

MODELLING WEBSITE USER BEHAVIOR FROM WEB ACCESS LOGS

Ganihachchi Pathirannehelage Don Madhuka Udantha

(148016F)



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Dissertation submitted in partial fulfillment of the requirements for the degree Master
of Science

Department of Computer Science & Engineering

University of Moratuwa
Sri Lanka

March 2016

DECLARATION

“I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date:

Name: G P D M Udantha



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The above candidate has carried out research for the Masters Dissertation under my supervision.

Signature of the supervisor:

Date:

Name of the supervisor: Dr. Surangika Ranathunga

Signature of the co-supervisor:

Date:

Name of the supervisor: Prof. Gihan Dias

ABSTRACT

Mining web access log data is a popular technique to identify frequent access patterns of website users. Web logs can provide a wealth of information on the user access patterns of the corresponding website, if and when they are properly analyzed. However, finding interesting patterns hidden in the low-level log data is non-trivial due to large log volumes, and the distribution of the log files in cluster environments.

Existing clustering techniques have not focused on identifying infrequent patterns and most of the clustering techniques suffer from cluster parameter issues, when it comes to web usage mining. This thesis presents the application of Density Based Spatial Clustering of Applications with Noise (DBSCAN) and Expectation Maximization (EM) algorithms in an iterative manner for clustering, which is not a technique that has been used before. Each cluster corresponds to one or more web user activities. For clusters that did not have a unique access pattern, frequent pattern mining and sequence pattern mining techniques were used to identify the unique user access patterns.

Secondly, this thesis solves another issue in web usage mining – detecting slight changes between web user access sessions. This thesis introduces a method to identify these access patterns at a much lower level than what is provided by traditional clustering techniques, such as nearest neighbor based techniques and classification techniques. This technique makes use of the concept of episodes to represent web sessions. These episodes are expressed in the form of regular expressions. To the best of our knowledge, this is the first time that the concept of regular expressions are applied to identify user access patterns in web server log data.

We demonstrate that the implemented system is capable of not only identifying common user behaviors, but also in identify anomalous user behavior.

ACKNOWLEDGEMENTS

I would like to dedicate my sincere thanks to my supervisors Dr. Surangika Ranathunga and Prof. Gihan Dias for their dedicated support for the success of this research. This would not have been a success without your support from the initial stage to the final phase of the research.

This research was supported by the LK Domain Registry, Sri Lanka. I thank our colleagues from the Research Division of LK Domain Registry who provided insight and expertise that greatly assisted the research.

I would like to thank the entire academic and non-academic staff of the Department of Computer Science and Engineering for their kindness extended to me in every aspect.

Last but not least, I thank my parents, my wife and all my friends who supported me for the success of this piece of work. Your support was very precious.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

TABLE OF CONTENTS

DECLARATION	i
ABSTRACT.....	ii
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS.....	iv
LIST OF FIGURES	vii
LIST OF TABLES	ix
LIST OF ABBREVIATIONS	x
LIST OF APPENDICES	xi
1. INTRODUCTION	1
1.1. Motivation	2
1.2. Objectives	3
1.3. Contributions.....	4
1.3.1. Refereed Articles.....	5
1.4. Organization of the Thesis	5
2. LITERATURE SURVEY.....	7
2.1. Overview	7
2.2. Web Mining.....	7
2.3. Web Usage Mining and Applications	8
2.4. Data Preprocessing in Web Usage Mining.....	11
2.4.1. Data Selection	12
2.4.2. Web Log Data Cleaning.....	12
2.4.3. User Identification and Session Identification	14
2.5. Pattern Discovery	15
2.5.1. Association Rule Mining	16
2.5.2. Clustering	18
2.5.3. Sequential Patterns Discovery	21
2.5.4. Statistical Analysis for Web Usage Mining	24

2.6.	Anomaly Detection Techniques	24
2.6.1.	Anomaly Detection using Clustering Techniques	25
2.6.2.	Anomaly Detection using Classification Techniques	26
2.6.3.	Statistical Anomaly Detection Techniques	26
2.6.4.	Anomaly Detection using Association Rule Mining	27
2.6.5.	Other Techniques for Anomaly Detection	27
2.7.	Using Episodes for Web Usage Mining	27
2.8.	Suffix Array to Locate the Substring Pattern	30
2.9.	Regular Expressions	31
2.9.1.	Regular Expression Engines	31
2.10.	Discussion.....	32
3.	USING HYBRID CLUSTERING TO IDENTIFY WEBSITE USER ACCESS PATTERNS	33
3.1.	Overview	33
3.2.	Terminology and the Data Model.....	34
3.3.	Preprocessing Engine.....	36
3.3.1.	Implementation of Preprocessing Engine.....	36
3.4.	Hybrid Clustering for Web Usage Mining	39
3.4.1.	Justification for EM+DBSCAN.....	39
3.4.2.	EM+DBSCAN Algorithm Implementation	41
3.5.	The Signature Module	42
4.	EPISODE BASED APPROACH.....	44
4.1.	Overview	44
4.2.	Detecting Slight Changes	44
4.2.1.	Data Models for Detecting Slight Changes.....	44
4.2.2.	Slight Changes between Web User Sessions	46
4.2.3.	Design	46
4.3.	Episode	47
4.4.	Regular Expressions to Represent Episodes.....	49
4.5.	Regular Expression-Based Episode Representation.....	53
4.6.	Episode Clustering	53
4.7.	Summary	54
5.	EVALUATION AND DEMONSTRATION	55

5.1.	Overview	55
5.2.	Data Set for Evaluation	55
5.3.	Evaluation of the EM+DBSCAN Approach	56
5.3.1.	Evaluating Clustering Algorithms	56
5.3.2.	Evaluating Cluster Signature Uniqueness	58
5.3.3.	Evaluation of Effects of Temporal Website Changes	60
5.3.4.	Demonstrating Social Media Impact on Site Access	63
5.3.5.	Attack Detection.....	64
5.4.	Evaluation of the Episode based approach.....	65
5.4.1.	Improving Clustering with Episodes.....	65
5.4.2.	Evaluation of Memory Usage	68
5.4.3.	Identifying Attacks on a Website.....	69
5.4.4.	Common User Patterns	71
6.	Discussion.....	73
6.1.	Contributions	73
6.2.	Usability	74
6.3.	Scalability	74
6.4.	Accuracy	75
7.	CONCLUSION & FUTURE WORK.....	76
	APPENDIX A: SOURCE CODE	86



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF FIGURES

Figure 1.1: Web mining categories	1
Figure 2.1: General Architecture for web usage mining (source [15])	9
Figure 2.2: The web usage mining steps (source [29])	11
Figure 2.3: Data preprocessing phases	11
Figure 2.4: Pattern discovery techniques and methods	16
Figure 2.5: Categorizing patterns for sequential mining algorithms.....	22
Figure 2.6: Sample query of WEBMINER	23
Figure.2.7: Clustering based anomaly detection techniques	25
Figure 2.8: Preprocessing steps with episode identification (source [29])	28
Figure 2.9: Detailed web usage mining process (source [1]).....	29
Figure 2.10: Two types of episodes (Source [29]).....	30
Figure 3.1: Hybrid clustering algorithm approach	33
Figure 3.2: Episode based approach.....	33
Figure 3.3: Log record in access log file	35
Figure 3.4: Functionality of the data preprocessing engine	36
Figure 3.5: sample access log file	37
Figure 3.6: Session file and Mapping file	38
Figure 3.7: Component architecture of the data preprocessing engine	39
Figure 3.8: EM +DBSCAN algorithm	41
Figure 3.9: Cluster matrix with session numbers and page occurrences	43
Figure 3.10: Cluster signature module	43
Figure 4.1: Data model types in web usage mining	45
Figure 4.2: An example of a slight change in web session	46
Figure 4.3: System architecture.....	47
Figure 4.4: Episode structures.....	49
Figure 4.5: Regular expressions generator	50
Figure 4.6: Sample session with page sequence	53
Figure 4.7: Session with episode representation	53
Figure 5.1: Evaluating cluster mechanisms and EM+DBSCAN for website N	58
Figure 5.2: Evaluating cluster mechanisms and EM+DBSCAN for website U	58
Figure 5.3: Evaluating cluster mechanisms and EM+DBSCAN for website F	58

