**INTRODUCTION**

The deep (or hidden) web refers to the contents lie behind searchable web interfaces that cannot be indexed by searching engines. Based on extrapolations from a study done at University of California, Berkeley, it is estimated that the deep web contains approximately 91,850 terabytes and the surface web is only about 167 terabytes in 2003. More recent studies estimated that 1.9 petabytes were reached and 0.3 petabytes were consumed worldwide in 2007. An IDC report estimates that the total of all digital data created, replicated, and consumed will reach 6 petabytes in 2014. A significant portion of this huge amount of data is estimated to be stored as structured or relational data in web databases — deep web makes up about 96% of all the content on the Internet, which is 500-550 times larger than the surface web. These data contain a vast amount of valuable information and entities such as Info mine, Clusty, Books in Print may be interested in building an index of the deep web sources in a given domain (such as book). Because these entities cannot access the proprietary web indices of search engines (e.g., Google and Baidu), there is a need for an efficient crawler that is able to accurately and quickly explore the deep web databases.

It is challenging to locate the deep web databases, because they are not registered with any search engines, are usually sparsely distributed, and keep constantly changing. To address this problem, previous work has proposed two types of crawlers, generic crawlers and focused crawlers. Generic crawlers, fetch all searchable forms and cannot focus on a specific topic. Focused crawlers such as Form-Focused Crawler (FFC) and Adaptive Crawler for Hidden-web Entries (ACHE) can automatically search online databases on a specific topic. FFC is designed with link, page, and form classifiers for focused crawling of web forms, and is extended by ACHE with additional components for form filtering and adaptive link learner.

**PROBLEM STATEMENT**

The existing search engines, such as Google, Baidu, Bing etc., to find center pages of unvisited sites. This is possible because search engines rank webpages of a site and center pages tend to have high ranking values. Other Web service classification approaches include intelligent clustering method, support vector machine, automatic semantic annotation and ensemble learning method, QoS-aware service classification and recommendation method, etc. The above approaches use different mechanisms to classify services from different levels. Some approaches do not consider the classification from the semantic level, and it will influence the accuracy. Other semantic-based methods process the data from the qualitative point of view, but it lacks the support of mathematical theory.

* Web Crawlers are built to crawl different web pages based on user query.
* Crawling defines the basic motive of providing user with search results from web databases like Google.
* The main problem of web crawlers is how to rank web pages so that user can be provided with best results.
* The ranker algorithms indexes search results based on the algorithm defining the rank for each page.
* To provide user with proper results I propose a three phase search engine crawler which will rank user pages based on word counts and frequency generations. I then also compare the working of proposed algorithm with existing work.

**OBJECTIVES**

The main objectives of the study are listed below:

* To access data from Google using JSON API for Top Results.
* To Perform KNN on top results with pre-processing as well.
* To Perform HTML Extraction using JSOUP API
* To Perform Frequency Calculation using Naïve Bayes Algorithm
* To Performing Comparative Study of both algorithms.

**Applications**

* Real Time Prediction
* Multi Class Prediction
* Spam Filtering / Text Classification / Sentiment Analysis
* Smartphone Search Engines
* Recommendation System

**LITERATURE REVIEW**

**1. Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin “Smart Crawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces” in IEEE Transactions On Services Computing, Vol. 9, No. 4, July/August 2016.**

In this paper, author proposed, deep web grows at a very fast pace, there has been increased interest in techniques that help efficiently locate deep-web interfaces. However, due to the large volume of web resources and the dynamic nature of deep web, achieving wide coverage and high efficiency is a challenging issue. Here propose a two-stage framework, namely Smart Crawler, for efficient harvesting deep web interfaces. In the first stage, Smart Crawler performs site-based searching for center pages with the help of search engines, avoiding visiting a large number of pages. To achieve more accurate results for a focused crawl, Smart Crawler ranks websites to prioritize highly relevant ones for a given topic. In the second stage, Smart Crawler achieves fast in-site searching by excavating most relevant links with an adaptive link-ranking.

