International Journal of Web Portals
Volume 11 • Issue 2 • July-December 2019

Personalized Content Extraction and Text Classification Using Effective Web Scraping Techniques

Karthikeyan T., CSE, Sri Balaji Chockalingam Engineering College, Arni, India

Karthik Sekaran, Vellore Institute of Technology, Vellore, India

https://orcid.org/0000-0002-1969-7632

Ranjith D., Meenakshi Academy of Higher Education and Research, Chennai, India

Vinoth kumar V, MVJ College of Engineering, Bangalore, India

Balajee J M, Vellore Institute of Technology, Vellore, India

ABSTRACT

Web scraping is a technique to extract information from various web documents automatically. It retrieves the related contents based on the query, aggregates and transforms the data from an unstructured format into a structured representation. Text classification becomes a vital phase to summarize the data and in categorizing the webpages adequately. In this article, using effective web scraping methodologies, the data is initially extracted from websites, then transformed into a structured form. Based on the keywords from the data, the documents are classified and labeled. A recursive feature elimination technique is applied to the data to select the best candidate feature subset. The final data-set trained with standard machine learning algorithms. The proposed model performs well on classifying the documents from the extracted data with a better accuracy rate.

KEYWORDS

Back-Propagation Neural Networks, Content Retrieval, Machine Learning, Recursive Feature Elimination, Text Classification, Web Harvesting, Web Scraping

1. INTRODUCTION

In most of the web sites, the users are allowed only to view the content. Access to download or copy the material will be restricted to avoid discrepancies. However, in some cases, the data can be fetched manually, but it is a time-consuming process. So, to make this as an automated function, Web Scraping technique is introduced (Vargiu, 2013). It is an intelligent program or a web script that helps to extract the content from the webpages. Furthermore, it could store in a structured format in the local system for future analysis. It enables rapid, in-depth retrieval of relevant data. This technique can be useful to a different set of peoples from all the fields to capture the humongous amount of

DOI: 10.4018/IJWP.2019070103

information from the internet. In recent days, the data from the web turned into spreadsheets as structured content using web scraping tools.

The process of scraping content from web overloads the target site with high demand. The server of the target sight might temporarily go down when a web crawler sends too many service requests to the specific site. Most sophisticated sites are powered up by anti-scraping techniques to resist the attacks from such web scraping bots (McKenna, 2016).

Automated web scraping technique is inevitable to extract knowledgeable content as the amount of data generated over time becomes increasing (Ikonomakis, 2005).

The existing approaches have few drawbacks in it, and some of them are ineffective scraping methods, inadequate server response, and irregular data transformation. The proposed scraping technique in this paper performs effectively in data extraction and conversion into well-structured form.

The rest of the paper organized as follows. Section 2 discusses the existing literature related to this study. Scraping methodologies and its types briefed in section 3. The proposed model clearly stated in the modules in section 4. In the next section, the results depicted through graphs and tables, and section 6 concludes the work by highlighting its significance.

2. BACKGROUND

A personalized recommendation system is developed from the user's search by (Liang et al., 2008) to provide customized content. Similarly, news categorization is made personally to the target users of different groups with scalable document classification techniques (Ioannis et al., 2006).

Collaborative tagging improves the keyword extraction process with better outcomes. Content-based tagging system represents the capabilities of search systems (Nirmala et al., 2010). The personalized blog recommendation system developed (Chiu et al., 2018) for mobile phone users. User history and browsing content are analyzed to provide targeted recommendations.

A personalized web-bot is created to assist the user based on their interest to view specific content and webpages (Jung et al., 2004). This system is developed by fusing collaborative filtering and hybrid content-based filtering techniques. Web browsing classification system on mobile interfaces is developed with six standard perspectives (Roudaki et al., 2015).

Genetic algorithm based document clustering method is proposed to mine the text from a large amount of biomedical information (Wahiba et al., 2016). A structured meta-data extraction method is deployed to fetch information from scientific studies (Tkaczyk et al., 2015) and available for researchers under open source license.

