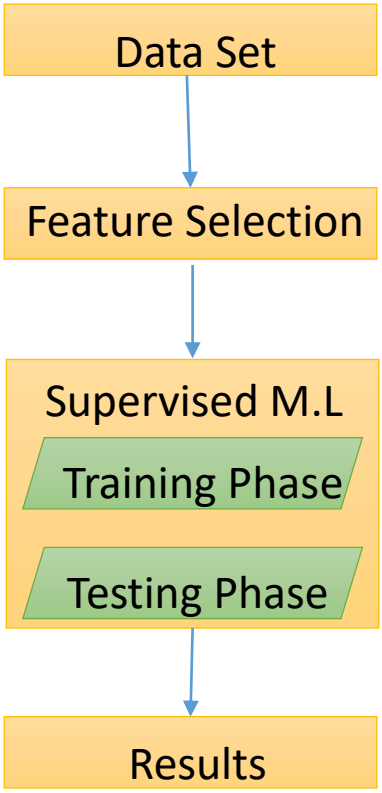


Fig 6 Flow Chart of Suggested solution

From the above flow diagram, we can see that after taking the data set we are going to use various algorithms which will be implemented like K-means clustering and PAM. In supervised ML there are two phases

1. Training Phase:- In this phase machine will be trained with help of some data present in the dataset to implement various algorithms and compute the results.
2. Testing Phase:- This phase will be useful to make predictions and detect whether the device entering our site is a bot or human.

After the implementation of all algorithms, we will compare the results to check which algorithm is best suitable for detecting the bots among human.

Chapter-4 PERFORMANCE ANALYSIS

4.1 Technique Used

4.1.1. K-means clustering:- is a unsupervised learning technique which is used to categorize the items into groups. It aims to partition the n observation into k clusters and is used when the data is unlabeled. The algorithm work iteratively to assign each data point to one of the K group based on the provided features.

ALGORITHM

1. Clusters the data into *k* groups where *k* is predefined.
2. Select *k* points at random as cluster centers.
3. Assign objects to their closest cluster center according to the Euclidean distance/Manhattan distance function.
4. Calculate the centroid or mean of all objects in each cluster.
5. Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.

4.2 Dataset

	IP Address	Time Spent	Number of requests	Average Time	Byte Entropy	Time Entropy
2	220.181.7.76	20	1	20	1	1
3	220.181.7.116	6	1	6	0.968062152	1
4	209.85.228.85	8	1	8	1	1
5	209.85.228.84	307	1	307	1	1
6	125.22.2.42	39	1	39	1	1
7	98.155.184.203	13	1	13	1	1
8	61.135.216.104	75	1	75	0.968062152	1
9	99.168.127.53	3628	8	453.5	2.596506103	2.725481
10	66.249.65.40	48	3	16	0.996099652	2
11	67.195.114.50	123	1	123	0.030175579	1
12	75.139.202.90	3628	7	518.2857143	2.104330872	2.405639
13	66.249.65.40	11	1	11	0	1
14	81.170.208.83	27	50	0.54	0	1.629756
15	67.195.114.50	71	1	71	0.004322274	1
16	193.189.143.44	115	1	115	0.968062152	1
17	220.181.7.76	6	1	6	0.999520389	1

4.2.1 BASIC FEATURES

1. **IP Address:-** This field represents the location from which the user is accessing the data.

2. **Time Stamp:-** This field represents the date and time at which various requests are done for particular links.

3. **Method:-** This field represents whether the user is requesting using GET method or POST method.

4.2.2 DERIVED FEATURES

1. **Total Pages:-** No. of pages requested

2. **Total Time:-** Total time of a session

3. **Average Time:-** Average time between two HTML requests

4. ENTROPY: Entropy is the measure of the randomness present in the information that is being processed. Therefore, a higher value of Entropy signifies large randomness in the data. This randomness makes it harder to extract any patterns or conclusions from the information available to us. Mathematically, Entropy can be calculated using a simple mathematical formula.

$$E = -\sum_{i=1}^{c-1} p(i|t) * [\log_2 p(i|t)]$$

$$where\ p(i|t): fractions\ of\ records\ belonging\ to\ ith\ class$$

4.3 Results

1. We have plotted the time entropy and byte entropy with respect to number of requests and obtained the following results.

