SSRG International Journal of Computer Science and Engineering
ISSN: 2348–8387 / https://doi.org/10.14445/23488387/IJCSE-V11I6P103

Volume 11 Issue 6, 11-19, June 2024
© 2024 Seventh Sense Research Group®

Original Article

Intelligent Web Mining Technique using Sequential Pattern

Elliot, S. J[1], Bennett, E.O[2], Nwiabu, N. D[3], Matthias, D.[4]

[1,2,3,4]*Department of Computer Science, Rivers State University, Port Harcourt, Nigeria.*

Corresponding Author : sobestman@yahoo.com

Received: 16 April 2024 Revised: 20 May 2024 Accepted: 05 June 2024 Published: 19 June 2024

Abstract *- As organizations expand and share more information about their operations online, the website data produced by these organizations becomes an invaluable resource for studying innovations. Effectively managing this vast volume of data and presenting relevant information to users is paramount. It is not practical to analyze and retrieve data manually from large databases. Addressing this challenge requires automated extraction tools enabling users to sift through billions of web pages and unearth pertinent information. This mechanism allows individuals and organizations to analyze data patterns within web contents and page structures, facilitating the discovery of valuable insights and knowledge. It aids in predicting user behavior during their online interactions, uncovering navigation patterns, and extracting useful information from user engagements, thereby enhancing our comprehension of consumer behavior. This paper focuses on extracting patterns of web access. Generally, a weblog can be seen as a series of user identifiers and event pairs. In this paper, web log files are segmented based on mining objectives. Preprocessing techniques are employed on the original web log files to extract segments. Each segment represents a sequence of events from a single user or session, arranged in ascending timestamp order. The model interprets these segments as event sequences and identifies sequential patterns exceeding a certain support threshold. This paper presents the mining of a sequential list of papers from the Neural Information Processing Systems (NIPS) website using the PrefixSpan algorithm. The system is implemented in Matlab programming language. Matlab programming language has been used in web mining to harvest useful data from the web, such as user logs and content. The system is tested and evaluated using accuracy and accessibility.*

Keywords *- Data mining, Web mining, Sequential patterns, Frequent patterns, Web logs.*

1. Introduction

Data Mining has emerged as a fundamental discipline within computer science. The term first began to circulate in the late 1980s, primarily within the research community. Initially, there was ambiguity regarding the scope of data mining, a contention that persists to some extent today. Broadly, the techniques for discovering concealed insights from data using various mechanisms and techniques implemented in software are referred to as data mining.

By the early 1990s, data mining was recognized as a subset of Knowledge Discovery in Databases (KDD), a broader process aimed at pinpointing valuable, innovative, and understandable patterns in data. KDD encompasses several sub-processes, including data preparation (e.g., warehousing, cleaning) and result analysis/visualization. Although KDD and data mining are often used interchangeably, technically, data mining is an essential component of the overarching KDD process.

Web mining software is utilized to identify patterns on the internet through the implementation of data mining methodologies. Web mining is a critical domain within computer science and information science in the contemporary era, characterized by an overwhelming amount of diverse, recurrent, and abundant data. E-commerce, web analytics, filtering, and information retrieval are merely a few of its numerous potential applications.[1].

Web mining deals with the automated process of extracting knowledge via web data, encompassing web documents, document hyperlinks, and website utilization records, through the implementation of data mining algorithms.

Web mining employs several data mining methodologies, such as classification, association rule mining, clustering, and frequently utilized item set analysis. Subtasks consist of the following: identifying pertinent resources, choosing and preprocessing data, and ultimately, deriving conclusions.

Web mining is differentiated from traditional data mining by the heterogeneous, unstructured, or semi-structured character of Web data, which includes text, images, and videos, among other things. To effectively manage these enormous data sets and extract actionable insights, it is therefore essential to develop new techniques and instruments.

In situations where objects outside of a cluster exhibit dissimilarities while objects within the cluster share similarities, clustering algorithms can identify these groups and subsequently group the objects.

However, heuristics are required because the development of clustering algorithms can be difficult due to the potential for substantial memory or processing requirements. Sequential pattern mining is regarded as an effective technique for scraping web utilization due to its capability to handle profiles induced by crawling [2].

Due to the multitude of elements and frequently unstructured structure of HTML web pages, retrieving data is a challenging endeavour. Numerous algorithms have been proposed in an effort to address this issue; among them is the HÜ-PLWAP miner, which uncovers ambiguous sequential patterns by utilizing internal and external utility information. [3]. Despite its promise in innovation research, there is a dearth of published literature concerning the validity and utility of web mining, particularly with regard to the use of company websites as a data source.