A deep learning method is applied for text classification using the softmax regression technique and deep belief networks (Jiang et al., 2018) on scientific research databases. An automated system for climatic data scraping, cleaning, and display the results is proposed to improve the climatic information processing (Yang et al., 2010).

Deep learning methodologies provide state-of-art methods on heterogeneous, sophisticated text analytics. In-text classification, many methods are proposed to enhance model performance. A character-level text classification system proposed with convolutional networks. It is constructed with many large scale datasets to highlight the significance of the work through its competitive results. These models compared with a bag of words, TFIDF technique, n-grams, and other deep learning approaches (Zhang, 2015).

Recurrent neural networks are most popular among text classification models. A recurrent convolutional model is developed to improve the model accuracy on classifying various text data (Lai, 2015). Through the multi-task learning strategy, a novel framework deployed for useful text classification in different layer schemes (Liu, 2016).

Sentiment review classification system provides an intuition about blogs, reviews about any subjective topic, and comments posted in online portals with the support of machine learning algorithms. An n-gram based technique enhances the predictive probability of the system with other NLP related functionalities incorporated together (Tripathy, 2016).

3. SCRAPING TECHNIQUES

Scraping is an activity of retrieving information on various web sites. It could be done with or without permission from the site owners. This process could be automated or manually done. Content scraping is mostly used to perform malicious activities. Stealing of data through web-scraping is unethical (Suchacka et al., 2015). The site owner needs to manage and defend the site against malicious programs to keep their online business one step ahead of others to be on a competitive edge in their market world.

3.1. Types of Scraping

3.1.1. Screen Scraping

Capturing the screen data from the display of one application and transforming it into another application to display the same information effectively.

3.1.2. Report Mining

Extraction of information from the computer reports that of human-readable form.

3.1.3. Web Scraping

It is the process of retrieving necessary information from a website and converting it into a structured form for future analysis.

3.2. Web Scraping

Most websites display data that can view with typical web browser software. The functionality of saving a copy of the data is mostly restricted. The next option is to perform the copy/pasting of data from the site manually, but the process is tedious. It sometimes takes some days to complete the task.

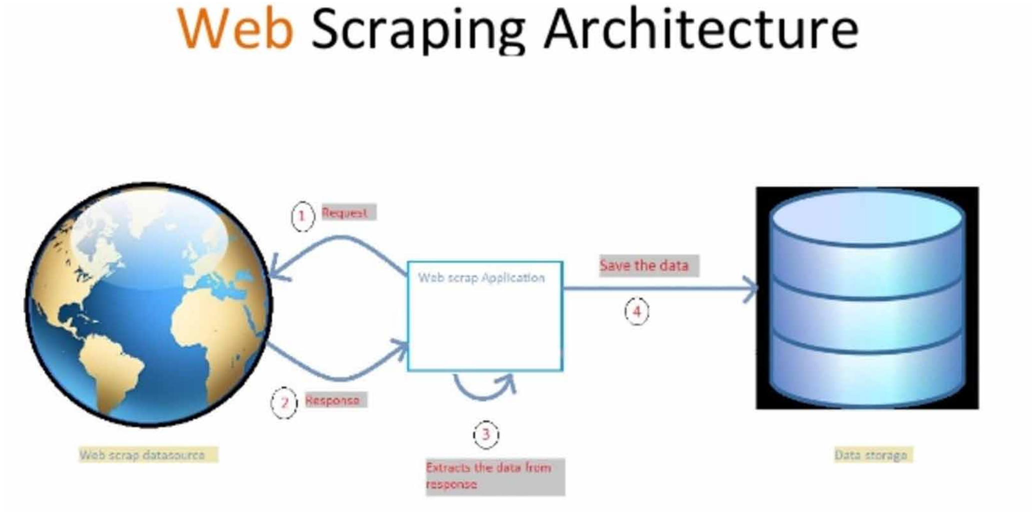
Web scraping is a technique used to extract information from the websites. It turns the unstructured data into a structured form. These data can be stored anywhere in the local computer or remote servers. This process is often efficient, faster, and less prone to errors when automated. In recent times, many researchers are widely using this technique to create their own data sets for journal article information extraction, text mining-related projects, etc (Panta et al., 2015).