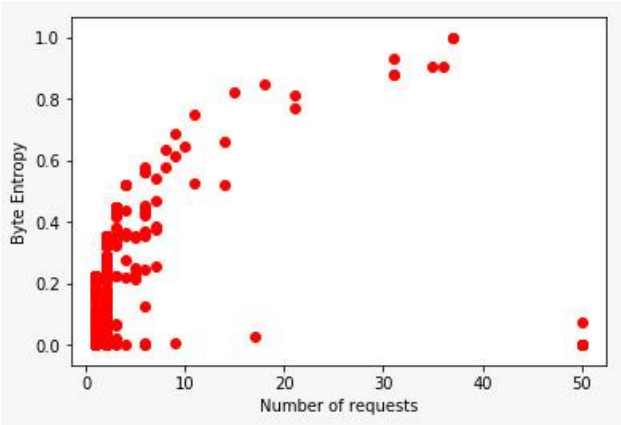


Fig 7 Byte Entropy to Number of Request

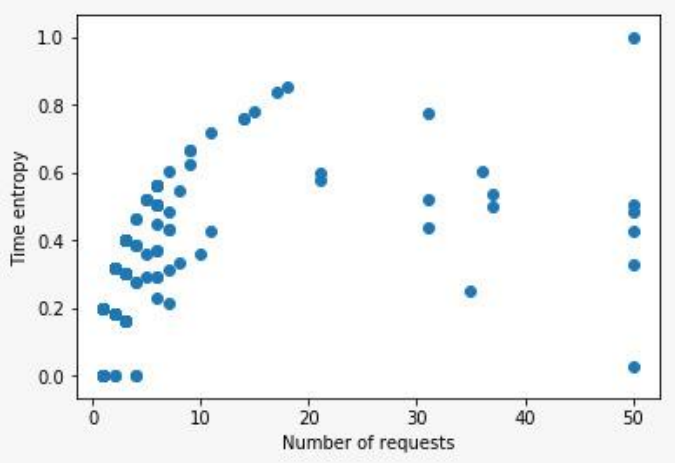


Fig 8 Time entropy to Number of request

We have considered the following conditions to illustrate the difference between humas and bots:-

Entropy > 0.6 →Bot

Entropy < 0.5 →Human

2. Now we are using above mentioned clustering technique to train and test our dataset in order to differentiate between bots and human.

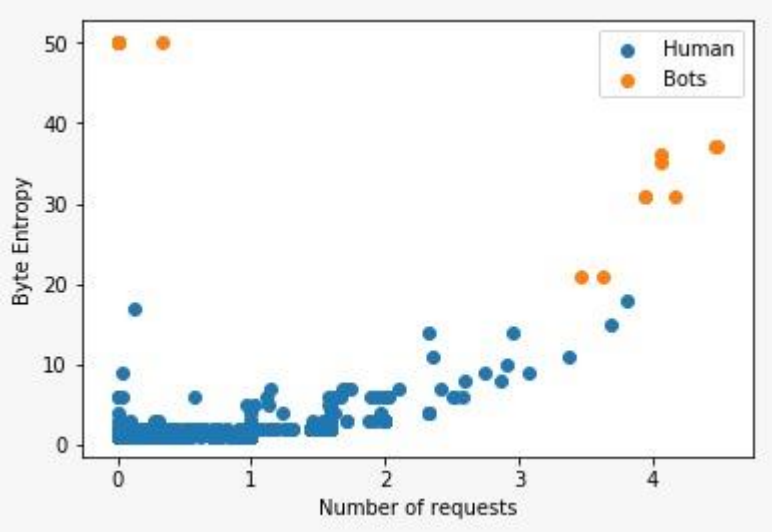


Fig 9:- K-means clustering using Euclidean Distance

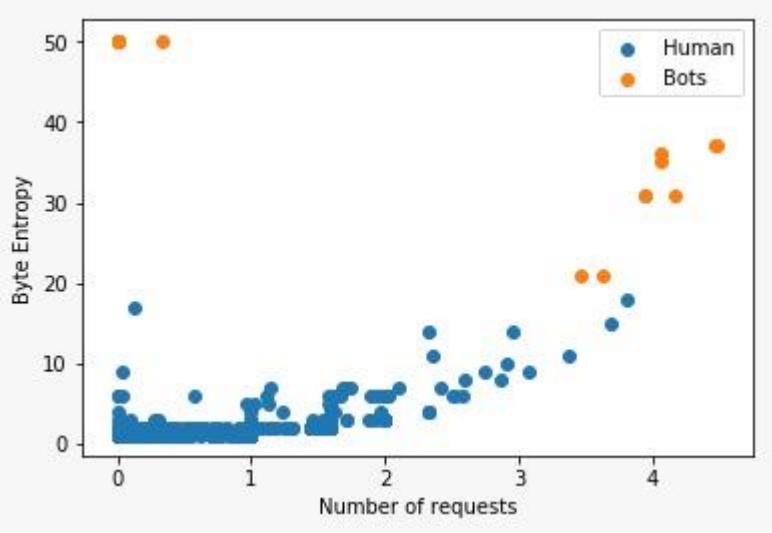


Fig 10:- K-means clustering using Manhattan Distance

Distance Formula Used	Purity	Mean Square Error
Euclidean Formula	0.630152	2.050191
Manhattan Formula	0.630152	1.320987