In addition, web mining is highly dependent on clustering algorithms, which categorize elements based on similarities in their data. However, the efficacy of cluster analysis is constrained by the challenge of determining the optimal quantity of clusters [4].

This research aims to extract web content for sequential pattern analysis utilizing the PrefixSpan algorithm. PrefixSpan examines the entire projected database to identify frequent patterns, mining the entire collection of patterns while significantly reducing the effort required for generating candidate subsequences. The choice of the PrefixSpan algorithm is based on its innovative approach, which enhances speed and efficiency in mining large itemsets. The system uses less memory in comparison to the GSP and Markov models.

In summary, web mining exhibits considerable potential in extracting valuable insights from the vast expanse of the internet; however, further research and advancement are required to surmount its constraints and fully exploit its capabilities across diverse domains.

2. Related Literature

2.1. Web Mining

The objective of web mining is to autonomously detect and extract data from online publications and services through the utilization of data mining methodologies. Its primary objective is to locate and evaluate effective information on the World Wide Web.

Web utilization mining is included in this. It involves the automatic identification of user access patterns to web servers, in addition to web content mining, which involves the automated retrieval of information resources from the internet.

2.1.1. Applications of Web Mining

Data Mining (DM) finds application in various domains such as image mining, web minng, text mining and sequential pattern mining. Amongst these, Sequence Data Mining (SDM) stands out as a fundamental operation in computational biology, revealing sequential relationships and hidden knowledge within vast sequence datasets. For instance, the BiRen algorithm utilizes deep learning to predict enhancers solely from Deoxyribonucleic Acid (DNA) sequences [5]. Additionally, Lim [6] introduced an automated information extraction system based on Support Vector Machines for text mining of microbial interactions. SDM boasts diverse applications, including DNA/protein sequencing/RNA, web access patterns, weather forecasting, customer purchase analysis, business and medical data analysis, and security. In bio-data analysis, critical tasks include similarity searches and comparisons among biological sequences and structures. Sequence analysis involves subjecting RNA, peptide sequences or DNA to aligning sequences, databases, repeated searches of sequences, or another method of bioinformatics on computers.

As costs decrease, bioinformatics computing advancements and the availability of complete sequences of the genome, bioinformatics offers both theoretical foundations and methods that are practical for understanding systemic cellular and organismal behaviors. Within RNA, protein sequence and DNA analysis, SDM techniques are employed for searching, classification and aligning of sequence. Classification of Protein sequence remains a popular research area. Other areas include Personalization of web content, E-commerce, Prefetching and Catching, etc.

Choudhary [7] suggested Exploring the Landscape of Web Data Mining. Through the presentation of the most recent advancement and the provision of web mining techniques on web data, both practitioners and researchers can get insight into the current state of the field and pinpoint possible areas requiring more investigation. Along with summarizing and comparing several web data mining techniques with applications, their paper provides an overview of research developments and highlights certain key research questions.

An image search engine called iFind was created by the researchers, and it works better than conventional methods. IFind, such as log mining, relevance feedback, query by example, and keyword-based search, support numerous search possibilities. Their study provides a thorough grasp of previous accomplishments and paves the path for future research and practice in web data mining, making it an invaluable resource.

2.1.2. Web Mining Categories

The technique of web mining exists in three forms: web usage, web content, and web structure mining. Consumer behavior management, text mining, and e-commerce web mining are a few of the numerous applications.

Web Structure Mining

Web structure mining The procedure of extracting information regarding the structure of a website is referred to as “web structure mining. The structure of the web graph comprises web pages represented as nodes, interconnected by hyperlinks forming edges. This methodology proves particularly beneficial in establishing connections between two commercial websites and unveiling the document structure, thereby enabling comparisons between web page layouts.

Web Content Mining

Automatically finding web-based information is challenging due to the absence of structure in the sources of information accessible via the World Wide Web (WWW). Classic search engines like WebCrawler, MetaCrawler, Lycos, ALIWEB, Alta Vista, and others give users some comfort, but they typically don’t categorize, filter, or interpret documents or offer structural information.

Web Usage Mining

Refers to the automated process of uncovering patterns in user interactions with web servers. Organizations accumulate extensive data through daily operations, automatically generated by web servers and stored in server access logs. Additional sources of user information include referrer logs containing details about pages that refer to each page accessed and data from user registrations or surveys gathered via CGI scripts. In the realm of online advertising, analyzing user access patterns aids in targeting advertisements to specific user groups.

