CHAPTER – 4

WEB CRAWLER MODEL FOR SECURE PAGES

<table>
<thead>
<tr>
<th>S.No</th>
<th>Contents</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>76</td>
</tr>
<tr>
<td>4.2</td>
<td>Associated Work</td>
<td>78</td>
</tr>
<tr>
<td>4.3</td>
<td>The Approach And Working</td>
<td>82</td>
</tr>
<tr>
<td>4.4</td>
<td>Observations And Results</td>
<td>87</td>
</tr>
<tr>
<td>4.5</td>
<td>Chapter Summary</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>88</td>
</tr>
</tbody>
</table>
CHAPTER - 4

WEB CRAWLER MODEL FOR SECURE PAGES

4.1 Introduction

4.1.1 Web Crawler

A web agent must take into account an array of parameters in order to execute a search query. The working of a deep web crawler differs with the working of a traditional web crawler in several aspects, initially the web, taken as a graph by the web crawler has to be traversed in a different path with diverse authentication and permission to enter a secure and restricted network. The process of doing so is not simple, as it involves structuring and programming the web crawler to do so.

4.1.2 Web Crawler Categories

Basically the web crawlers are divided into several categories listed below

(i) **Dynamic Web Crawler**: The crawler returns dynamic content in response to the submitted query or completed form. The primary search attribute for this kind of web crawler is text fields.

(ii) **Unlinked Pages/Content**: several pages over the web are independent and are not connected to any other in/back links preventing them to be found by search engines. These contents are referred to as back links.
(iii) **Private Pages/Web**: Several sites that are administered by organisation and contain certain copyrighted material needs a registration to access it. There is also a possibility of the website to ask the user to authenticate. Most of these pages are encrypted and may also require Digital Signature for the browser to access.

(iv) **Context Oriented Web**: These web pages are accessible only by a range of IP addresses are kept in the intranet, that are ready to be accessed by internet too.

(v) **Partial Access Web**: Several pages limit the access of their pages to avoid search engine to display the content in a technical way, by the use of Captcha code and restriction of Meta data, preventing the web crawler’s entry.

(vi) **Scripted Web Content**: pages are accessible only through the link provided by web servers or name space provided by the cloud. Some video, flash content and applets will also falls under this category

(vii) **Non-HTML Content**: Certain content embedded in image and video files are not handled by search engines.

Other than these categories of contents, there are several different formats of data that are inaccessible by any of the web crawlers. Most of the internet search happens through the Hyper Text Transfer Protocol (HTTP), the existence of other protocols like Gopher, File Transfer Protocol (FTP); Hyper Text Transfer Protocol Secure (HTTPS) also restrict the content to be searched by traditional search engines.
This chapter deals with the techniques by which these above mentioned information known as deep-content or hidden content or invisible content for web crawlers can be included in the search outcomes of a traditional web crawler.

4.1.3 Traditional Web and Hidden Web

The entire web can be categorised into two types, the traditional web and the hidden web as in [87,88,89]. The traditional web is the one, which is normally deployed by general purpose search engine. The hidden web which has got abundant and important information, but cannot be traversed directly by a general purpose search engine as it has certain security concerns on the crawlers. Internet survey says that there are about 3,00,000 Hidden Web databases as in [90]. Few qualities of the hidden web contains are containing high quality contents exceeding all print data available.

4.2. Associated Work

There exist several other web crawlers that are intended to search hidden web pages. A periodical survey of such web crawler is being done here in order to know their limitations and constraints to overcome the same in the proposed framework. By the way of setting apart loud and insignificant blocks from the web documents can make easy search and to improve the web crawler has been proved. This way can facilitate even to search hidden web pages as in [91].
The most popular ones are DOM-based segmentation as in [92], Location-Based Segmentation as in [93] and Vision-Based Page Segmentation as in [94]. This chapter deals with capability of differentiating features of the web page as blocks. Modelling is done on the same to find some insights to get the knowledge of the page by using 2 techniques based on neural network and Support Vector Machine (SVM) facilitating the page to be found.

The accessibility of robust, supple Information Extraction (IE) systems for transforming the internet documents into algorithm, program readable structures like one as relational database that will help the search engine to search easily as in [95]. The difficulty of getting website frame, i.e. getting the fundamental hyperlink structure used to sort out the data documents in a taken website. They have proposed an automated Back On Topic (BOT) like algorithm that has the functionality of discovering the skeleton of a given website.

The Search Engine Watch (SEW) algorithm, it inspects hyperlinks in gatherings and recognizes the route connections that focus to pages in the following level in the site structure. Here the whole skeleton is then built by recursively bringing pages sharp by the revealed connections and dissecting these pages utilizing the same methodology is explained as in [97].

