# SVM based Mobile Application for Deep Web Classification and Extraction

**S. Suneetha[1], Dr. M. Usha Rani[2]**
**Research Scholar[1], Professor & BOS Chair Person[2],**
**Department of Computer Science, SPMVV, Tirupati.**
**suneethanaresh@yahoo.com s.suneethanaresh@gmail.com[1], musha_rohan@yahoo.com[2]**

***Abstract*—By virtue of perpetual materialization of multitudinous internet and broadcasting technologies, Smartphone based Internet usage has been rapidly hoisting from 2012. In today's Smartphone driven world, the need for accessing Deep Web through mobile is inevitable. In this paper, we proposed a mobile application for Android mobiles to extract and classify contents from Deep Web. This application works in two steps: Once the app has been launched, the app will acquire a connection establishment from TOR. After the Relay has been connected successfully, it will obtain the search query and will return the results. Then, using Support Vector Machine (SVM) indexing algorithm, the results will be indexed and displayed.**

***Index Terms*—Deep Web, TOR (The Onion Router), SVM (Support Vector Machine) Indexing**

## I. Introduction

In this contemporary World of Internet, people wish to search and share abundant information. 'Deep Web' refers to the vast portions of the Internet that are hidden and not accessible via regular search engines and Web browsers. The size of the Deep Web is about 500 times the size of the Surface Web that we know.

The Deep Web is rich in information and is described as a 'Pool of Information' by recent researchers. Searching the Deep Web provides access to the sites that are not indexed by the standard search engines such as, Google, Yahoo, and Bing. A Deep Web search engine's chief advantage is the depth and thoroughness of its results. They are more efficient, and can retrieve high quality as well as more relevant content**.** A standard Web search considers only the site's introduction and supplied keywords, yielding results, many of the returned links being commercial sites or repetitive material. Deep Web searches analyze each page's entire content; ensuring returned results have a higher relevance to the desired search string. Professionals especially research scholars have several advantages in surfing through the Deep Web. [10] [11]

Deep Web URLs have **.onion** as the extension. Domain names on the Deep Web are so random and meaningless. These links have a combination of random letters and numbers. Eg:- http://xmh57jrzrnw6insl.onion/ Such [websitename].onion domains of Deep Web can be accessed with special software called 'The Onion Router' referred to as TOR.

Tor browser is free open source software that makes use of onion routing process. It is easy to install, setup and use. It can be used on all major operating systems such as, Windows, Mac OS X, or Linux and is portable. It can work on all type of internet protocols like, HTTP, Https, FTP and gopher. Tor browser masks the original IP address, provides security and allows surfing of Internet anonymously. Tor browser is a gateway into the Deep Web. Deep Web Search engines such as, SurfWax, IceRocket, TechDeepWeb communicate with the onion service via Tor and relays, resolve the onion links and then deliver the final search result or output to the browser on the Surface Web.

All the Deep Websites can be accessed through portal sites, in which users need privileged access for accessing the content. Due to the access privilege issues, search engines can't crawl and index the vast amount of data that is not hyperlinked or accessed via public DNS services.

Most of the Deep Websites are either un-indexed or encrypted in order to maintain the anonymity. Moreover, it is dynamically expanding at an exponential rate. Exploring the Deep web is and it's been a complicated issue because no ruling authority regulates the Deep Web for accessing the un-indexed Web data. Several issues have to be dealt with, in order to access the Deep Web contents. 'Indexing the Deep Web' is the process of collecting data from the hidden sources by issuing a query through various techniques such as, APIs, HTML forums, IP based identification. It requires appropriate queries for retrieving data from the data sources in less time. [1] [6]

The purpose of indexing is to optimize the speed and performance in finding relevant information for a search query. SVM (Support Vector Machine) is widely used in various applications including search engines or relevance feedback systems for ranking.

In this paper, we proposed a mobile application for accessing the Deep Web to work on Android mobiles. This paper is organized as follows. We first discuss about the related works in the second section. Section III provides the details of our proposed mobile application. The fourth section presents empirical study. Section V concludes our study with the future works.

## II. Review of Related Literature

### *A. Crawling Deep Web Using a New Set Covering Algorithm*

Yan Wang et al. [13], proposed a set covering algorithm for crawling Deep Websites. The proposed algorithm consists of a data management technique with covering database for using the query DB. Unlike the existing deep web crawling methods that made use of greedy set covering algorithms, weights are added to the greedy strategy and the weighed version exhibited consistent and increased performance when compared to the un-weighted classification techniques for indexed documents. The distribution of indexed documents has also been improved as compared to the traditional algorithms.