Applications of Web Usage Mining (WUM)

WUM applications’ are proliferating due to the growing interest of organizations in e-commerce websites and other web marketing tools. Web semantic mining is a burgeoning field that is currently garnering significant interest from the web semantics community. This bodes well for applications related to WUM. This category includes customization, business intelligence, site modifications, and system enhancements, among other things.

2.2. Sequential Pattern Mining

The discovery of sequential patterns is a data mining method. A significant proportion of the objects demonstrate sequential patterns. The predominant objective of the suggested algorithms for SPM is to identify all sequential patterns whose support levels surpass a minimal threshold specified by the user.

Agrawal [8] initially presented sequential pattern mining in 1995, and they suggested the following three algorithms: DynamicSome, AprioriAll, and AprioriSome. The definition of sequential pattern mining is then expanded upon using several factors, including time limits, sliding window times, and user-defined criteria, in order to present an enhanced Apriori-based technique known as GSP (Generalized Sequential Patterns). Zaki [9] mentioned the SPADE algorithm, which is predicated on class equivalency. All it was doing was broadening the sequential pattern mining approach for vertical data formats. There are later ways of pattern growth developed. Han suggested FreeSpan and PrefixSpan as two pattern growth methods [10]

Numerous domains, including but not limited to scientific investigations, medical treatments, protein synthesis, natural disaster analysis, and internet usage patterns of customers, stand to gain significantly from resolving the critical data mining challenge of sequential pattern mining. Recurring patterns, or sequences that occur frequently, are what a sequential pattern mining algorithm attempts to discover. This information could potentially be employed by users or upper management to facilitate future marketing campaigns, company reorganization, planning, or simply identify interconnections. Web use mining has become a significant domain for applying sequential pattern mining as a result of the widespread adoption of online services and electronic commerce.

2.2.1. Improving Efficiency in Sequential Pattern Mining Process

Due to the typically tremendous volume of processed data, it is critical to discover efficient methods for mining sequential pattern data. Essentially, two primary methods can be employed:

- Creating algorithms to increase efficiency
- Increasing effectiveness through the provision of maintenance tools.

The first approach is to design efficient algorithms. These algorithms can be divided generally into three classes:

- Horizontal, apriori-based formatting techniques like Generalized Sequential Pattern (GSP).
- Vertical formatting techniques, like SPADE, that are based on apriori.
- Projection-based techniques for pattern expansion, such as PrefixSpan.

The second approach is to build up a maintenance mechanism of sequential patterns. In real-life applications of sequential pattern mining, users always confront the following two situations:

- The sequence database will be updated with new transactions.
- The user is always adjusting the support value and is unable to locate the right support threshold at once. However, both of them may lead to a change in sequential patterns. It motivates to design of the maintenance mechanisms to efficiently uncover patterns in these situations without scanning the sequence database again and performing the entire mining procedure again.

2.3. Prefixspan Algorithm

The PrefixSpan algorithm, which embodies the pattern-growth methodology, first scans the sequence database once in order to identify the frequently occurring items. Subsequently, the larger database is subdivided into multiple small databases according to frequently occurring entries. Each projected database discovers the complete set of sequential patterns by expanding subsequence fragments recursively. Despite the successful identification of patterns using the division-and-conquer strategy of the PrefixSpan method, its implementation potentially led to a substantial memory space requirement as a consequence of the construction and processing of numerous projected sub-databases.

A single scan of the original sequence database can produce both the projected and original databases when utilizing PrefixSpan. Maintaining the lexicographic order is of utmost value. Although Split-and-Project has its applications, storing all the projections can be memory intensive, particularly when recursion is present. In order to tackle this concern, the concept of pseudo-projections was introduced [11]. Later on, this was utilized to produce algorithms such as SPARSE [12] and LAPIN_Suffix [13]. Here, there is no physical memory storage for the projected database; instead, the places where different projections are inside the sequence database in memory are specified using a pointer and an offset. Using bi-level projection, which FreeSpan and PrefixSpan represent, is another quicker method once the projected database has been established.

［14］ presented an effective method that just necessitates two database searches for finding frequent patterns in sequence databases. In the initial scan, support counts are acquired for subsequences of length two. A compressed frequent sequences tree structure (FS-tree) is used to represent the potentially frequent sequences of any length that are extracted in the second scan.

Support counting can be eliminated using the Direct Sequence Comparison (DISC) method and the DISC-all algorithm, which compare irregular sequences with another sequence of identical length. A DISC-all for web log extraction does not exist at this time.