This web scraping software's loads the multiple pages of the website automatically and extracts useful information out from it. The content extracted from the site depends on the type of requirement. It can be configured anytime and based on the type of the website, and it is customized. With a single click, the data can be scraped, formatted, and stored as a file in the system. (Russell et al., 2013). The data can be replicated anywhere with the help of the scraper software. It is different from screen scraping where the pixels of the display will capture, but in web scraping the underlying data inside the HTML tags can be retrieved. In recent times, many intelligent bot scripts are deployed to do such tasks (Figure 1).

3.3. Prerequisites for Web Scraping

The preliminary step of web scraping is to keep the existing HTML scripts, and with the help of a web scraper program, the data can be extracted and converted into a useful form. The last step is to store the data either in JSON or another format. These scrapers can be built using any technology such as PHP, Node JS, and Python etc. In most cases, python will be the choice to develop such programs, which is useful in terms of flexibility. The knowledge on the structure of HTML scripts and data formats such as JSON is highly essential to build robust web scraper software. The basic requirements to develop a web scraping tool are:

Figure 1. A typical architecture of web scraping process



1. Python Libraries
2. Data Formatting
3. HTML Constructs

3.4. Manual Scraping

Copying a part or complete web content from a site and pasting manually is called manual scraping. However, this technique is ineffective as it needs much effort, and the process is repetitive (Petta et al., 2013). This method works well when a web site protected with anti-scraping techniques.

3.5. Automated Web Scraping Techniques (Table 1)

3.5.1. HTML Parsing

HTML parsing is a common technique. It can be done through JavaScript that targets nested and linear HTML pages (Malik et al., 2011). It works faster and identifies HTML tags from webpages.

3.5.2. DOM Parsing

DOM otherwise Document Object Model defines the style, structure and the content of an XML document. The internal functionality of a website must be transparent to the scrapers for content extraction, and DOM delivers its parsing modules to do it (Lourenço et al., 2013). The nodes organized with DOM parsers and XPath and other related tools help to retrieve the content from the sites. DOM parsers can even work well with dynamic websites.

Table 1. Techniques adopted for effective web scraping

Techniques	Advantages
Scripts	Automates tasks, Perform complex functions
Command Line Codes	Processes huge files
DOM Method	Identifies and extracts data inside scripts
Regular expression	Identifies and extracts data through standard protocols

3.5.3. Vertical Aggregation

Industries having high computational power targets specific sites and create vertical scalable platforms. Few might have run even in cloud platforms (Sirisuriya et al., 2015). To monitor such platforms, bots created and the need for human intervention becomes null in such cases. These bots could efficiently perform from the existing knowledge base of the system.

3.5.4. XPath

XML Path Language (XPath) is a query language, used when the data is being extracted from the nodes of the XML documents. XML files follow a hierarchical tree-like form. XPath provides a simple way to handle exact nodes and to extract data from the nodes (Myllymaki et al., 2002). It is used with DOM parsing technique to retrieve data from the webpages.

3.5.5. Text Pattern Matching

It is a regex (regular expression matching) technique using the grep command from UNIX systems (Johnson et al., 2012). It is often integrated with standard programming languages such as Python, Perl, etc.

3.6. Extracting Data Unstructured Content

Web data extraction is simply an automated technique to fetch necessary data from a website. Then it transforms the data from unstructured representation to structured form and can be stored in a warehouse or a database. The Internet holds a massive amount of unstructured data (Herrouz et al., 2013). The data can be turned into useful insights after they framed into a unique form that simplifies the analytical process.

3.7. Web Scraping Tools

Open Source becomes a buzzword in technology and fuels up the technology industry to share and experience things. Most of the tools nowadays are open-sourced, and web scraping tools become a part of it (Hofmann, 2016). These tools and scraping frameworks effectively parse the sites and fetches the contents. Some handy scraping tools discussed in Table 2.