Figure 5.4: Effect of training session addition to the non-profit organization site	60
Figure 5.5: Detecting major changes in the website using EM+DBSCAN clustering	61
Figure 5.6: Detecting major changes in the website using DBSCAN	61
Figure 5.7: Detecting major changes in the website using k-means with domain expert.....	62
Figure 5.8: Detecting major changes in the website using EM.....	62
Figure 5.9: Detecting major changes in the website using K-means with a cluster count of 15	63
Figure 5.10: Impact of social media on the user behavior model	64
Figure 5.11: Generated new cluster that represents the sessions of an attack.....	65
Figure 5.12: Comparing completeness of clustering algorithms with episodes.....	66
Figure 5.13: Intra-cluster distance of clusters (website N)	67
Figure 5.14: Nearest-cluster distance of clusters (website N).....	68
Figure 5.15: Suffix array length growth for the two suffix array versions in website N	69
Figure 5.16: Suffix array length growth for the two suffix array versions in websites U (left) and F (right).....	69
Figure 5.17: (a) Normal user pattern (b) Attacker pattern	70
Figure 5.18: Slight change in cluster	70
Figure 5.19: Sample web session attack	71
Figure 5.20: Search user access pattern	71
Figure 5.21: Article readers access patterns in website N	72



LIST OF TABLES

Table 2.1: Web log preprocessing techniques and algorithms.....	13
Table 3.1: Factors that affect user navigation in a website	40
Table 4.1: Suffix Array on a sample user session.....	51
Table 4.2: Sorted Suffix Array.....	51
Table 4.3: n-grams of the user session.....	51
Table 4.4: Sorted suffix array from n-gram	52
Table 5.1: Dataset for evaluation	56
Table 5.2: User behavior model count by domain experts and the system.....	57
Table 5.3: Cluster distribution and signature uniqueness	59



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF ABBREVIATIONS

API	Application Programme Interface
CEP	Complex event processing
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer protocol
HTTPS	HTTP over TLS
KDD	Knowledge Discovery and Data Mining
W3C	World Wide Web Consortium
WUM	Web Usage Mining



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

LIST OF APPENDICES

Appendix	Description	Page
Appendix - A	Source Code	86



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

1. INTRODUCTION

This thesis focuses on modelling user access behavior from web access logs. Adding websites to the World Wide Web (WWW) has drastically increased in the last decade, along with web usage. Websites keep improving, and incorporating many features to attract more users. As the size and complexity of a website increases, the simple statistics provided by existing web log analysis tools are inadequate in providing meaningful insight into how the website is being used [1, 2].

In a website, web access logs record user navigations and other activities such as searching, site registering and editing web pages. Access logs are semi-structured files. Web access log files contain millions of log records. Log records consist of the user Internet Protocol (IP) address, HTTP status code, timestamp, web request URI, user agent, HTTP referrer and HTTP request type. The log records are distributed and occupy a large volume.

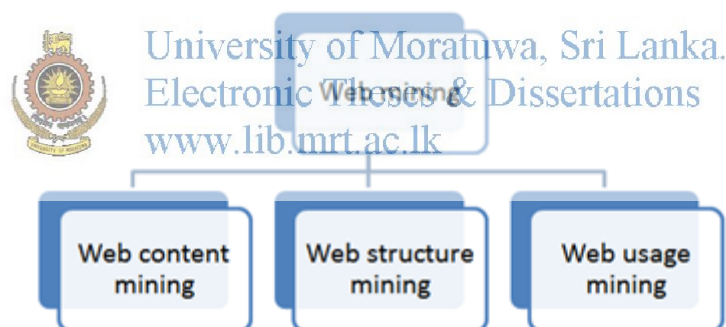


Figure 1.1: Web mining categories

The low level data included in web logs contain valuable information about the users of the system. Web mining is the application of data mining techniques used to discover patterns from the websites users [3, 4]. Web mining can be divided into three different types as shown in Figure 1.1: web content mining, web structure mining, and web usage mining [5]. Web content mining extracts information and knowledge from web page content. Web structure mining is the process of analyzing the structure of a website. Web usage mining discovers interesting usage patterns from web data in order to understand and better serve the needs of web-based applications. Usage data capture the identity or origin of web users along with their browsing behavior at a website.

Web usage mining consists of three phases; preprocessing, pattern discovery and pattern analysis. The preprocessing phase includes web log data cleaning, web user identification and user session identification. The pattern discovery recognizes the access patterns by applying data mining techniques such as path analysis, association rule mining, clustering and classification. The final stage of web usage mining using the pattern analysis phase is to analyse the patterns found during the pattern discovery step [1, 4].

The recent years have seen the flourishing of research in the area of web usage mining. Published papers on web usage mining exceed the 150 mark, showing a dramatic increase in this area since the year 2000 [5]. Consolidated statistical analysis techniques are exploited in most of the commercial applications of web usage mining. In contrast, research in this area is mainly focused on the development of knowledge discovery techniques for the analysis of web usage data. Association rules, sequential patterns, neural networks, fuzzy logic, genetic algorithms and clustering are common techniques in web usage mining. Among these, clustering techniques are the most popular. Clustering techniques look for groups of similar items among a large amount of data based on a general idea of a distance function, which computes the similarity between groups.

Interesting information about user navigation patterns are generated from web usage mining. This information can be exploited later to improve website interactions from the user's viewpoint. The results produced by the mining of web logs can be used for various purposes [7]: to personalize the delivery of web content, to improve user navigation through pre-fetching and caching, to improve web design in e-commerce sites, and to improve customer satisfaction [5].

1.1. Motivation

Even though web log files contain valuable information, it is hidden among the low-level log data. Retrieving meaningful information from web log data is difficult since logs are semi structured with text and numbers and have large volumes of data.

The high-level information reveals interesting patterns of user behaviours. A method to extract this valuable high level information would help detect anomalies and help website administration on Human Computer Interaction (HCI) improvements [8].

In web usage mining, there are many methodologies that can be used to identify user access patterns. These include clustering, Apriori algorithms, web access pattern tree (WAP-tree)

and mining frequent patterns [1, 3, 7, 8, 9]. These methods group the same access patterns into one group. However, it is difficult to determine how each session deviates from other sessions within the same group or cluster.

Anomaly detection solves the problem of finding patterns in data that do not conform to normal behavior. These non-conforming patterns are often referred to as anomalies, discordant observations or exceptions. Anomalies might be induced in the data for a variety of reasons such as malicious activity, fraud, cyber-intrusion or the breakdown of a system [12]. It is important to find the slight differences between sessions. These seemingly insignificant changes could be the most important for a domain expert as they may resemble anomalies.

Clustering techniques are used to discover clusters and are not optimized to find anomalies. These techniques do not detect slight changes. Anomalies can also reside inside a cluster.

One of the limitations in k-means [13] and other clustering mechanisms is that they remove noise from the data [14]. Here, important data or patterns might be lost. For example, an attacker's access sequence might be discarded as noise since it may be considered as an outlier. When data contain noise, the completeness of the system lowers due to data being lost as noise. One technique to identify these anomalies is by only considering outliers, but this will not work if the pattern is inside a cluster. If inside a cluster, the slight difference between the attacker's access pattern and normal user access pattern is not obvious. Anomaly detection systems are optimized to find anomalous behaviour in users but not the common patterns. Web usage mining is more focused on categorizing the common user behaviours. If both functions can be achieved by the same system, it will be useful for security experts, web administrators, web user interface developers and marketing strategy designers to take better decisions.