A technique for database partitioning was proposed by [15]. By independently mining each segment of the sequence database until all frequently occurring sequences from all segments are combined, the database can be partitioned into numerous independent parts that are capable of fitting into the main memory.

SPAM, an algorithm that is advantageous in sequence mining, incorporates a Generate-and-test functionality. When combined with

Partitioning the search space and projecting databases without an Apriori-generate join, this technique is frequently implemented during specific mining phases. In this scenario, memory utilization is diminished by a magnitude's order.

Apriori-GST, an algorithm, was presented by [16]. The number of encoded subsequences, on the other hand, is ascertained through the utilization of a generalized suffix tree as a hash index. A closer examination of Apriori-GST raises the question of why the suffix tree was not utilized for mining as opposed to relying on a variant of Apriori and retaining the support data.

A PSP algorithm was proposed by [17]. The prefix tree is utilized to store candidate sequences, where the number of support for each sequence is indicated at the terminus of its corresponding branch. Web log mining is not the most optimal application for this technology (or GSP, the company that adopted it) due to the inefficiency it experiences when support is inadequate.

An online wine shop's (e-shop) web log information was analyzed, as stated in reference [18]. They developed a preprocessing interface. It was observed that the initial size of the data was diminished as a result of preprocessing, specifically the elimination of photographs. This is to be expected considering the characteristics of the evaluated website; in fact, e-commerce platforms and online stores commonly exhibit a multitude of product images. Additionally, the authors noted the noteworthy fact that a conventional log analyzer application could only achieve a 15% reduction in the log file size. This underscores the criticality of employing appropriate tools during the compilation phase.

Approaches to identify episodes, users, sessions, and page views, as well as path completion, were presented in [19]. Additionally, some of the suggested criteria may not be applicable to websites that are more intricate and extensive. This is what the term "user session" refers to. By examining each requested page in chronological order, the proposed heuristic attempts to distinguish between users whose IP addresses are identical. If the requested page is not mentioned in any preceding page, it signifies the beginning of a new user session.

The conceptual hierarchy of services was utilized to represent the query capabilities of the online catalog [20]. Schools across the globe can utilize the perusing and searching capabilities of this website. Instead of centering their attention on the content of the generated webpage, the author's objective was to evaluate the inquiries. Hence, a conceptual framework organized according to service orientation is implemented.

One of the three heuristics proposed by the authors for the preprocessing phase is analogous to the Browsing Speed. The frequency of requests lacking the referrer field and requests that are duplicated from the same site for the same resource are the subject of two additional heuristics. However, to validate the effectiveness of the latter, it is necessary to compare it to other heuristics.

2.4. Research Gap

Markov models are powerful tools for modeling stochastic processes, but they have several limitations and gaps (Tran, 2019). Markov's models are not well suited for sequences of varying lengths as they typically require a fixed state transition framework. It can struggle with capturing contextual information and longer patterns within sequences, leading to the potential loss of important sequence information. Also, Markov models are prone to overfitting, capturing noise rather than underlying patterns.

GSP involves generating a large number of candidate sequences and repeatedly scanning the database, which can be computationally expensive and time-consuming. GSP generates numerous candidates' sequences, many of which may not contribute to meaningful patterns, leading to inefficiencies.

By addressing these research gaps and leveraging solutions like PrefixSpan, the effectiveness and applicability of sequential pattern mining can be significantly improved.

3. Methodology

The research methods implemented in the research consist of constructive research and Object-Oriented Design Analysis (OODA). Even with contemporary techniques, the process of extracting sequential patterns from enormous datasets is still time-consuming. The Prefixspan algorithm is utilized in this process. Sequential pattern mining is implemented in Matlab programming language.

3.1. System Architecture

System architecture is the process of outlining a system's components, functions, and viewpoints. To find sequential patterns, the suggested approach makes use of prefix techniques. Figure 1 shows the system's constituent parts.

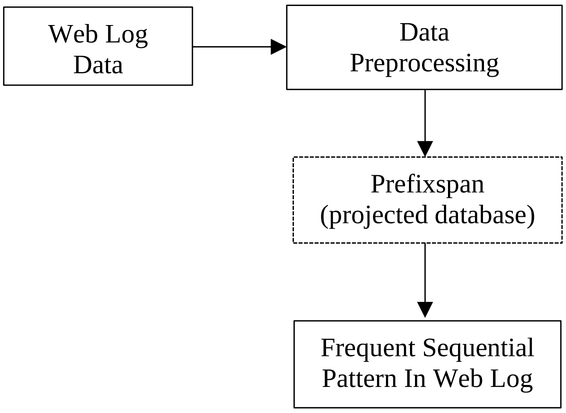
Web Log Data
Data Preprocessing
Prefixspan (projected database)
Frequent Sequential Pattern In Web Log