The list of various web scraping frameworks or languages and the best open source web scraping tools available in each language or platform.

4. MATERIALS AND METHODS

Text classification is an integral part of natural language processing (NLP). It automatically discriminates these documents by analyzing its content using a sophisticated machine learning algorithms. In this work, using an effective web scraping method, the content is retrieved from online sites and is personalized in its way and type of information. The collected data is transformed into a structured form to simplify the analytical process. Using machine learning algorithms, the data trained

Table 2. Tools for web scraping

Tool Name	Functions
OpenRefine	Data Cleaning and Processing
cURL (CLI)	Data can be retrieved through API’s
Wget (CLI)	Retrieves the webpages recursively
Web Scraping service	Makes data acquisition process simple

and evaluated to identify the exactness of the proposed system on categorizing the documents. This process detailed in the upcoming sections.

4.1. Content Scraping

In this phase, the target site defined from where the content is being extracted, and useful data fetched from the scraped information (Figure 2).

4.1.1. Requesting Data From the Website

The site is accessed through standard protocols and using objects, and the data stored in HTML format. The extracted data is then processed to eliminate HTML constructs, and data alone will be separated.

4.1.2. Extracting Text From HTML Page

The process of extracting content from HTML pages can be done with available python libraries. BeautifulSoup is a library or package used to fetch the data from HTML files. Similarly, html2text module works better for the same task. Using these packages, the text data scraped and stored in a structured spreadsheet format for next level processing.

A snippet of the code is given below:

```
b_soup = BeautifulSoup(html_script, ‘html_script.parser’)
text_info = b_soup.findAll(text = True)
text_from_html’ ‘.join(text_info)
```

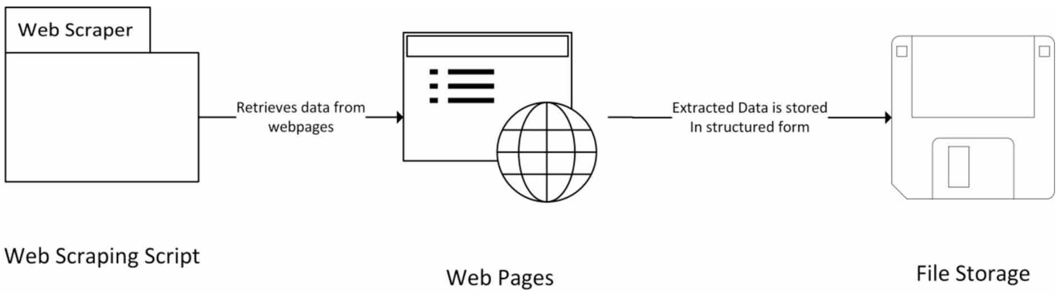
In the above snippet, BeautifulSoup library parses through the HTML script and the data is assigned to b_soup object. In the next line, findAll() function returns the strings present in the script and the following line, join() function binds all the individual strings together.

4.2. Keyword Matching

Any classification algorithm needs the data to be in a labeled format to train and evaluate the performance of the model. To label the extracted samples, relevant keywords are manually prepared and grouped that are related to the scraped data. The samples are labeled based on the matching score calculated using the keywords. The formula is given in Equation 1:

$$matchingscore = \frac{\text{Number of matched keywords under a category}}{\text{Total number of keywords matched}} \quad (1)$$

Figure 2. Basic process of web scraping


Web Scraper
Retrieves data from webpages
Extracted Data is stored In structured form
Web Scraping Script
Web Pages
File Storage

With the help of KeywordProcessor library, a matching score calculated. Now, based on the score, the samples are labeled into its categories.

4.3. Dataset Information

The dataset prepared by scraping the content from technical, fashion, and news blogs. Datasets of 1870 samples are collected, and each one is categorized and labeled from the previous steps.

4.4. Document Classification

The samples are initially pre-processed, and the best features are selected using Logistic Regression – Recursive Feature Elimination (LR-RFE) before feeding them into Back-Propagation Neural Networks (BPNN). The preprocessing steps include tokenization, stemming, and bag-of-words.