**2. Jianxiao Liu, Zonglin Tian, Panbiao Liu, Jiawei Jiang, “An Approach of Semantic Web Service Classification Based on Naive Bayes” in 2016 IEEE International Conference On Services Computing, September 2016**

In this paper, author proposed, how to classify and organize the semantic Web services to help users find the services to meet their needs quickly and accurately is a key issue to be solved in the era of service-oriented software engineering. This paper makes full use the characteristics of solid mathematical foundation and stable classification efficiency of naive bayes classification method. It proposes a semantic Web service classification method based on the theory of naive bayes. It elaborates the concrete process of how to use the three stages of Bayesian classification to classify the semantic Web services in the consideration of service interface and execution capacity.

**3. Bo Tang, Student Member, IEEE, Steven Kay, Fellow, IEEE, And Haibo He, Senior Member, IEEE “Toward Optimal Feature Selection in Naive Bayes for Text Categorization” In IEEE Transactions On Knowledge and Data Engineering, 9 Feb 2016**

In this paper, author proposed, automated feature selection is important for text categorization to reduce the feature size and to speed up the learning process of classifiers. In this paper, author present a novel and efficient feature selection framework based on the Information Theory, which aims to rank the features with their discriminative capacity for classification. Author first revisit two information measures: Kullback-Leibler divergence and Jeffreys divergence for binary hypothesis testing, and analyze their asymptotic properties relating to type I and type II errors of a Bayesian classifier.

**4. Amruta Pandit, Prof. Manisha Naoghare, “Efficiently Harvesting Deep Web Interface with Reranking and Clustering”, in International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 1, January 2016.**

In this paper, author proposed, the rapid growth of the deep web poses predefine scaling challenges for general purpose crawler and search engines. There are increasing numbers of data sources now become available on the web, but often their contents are only accessible through query interface. Here proposed a framework to deal with this problem, for harvesting deep web interface. Here Parsing process takes place. To achieve more accurate result crawler, calculate page rank and Binary vector of pages which is extracted from the crawler to achieve more accurate result for a focused crawler give most relevant links with a ranking. This experimental result on a set of representative domain show the agility and accuracy of this proposed crawler framework which efficiently retrieves web interface from large scale sites.

**5. Anand Kumar, Rahul Kumar, Sachin Nigle, Minal Shahakar, “Review on Extracting the Web Data through Deep Web Interfaces, Mechanism”, in International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 1, January 2016.**

In this paper, author proposed, web develops at a quick pace, there has been expanded enthusiasm for procedures that assistance effectively find profound web interfaces. Be that as it may, because of the expansive volume of web assets and the dynamic way of profound web, accomplishing wide scope and high proficiency is a testing issue. Author propose a two-phase system, to be specific Smart Crawler, for productive gathering profound web interfaces. In the primary stage, Smart Crawler performs site-based hunting down focus pages with the assistance of web crawlers, abstaining from going to a substantial number of pages.

**6. Sayali D. Jadhav, H. P. Channe “Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques” in International Journal of Science and Research, Volume 5 Issue 1, January 2016.**

In this paper, author proposed, Classification is a data mining technique used to predict group membership for data instances within a given dataset. It is used for classifying data into different classes by considering some constrains. The problem of data classification has many applications in various fields of data mining. This is because the problem aims at learning the relationship between a set of feature variables and a target variable of interest. Classification is considered as an example of supervised learning as training data associated with class labels is given as input. This paper focuses on study of various classification techniques, their advantages and disadvantages.

**7. Akshaya Kubba, “Web Crawlers for Semantic Web” in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 5, May 2015. [7]**

In this paper, author proposed, Web mining is an important concept of data mining that works on both structured and unstructured data. Search engine initiates a search by starting a crawler to search the World Wide Web (WWW) for documents. Web crawler works in a ordered way to mine the data from the huge repository. The data on which the crawlers were working was written in HTML tags, that data lags the meaning. It was a technique of text mapping. Semantic web is not a normal text written in HTML tags that are mapped to the search result, these are written in Resource description language. The Meta tags associated with the text are extracted and the meaning of content is find for the updated information and give us the efficient result in no time.

**8. Monika Bhide, M. A. Shaikh, Amruta Patil, Sunita Kerure, “Extracting the Web Data Through Deep Web Interfaces” in INCIEST-2015.**

In this paper, author proposed, the web stores huge amount of data on different topics. The users accessing web data vastly in now days. The main goal of this paper is to locating deep web interfaces. To locating deep web interfaces uses techniques and methods. This paper is focus on accessing relevant web data and represents significant algorithm i.e. adaptive learning algorithm, reverse searching and classifier. The locating deep web interfaces system works in two stages. In the first stage apply reverse search engine algorithm and classifies the sites and the second stage ranking mechanism use to rank the relevant sites and display different ranking pages.