1.2. Objectives

Improving web usage mining has many aspects. A system that addresses all these different aspects goes well beyond the scope of a single thesis. Therefore, the focus of this thesis is only on modelling website user access behaviour from web access logs.

The first objective of this research is to design a system that is able to group similar access behaviours and detect anomalies from web access logs. The designed system should be able

to work with different domain websites, for example, educational websites, media websites and e-commerce websites, without causing any conflict in the system execution.

The second objective is to find a unique signature for similar access pattern clusters. The uniqueness of a signature is measured using its coverage and discrimination. Signature coverage represents occurrences of the signature in clusters, while discrimination indicates the absence of the occurrence of the signature patterns in the rest of the clusters.

The third objective is to develop a mechanism to identify interesting slight changes in user access patterns that may reside inside or outside a cluster. In essence, this will be a solution to the problem of detecting anomalies and clustering common user behaviours using a single system.

No solution is comprehensive until it has been implemented and tested. Therefore, the final objective of this thesis is to implement a system to model user behavior in websites using access logs.

1.3. Contributions

The main contributions of this thesis are as follows:

- This thesis presents a model to detect website user access patterns from web access logs. As described in chapters 3 and 4, this system clusters the common access patterns in web session data. The system identifies common patterns that group the sessions into the same clusters. The system outputs the sub-sequences of access patterns that group the sessions into a single cluster and the access patterns that differentiate two clusters.
- This thesis implements an episode based approach for web session mining [1]. Researchers have theoretically presented episode based web usage mining upto now. However no one has implemented and evaluated episode based web usage mining. Chapter 5 demonstrates results obtained using the episode based approach and show that episode based web usage mining outperforms normal web usage mining.
- This thesis introduces a novel technique, the application of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and Expectation Maximization (EM) algorithms in an iterative manner for clustering web user sessions. EM+DBSCAN overcomes the limitations in clustering algorithms such as inserting

parameters cluster count and minimum count for a density region. EM+DBSCAN is used to identify user access patterns in web logs.

- This thesis presents the concept of using regular expressions to represent user access patterns. Regular expression is a technique very commonly used for similar tasks in domains such as genetic algorithms and text searching. It is used for the first time in this study for web session mining. Regular expressions are used to represent web session episodes.
- This thesis demonstrates the improvement of the web log preprocessing stage. Most of the preprocessing engines disregard the sequential information in web sessions as they only need to pass integers for pattern discovery in web usage mining. When an episode is represented as a regular expression, sequential information of web sessions is preserved.

1.3.1. Refereed Articles

This research so far has produced the following publications:

Published:



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations

- Madhuka Udantha, Surangika Ranathunga and Gihan Dias, "An Episode-based Approach to Identify Website User Access Patterns", Proceedings of the International Conference on Pattern Recognition Applications and Methods (ICPRAM 2016), Rome, Italy, 24-26 February, 2016, 343-350.
- Madhuka Udantha, Surangika Ranathunga and Gihan Dias. "Modelling Website User Behaviours by Combining the EM and DBSCAN Algorithms." Moratuwa Engineering Research Conference, 2016.

1.4. Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 reviews related work regarding web log preprocessing, web session clustering, web usage mining, regular expressions, episodes (sub sequences) and anomaly detection. It also discusses existing web usage mining applications and their features.

Chapter 3 explains the approach and the design of the proposed system that addresses the limitations of existing clustering techniques.

Chapter 4 presents the implementation of the episode based web usage mining system with regular expressions. It answers the problems pointed out in the literature review.

Chapter 5 demonstrates and evaluates the work presented in the thesis. It contains main subsections for evaluating the EM+DBSCAN approach and evaluating episode based approach.

Chapter 6 provides a discussion on the thesis contributions, and possible research improvements respective to usability and scalability.

Chapter 7 concludes the thesis with future work.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

2. LITERATURE SURVEY

2.1. Overview

This chapter provides the concepts and techniques of web usage mining, data mining and pattern recognition. It also provides a comprehensive explanation of what motivated this research.

Section 2.2 provides an overview of web mining in general. A discussion on the history and the present state of web usage mining techniques is followed in section 2.3. Research motivation facts and web usage mining applications are also explained in this section. There are many web usage mining techniques and this section carries a comprehensive description on each technique.

In web mining, the data preprocessing phase is stated first, and section 2.4 explains web log cleaning and log data preprocessing steps in web mining in the present research. It contains web log data cleaning, user detection models and web session detection methods. Data transformation phases and data structures for handling web session data are also discussed.

The next section (section 2.5) provides the current research status in pattern discovery in web usage mining. There are four subtopics where each existing pattern discovery method is discussed. Detecting slight changes in web sessions is also important in web usage mining as they can be interesting to security experts and web administrators since they can be web or network attack sessions. Anomaly detection techniques are also reviewed in section 2.6.

Section 2.7 and section 2.9 discuss theoretical mechanisms related to web usage mining with episodes, and regular expressions, respectively.

2.2. Web Mining

Web mining contains a widespread range of applications aimed at discovering and extracting information from web pages and other documents found on the web. The web mining process uses data mining techniques and algorithms for detecting patterns in web data in order to gain insight into trends and web users. Providing a mechanism to make the data access more efficient is another objective in web mining. Discovering hidden patterns and the information on user activities is also an important task in web mining. As explained in Figure 1.1, web

mining has three categories; web content mining, web structure mining and web usage mining.

- Web content mining - This is the process of mining, extracting and integrating useful data from the contents of web pages and web documents. There are two types of techniques that are used in this discipline, which have two different points of view. In information retrieval (IR), unstructured data and semi-structured data are represented as a vector space model, bag of words or a statistical model. Information retrieval involves retrieving a single word from the bag of words or training corpus. From a database perspective, information retrieval involves transforming a website into a database in order to have better information management and querying on the web.
- Web structure mining - Generating the structure of a website by analyzing the nodes and connections in the website is web structure mining. It is achieved by extracting patterns from hyperlinks in the website or mining the web document structure. Two things can be obtained from this; the structure of a website in terms of connection links to other sites, and the document structure inside the site.
- Web usage mining - This is a method to extract patterns on user navigation. It processes the web log files to detect those patterns, which gives insight on user activity including where the users access. This information is regularly gathered automatically into access logs via the web server. Different levels of logs are used in the analysis to give more value and semantic meaning for user behavior.

2.3. Web Usage Mining and Applications

In this thesis, we focus only on web usage mining, which is a subpart of web mining as explained in section 2.2. Web usage mining is the process of discovering and analyzing patterns in log files, which are generated on a daily basis. The discovered user access patterns represent collections of web pages or resources that are normally accessed by groups of users with common necessities or interests. Figure 2.1 shows the general architecture of a web usage mining system. The data preprocessing phase extends from data cleaning to data transformation. Pattern discovery is centered around the architecture as it is a fundamental component in the system [15]. Pattern analysis is located at the end of the process.