Fig. 1 Architecture of system

a) A Weblog: A weblog is a file that is updated by the Web server whenever a user requests a webpage. Using a list of the tasks it completed, it was automatically generated and kept up to date by a server.

b) Preprocessing: An operation that is performed on raw data to prepare it for the subsequent segment of data processing is referred to as data preparation. It has historically been a crucial preliminary stage in the process of data extraction. The preprocessing step is employed to convert the unprocessed data into a pristine data set. Unfortunately, performing analyses on data obtained in its unprocessed state from numerous sources is not feasible.

c) PrefixSpan: A technique utilized in sequence databases to identify patterns of prefixes.

d) Sequential patterns are regular patterns that appear in one or more transactions with several input sequences that follow one another.

3.2. Functional Design of Component

By analyzing the content of web pages, web archives can be extracted. A collection of weblogs extracted from the articles presented at the NIPS 2015 conference will be utilized for this investigation. The input variables are arranged in the following manner, as shown in Table 1: host IP address, articles, authors, affiliates, and coauthors; request date and time; and Uniform Resource Locators (URL) status bytes.

Table 1. Input variables

Input Variables	Description
10.2.1.40—(30/aug/2022:9:15:12+0530)	Date and Time
http://WWW.ncpapers.com/favicon.ico HTTP?!>! 302460	URL
TCP_MISS:FIRST_UP_PARENT	Host IP
Article Names	Deep learning, Reinforcement learning, Neural network
Authors	Samy Bengio, Danilo Rezende, Bill Dally
Coauthors	Nihar Shah, Dengyong Zhou, Jonathan Vacher
Affiliation	NICTA, UC Berkeley, MSR

3.3. Processing Design

Converting raw data into a clean data set is the idea behind data preprocessing. Preprocessing is performed on the dataset before its input into the algorithm to eliminate any missing values, noise, or other inconsistencies. Utilizing unprocessed web log data for pattern mining presents challenges due to its frequently diverse, fragmented, and poorly organized nature. The processes involved in its processing are illustrated in Figure 2.

4. Results and Discussion

The details pertaining to the dataset and extraction pattern are furnished for this assignment. For statistical analysis, website data and extraction patterns are included in Table 2.

Tags are element of a web page that instructs a web browser on how to format and display the page's content. The following are instances of tag names;

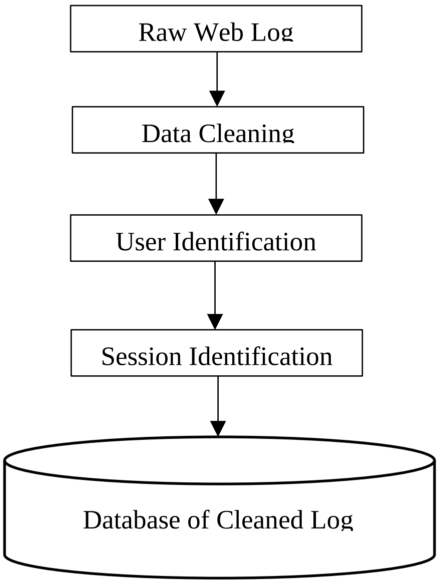

Fig. 2 Data preprocessing sequence

- The H tag is the header tag that helps organize websites. They separate subsections of the content. It can be in this format: H: <h1>, <h2>, <h3>
- Div - Known as Division tag used to group sections of a web page together
- Span tag used to wrap sections of text for styling purposes, p tag used for paragraphs that appear between opening<p> and closing</p> tags, ul tag for unordered list element for unordered list item.
- MFST, which is the main fieldset section table. Abbreviated as MFST: <main>, <fieldset>, <section>, <table>

During web mining, the system can extract relevant content without using any additional information. In this case, the mining is performed from the top of the document to the index of a document. This is the way simple extraction is performed. The second way, which is the default way of the approach, predicts the additional information, including startingPosition, innerTagCount, and repetition during the mining process.