4.4.1. Text-Preprocessing

4.4.1.1. Tokenization

In a sentence, paragraph or a document, the words divided into individual pieces and each of them is called 'tokens.' It generally normalizes a statement with complex representation and punctuations. Each sample in the dataset converted into individual tokens in this phase.

4.4.1.2. Stemming

Stemming is the concept of eliminating inflected word to its root form otherwise, stem. It transforms the related set of words from its common form to unique base form. The tokens in the samples are stemmed up into its base representation.

4.4.1.3. Bag-of-Words

In information retrieval and NLP, bag-of-words (BoW) technique remains as an essential phase for text analysis. The data given in sentence or individual tokens can be combined in different unit-level form Bows. Disregarding the order of words or grammar, the number of words in each bag might be anything more than one. However, the standard count lies between the range 3-5 and up to 7 in some cases. In this system, the number of words in a bag contains five words each.

4.4.2. Logistic Regression – Recursive Feature Elimination (LR-RFE)

To make a better prediction over input data, selecting the best features is highly significant. In this work, LR-RFE, a wrapper based feature selection technique, is applied on the dataset to select optimal feature subset. It is a greedy approach, where it initially uses a learning model (in this case, LR) to train the samples with initial feature subset (Chen, 2007). In an iterative process, the features are randomly selected and trained. During every iteration, the worst-performing features eliminated and the best features are finally sorted out. Even though this method is computationally expensive, the outcome of the process is better than other techniques.

After applying this technique, the number of features is significantly reduced and simplifies the classification task (Figure 3).

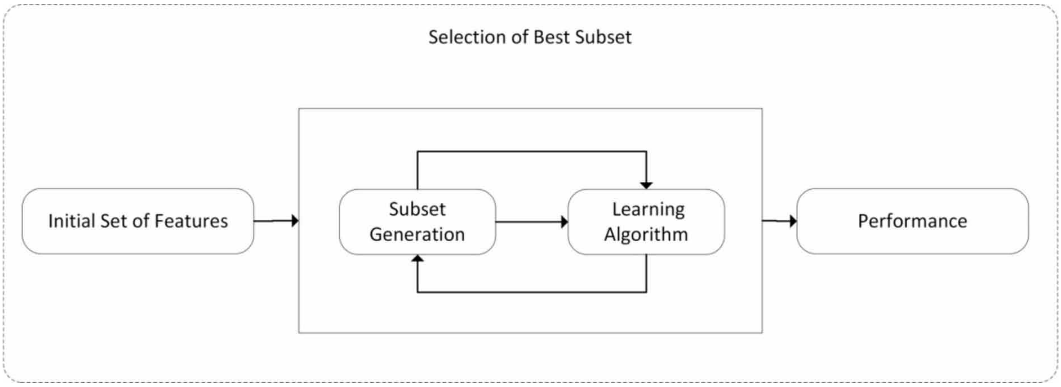
4.4.3. Classification

Machine Learning algorithms are widely used in many applications to develop robust predictive models. Due to its high success rate, the fields that are highly interrelated with intelligent computing adopt machine learning methods. In this work, three supervised algorithms are deployed to train the model with the prepared dataset. Those algorithms are Support Vector Machines (SVM), Random Forest (RF) and Back-Propagation Neural Networks. Each algorithm has its advantages and features that work well on different data. In general, the neural network model works well on text data when

Algorithm 1. LR-RFE

```
Step 1: Train logistic regression with training data set
$P = \frac{1}{1 + e^{-a+bX}}$
Step 2: Calculate the performance of the model
Step 3: Find the variable importance
Step 4: For each subset $S_{i,}$ I = 1…S do
        i. Keep the best feature from the subset $S_i$
        ii. Train the model on the training set with new subset $S_i$
        iii. Evaluate the model performance.
End
Step 5: Calculate the performance of finite subset, $S_i$
Step 6: Determine the number of predictors
Step 7: To the optimal subset $S_{i,}$ find a corresponding learning model.
```

Figure 3. Process of LR-RFE Algorithm



compared with the other models. As it comes true, in this experiment, the BPNN model outperformed RF and SVM models. The parameters of the BPNN model given in Table 3.