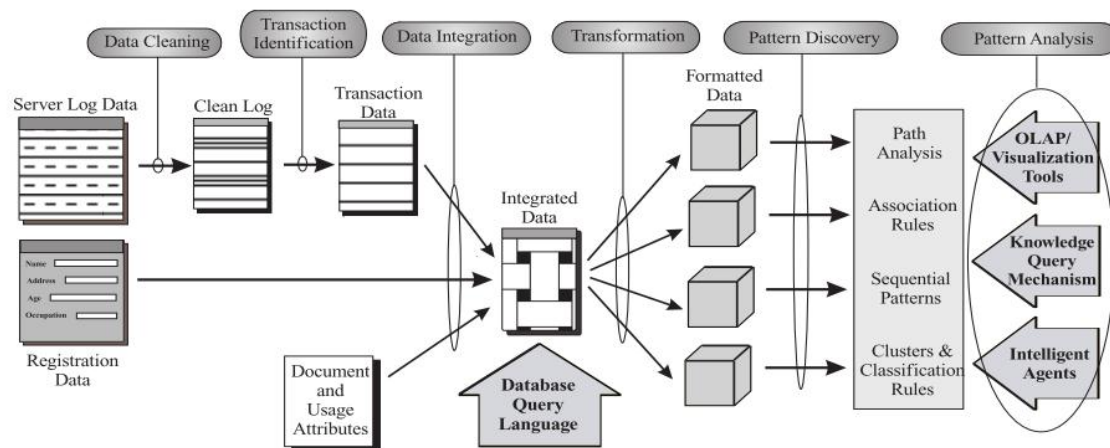


Figure 2.1: General Architecture for web usage mining (source [15])

Usage mining can be used for system improvement, site modification, network security, business intelligence and personalization. Companies understand customers' trends and needs at the right time through web usage mining. It helps them to complete promotional campaigns effectively, increase sales, and implement better marketing strategies. It enables web-based systems to provide a better service, have more user-friendly interfaces and provide the best access paths to services.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations

www.hb.mru.ac.lk

By mining web logs, web administrators can do effective site modification by verifying that they cater to the changing user requirements. Web usage mining provides a user behavioral model for the website so that web designers can improve navigations and user interfaces by looking at actual customer needs. PageGather is an algorithm used to group related web pages in the website using web access logs [16]. It applies cluster mining to log data and produces clusters that contain pages and documents, which the web admin can evaluate. However, the fact that PageGather depends on human web master evaluation is a limitation.

A personalized and proactive view of the web services is deployed using the extracted knowledge from web usage mining [17]. By looking over navigational behavior of the user or visitor, the system is able to identify the need or the user's interest and guide them down the correct path. There are tools such as Site Helper and Web Watcher to handle personalized features in websites. User preferences are identified by Site Helper by looking at page accesses. It uses a list of keywords from pages that a user has spent a significant amount of time on. Web Watcher observes user navigational links and marks interesting links [18]. Later, when a new page is added to the system, inheriting similarity is compared between the new page and user. Researchers also try to automate personalization via web usage mining.

The primary components of web personalization are web objects (pages, etc.) and subjects (users), matching and turning models across the objects and subjects, and providing a set of actions to be suggested for particular users [15,16].

Network security is improved by using the concept of web usage mining. Online attacks that occur on the network are detected using web agents, basically web robots, rather than using manual efforts [21]. Security analysis based on web usage mining is conducted to figure out security vulnerabilities of the website [22]. Security events are extracted from the logs by rule matching and statistical analysis, and a sequential pattern of attacks is detected using a sequential mining method called Prefix Span [22]. Later in 2016, researchers have used the log mining algorithm based on clustering to find the frequent attack sequences from log files. Several properties related to network security are considered in the paper [23]. These are the start time of attacking, source IP of attackers, and network protocol. Experiments have shown that detecting security anomalies using the signature-based approach is effective [20,21]. Web usage mining delivers patterns that are useful for detecting high level events such as intrusion, fraud, hacking and cybercrimes.

Web usage mining has become very critical in business intelligence applications and sites [26]. Intelligent miner (called i-Miner) is a hybrid framework for usage mining used to find hidden patterns using log files from web servers. A thorough illustration on the evolution of data mining techniques that improve web usage mining into business intelligence applications was given by Abraham [27]. SurfAid Analytics is an industrial tool from the IBM e-business service. It provides business intelligence for websites. SurfAid identifies visitor attributes such as customer retention, navigation patterns and buying habits. Logs are sent to the IBM SurfAid facility for processing and it runs offline. SurfAid is being used to analyze e-commerce events from click streams, which are site access logs [28]. It provides usage statistics, page view statistics and uses path analysis using the IBM SurfAid processing engine through a data cube and also clusters web users.

As explained above, web usage mining has contributed to many industrial applications in different domains such as business intelligence, network security, personalizing websites and site modification systems and is used for knowledge discovery research as well.

The next few paragraphs explain the components in the web usage mining process shown in Figure 2.1.

2.4. Data Preprocessing in Web Usage Mining

Web usage mining refers to the automatic discovery and analysis of patterns from web data, in order to understand and serve to the needs of web based applications. Web usage mining contains three main steps, preprocessing, pattern discovery, and pattern analysis.

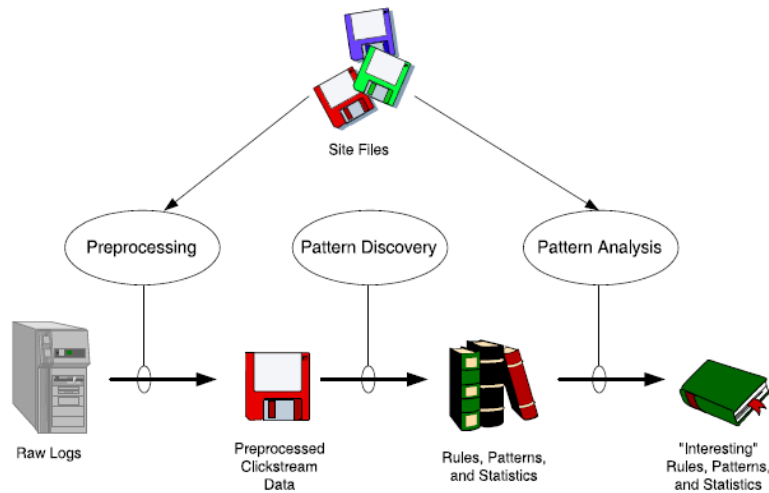


Figure 2.2: The web usage mining steps (source [29])

Data preprocessing is a key step in the data mining process and in web usage mining [30]. Preprocessing steps are critical, as all other processes and steps depend on this phase. The main goal of preprocessing in web usage mining is to transform the raw log data into a set of user sessions or user profiles [30]. The preprocessing phase contains data selection, cleaning and transformation and the identification of user sessions [31] as shown in Figure 2.3.

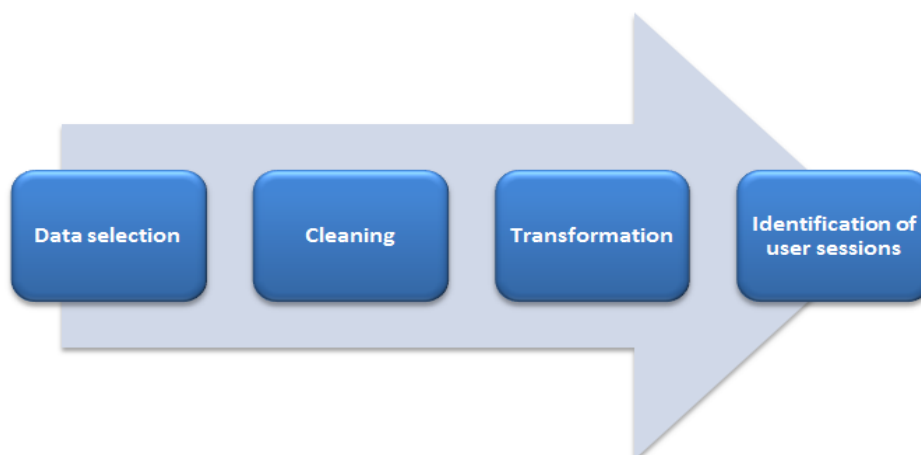


Figure 2.3: Data preprocessing phases

2.4.1. Data Selection

Logs can be collected at different levels; network level, site level, proxy level and client level [5]. Each of these logs presents different information; for instance, network utility and network traffic are given in the network level log. The server level log gives server information. A single user's behavior in a multi-site environment is shown in client side logs. Site level logs contain multiple users in a single site and are known as server logs or web access logs [32]. Access logs are mostly used in web usage mining as they record all users' web requests [33]. Access logs include navigation paths of all users ordered by requested time. Access logs are the most valuable sources to track user navigation paths.