Table 1 showing the comparison between two distance formulas

Distance formula will not affect the clusters but only reduce the error as we are able to see in the above table

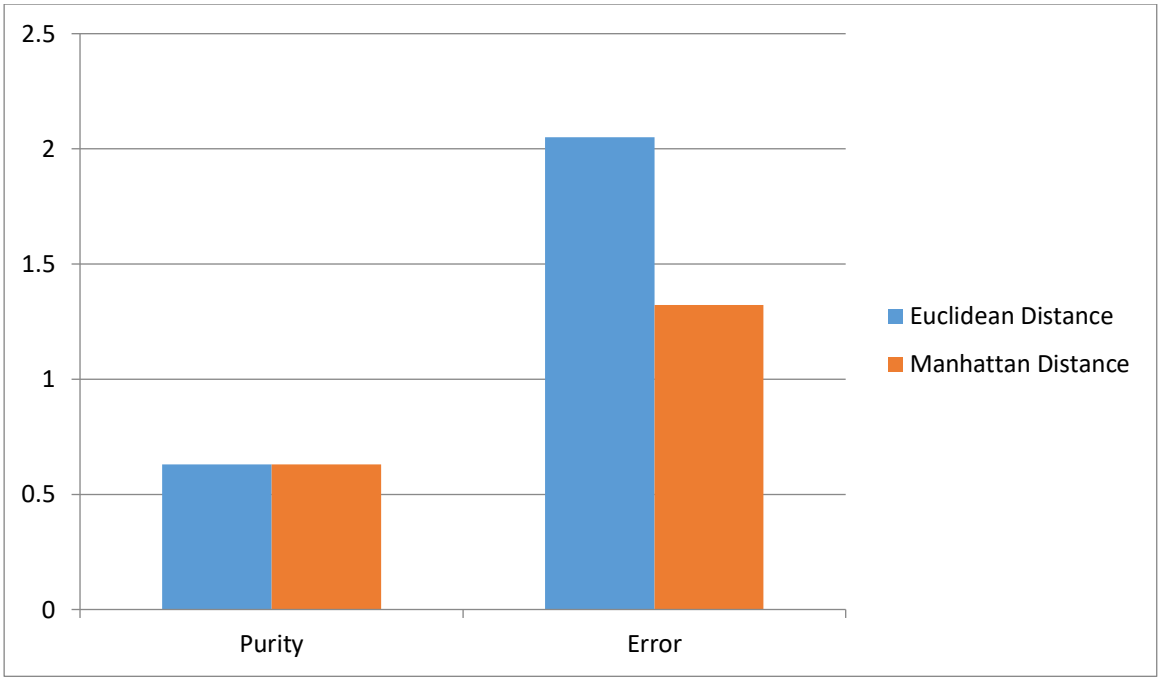


Fig 11:- Graphical representation of comparison between two techniques

Chapter-5 CONCLUSIONS

The first and most significant thing to be cautious about when composing a web scraper is that it regularly includes questioning a site more than once and getting to a possibly extensive number of pages. For some sites web scraping is considered to be illegal and you have to keep in mind the copyrights of any site you are going to scrape. Copyrights and data privacy laws vary country to country. For eg;- in Australia, it can be illegal to scrape and store personal information such as names, phone numbers and email addresses, even if they are publicly available. Different types of techniques are used to scrape the data mentioned in the above chapters but all those techniques can only be used if any person is using them in the right way without breaking any law.

References

1. Discovery of web robots sessions based on Navigational Pattern by Pang-Ning Tan and Vipin Kumar.

2. DETECTION OF WEB SCRAPING USING MACHINE LEARNING
Kaushal Parikh, Dilip Singh, Dinesh Yadav, Mansingh Rathod
Student, Department of Information Technology, KJSIEIT, Mumbai-400022, India1
Student, Department of Information Technology, KJSIEIT, Mumbai-400022, India2
Student, Department of Information Technology, KJSIEIT, Mumbai-400022, India3
Professor, Department of Information Technology, KJSIEIT, Mumbai-400022, India4

3. The Use of Web-scraping Software in Searching for Grey Literature by Neal R. Haddaway

4. Exploiting web scraping in a collaborative filtering based approach to web advertising Eloisa Vargiu1, 2, Mirko Urru1 1. Dipartimento di Matematica e Informatica, Università di Cagliari, Italy. 2. Barcelona Digital Technology Centre, Spain

5. Constrained K-means Clustering with Background Knowledge Kiri Wagstaff Claire Cardie Department of Computer Science, Cornell University, Ithaca, NY 14853 USA

6. Using Diverse Detectors for Detecting Malicious Web Scraping Activity by Pedro Marques, Zayani Dabbabi , Miruna-Mihaela Mironescu , Olivier Thonnard , Alysson Bessani , Frances Buontempo , Ilir Gashi