Table 2. Information about extraction patterns

Tag Names	ID Average	Number of characters	Class	Fixed Inner Elements	Repetition
H	5	24	47	66	4
Div	34	35	138	78	3
Span, p, ul	2	28	17	13	3
MFST	3	32	15	13	-

Table 3. Effects of tags

Mining Time			
Parameters	**Outcome (millisec)**	**Appropriate (ms)**	**Not Appropriate**
H	0.008	0.008	N/A
Div	0.081	0.012	0.151
Ul, p, span	0.028	0.028	0.029
MFST	0.041	0.010	0.224

Presents the results of the average extraction time analysis, indicating that the most optimal average extraction time seen is 0.008 ms when using the <h1>, <h2>, <h3>, and <header> (H) elements. The mean value for the character count of material within these tags is around 173.26. The observed value is considerably lower than the mean number of characters exhibited by alternative tags. The use of these tags facilitates expedited extraction, as anticipated, due to their reduced character count. As the quantity of characters within the HTML elements, such as <p>, <span>, and <ul>, grows, there is a noticeable decrease in the speed of extraction. The aforementioned tags have an average extraction time of 0.028 milliseconds, while the average character count inside these tags amounts to 173.26. The findings presented in this study demonstrate the relationship between the duration of extraction and the quantity of characters. Nevertheless, the HTML tags main>, field>, section>, table> (MFST), and the div> tag do not provide the anticipated results. The mean duration for extracting the MFST tags is around 0.041 milliseconds, while the average character count inside the content of these tags is 17391.97. The character count of the material within the <div> element is lower than the character count of the content within the MFST tags. Nevertheless, the mean duration for extracting the <div> tag is around 0.081 milliseconds, indicating a slower extraction rate compared to the average extraction time of the MFST. Due to its sluggishness, a <div> element has the capability to encompass several additional <div> elements. The aforementioned tag can be characterized as a nested structure.

Multiple coauthors were contained in a single string; however, they were separated into a list of coauthors and their affiliations, as shown in Table 4. Coauthor is a person who has made a significant contribution to a journal article. Affiliation is the institution where research is conducted.

Table 4. Extracted coauthors with the affiliation

Coauthors	Affiliation
Nihar Shah	UC Berkeley
Dengyong Zhou	MSR
Brendan van Rooyen	NICTA
Aditya Menon	NICTA
Robert Williamson	NICTA
Avrim Blum	NICTA
Julian Yarkony	NICTA
Jonathan Bassen	NICTA
Jonathan Vacher	NICTA

Table 5. Frequent sequential accessed pattern for articles from 2015 – 2018

Articles from NIPS	Term Frequencies	Support
Neural network	22	11
Deep learning	18	9
Reinforcement learning	14	7
Monte Carlo	4	2
Variational inference	4	2

From Table 5, it was seen that within the corpus of examined literature, the concepts of neural network and deep learning were found to be of significant importance, with respective frequencies of occurrence amounting to 22 and 18 instances. Furthermore, it was determined that there was a notable demand for papers pertaining to Reinforcement learning, as evidenced by its occurrence 14 times.

In Fig 3, the graph depicting the relationship between articles and coauthors may be observed. An incoming edge represents each coauthor of an article. A coauthor possesses several outgoing edges if they have made contributions to many works. The graph generated has various linked components, illustrating the intricate network of relationships among coauthors, which may encompass numerous connections.

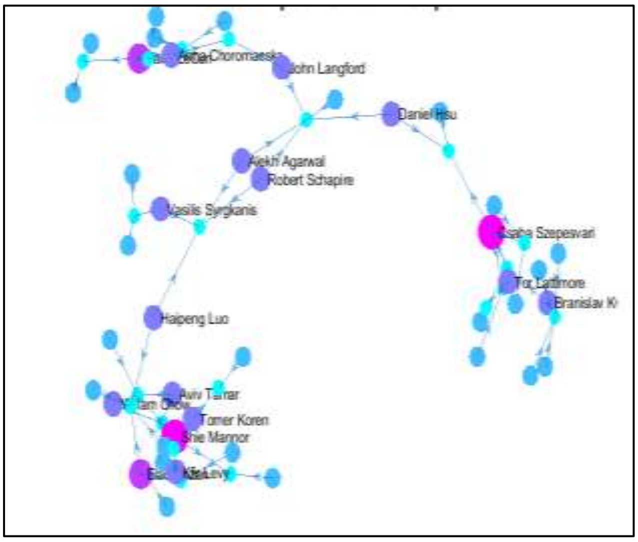

Fig. 3 Papers – Coauthors graph

Figure 4 presents the word cloud created to identify the common subjects among the publications within the largest linked component. In addition to gathering data on the frequencies of bigrams, we employed a WordCloud visualization technique to depict the most commonly occurring bigrams in our text. To do this, we replaced spaces with underscores. Remarkably, the term neural network emerged as the most often occurring bigram in the analyzed publications, appearing a total of 14 times. This observation underscores the increasing significance placed on this particular field within the realm of computers.