Commonly, access logs are analyzed with static web analysis tools, which output the number of visitors (unique visitors) to pages, requests count for each page and usage statistics report in terms of time of day, day of week, and seasonality [34]. A typical access log format contains the Internet Protocol (IP) address of the client computer, the identity of the user (identity of the user field in access log is not used since it is not reliable), the user name determined by HTTP authentication, the timestamp at which the HTTP request was received, the HTTP request URL or query, the request line from the client, the HTTP status code sent from the server to the client, the size of the response to the client (in bytes) and the user-agent identification string [35]. The access log contains a semi structure with text, timestamp and numerical data. The access log is the most preferred data source to track user behavior high level events as access logs contain low level access data.

2.4.2. Web Log Data Cleaning

Log data cleaning is included in the log data preprocessing module. When a web application is deployed in a cluster architecture, log files are distributed over multiple servers or load balanced in the frontend of the application. Web requests are catered to by different servers located in different locations. The log files are also distributed, and the log records are distributed among those files. The web log data cleaning process merges those logs ordering according to the time sequence. The data cleaning process includes the removal of records of images, videos, scripts, format information, records with a failed HTTP status code and robots cleaning [34, 35].

Table 2.1 compares preprocessing steps and algorithms used in log data preprocessing for web usage mining. User identification and session identification steps are the most common

and the most important in web log preprocessing. Zheng [38] improves data preprocessing technology for web log mining in order to solve some existing problems in traditional data preprocessing technology. His identification strategy based on the referred web page is adopted at the stage of user identification, which is more effective than the traditional one based on website topology. His proposed strategy is now used widely. Data preprocessing algorithm based on collaborative filtering is introduced for web log mining to identify web user sessions [39]. This approach has proven to be fast and flexible judging by the experimental results [2, 37].

Table 2.1: Web log preprocessing techniques and algorithms

Author	Source of Log File	Preprocessing Technique	Applied Algorithm
Yan LI, Boqin FENG and Qinjiao MAO [40]	Access log file (English study website)	Data Cleaning User Identification Session Identification Path Completion Transaction Identification	Maximal Forward References (MFR), Reference Length
Tasawar Hussain, Sohail Asghar and Nayyer Masood [41]	Web server log	Data Cleaning Log File Filtering User Identification Session Identification	None
Doru Tanasa and Brigitte Trousse [42]	INRIA website log file	Data Fusion Data Cleaning Data Structuration Data Summarization	None
Ling Zheng, Hui Gui and Feng Li [38]	ISS server log file	Data Cleaning User Identification Session Identification Path Completion	Based on referred web page and fixed priori threshold

JING Chang-bin and Chen Li [39]	Web server log	Data Preprocessing	Based on Collaborative Filtering
Fang Yuankang and Huang Zhiqiu [43]	Chizhou college website log	Data Filtering Session Identification	Frame page and Page Threshold
Hongzhou Shaa, Tingwen Liub, Peng Qinb, Yong Sunb and Qingyun Liu[44]	Proxy logs	EPLogCleaner (StandardFilter TimeFilter DayStandardLog Filtering By Prefix)	Prefixes algorithm
J. Vellingiri and S. Chenthur Pandian [45]	College website log	Data Cleaning User Identification Session Identification Path Completion Transaction Identification	MFR RL & Time Window
Bhawesh Kumar Thakur, Syed Qmar Abbas, Mohd Rizwan Beg and Sheenu Rizvi[46]	Client side and server side logs	Data cleaning User identification Session identification Formatting	None

2.4.3. User Identification and Session Identification

Identification of individual user sessions is an important step and various methods to identify users are described below [45].

- Using IP address & agent - Assume each unique IP address and agent pair is a unique user. In the access log, IP and agent data are always available. It does not require additional technology [30].

- Using embedded session ID - This is valid for websites that have dynamically generated pages so all the URLs will have unique IDs or request header tokens.
- Using registration - If users explicitly sign-in to the site and unknown access is prohibited, then it only filters the user ID or name [34].
- Using timeout - Another commonly used method when no information exists about leaving the website. A thirty minute timeout is considered a session [5].

For many websites, users do not need to register or sign-in to use or visit the website. Therefore, the first user identification method using the IP address and agent is appropriate if the web usage mining tool is to be tested in many different domain websites. Regarding web session identification, a previous research has evaluated the heuristics to reconstruct sessions from the server log data [47]. The activities were partitioned by users and then by the visits of the users in the site. Algorithm 2.1 is a common session identification algorithm used in a lot of experiments for web log preprocessing and web user profiling, as it is stable [44, 46].

Algorithm 2.1: Session identification heuristic [47]

1. Let $H_i = \{f_1, \dots, f_n\}$ denote a time order session history
2. Let l_j, f_j, r_j , and t_j denote a log entry, referrer, request and time respectively.
3. Let T denote the session timeout
4. Sort data by IP address, agent and time
5. For each unique IP / agent pair do
6. for each l_i do
7. if $(t_j - t_{j-1}) > T \vee r_j \notin \{H_0, \dots, H_m\}$ then
8. Increment i
9. Add l_j to H_i
10. else
11. assign = Distance (H_i, r_j)
12. Add l_i to H_{assign}
13. end;

2.5. Pattern Discovery

A pattern is a feature that occurs repeatedly in sequences, typically more often than expected at random. A pattern in web usage mining is a sequence of web pages that a user has accessed in a web session. Since there are multiple web pages in a website and there are many

navigational paths available, finding semantically meaningful and repeated page access is pattern discovery. The emerging tools and applications for user pattern discovery use sophisticated techniques. Pattern discovery is a main component in web usage mining and the applications of web usage mining are classified into two main categories: learning user profiles or user modeling in adaptive interfaces (personalized) [49], and learning user navigation patterns (impersonalized) or user behavior models [6]. Pattern discovery methods and techniques are affected by this categorization and most research aligns with learning user access patterns rather than user profiling.

Pattern discovery is spread in several fields and develops many methods and algorithms using statistical, data mining, machine learning and pattern recognition techniques [3, 48], as shown in Figure 2.4. This section describes the pattern discovery activities that have been applied to the web usage mining domain. Statistical methods, clustering, classification, dependency modeling and sequential pattern mining are the most popular techniques in pattern discovery in the web usage domain.

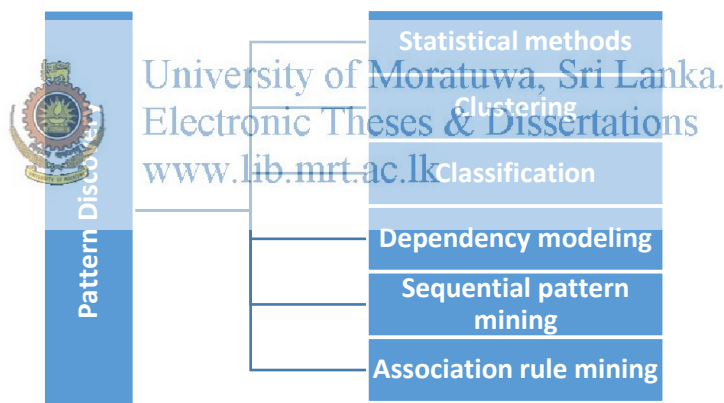


Figure 2.4: Pattern discovery techniques and methods

The next sub sections cover the five pattern discovery techniques, how they are used in pattern discovery, a comparison of the techniques, and the limitations and advantages of each technique.

2.5.1. Association Rule Mining

Association rule mining discovers interesting relations between items in large datasets and captures the relationships between items based on their patterns of occurrence across item sequences or transactions [51]. Association rule mining is a data mining function that

discovers the probability of the co-occurrence of items in a collection. The relationships between co-occurring items are expressed as association rules.

In order to select interesting rules from the set of all possible rules, constraints on various measures of significance and interest are used. The best known constraints are minimum thresholds on support and confidence. Support is defined per item set and gives the proportion of transactions, which contain X. It is used as a measure of significance (importance) of an item set. Since it basically uses the count of transactions, it is often called a frequency constraint. An item set with a support greater than a set minimum support threshold, $\text{sup}(X) > \sigma$, is called a frequent or large item set [51].