### *B. Towards XML Schema Extraction from Deep Web*

Yasser Saissi et al. [14], proposed an approach to extract relational schema describing the hidden content of the analyzed Deep Web source. The XML schema extracts the necessary information needed to integrate the associated deep web source in the mediation system. The XML schema is extracted by performing static and dynamic analysis of the HTML forms and HTML tables extracted from the selected Deep Web source.

### *C. Deep Web performance enhance on search engine*

Deepak Kumar et al. [5], proposed a framework for enhancing the Deep Web performance. The framework comprised of uploading non-hypertext content with the implementation of sitemap, metadata and text for improving the indexed results on search engine.

### *D. Smart Crawler : a two stage crawler efficiently harvesting Deep Web routing*

Feng Zhao et al. [7], discussed about two stage crawling method for extracting Deep Web interfaces. It consists of two parts: Site Location and In-Site Exploration. Site location part is used to locate and rank the website in Deep Web. In second step, the located sites are linked and the databases will be indexed and ranked based on query the database.

### *E. Dynamic Integration for Deep Web Search Results*

Bo Liu et al. [2], discussed about a deep web interface query method that handles feature extraction. At first, the data will be collected from HTML forms. The data KNN classifier will be used to classify the indexed forms for further processing. In this method, a heuristic filter is used for applying heuristic rule and identifying the forms for classification. The KNN classifier algorithm has shown improved results than the naïve Bayesian model and C5 model.

## III. PROPOSED WORK

In this paper, we proposed a Mobile Application to access the Deep Web pages for Android mobiles. Only few mobile apps are available for accessing Deep Web, each with their own limitations. The major issue faced by the mobile users is that the working of the existing mobile apps is based on mobile architecture and OS restrictions.

The proposed mobile application is used to access and rank the Deep Web pages. The application will be working on the following steps: Firstly, the app needs the user needs to be logged in and the connection for TOR has to be successful. Once the connection relay has been successful, the user can type the search query. Once the query has been submitted, the app will get the results from TOR. If the search query is lengthy then the keyword will be split into multiple words. Afterwards, the results are retrieved from the TOR interface. Subsequently, the results will be classified and ranked by using Support Vector Machine (SVM) indexing algorithm. [7] [9] [12]

### *A. Framework Description*

Initially, the authorized user has to log in. Registered users only can use this app since it is in beta version.