Table 4.1: Features of Web Crawlers

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Types of Webcrawlers</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
| 1 | Gnu Wget | - Can resume aborted download.  
- Can use filename wild cards and recursively mirror directories.  
- Supports HTTP proxies and HTTP cookies.  
- Supports persistent HTTP connections. |
| 2 | WebSphinx | - Multithreaded Web page retrieval  
- An object model that clearly represents documents and links  
- Supports for reusable document information classifiers  
- Support for the robot exclusion standard |
| 3 | Heritrix | - Ability to run multiple crawl jobs simultaneously.  
- Ability to browse and modify the configured Spring beans.  
- Increased scalability.  
- Increased flexibility when modifying a running crawl |
| 4 | J-Spider | - Checks sites for errors.  
- Outgoing and/or internal link checking.  
- Analyze site structure.  
- Download complete web sites. |

### 4.2.1. Alternative Techniques for Web Crawlers

The issue of extraction of search term for over millions and billions of information have touched upon the issue of scalability and how approaches can be made for very large databases as in [97]. These chapters have focused completely on current day crawlers and their inefficiencies in capturing the correct data. This analysis covers the concept of present robots retrieving content only from the Publicly Index able Web (PIW), the documents available only by following hypertext links and
neglecting the documents that require certain authorization or prior registration for viewing them as in [98].

The different characteristics of web data, the basic mechanism of web mining and its several types are summarized. The reason for the usage of web mining for the crawler functionality is well explained here in the chapter. Even the limitations of some of the algorithms are listed. The chapter talks about the usage of fields like soft computing, fuzzy logic, artificial networks and genetic algorithms for the creation of crawler. This chapter gives the reader the future design that can be done with the help of the alternate technologies available as in [99].

4.2.2 Intelligent Web Agents

The later part of the chapter deals with describing the characteristics of web data, the different components, types of web mining and the limitations of existing web mining methods. The applications that can be done with the help of these alternative techniques are also described. The survey involved in the chapter is in-depth and surveys all frameworks which mean to powerfully extricate qualified information from new assets.

Shrewd web executors are ready to hunt down significant informative content utilizing attributes of a specific dominion got from the client profile to form and decipher the uncovered informative data. There are several available agents such as Harvest as in [100], FAQ-Finder as in [100], Information Manifold as in [101], OCCAM as in [102], and Parasite that rely on the predefined domain specific template information and are experts in finding and retrieving specific information.
The Harvest as in [100] system depends upon the semi-structured documents to extract information and it has the capability to exercise a search in a latex file and a post-script file. At most used well in bibliography search and reference search, is a great tool for researchers as it searches with key terms like authors and conference information. In the same way FAQ-Finder as in [100], is a great tool to answer Frequently Asked Questions (FAQs), by collecting answers from the web. The other systems described are ShopBot as in [103] and Internet Learning Agent as in [104] retrieves product information from numerous vendor website using generic information of the product domain. The Features of different Web Crawlers are as shown in table1.

4.2.3 Ranking

The evolving web architecture and the behaviour of web search engines have to be altered in order to get the desired results as in [105]. As in [106] the authors’ talk about ranking based search tools like Pub Med that allows users to present highly easy-to-read Boolean keyword queries, but ranks the query results by date only. A proposed approach is to submit a disjunctive query with all query keywords, retrieve all the returned matching documents, and then re rank them. The user fills up a form in order to get a set of relevant data. The process is tedious for a long run and when the number of data to be retrieved is huge, is discussed as in [108]. In the thesis by Tina Eliassi-Rad, several works that retrieve hidden pages are discussed.

There are many proposed hidden pages techniques, which are a unique web crawler algorithm to do the hidden page search as in [110]. An architectural model for
extracting hidden web data is presented as in [109]. The end of the survey circumstances that much less work has been carried out an advanced form based search algorithm that is even capable of filling forms and captcha codes.

4.3 The Approach and Working

Consider a situation, where a user is to search a term “ipad”. The main focus of a traditional crawler will be to list a set of search results mostly consisting of the information about the search term and certain shopping options for the search term “ipad”. It might omit several websites with best offer on the same search term “ipad” as it involves, only a registered user to give authentication credentials to view the product pricing and review details. The basic need of the search engine is to enter into such type of web pages, after filling the username and password. Enabling the web crawler to do the same is the primary importance given in the chapter.