Confidence is defined as the probability of seeing the rule's consequent under the condition that the transactions also contain the antecedent. Confidence is directed and gives different values for the rules $X \Rightarrow Y$ and $Y \Rightarrow X$. Association rules have to satisfy a minimum confidence constraint, $\text{conf}(X \Rightarrow Y) \geq \gamma$.

The Apriori algorithm reduces the size of candidate sets. However, it can suffer from two non-trivial costs: (1) generating a huge number of candidate sets, and (2) repeatedly scanning the database and checking the candidates via pattern matching [52].

Researchers have highlighted the limitation in association rule mining and argue that algorithms for association rule mining such as Apriori are not efficient when applied to long sequential patterns, which is an important drawback when working with web logs [9]. Another frequent data mining research group has mentioned that there are still some challenging research issues that need to be solved before frequent pattern mining can claim a cornerstone methodology in data mining applications [52].

2.5.1.1. Association Rule Mining for Web Usage Mining

Association rule mining is used in web usage mining in many occasions. Some recognized algorithms for mining association rules have been improved to extract sequential patterns. For instance, researchers used AprioriAll and GSP, two extensions of the Apriori algorithm, for association rule mining in web usage mining domain [3, 4]. The same authors in a later work presented the GSP algorithm that outperforms AprioriAll by up to 20 times [55]. GSP is efficient when the sequences are not long and the transactions are not large in web sessions.

The Association Rule Hypergraph Partitioning (ARHP) technique efficiently clusters high dimensional data sets without requiring dimensionality reduction as a preprocessing step [3]. ARHP is used to build automatic personalization based on web usage mining [56]. It combines association rule mining and clustering into the ARHP method. First, association rules are used to extract frequent patterns from user sessions; then the frequent patterns are used to build a model.

2.5.2. Clustering

In the context of web usage mining, clustering is used to cluster similar web user sessions to determine general site access behaviors [57]. The task of clustering web sessions is to group web sessions based on similarity and consists of maximizing the intra-group similarity while minimizing the inter-group similarity. From a real world perspective, clustering plays an outstanding role in data mining applications and tools. Clustering has been extensively used in web usage mining to group together similar web user sessions [5].

2.5.2.1. K-Means

The k-means algorithm is an algorithm used to cluster n number of objects based on attributes into k number of partitions, where $k < n$. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. The k-means algorithm is not able to handle the clusters of arbitrary shape. In web usage mining data clusters can take arbitrary shapes as the dataset corresponds to different user access patterns. Therefore, it is hard to achieve better results from only a standalone k-means algorithm.

2.5.2.2. Density-Based Spatial Clustering of Applications with Noise

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is one of the most common clustering algorithms. DBSCAN requires two parameters: ϵ (epsilon, which specifies how close points should be to each other to be considered a part of a cluster) and minPts (the minimum number of points required to form a dense region). DBSCAN does not require the user to specify the number of clusters in the data a priori, as opposed to the k-means algorithm. DBSCAN can find arbitrarily shaped clusters. DBSCAN has a notion of noise, and is robust to outliers [62, 63].

The minPts can be derived from the number of dimensions in the data set, as $\text{minPts} \geq D + 1$. In web session mining, there will be 100 to 200 dimensions and it varies according to the website. Since those dimensions are a representation of web page views, the minPts is less than the dimensions count. When the lower value of minPts is 1, every point on its own is already a cluster and is not a valid value for minPts. When minPts has a value less than or equal to 2, the result is the same as that of hierarchical clustering with the single link metric [58]. Therefore, minPts with a minimum value of 3 must be chosen. Larger values are usually better for data sets with noise. Since web user access patterns have lots of combinations and vary between websites, it is hard to pick a correct value for the minPts. Choosing a larger value of minPts is a good practice in DBSCAN in larger data sets and where web session data sets are always huge.

Every data mining task and clustering algorithm has the problem of parameters. Every parameter influences the algorithm in specific ways [64, 65]. For DBSCAN, the parameters ϵ and minPts are needed. The parameters must be specified before algorithm execution begins and those parameters values are assigned by the user. Preferably, the value of ϵ is given to solve problems such as the physical distance between nodes, and minPts is then the desired minimum cluster size.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

2.5.2.3. Expectation Maximization (EM)

The other clustering approach is the EM algorithm [62]. Unlike DBSCAN, this algorithm works well even if there are large differences in densities. In machine learning, data mining and computer vision, EM is often used for data clustering and used in web usage mining [63].

In EM clustering, each instance is assigned a probability distribution by EM. The given probability distribution indicates the probability of it belonging to each of the clusters. EM has the capability to decide the cluster count by cross validation, where the k-means clustering algorithm is needed for the cluster count beforehand. In EM clustering, the user may specify the cluster count to be generated if the count is known.

The number of clusters is determined by cross validation in EM. Cross validation is also known as rotation estimation. Cross-validation is a way to forecast the fit of a model to a theoretical validation set when a clear validation set is not available. In EM, the number of clusters is set to 1 initially and the training set is split randomly into 10 folds. The EM algorithm is also executed 10 times using the 10 folds as the standard cross validation manner

[64]. Then the log likelihood is averaged over all 10 results. If the log likelihood has increased, the number of clusters is increased by 1 and the program continues splitting the training set again. Since EM can handle large datasets with high dimension counts, the number of folds is fixed to 10. However, the instance count in the training set need not be smaller than 10. If the training set instance count is less than 10, the number of folds is set equal to the number of instances in the training set.

2.5.2.4. Hybrid Clustering

Due to the limitations in these clustering techniques, hybrid clustering is also used. For example, K-SVMMeans algorithms are introduced by combining two clustering algorithms [65]. K-SVMMeans is used to identify distinct clusters of documents in text collections and authorship analysis. K-SVMMeans is a clustering algorithm integrating K-Means clustering with Support Vector Machines. SyMP (Synchronization with Multiple Prototypes) is a clustering approach that identifies clusters of arbitrary shapes in large data sets and uses Gaussian components to determine the optimum number of clusters [66]. The hybrid approach is tried in clustering web access patterns after transforming logs into a fuzzy vector. Those vectors are in a predetermined dimension and determining dimensions are not easy in data mining. Grouping the patterns into a number of clusters is done by learning vector quantization. Secondly, the weighted fuzzy c-means is developed to deal with the results of learning vector quantization.

2.5.2.5. Use of Clustering Mechanisms for Web Usage Mining

Xie and Phoha [67] suggested that the focus of web usage mining should be shifted from single user sessions to groups of user sessions. They were the first to apply clustering to identify clusters of similar sessions, which were produced by web access logs. Cooley et al [15] proposed a taxonomy of web mining and identified further research issues in web usage mining.

The similarity graph in conjunction with the time spent on web pages to estimate group similarity in concept-based clustering was introduced by Banerjee and Ghosh [68]. A Genetic algorithm was used to improve the results of clustering through user feedback [69]. The multi-modal clustering technique was applied to build clusters by using multiple information [70]. Later, a United Kingdom research group presented a methodology for social event detection as a multi-modal clustering task with density based clustering, where they combined

items from social media streams [71]. The research combined association rule mining and clustering into a method called association rule hypergraph partitioning, as explained in section 2.5.1 [3].

There are major limitations in clustering techniques with regard to web usage mining. The number of clusters, the initial data points of the respective clusters, and the criterion function definition are the 3 key points and difficulties that deserve consideration in web session clustering [72]. Researchers have highlighted one of the main problems in clustering with regard to web session mining [73]. In web usage mining or web session mining, the knowledge on the dataset is lower even for domain experts. Therefore, when clustering techniques are used, it is not possible to find values for cluster parameters such as cluster count, initial data points, minimum density for clustering or any other parameters.