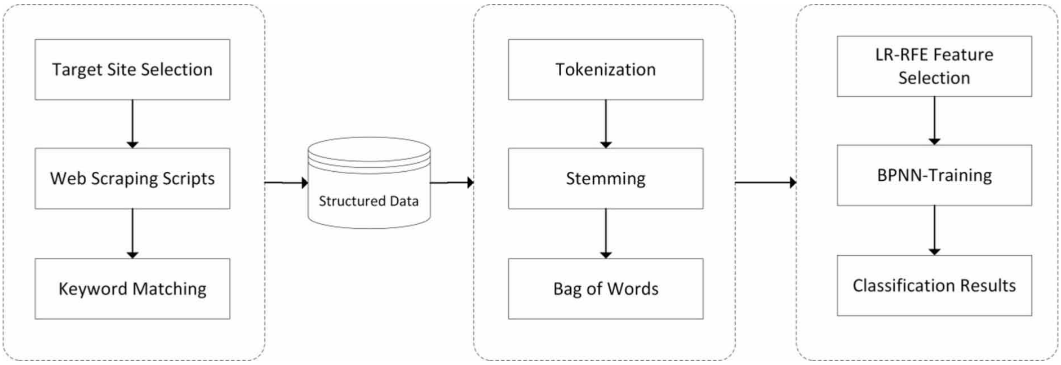
The pipeline shown in Figure 4 represents the workflow of the proposed model. Initially, the site from the data to be scraped is identified. Through web scraping tools and snippets given in section 4.1.2, the required data is collected and recorded. The scraped data labeled by finding the proper keyword from the data. The labeled data stored in a structured format.

The data undergone NLP operations such as tokenization, stemming, and a bag of words each to convert the data into tokens, pruning the prefixes and suffixes of a word and bagging of terms

Table 3. Parameters of BPNN model

Parameters	Values
Activation Function	Rectified Linear Unit (ReLU)
Learning Rate	0.01
Momentum	0.60
Epochs	500
Hidden Layers	2

Figure 4. Pipeline of the proposed work



respectively. The altered dataset is prepared to perform feature selection on it. The proposed LR-RFE algorithm identifies the feature subset from the dataset. Then, the optimal parameters are inputted into the BPNN model for training. The performance of the model evaluated with standard metrics, and classification accuracy calculated. The proposed model achieved 94.63% accuracy outperformed SVM and RF classifiers.

5. RESULTS

The experimental setup made to develop this model includes the following. Windows 10 Operating System, Anaconda Python Distribution (Python version 3.6) with NVIDIA GEFORCE GTX 950 MX 4GB Graphic Processor and Intel core 7th Generation 8GB processor. The proposed pipeline (LR-RFE + BPNN) achieved 94.62% accuracy on the test data. The evaluation performed with KFold Cross-Validation technique with 10 Folds each. Accuracy measure is used to calculate the classification rate of the model. The results obtained with three machine learning algorithms given in Table 4.

Moreover, the performance of the model before and after feature selection is also calculated and projected in Figure 5.

From the above graph, it is observed that there is a significant improvement in the model after performing feature selection on the dataset. The proposed LR-RFE-BPNN model accurately discriminates the samples well than the other benchmarked algorithms on text classification with 94.63% accuracy.

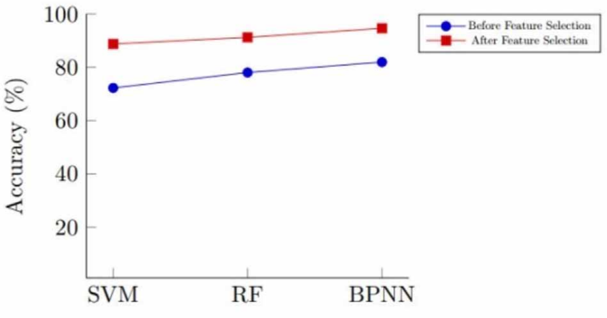
6. CONCLUSION

In this paper, a robust text classification model is proposed using various NLP and machine learning techniques. Through web scraping the data is fetched and transformed into a structured form for further analysis. The data fetched through web scraping is made personalized concerning its content. With the

