# Our Proposed Title: - The Dual Stage well-dressed Crawler using adaptive learning algorithm on web interfaces

# **Abstract**

As 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. We 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. To eliminate bias on visiting some highly relevant links in hidden web directories, we design a link tree data structure to achieve wider coverage for a website. Our experimental results on a set of representative domains show the agility and accuracy of our proposed crawler framework, which efficiently retrieves deep-web interfaces from large-scale sites and achieves higher harvest rates than other crawlers.

**EXISTING SYSTEM**

Existing hidden web directories usually have low coverage for relevant online databases, which limits their ability in satisfying data access needs. Focused crawler is developed to visit links to pages of interest and avoid links to off-topic regions. A recent study shows that the harvest rate of deep web is low; they just search in Search Index.

**Disadvantages:**

1. Consuming large amount of data’s.

2. Time wasting while crawl in the web.

.

**PROPOSED SYSTEM:**

In this paper, we propose an effective deep web harvesting framework, namely Smart Crawler, for achieving both wide coverage and high efficiency for a focused crawler. Based on the observation that deep websites usually contain a few searchable forms and most of them are within a depth, our crawler is divided into two stages: site locating and in-site exploring. The site locating stage helps achieve wide coverage of sites for a focused crawler, and the in-site exploring stage can efficiently perform searches for web forms within a site. Our main contributions are:

* We propose a novel two-stage framework to address the problem of searching for hidden-web resources.
* We propose an adaptive learning algorithm that performs online feature selection and uses these features to automatically construct link rankers

**MODULE DESCRIPTION:**

# Number of Modules:

After careful analysis the system has been identified to have the following modules:

1. **two-stage crawler.**

**2. Adaptive learning**

**3. Admin**

**4. User**

**1 Two-stage crawler.**

We propose a two-stage framework, namely SmartCrawler , for efficient harvesting deep web interfaces. In the first stage, SmartCrawler 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, SmartCrawler ranks websites to prioritize highly relevant ones for a given topic. In the second stage, SmartCrawler achieves fast in-site searching by excavating most relevant links with an adaptive link-ranking. To eliminate bias on visiting some highly relevant links in hidden web directories, we design a link tree data structure to achieve wider coverage for a website.

**2. Adaptive learning**

Adaptive learning algorithm that performs online feature selection and uses these features to automatically construct link rankers. In the site locating stage, high relevant sites are prioritized and the crawling is focused on atopic using the contents of the root page of sites, achieving more accurate results. During in site exploring stage, relevant links are prioritized for fast in-site searching. We have performed an extensive performance evaluation of Smart Crawler over real web data in representative domains

**3. Admin**

In our proposed architecture admin is a data owner, and perform site ranker and adaptive link ranker. He search site links from the google search engine according to some topics, and choose links for smart crawling. He maintains the site database.

**4. User**

In our proposed architecture user is end user of our application, and data user. Whenever he wants the data he can search from our application, data retrieve from the google search engine, but when sites are matched with our seed sites then smart crawl results he can get according to ranking.

**System Configuration:**

**HARDWARE REQUIREMENTS**:

Hardware - Pentium

Speed - 1.1 GHz

RAM - 1GB

Hard Disk - 20 GB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

**SOFTWARE REQUIREMENTS**:

Operating System : Windows Family

Technology : Java and J2EE

Web Technologies : Html, JavaScript, CSS

Web Server : Tomcat

Database : My SQL

Java Version : J2SDK1.5

**Implemented by**

Development team : Cloud Technologies

Website : <http://www.cloudstechnologies.in/>

Contact : 8121953811, 040-65511811