Secondly, most of the algorithms presented in the literature dealing with clustering web sessions treat sessions as sets of visited pages within a time period and do not consider the sequence of the click-stream of visitors. This has a significant consequence when comparing similarities between web sessions [14]. The research group pointed out that many clustering techniques applied in web usage mining only consider the page occurrence or page frequency in web sessions and do not consider the page access order. However, web session is a sequence of web pages the user has accessed. By increasing a dimension in web session against time, the sequential nature of data can be preserved. Since the volume of data will increase drastically, it limits the applicability of clustering techniques. Interesting high level events may be missed since clustering techniques disregard the sequence as they used only page occurrences in web sessions.

The two clustering algorithms discussed in section 2.5.2.2 and section 2.5.2.3 do not require one to specify the number of clusters in the data a priori, as opposed to k-means, but DBSCAN requires the minimum number of points required to form a dense region [58]. It is robust to outliers. Therefore, a lot of web usage mining research and applications have used DBSCAN for pattern discovery [63, 64, 65, 66, 67, 68].

2.5.3. Sequential Patterns Discovery

Sequential pattern mining is a topic of data mining concerned with finding interesting patterns in a dataset where the values are delivered in the form of a sequence. In a web user session, pages are recorded in sequences as website users access or request them. Therefore,

sequential pattern mining is used in web usage mining and there are tools and research applications with sequential pattern mining. Sequential mining algorithms are designed to discover patterns appearing in a single sequence, across sequences or common to multiple sequences, as shown in Figure 2.5.

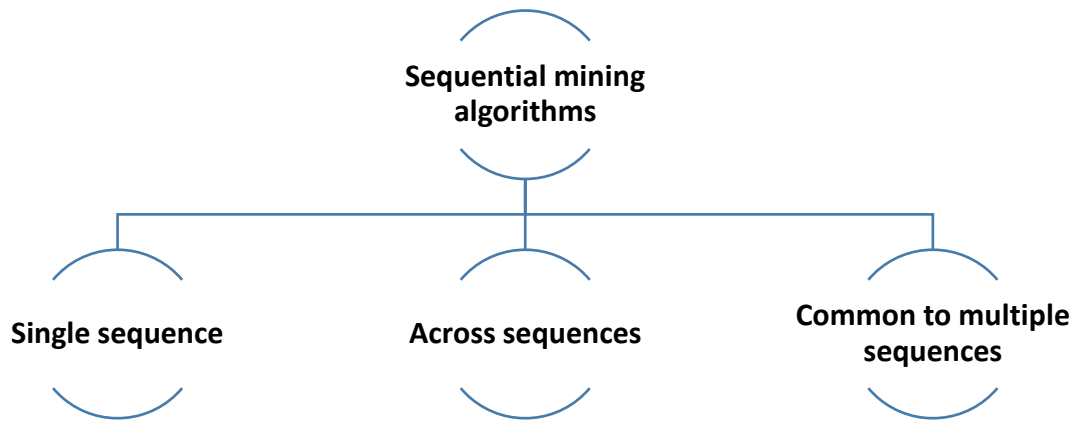


Figure 2.5: Categorizing patterns for sequential mining algorithms

Web session mining belongs to the third category of sequential pattern mining, as there are multiple sessions with a sequence of pages [79]. A web session contains sequences of user access pages and multiple web sessions with common sequences are considered in web usage mining.

The present state-of-the-art sequential pattern mining algorithms rely on a pattern-growth methodology to discover sequential rules. A weakness of this method is that it repeatedly executes an expensive database projection procedure, which declines performance for datasets containing dense or long sequences.

2.5.3.1. Sequential Patterns Discovery for Web Usage Mining

In the early stages of web usage mining, the problem when discovering sequential patterns is to find inter-transactional patterns such a way that the presence of a set of items (pages) is followed by another item (page) in the time-stamp order [1, 53, 77]. Mannila [81] presented a framework that discovers frequent episodes in sequential data. The framework is capable of defining episodes as partial orders, and looking at windows of the sequence. The algorithm finds all episodes from the occurrence count. Episodes have to be predefined in the system and incremental checking is done on the windows of time in sequences. The searching episode being a frequent sub-episode is a must and is a restraint of the system.

Cooley et al. [15] presented a web usage mining application tool called WEBMINE. It automatically discovers association rules and sequential patterns from server access logs. It proposed an SQL like query mechanism for discovering knowledge [15]. The sample query of WEBMINER is shown in Figure 2.6.

```
SELECT association-rule(X*Y*Z)
FROM log.data
WHERE date>=20100310 AND domain = ".lk"
AND support = 50.0 AND confidence = 80.0
```

Figure 2.6: Sample query of WEBMINER

The sample query in Figure 2.6 extracts rules involving the ".lk" domain after March 10th, 2010, that start with URL X, and contain Y and Z in that order, and that have a minimum support of 50% and a minimum confidence of 80%.

Huang [53] implemented the Apriori algorithm to learn association rules and the Apriori-All algorithm to learn sequential patterns from the session file. The application identified interesting rules and patterns in Livelink, which was a collaborative intranet. The system gave some uninteresting rules and the same research group used background knowledge on objects in Livelink to prune out the uninteresting rules.

Nanopoulos et al. [33] introduced a method based on signature tree, by efficiently mining web logs and log data that are stored in databases using proficient pattern queries. The proposed encoding scheme was constructed considering the order among the elements of access sequences. A tree structure called a WAP-tree (Web Access Pattern tree) was presented as an access web page sequence and was devised to access sequences and corresponding counts closely [9]. This structure was twice optimized to exploit the sequence mining algorithm (which was developed by the same authors).

In 2001, the same authors proposed a novel sequential pattern mining method called PrefixSpan (Prefix-projected Sequential pattern mining), where it only explored local frequent patterns. Due to prefix-projection, the size of the database is substantially reduced, which leads to efficient processing [82]. Later, in 2004, the performance improvement of the PrefixSpan algorithm, which was based on bi-level projection, was reported not to be so effective in certain cases [83]. Therefore, they improved the PrefixSpan algorithm and it was integrated with pseudo-projection. Since PrefixSpan-2 explored ordered growth by prefix-

ordered expansion, it resulted in less growth points and reduced projected databases in comparison.

In the same year, Ernestina, et al [84] stored log data in another tree structure, the FBP-tree (Frequent Behavior Paths tree), to improve the sequence pattern by isolating useful sub-sessions within web page access sessions, where each sub-session represented a frequently traversed path indicating high-level user activity. However, patterns were discovered without considering time information. The FBP-tree was used to compute the frequent access path and isolate the sub session in PathSearch-BF. The system constructed a smart access path that is presented to users, assisting them during their navigation in the websites [85]. Currently, they are also working on an extension to this algorithm by incorporating time information in order to develop more sophisticated rules [85].

2.5.4. Statistical Analysis for Web Usage Mining

From an academic perspective, statistical analysis focuses on the following aspects; individual statistics: analysis over an individual website accessed by the user, relative statistics: analysis over collective users for collective sites, and general statistics: analysis of the access pattern by the category. Web usage mining belongs to general statistics as identified by Bommepally et. al. [86].

Chakrabarti [87] carried a survey of data mining for hypertext. He primarily surveyed the statistical techniques for web mining. Another survey on the use of web mining for web personalization highlighted that interesting usage patterns were discovered by extracting statistical information such as diagnostic statistics (for instance, server errors, and “Page Not Found” errors), server statistics (for instance, top pages visited, entry/exit pages, and single access pages), referrers statistics (for instance, top referring sites, search engines, and keywords), user demographics (such as top geographical location, and most active countries/cities/organizations) and client statistics (visitor’s web browser, operating system, and cookies) [48]. Many web analytics tools also provide this level of information.

2.6. Anomaly Detection Techniques

Anomaly detection is a main problem that has been researched within diverse research spaces and industrial application domains in web usage mining [21]. Anomaly detection denotes the identification of patterns in data that do not present the expected behavior. These non-

conforming patterns are often stated as anomalies, exceptions, aberrations, surprises, outliers or discordant observations [12]. Many techniques discussed in section 2.5 are also used in anomaly detection. Detecting minor changes in patterns is useful for many domains and the sections below discuss the approaches in detecting minor changes in anomaly detection researches. The next few paragraphs discuss each method, their limitations and benefits for anomaly detection.