**9. Raju Balakrishnan, Subbarao Kambhampati, “SourceRank: Relevance and Trust Assessment for Deep Web Sources Based on Inter-Source Agreement” in WWW 2011, March 28–April 1, 2011.**

In this paper, author proposed, selecting the most relevant web databases for answering a given query. The existing database selection methods (both text and relational) assess the source quality based on the query-similarity-based relevance assessment. When applied to the deep web these methods have two deficiencies. First is that the methods are agnostic to the correctness (trustworthiness) of the sources. Secondly, the query based relevance does not consider the importance of the results. These two considerations are essential for the open collections like the deep web. Since a number of sources provide answers to any query, author conjuncture that the agreements between these answers are likely to be helpful in assessing the importance and the trustworthiness of the sources.

**10. Luciano Barbosa, Juliana Freire “An Adaptive Crawler for Locating Hidden Web Entry Points” in WWW 2007.**

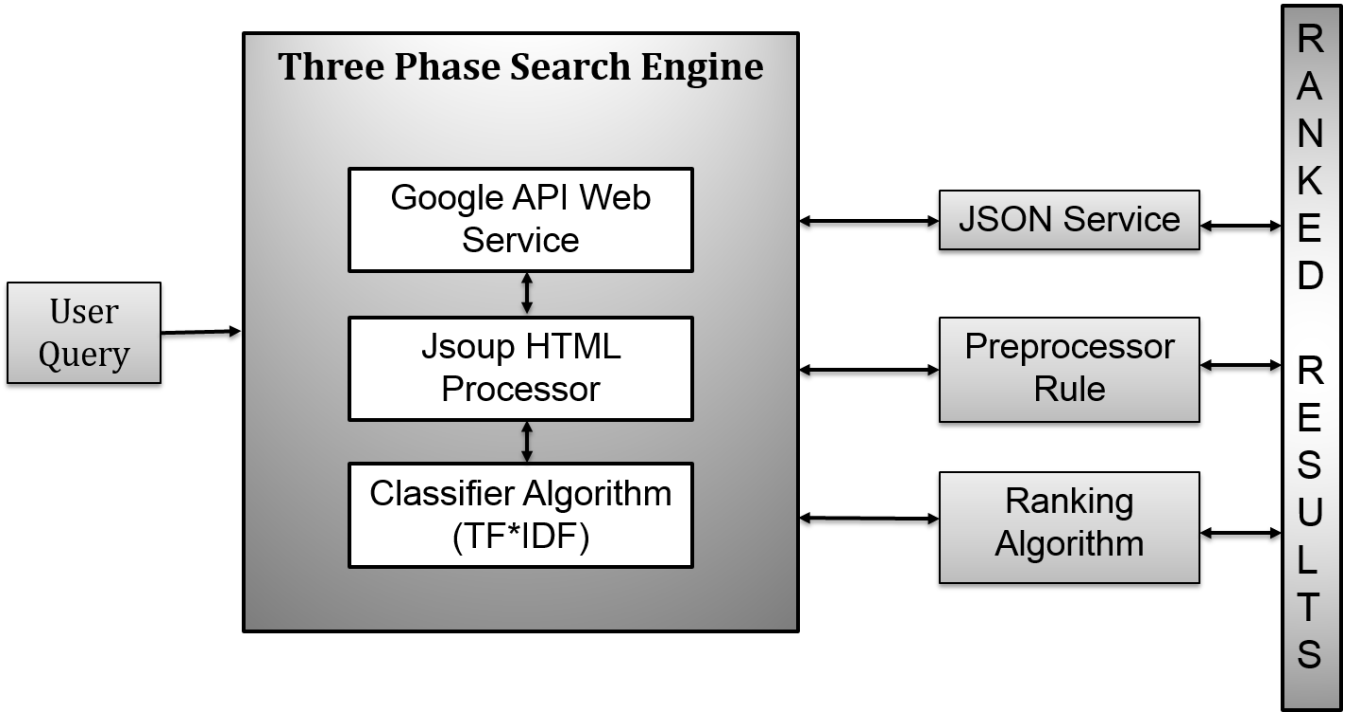
In this paper, author proposed, describe new adaptive crawling strategies to efficiently locate the entry points to Hidden-Web sources. The fact that Hidden-Web sources are very sparsely distributed makes the problem of locating them especially challenging. Author deal with this problem by using the contents of pages to focus the crawl on a topic; by prioritizing promising links within the topic; and by also following links that may not lead to immediate benefit. Author propose a new framework whereby crawlers automatically learn patterns of promising links and adapt their focus as the crawl progresses, thus greatly reducing the amount of required manual setup and tuning.