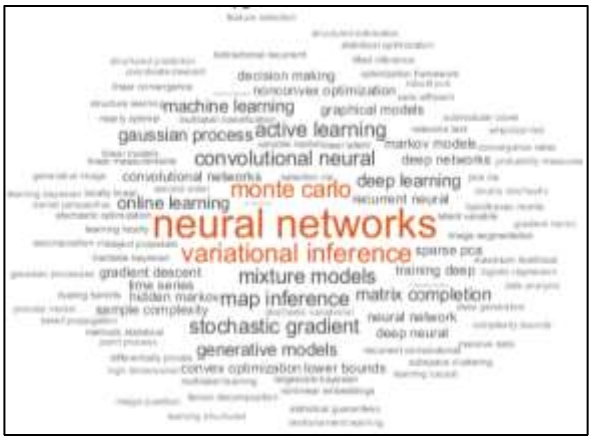

Fig. 4 Papers - Word cloud on titles

5. Comparison Analysis

PrefixSpan algorithm uses a projected prefix database. The PrefixSpan algorithm uses less memory in comparison to the Generalized Sequential Pattern (GSP). The PrefixSpan algorithm has more performance than the Markov model and GSP algorithm, with an execution time of 2.35 seconds, as shown in Table 6. It is clear that the GSP algorithm is efficient. On the other hand, the PrefixSpan Algorithm performs better in terms of space usage and execution time.

Table 6. Comparison with existing system

Model	Number of sessions	Execution Time (Sec)	Memory Usage (MB)
PrefixSpan	2650	2.35	20
Markov model and GSP algorithm	1446	62	170

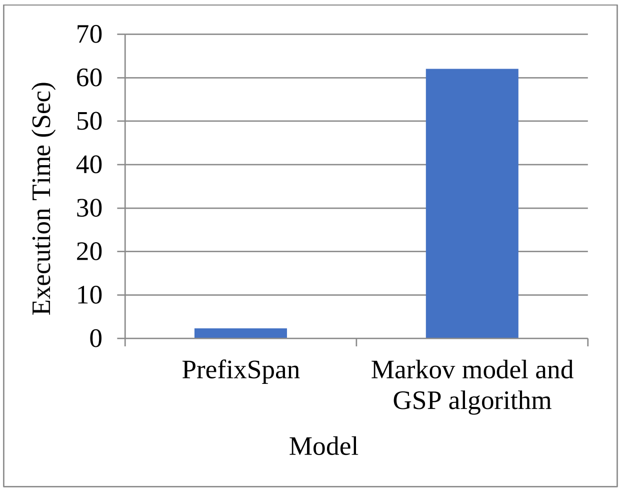

Fig. 5 Graphical Representation of Execution Time of PrefixSpam and Markov Model + GSP

6. Conclusion

The method presented in this study addresses the challenge of extracting valuable insights from vast quantities of web log data obtained from various web pages. This paper aims to demonstrate the applicability of frequent sequential pattern discovery tasks on web log data for the purpose of extracting valuable insights into user behaviour. In order to enhance mining efficiency, the algorithm under consideration adopts a strategy of identifying common prefix patterns rather than suffix patterns.

The majority of prior research on web mining has focused on the automated extraction of web material, sometimes disregarding considerations of time efficiency. Traditional extraction methods rely on a Document Object Model (DOM) tree, which encompasses all items present on a given web page. Nevertheless, this particular strategy results in an augmented time expenditure during the extraction phase. PrefixSpan examines only the prefix subsequences and projects only their corresponding postfix subsequences into projected databases. This way, sequential patterns are grown in each projected database by exploring only local frequent sequences. The prefixspan algorithm generates frequently accessed web pages sequenced. The comparative analysis of mining time in different extraction patterns has been demonstrated using H tag, div tag, span tag, p tag, ul tag, MFST, innerTag count, starting position, and repetition. The Matlab programming language has been used for extracting valuable data from the internet, including user logs and content. The utilization of the proposed method has been implemented in order to optimize both efficiency and accuracy in the process.

In comparison with another system, the PrefixSpan algorithm has used less memory in comparison to GSP. The PrefixSpan algorithm has more performance than the Markov model and GSP algorithm, with an execution time of 2.35 seconds. The GSP algorithm is efficient. However, the PrefixSpan Algorithm is more efficient with respect to running time and space utilization.

This system presents a method that enhances time efficiency by using supplementary information extracted from web pages inside a website throughout the data mining process. The system is introduced in conjunction with sequential pattern analysis.