Table 4. Accuracy of the models on different classifiers trained after LR-RFE

Model	Accuracy
Support Vector Machine	88.72%
Random Forest	91.20%
Back Propagation Neural Network	94.63%

Figure 5. Accuracy of the model before and after feature selection



help of NLP techniques, the data is processed and simplified for the next process. Text classification is made from standard machine learning techniques to train the model with the processed data. The outcome of this work revealed the best performing pipeline for text classification. LR-RFE-BPNN model produced a better result for this system.

Web scraping becomes an essential technique to retrieve data from web sites effectively. It is an essential skill that can be combined with sophisticated analytical techniques to develop intelligent text classification systems. In today's digitalized world, the rate of data generation is at a peak. The advent of big data simplifies the process of storing and analyzing massive data. Most of the content available online is unstructured. Web scraping tools could be helpful to extract essential information and makes them into structured form. These automated tools could aggregate the scraped data, transform them into useful insights, and empowers the business world with intelligent decision systems by analyzing the real-world data.

REFERENCES

Antonellis, I., Bouras, C., & Poulopoulos, V. (2006, January).Personalized news categorization through scalable text classification. *Proceedings of the Asia-Pacific Web Conference* (pp. 391-401). Springer. doi:10.1007/11610113_35

Chen, X. W., & Jeong, J. C. (2007, December).Enhanced recursive feature elimination. *Proceedings of the Sixth International Conference on Machine Learning and Applications (ICMLA 2007)* (pp. 429-435). IEEE. doi:10.1109/ICMLA.2007.35

Chiu, P. H., Kao, G. Y. M., & Lo, C. C. (2010). Personalized blog content recommender system for mobile phone users. *International Journal of Human-Computer Studies*, *68*(8), 496–507. doi:10.1016/j.ijhcs.2010.03.005

Glez-Peña, D., Lourenço, A., López-Fernández, H., Reboiro-Jato, M., & Fdez-Riverola, F. (2013). Web scraping technologies in an API world. *Briefings in Bioinformatics*, *15*(5), 788–797. doi:10.1093/bib/bbt026 PMID:23632294

Herrouz, A., Khentout, C., & Djoudi, M. (2013). Overview of web content mining tools.

Hofmann, M., & Chisholm, A. (Eds.). (2016). *Text mining and visualization: case studies using open-source tools* (Vol. 40). CRC Press. doi:10.1201/b19007

Ikonomakis, M., Kotsiantis, S., & Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS Transactions on Computers*, *4*(8), 966–974.

Jiang, M., Liang, Y., Feng, X., Fan, X., Pei, Z., Xue, Y., & Guan, R. (2018). Text classification based on deep belief network and softmax regression. *Neural Computing & Applications*, *29*(1), 61–70. doi:10.1007/s00521-016-2401-x

Johnson, F., & Gupta, S. K. (2012). Web content mining techniques: A survey. *International Journal of Computers and Applications*, *47*(11).

Jung, K. Y., & Lee, J. H. (2004). User preference mining through hybrid collaborative filtering and content-based filtering in recommendation system. *IEICE Transactions on Information and Systems*, *87*(12), 2781–2790.

Karaa, W. B. A., Ashour, A. S., Sassi, D. B., Roy, P., Kausar, N., & Dey, N. (2016). Medline text mining: an enhancement genetic algorithm based approach for document clustering. In *Applications of Intelligent Optimization in Biology and Medicine* (pp. 267–287). Cham: Springer. doi:10.1007/978-3-319-21212-8_12

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015, February).Recurrent convolutional neural networks for text classification. *Proceedings of the Twenty-ninth AAAI conference on artificial intelligence*. AAAI Press.