**Drawbacks of Existing System: -**

1. Tree Structure prone to sampling.
2. Tree Splitting is locally greedy.
3. Optimal decision tree is NP-complete problem.
4. Information is not exact
5. Time Consuming, valuable resources used

**PROPOSED SYSTEM**

The proposed work is planned to be carried out in the following manner

****

**Fig 1.1: Proposed System Architecture**

To efficiently and effectively discover deep web data sources, Crawler is designed with a three-stage architecture, site locating and in-site exploring, as shown in above Figure. The first site locating stage finds the most relevant site for a given topic, the second in-site exploring stage uncovers searchable forms from the site and then the third stage apply naïve base classification ranked the result.

Specifically, the site locating stage starts with a seed set of sites in a site database. Seeds sites are candidate sites given for Crawler to start crawling, which begins by following URLs from chosen seed sites to explore other pages and other domains. When the number of unvisited URLs in the database is less than a threshold during the crawling process, Crawler performs “reverse searching” of known deep web sites for center pages (highly ranked pages that have many links to other domains) and feeds these pages back to the site database. Site Frontier fetches homepage URLs from the site database, which are ranked by Site Ranker to prioritize highly relevant sites.

**Data Flow Diagram**

**Level 0:**

Result

Query

**User**

**User**

**Level 1:**

Query

Result

**User**

**User**

**Counter**

**Non Html Data**

**Non Html Data**

**Ranker**

The Site Ranker is improved during crawling by an Adaptive Site Learner, which adaptively learns from features of deep-web sites (web sites containing one or more searchable forms) found. To achieve more accurate results for a focused crawl, Site Classifier categorizes URLs into relevant or irrelevant for a given topic according to the homepage content.

**Level 2:**

Query

Result

**User**

**User**

Result

Result

**User**

**User**

**Software Requirements**

* JDK for Java Development
* JSON API

**1.** **Software Utilization**

* JDK 6.0 or above

Java is a general purpose computer programming language that is concurrent, class based, object oriented and specifically designed to have as few implementation dependencies as possible.

* JSON API

JSON or JavaScript Object Notation is a lightweight text-based open standard designed for human-readable data interchange. The JSON format was originally specified by Douglas Crockford, and is described in RFC 4627. The official Internet media type for JSON is application/json. The JSON filename extension is .json. This tutorial will help you understand JSON and its use within various programming languages such as PHP, PERL, Python, Ruby, Java, etc.

* It is used while writing JavaScript based applications that includes browser extensions and websites.
* JSON format is used for serializing and transmitting structured data over network connection.
* It is primarily used to transmit data between a server and web applications.
* Web services and APIs use JSON format to provide public data.
* It can be used with modern programming languages.

**Hardware Requirements**

* Processor: 1 gigahertz (GHZ) or Faster
* RAM: 1gigabyte (GB) (32-bit) or 2 GB (64-bit)
* Free Hard Disk Space: 16GB
* Graphics card: DirectX 8 or Latest
* Internet Access

**Advantages:**

* **Feature learning**: A randomly initialized system and trained on some datasets will eventually learn good feature representations for a given task. Classical approaches involved handcrafting features by an expert human. This took several years of painstakingly fine tuning several parameters to get it right. Nowadays machine learning is used to discover relevant features in otherwise disordered datasets. Such features can be useful for things such as face detection, face recognition, speech recognition or image classification. Deep learning in particular aims to build higher-level abstract feature representation of data layer by layer. These features can be very powerful in speech and image recognition.
* **Parameter optimization**: This is similar to feature learning as a group of tuneable parameters can be visualized as a feature. Machine learning mostly employs a gradient based method of optimizing a large array of parameters. Again such parameters may be large in number for example; a deep neural architecture can have billions of tuneable parameters. These parameters when well set can result in a system working properly.