References

[1] S.K. Pal, V. Talwar, and P. Mitra, “Web Mining in Soft Computing Framework: Relevance, State of The Art and Future Directions,” *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1163-1177, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[2] Runa Bhaumik, Robin Burke, and Bamshad Mobasher, “Effectiveness of Crawling Attacks Against Web-Based Recommender Systems,” *Proceedings of the 5th Workshop on Intelligent Techniques for Web Personalization (ITWP-07)*, pp. 1-10, 2007. [Google Scholar] [Publisher Link]

[3] R. Agrawal, and R. Srikant, “Mining Sequential Patterns,” *Proceedings of the Eleventh International Conference on Data Engineering*, pp. 3-14, 1995. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sravya Vangala, “*Mining High Utility Sequential Patterns from Uncertain Web Access Sequences Using The PL-WAP*,” *Electronic Theses and Dissertations*, 2017. [Google Scholar] [Publisher Link]

[5] Pavel Berkhin, *A Survey of Clustering Data Mining Techniques*, Grouping Multidimensional Data, Springer, Berlin, Heidelberg, pp. 25-71, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[6] Bite Yang et al., “Biren: Predicting Enhancers with A Deep-Learning-Based Model Using the DNA Sequence Alone,” *Bioinformatics*, vol. 33, no. 13, pp. 1930-1936, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[7] Kun Ming Kenneth Lim et al., “@Minter: Automated Text Mining of Microbial Interactions,” *Bioinformatics*, vol. 32, no. 19, pp. 2981-2987, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[8] Laxmi Choudhary, and Shashank Swami, “Exploring the Landscape of Web Data Mining: An In-Depth Research Analysis,” *Current Journal of Applied Science and Technology*, vol. 42, no. 24, pp. 32-42, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mohammed J. Zaki, “SPADE: An Efficient Algorithm for Mining Frequent Sequences,” *Machine Learning*, vol. 42, pp. 31-60, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[10] Jiawei Han et al., “FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining,” *KDD '00: Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 355-359, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Jian Pei et al., “Prefixspan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth,” *Proceedings 17th International Conference on Data Engineering*, Heidelberg, Germany, pp. 215-224, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[12] C. Martins-Antunes, and AL Oliveira, “Sequential Pattern Mining Algorithms: Trade-Offs Between Speed and Memory,” *PKDD’04 Workshop on Mining Graphs, Trees and Sequences*, 2004. [Google Scholar]

[13] Zhenglu Yang, and M. Kitsuregawa, “LAPIN-SPAM: An Improved Algorithm for Mining Sequential Pattern”, *21st International Conference on Data Engineering Workshops (ICDEW'05)*, Tokyo, Japan, pp. 1222-1222, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[14] Maged El-Sayed, Carolina Ruiz, and Elke A. Rundensteiner, “FS-Miner: Efficient and Incremental Mining of Frequent Sequence Patterns in Web Logs,” *WIDM '04: Proceedings of the 6th annual ACM international workshop on Web information and data management*, pp. 128-135, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[15] Ashoka Savasere, Edward Omiecinski, and Shamkant B. Navathe, “An Efficient Algorithm for Mining Association Rules in Large Databases”, *VLDB '95: Proceedings of the 21th International Conference on Very Large Data Bases*, pp. 432-444, 1995. [Google Scholar] [Publisher Link]

[16] D. Tanasa, “Web Usage Mining: Contributions to Intersites Logs Preprocessing and Sequential Pattern, Extraction with Low Support,” PhD Thesis, Université De Nice Sophia-Antipolis, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[17] Florent Masseglia, Maguelonne Teisseire, and Pascal Poncelet, *Sequential Pattern Mining: A Survey on Issues and Approaches*, Encyclopedia of Data Warehousing and Mining, pp. 1-5, 2005. [CrossRef] [Publisher Link]

[18] Johan Huysmans, Bart Baesens, and Jan Vanthienen, “Web Usage Mining: A Practical Study,” *Twelfth Conference on Knowledge Acquisition and Management (KAM2004)*, Kule, Poland, 2004. [Google Scholar] [Publisher Link]

[19] Jaideep Srivastava et al., “Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data,” *ACM SIGKDD Explorations Newsletter*, vol. 1, no. 2, pp. 12-23, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Bettina Berendt et al., *The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis*, WEBKDD 2002 - Mining Web Data for Discovering Usage Patterns and Profiles, Springer, Berlin, Heidelberg, pp. 159-179, 2003. [CrossRef] [Google Scholar] [Publisher Link]