Liang, T. P., Yang, Y. F., Chen, D. N., & Ku, Y. C. (2008). A semantic-expansion approach to personalized knowledge recommendation. *Decision Support Systems*, *45*(3), 401–412. doi:10.1016/j.dss.2007.05.004

Liu, P., Qiu, X., & Huang, X. (2016).Recurrent neural network for text classification with multi-task learning.

Malik, S. K., & Rizvi, S. A. M. (2011, October).Information extraction using web usage mining, web scrapping and semantic annotation. *Proceedings of the 2011 International Conference on Computational Intelligence and Communication Networks* (pp. 465-469). IEEE. doi:10.1109/CICN.2011.97

McKenna, S. F. (2016). *Detection and classification of Web robots with honeypots*. Naval Postgraduate School Monterey United States.

Myllymaki, J. (2002). Effective web data extraction with standard XML technologies. *Computer Networks*, *39*(5), 635–644. doi:10.1016/S1389-1286(02)00214-1

Panta, D. (2015). Web crawling and scraping: developing a sale-based website.

Petta, D. L., & Mohs, B. K. (2013). U.S. Patent No. 8,595,847. Washington, DC: U.S. Patent and Trademark Office.

Pudota, N., Dattolo, A., Baruzzo, A., Ferrara, F., & Tasso, C. (2010). Automatic keyphrase extraction and ontology mining for content-based tag recommendation. *International Journal of Intelligent Systems*, *25*(12), 1158–1186. doi:10.1002/int.20448

Roudaki, A., Kong, J., & Yu, N. (2015). A classification of web browsing on mobile devices. *Journal of Visual Languages and Computing*, *26*, 82–98. doi:10.1016/j.jvlc.2014.11.010

Russell, M. A. (2013). *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More*. O'Reilly Media, Inc.

Sirisuriya, D. S. (2015). A comparative study on web scraping.

Suchacka, G., & Sobkow, M. (2015, June).Detection of Internet robots using a Bayesian approach. *Proceedings of the 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)* (pp. 365-370). IEEE. doi:10.1109/CYBConf.2015.7175961

Tkaczyk, D., Szostek, P., Fedoryszak, M., Dendek, P. J., & Bolikowski, Ł. (2015). CERMINE: Automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition*, *18*(4), 317–335. doi:10.1007/s10032-015-0249-8

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, *57*, 117–126. doi:10.1016/j.eswa.2016.03.028

Vargiu, E., & Urru, M. (2013). Exploiting web scraping in a collaborative filtering-based approach to web advertising. *Artificial Intelligence Review*, *2*(1), 44–54.

Yang, Y., Wilson, L. T., & Wang, J. (2010). Development of an automated climatic data scraping, filtering and display system. *Computers and Electronics in Agriculture*, *71*(1), 77–87. doi:10.1016/j.compag.2009.12.006

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In Advances in neural information processing systems (pp. 649-657). MIT Press.

Karthikeyan T. works as an Assistant Professor and as Head of Department Computer Science and Engineering at Sri Balaji Chockalingam Engineering College, Arni. He has completed his Bachelor of computer science degree in University of Madras, Chennai and Master of Computer Application degree from the same university, Chennai, Completed M. Phil from Periyar University, Salem and Finished ME Computer Science and Engineering from Anna University, Chennai. His current research interests include Big Data, machine learning, and IoT.

Karthik Sekaran is a research scholar from Vellore Institute of Technology.

Ranjith D. is working as an Assistant Professor/Placement Officer in the research institution.

Balajee Jeyakumar is currently pursuing a PhD in the School of Information Technology and Engineering, Vellore Institute of Technology University. He received his Bachelor of Computer Science degree in University of Madras, Chennai and Master of Computer Application degree from VIT University, Vellore and M. Phil degree Thiruvalluvar University, Vellore, respectively. He has worked as a Project Assistant for a project on Stability and aggregation of silver nanoparticles in natural aqueous matrices funded by the CSIR-Physical Sciences, Chennai, and Government of India. His current research interests include Big Data, Machine Learning, Deep learning, and IoT. He is the author/co–author of papers in conferences, book chapters, and journals.