An already available PIW crawler is taken and the automatic form filling concept is attached and the results are analysed using several different search terms. The proposed algorithm will be analysing most of the Websites and will tend to pull out the related pages of the search query. The URL’s of the pages are identified and are added to the URL repository. The role of parser comes to live at this moment and it sees for any extended URL’s from the primary source of URL. The analyser will be co-working with the parser and will extract finite information from the web page. It scans each page for the search terms by analysing each and every sentence by breaking them and retrieves the essential information before showing the page. The
composer will then compose the details of the web pages in a database. This is how a typical hidden-pages web crawler works.

The analyser sees for the web page with more number of terms relevant to the search query. It has a counter, which will be initialised and the counter increments as soon as some of the words in the web page are found similar to that of the search term. The web pages of web site with more counter value are analysed and numbered and they are projected in page-wise as search results.

The traditional mode of working of the hidden web crawler is taken into account as a skeleton and several improvements are done after finding out its limitations and constraints from the literature survey. The crawler has to be given capabilities to find out hidden pages better than the existing hidden crawlers. For the same, certain extra module has to be added with the existing modules of hidden crawler. The added module is named as structure module capable of filling authentication forms before entering the web site, if needed. This module facilitates the crawler to enter a Secure Hyper Text Mark-up Page. Almost all the e-shopping sites has https as their transport protocol and this ability will lead to get information form, for this kind of web sites, which are not visible to ordinary web crawlers. The web crawler writes down the websites found in a particular domain in text files, enabling easy access. The list divides the good and bad pages, according to certain attributes of the webpage. The proposed web crawler will also be legible to crawl through Ajax and java script oriented pages.

4.3.1. Design Modules
The blueprint modules for the example of WebCrawler are as below. The architecture of WebCrawler is shown in Figure 4.1.

Figure 4.1: The Web Crawler architecture

4.3.1.1 Analyser

The primary component of the web crawler is the analysis, capable of looking into the web pages. The module is after the structure module, which is a search form used by the user to give search term and also his credentials. The analyser will scan each and every page and will keep the vital information in a text file. The files got as an outcome of the analyser phase is a text file consisting of all the website information and is stored in a log database, for further use for another search query.

4.3.1.2 Parser and Composer

The primary function of the parser in the proposed approach is to take the document and splitting it into index-able text segments, letting it to work with different file formats and natural languages. Mostly linguistic algorithms are applied
as parser. Here a traditional parser algorithm is used. The composer will compose the data of the web pages in the database.

### 4.3.1.3 Indexer

The function of indexer is dependent on parser and builds the indexes necessary to complement the search engine. This part decides the power of the search engine and determines the results for each of the search word. The proposed indexer has the capability to index terms and words from secure as well as open web. The difference between the normal web crawler and hidden page web crawler is shown here. The Google’s web indexer is supposed to be the best and uses ranking algorithm and changes the terms of the web pages as per their popularity and updating, making it a dynamic indexer.

The proposed web indexer has the capability to fill search words within the web pages and find out results, as well as concentrating on secure pages with HTTPS too.

### 4.3.2.4 Result Analyser

The result analyser explores the searched results and give the same in a GUI based structure for the developer to identify and come out with modifications. It is done by inputting a web page and all the HTML tags of it are considered to be output. As part of implementation an open source web crawler was identified.

There are several open source web crawlers available and some of them are Heritrix, a web documents open-source, extensible, web-scale, archival quality web
robot that is web-scalable and extensible. WebSPHINX is a Website-Specific Processors for HTML Information Extraction and is worked on java and gives an interactive development environment for creating web crawlers. JSpider, is a highly configurable and customizable Web Spider engine, which is developed in java. Web-Harvest is an Open Source Web Data Extraction tool written in Java and focuses mainly on HTML/XML based web sites. JoBo is an easy program to download total websites to local computer. For the implementation of our specific method which can make use of a different pattern of search to mine the searches via HTTPs, HTTP and FTP and also has the capability of getting information from preregistration–then only access sites, GNU Wget is downloaded and modified. GNU Wget is a freely distributed, GNU licensed software package for retrieving files via HTTP, HTTPS and FTP. It is a command based tool. This tool when examined showed visible improvement and some resultant pages from HTTPs and a form filled web site. Figure 4.2 shows the comparison of crawlers.

![Comparison of focused web crawlers](image.png)

**Figure 4.2:** Comparison of crawlers
4.4. Observations and Results

The results are taken for several keywords to find out the proposed Hidden web page web crawler’s difference from the traditional web search engine and a better search is found, which includes several secure and hidden pages input in the search results. The results proved that the HiGwget shows better results.

4.5. Chapter Summary

With the advent of search is increasing exponentially people and corporate rely on searches for multiple decision making, search engine with newer and wider results including pages that are rare and useful. The proposed Hidden page web crawler makes use of integration of several secure web pages as a part of indexing and comes out with a better result. In future the same can be applied for a mobile search and can be extended for ecommerce application.