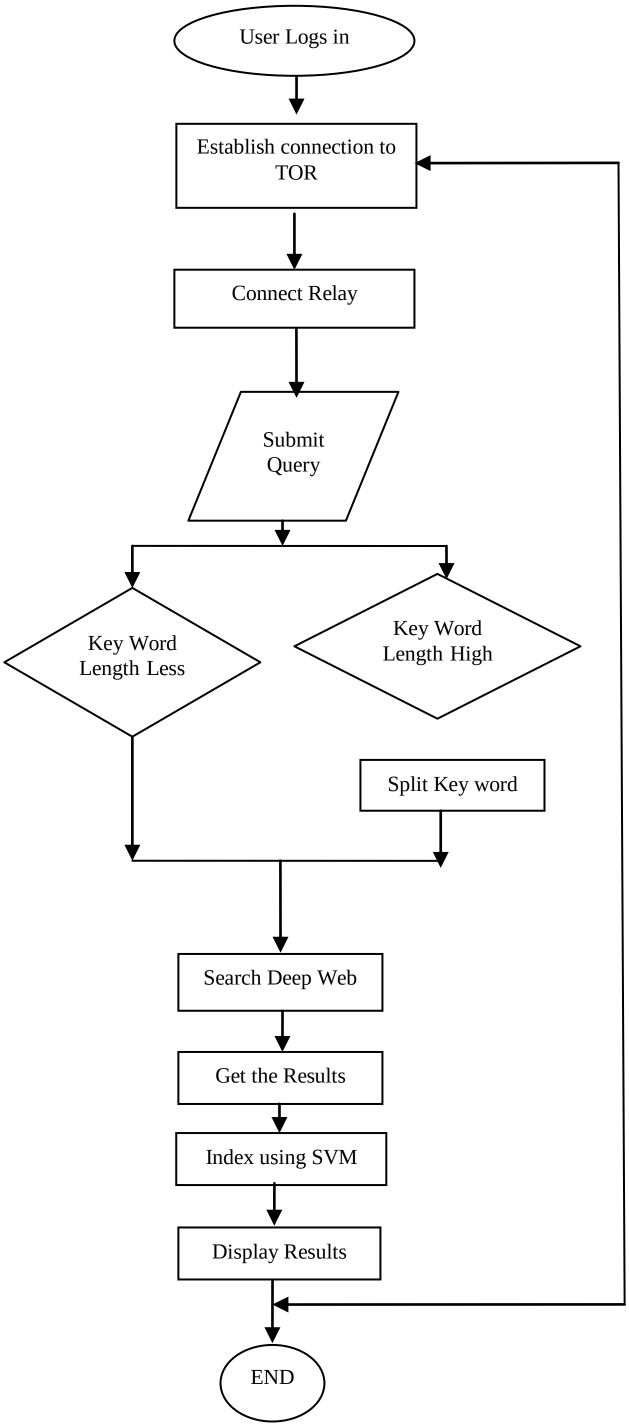


Figure 1: Architecture Flow Diagram for Mobile App

*TOR Interface:*

Once the user has been logged in, a connection to the TOR (The Onion Router) interface will be made. TOR is the most trusted and open source search engine for accessing deep web. The connection will be made once the mobile network has been connected to the TOR relay.

The bridge connections will be made after the relay gets connected.

*Search Indexing:*

Once the query has been submitted from the client side, the server will check the keyword length. If it exceeds the limit of 'n', keywords will be split and counted as multiple. After keyword segmentation, the search results from the TOR will be directed to the server. On the server side, the indexing algorithm will be implemented and ranking will be re arranged.

The mobile application will work on both the server and the client side. In the server side, the indexing algorithm will be implemented and comparative results will be saved on the server side.

The client is able to see the modified and original search results from the Deep Web search engines. With the comparative performance, the results will be ranked.

## IV. Empirical Study

SVM (Support Vector Machine) is a popular and actively researched methodology used for classification, regression, and ranking. SVM indexing algorithm is used here to classify and rank the search results obtained from the Deep Web using Tor. SVM gained incrementing popularity, as a binary relegation algorithm, because it has shown outstanding performance in many domains of relegation quandaries. The relegation function form of SVM is homogeneous to Artificial Neural Networks (ANN). Its output is a linear cumulation of some mid-layer nodes. Each mid-layer node corresponds to the inner product of input sample and a support vector. [3] [4] [8]

All the search results will be taken as,

$x^1, x^2, \ldots, x^d$

The input vector will be,

$x = (x^1, x^2, \ldots, x^d)$ and

the output is,

$$y = sgn\left\{\left(\sum_{i=1}^{n} \alpha_i^* y_i K(x_1 . x)\right) + b^*\right\}$$

The Kernel functions $K(x_i, x)$ will have the following forms:

Polynomial Kernel Function

$$K(x_i, x) = c[(x_i . x) + b]^n$$

Radial Basis Function Kernel Function, RBF

$$K(x_i, x) = \exp\{-c\frac{|x_i - x|^2}{\delta^2}\}$$

The figure of SVM tree is given below:

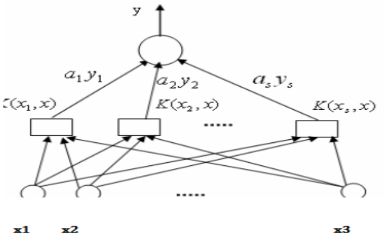
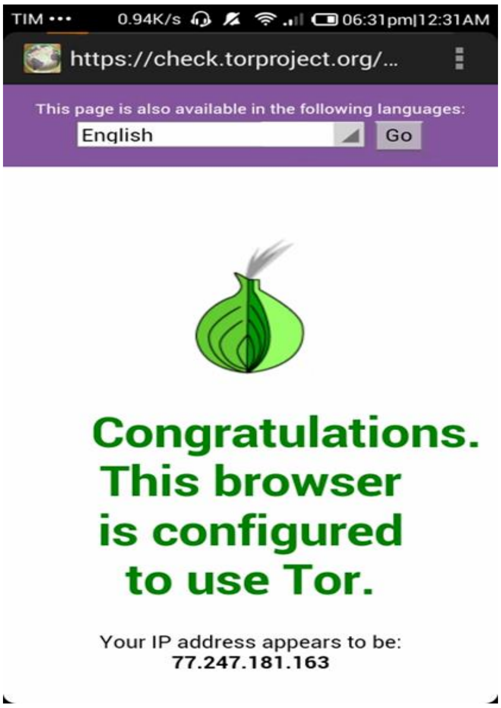


Figure 2: SVM Tree

In our experiments, we have chosen the SVM with the polynomial kernel function and RBF kernel function as the base classifier. In the polynomial kernel function, it is assumed that the exponent, *n* = 20; and in RBF kernel function constant, δ = 2. For SVM implementation, we used LIBSVM.