**Disadvantages:**

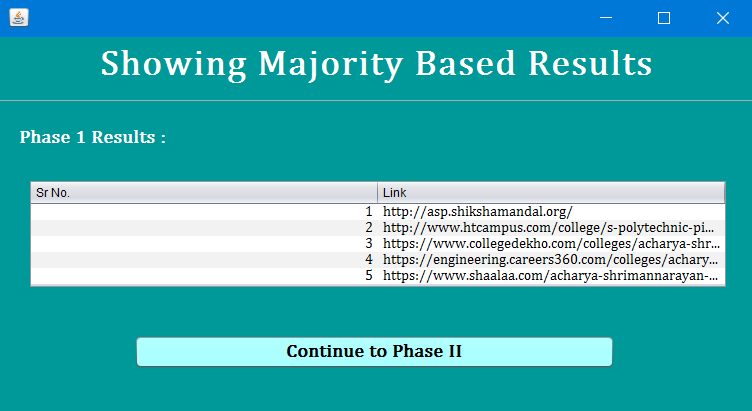
* **Works with continuous loss functions**: Non-differentiable discontinuous loss functions are hard to optimize using machine learning techniques. There are several reasons why discontinuous loss functions are important. In cases such as sparse representations, discontinuous loss functions can help to find such sparse representations. In many cases such non-differentiable loss functions are approximated by smooth loss functions without much loss in sparsity.
* **Limited**: It is not a guarantee that machine learning algorithms will always work in every case imaginable. Sometimes or most of the times machine learning will fail, thus it requires some understanding of the problem at hand in order to apply the right machine learning algorithm.
* **Large data requirements**: Some machine learning algorithms need a lot of training data such as deep learning algorithms. It might be cumbersome to work with or collect such large amounts of data. Fortunately, there are a lot of training data for image recognition purposes.

**SNAPSHOTS**

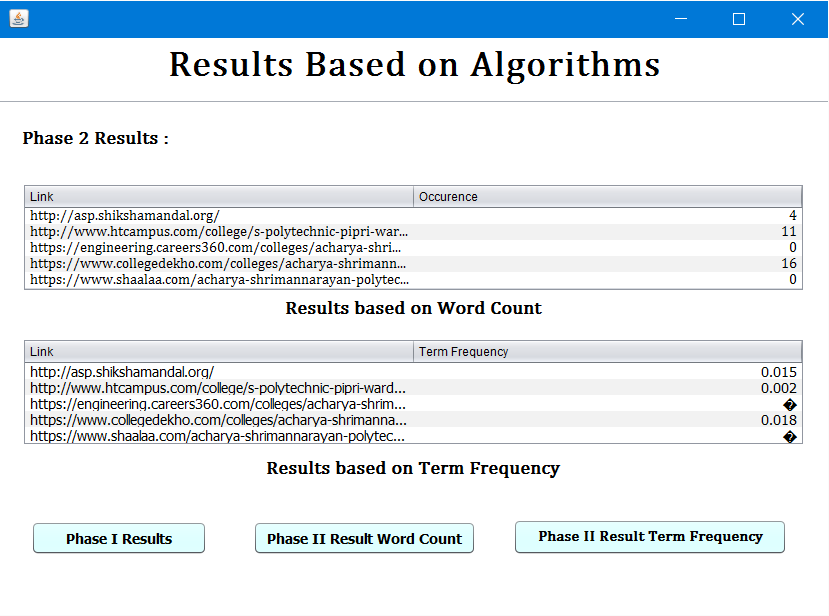
**Query Search: -**



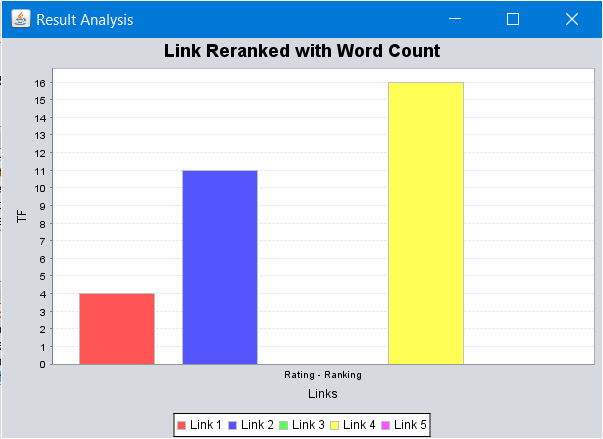
**Phase I: - Fetch Results from Google.**