We have successfully implemented a Mobile App based Indexing for Deep Web in Android.


TIM ••• 0.94K/s 06:31pm|12:31AM
https://check.torproject.org/...
English
Go

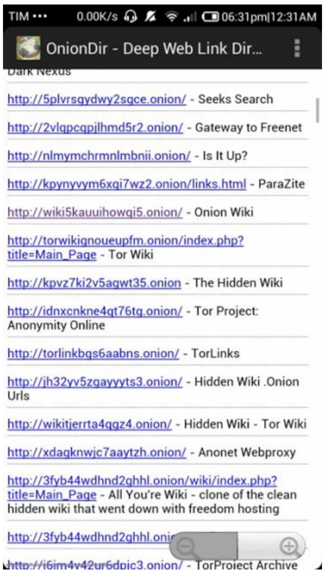
This page is also available in the following languages:

Congratulations. This browser is configured to use Tor.

Your IP address appears to be: 77.247.181.163

Figure 3: Mobile App Screenshot

*Six* different Web page representations are tested. The results from Support Vector Machine classifier based on polynomial kernel function are furnished below:


TIM ••• 0.00K/s 06:31pm|12:31AM
OnionDir - Deep Web Link Dir...

Dark Nexus

http://5plvrsgydwy2sgce.onion/ - Seeks Search

http://2vlqpcqpjlhmd5r2.onion/ - Gateway to Freenet

http://nlmymchrmnlmbnii.onion/ - Is It Up?

http://kpynyvym6xqi7wz2.onion/links.html - ParaZite

http://wiki5kauuihowqi5.onion/ - Onion Wiki

http://torwikignoueupfm.onion/index.php?title=Main_Page - Tor Wiki

http://kpvz7ki2v5agwt35.onion - The Hidden Wiki

http://idnxcnkne4qt76tg.onion/ - Tor Project: Anonymity Online

http://torlinkbgs6aabns.onion/ - TorLinks

http://jh32yv5zgayyyts3.onion/ - Hidden Wiki .Onion Urls

http://wikitjerrta4qgz4.onion/ - Hidden Wiki - Tor Wiki

http://xdagknwjc7aaytzh.onion/ - Anonet Webproxy

http://3fyb44wdhnd2ghhl.onion/wiki/index.php?title=Main_Page - All You're Wiki - clone of the clean hidden wiki that went down with freedom hosting

http://3fyb44wdhnd2ghhl.oni

http://i6im4v42ur6dpic3.onion/ - TorProject Archive

Figure 4: Indexed Results

The application is available in beta version. So, different indexing algorithms can be used in the mobile application, instead of SVM indexing.

## Conclusion and Future Work

The Deep Web is the largest expanding division of fresh unstructured and unindexed information on the Internet and is considered to be the next secured level of accessing Internet. When compared to standard search engines, Deep Web search engines are slow. Apart from the desired information, Deep Web searches may also return sensitive personal information creating ethical dilemmas and leaving individuals susceptible to fraud and identity theft. Searching the Deep Web also requires a more precise search string. So, Deep Web searches should be reserved only for serious, painstaking research, but not for simple and basic Web surfing.

Tor is software that allows the users to browse the Deep Web anonymously. Though Tor is as open source software that is freely available, it is not fool proof. While using Tor, the bandwidth speeds are reduced. Tor browser may use apps such as, Flash which are not protected and so cannot provide anonymity. Even, it doesn't encrypt the data. A powerful VPN (Virtual Private Network) is required on the top of Tor, as VPN + Tor Browser both can provide double layer security

'Indexing' is the process of collecting, parsing, and storing data to facilitate fast and accurate information retrieval. Without an index, the search engine has to scan every document in the corpus that requires considerable time and computing power. SVM indexing algorithm is used here to rank the results. As future work, we are planning to test the application with different indexing algorithms other than SVM. The algorithm that yields better results according to the requirement can be used in the place of SVM.

In future, we are also planning to carry on researches on various aspects of Deep Web page extraction that overcome the existing limitations and improve the performance further. The objective of the research work in future is to develop a mobile application that works irrespective of mobile architecture and OS restrictions with a customized indexing algorithm.

## REFERENCES

[1] Bo Liu, "Inconsistent Data Repairs in Database Integrations," Proceedings of 9th International Conference on Natural Computation, 2013, pp.1135-1139.

[2] Bo Liu, Huimin Zhang, Liuyan Liao, "Dynamic Integration for Deep Web Search Results", 3rd International Conference on Information Science and Control Engineering, 2016.

[3] Benjamin X. Wang and Nathalie Japkowicz, "Boosting Support Vector Machine for Imbalanced Data Sets", Springer, 2016.

[4] C. Cortes and V. Vapnik, "Support Vector Networks", Machine Learning, Vol. 30, no. 3, pp. 273-297, 1995.

[5] Deepak Kumar and Rajesh Mishra, "Deep Web performance enhance on search engine", In International Conference on Soft Computing Techniques and Implementations (ICSCTI), 2015, Publisher: IEEE.

[6] E.J. Glover, G.W. Flake, and S. Lawrence, "Improving Category Specific Web Search by Learning Query Modifications", Proc. 2001 Symp. Applications and the Internet (SAINT '01), pp. 23-31, 2001.

[7] Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin, "Smart Crawler: a two stage crawler efficiently harvesting Deep Web", IEEE transaction on Service Computing, 2016.

[8] Luciano Barbosa and Juliana Freire, "Combining classifiers to identify online databases", In Proceedings of the 16th International Conference on World Wide Web, pages 431–440. ACM, 2007.

[9] M. Marin-Castro Heidy, J. Sosa-Sosa Victor, L.A. Ivan. "A Tree-Based WQI Modeling Approach for Integrating Web Databases," Proceedings of 17th International Conference on Information Fusion (FUSION), Publisher: IEEE, 2014, pp. 1-8.

[10] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard P Fahringer, Peter Reutemann, and Ian H. Witten. The Weka Data Mining software: an update. SIGKDD Explorations Newsletter, 11(1):10– 18, November 2009.

[11] Sriram Raghavan and Hector Garcia-Molina, "Crawling the Hidden Web", In Proceedings of the 27th International Conference on Very Large Data Bases, pages 129–138, 2000.

[12] Y. Katsis, Y. Papakonstantinou, "View-based Data Integration," Encyclopedia of Database Systems, Publisher: Springer US, 2009, pp. 3332-3339.

[13] Yan Wang, Jianguo Lu, and Jessica Chen," Crawling Deep Web Using a New Set Covering Algorithm", IEEE transaction on Service Computing, 2013.

[14] Yasser Saissi Ahmed Zellou Ali Idri," Towards XML Schema Extraction from Deep Web", IEEE transaction on distributed Computing, 2016.

**AUTHORS**

Ms. S SUNEETHA received her Bachelor's Degree in Science and in Education, Master's Degree in Computer Applications (MCA) from SVU, Tirupati and M.Phil. in Computer Science from SPMVV, Tirupati. Currently, she is pursuing her Ph.D. in SPMVV, Tirupati. She is a life time member of ISTE. Her areas of interest are Data Mining, Software Engineering, Big Data and Cloud Computing. She has 23 papers in National/ International Conferences/ Journals to her credit. She also attended several workshops in varied areas. She served Narayana Engineering College, Nellore, Andhra Pradesh as Sr. Asst. Professor, heading the departments of IT and MCA.

Dr. M Usha Rani is Professor & BOS Chair Person in the Department of Computer Science, Sri Padmavati Mahila Viswa Vidyalayam (Women's' University), Tirupati. She did her Ph.D. in Computer Science in the area of Artificial Intelligence & Expert Systems.

She is in teaching since 1992. She presented many papers at National and International Conferences and published articles in National & International Journals. She has also written 4 books like Data Mining - Applications: Opportunities and Challenges, Superficial Overview of Data Mining Tools, Data Warehousing & Data Mining and Intelligent Systems & Communications. She is guiding M.Phil. and Ph.D. in areas like Data Warehousing and Data Mining, Computer Networks and Network Security and Cloud Computing.