**Phase II: - Ranked Results with Word Counts.**

****

**Phase III: - Graph Generation**



**Result Analysis**

|  |  |  |
| --- | --- | --- |
| **On Basis** | **Existing System** | **Proposed System** |
| ***Techniques used*** | Branching tree | Classifier technique |
| ***Speed*** | Low | High |
| ***Time*** | Consumes more | Less Time Consume |
| ***Information*** | Not Exact | Exact Information |
| ***Resources used*** | More | Less |

**Conclusion**

Machine learning approaches applied in systematic reviews of complex research fields such as quality improvement may assist in the title and abstract inclusion screening process. Machine learning approaches are of particular interest considering steadily increasing search outputs and accessibility of the existing evidence is a particular challenge of the research field quality improvement. Increased reviewer agreement appeared to be associated with improved predictive performance.

Machine learning is expected to bring about numerous changes to the best SEO practices. The motive behind Google’s use of machine learning is to solve complex problems that come with identifying and reducing spam. This will eventually help bring more quality results to users. But it makes it crucial for you to re-analyse your SEO efforts. Adapting best SEO practices according to Google’s standards can help you snag the top position in search results.

**References**

* Click through-based translation models for web search: from word models to phrase models. In CIKM.
* Russ Salakhutdinov and Hinton, Semantic hashing. in Proc. SIGIR Workshop Information Retrieval and Applications of Graphical Models.
* [Stanford Machine Learning](https://www.coursera.org/course/ml):  [Stanford CS229 course](http://cs229.stanford.edu/).
* [Programming Collective Intelligence: Building Smart Web 2.0 Applications](http://www.amazon.com/dp/0596529325?tag=inspiredalgor-20).
* [Data Mining: Practical Machine Learning Tools and Techniques, Third Edition](http://www.amazon.com/dp/0123748569?tag=inspiredalgor-20).
* https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/
* Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin “Smart Crawler: A Two-stage Crawler for Efficiently Harvesting Deep-Web Interfaces” in IEEE Transactions On Services Computing, Vol. 9, No. 4, July/August 2016.
* Jianxiao Liu, Zonglin Tian, Panbiao Liu, Jiawei Jiang, “An Approach of Semantic Web Service Classification Based on Naive Bayes” in 2016 IEEE International Conference On Services Computing, September 2016
* Bo Tang, Student Member, IEEE, Steven Kay, Fellow, IEEE, And Haibo He, Senior Member, IEEE “Toward Optimal Feature Selection In Naive Bayes For Text Categorization” In IEEE Transactions On Knowledge And Data Engineering, 9 Feb 2016
* Amruta Pandit , Prof. Manisha Naoghare, “Efficiently Harvesting Deep Web Interface with Reranking and Clustering”, in International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 1, January 2016.
* Anand Kumar, Rahul Kumar, Sachin Nigle, Minal Shahakar, “Review on Extracting the Web Data through Deep Web Interfaces, Mechanism”, in International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 1, January 2016
* Sayali D. Jadhav, H. P. Channe “Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques” in International Journal of Science and Research, Volume 5 Issue 1, January 2016.
* Akshaya Kubba, “Web Crawlers for Semantic Web” in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 5, May 2015.
* Monika Bhide, M. A. Shaikh, Amruta Patil, Sunita Kerure,“Extracting the Web Data Through Deep Web Interfaces” in INCIEST-2015.
* Raju Balakrishnan, Subbarao Kambhampati, “SourceRank: Relevance and Trust Assessment for Deep Web Sources Based on Inter-Source Agreement” in WWW 2011, March 28–April 1, 2011.
* Luciano Barbosa, Juliana Freire “An Adaptive Crawler for Locating Hidden Web Entry Points” in WWW 2007.