2.6.1. Anomaly Detection using Clustering Techniques

Clustering is used to group similar data instances into clusters [88]. Clustering based anomaly detection techniques can be categorized into three types as shown in Figure.2.7 and each type has different assumptions for anomalies [12].

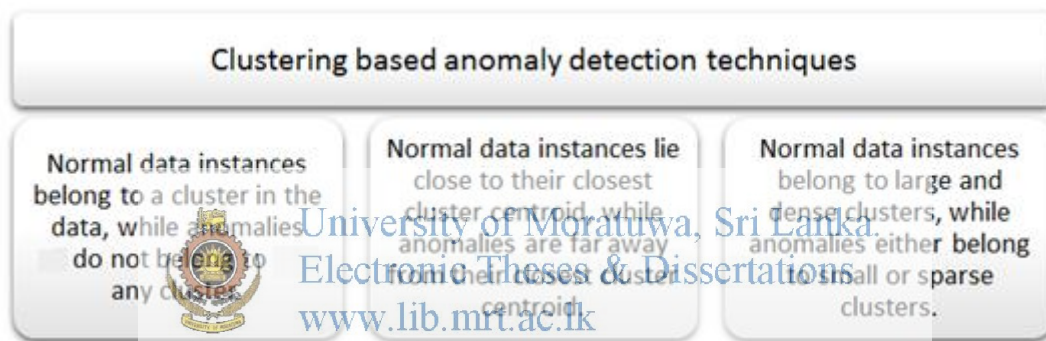


Figure.2.7: Clustering based anomaly detection techniques

In category one, normal data instances belong to a cluster, while anomalies do not belong to any cluster and are called outliers. Several clustering algorithms such as DBSCAN [58], ROCK [89], and SNN clustering [83] do not force all data instances to belong to a cluster. They detect clusters and non-clustered instances are declared as anomalies. Those techniques are not optimized to find anomalies, since the main aim of the underlying clustering algorithm is to find clusters.

The second category of clustering algorithms calculates the distances from cluster nodes to the closest cluster centroid where the distances are the anomaly scores. Self-Organizing Maps (SOM) [90], K-means Clustering[91], and Expectation Maximization (EM) [62] belong to this cluster type.

If the anomalies in the dataset form clusters by themselves, these clustering techniques are not able to detect such anomalies. To address this issue, a third category of clustering based

techniques has been proposed. This third type of cluster technique has threshold value that gives the minimum value for size and/or density of the anomaly cluster. Numerous distinctions of the third category of techniques have been proposed, such as cluster based local outlier factor (CBLOF) [92], an anomaly detection technique using k-d trees [93], and clustering and robust estimators to detect outliers [94]. Chaudhary et al. [93] proposed a technique using k-d trees that provide a partitioning of the data in linear time and they test it on astronomical data sets. Another research group proposed an indexing technique called CD-trees using the concept of Skew of Data (SOD) to efficiently partition data into clusters. The data instances that belong to sparse clusters are declared as anomalies [95].

Anomalies are detected as a by-product of many clustering techniques, and hence are not optimized for anomaly detection. Some clustering based techniques are effective only when the anomalies do not form significant clusters among themselves. Web usage mining may have high density instances when there is a Denial of Service (DOS) attack [96]. Therefore, those attacks are not detected using those techniques as the DOS attacks form a cluster.

2.6.2. Anomaly Detection using Classification Techniques

Classification anomaly detection techniques are based on classifiers that can distinguish between normal and anomalous classes that are learnt in the given feature space. In the training phase, classification techniques require a labelled data set [59, 60]. However, in web usage mining, getting a labelled dataset is a difficult task. Access logs contain web user requests and there is not enough information to generate labels for log data or web sessions automatically. Normally, there is a huge amount of web sessions, and web sessions have many combinations of pages and page sequences. Therefore, labelling those web sessions manually is not feasible.

2.6.3. Statistical Anomaly Detection Techniques

The underlying principle of any statistical anomaly detection technique is; “An anomaly is an observation which is suspected of being partially or wholly irrelevant because it is not generated by the stochastic model assumed” [99].

The key disadvantage of statistical techniques is that they rely on the assumption that the data is generated from a particular distribution. This assumption does not hold true for high dimensional real data sets. Choosing the best statistic is often not a straightforward task

[100]. Histogram based statistical techniques check if a test instance falls into any one of the bins of the histogram [96, 97]. If it does, the test instance is normal, otherwise it is anomalous. An anomaly might have attribute values that are individually very frequent, but their combination is very rare. An attribute-wise histogram based technique would not be able to detect such anomalies [12].

2.6.4. Anomaly Detection using Association Rule Mining

Anomaly detection rules are learnt by capturing the normal behavior. A basic multi-class rule technique involves two steps. The first step is to learn rules from the training data, whereas the second step is to test the instance by catching the test instances [61, 62].

Association rule mining has been used for one-class anomaly detection by generating rules from the data in an unsupervised fashion [95, 96]. Those techniques rely on the availability of accurate labels for normal classes, which are often not possible in web usage mining. Each test instance also needs to be labelled [12]. Therefore, rule based techniques are not that suitable for web usage mining.

2.6.5. Other Techniques for Anomaly Detection



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The spectral anomaly detection technique is another way to capture anomalies. Such techniques are based on the key assumption that data can be embedded into a lower dimensional subspace in which normal instances and anomalies appear significantly different [10, 19]. The spectral anomaly detection technique does not detect minor changes.

Since the main objective of web usage mining is to detect common user patterns, anomaly detection techniques are not often used in web usage mining. However a few research groups have used anomaly detection techniques in web usage mining to improve the system [19, 101].

2.7. Using Episodes for Web Usage Mining

An episode is defined by the W3C as a semantically meaningful subset of user sessions [42]. Episode identification is an optional preprocessing step that can be performed after the required preprocessing steps, as shown in Figure 2.8 [29].

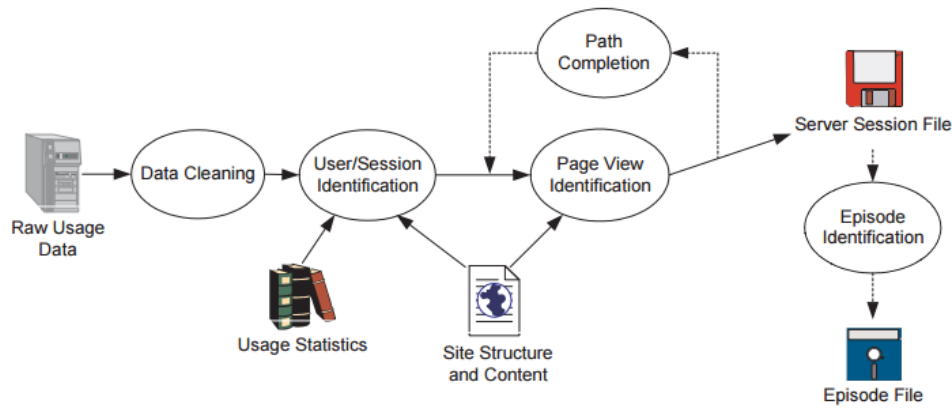


Figure 2.8: Preprocessing steps with episode identification (source [29])

The goal is to identify users' partial interests. For instance, we can define an episode as business page views on a news server or shopping cart pages on an e-shop. Besides these manual definitions, there are three general methods to recognize subsets of user sessions, based on the assumptions about user browsing behavior [29].

Web usage mining has three phases, as explained in section 2.3. Figure 2.9 shows the full details of the web usage mining process and covers methodologies that are used in web usage mining [29]. It also shows the episode identification module included in the preprocessing stage.



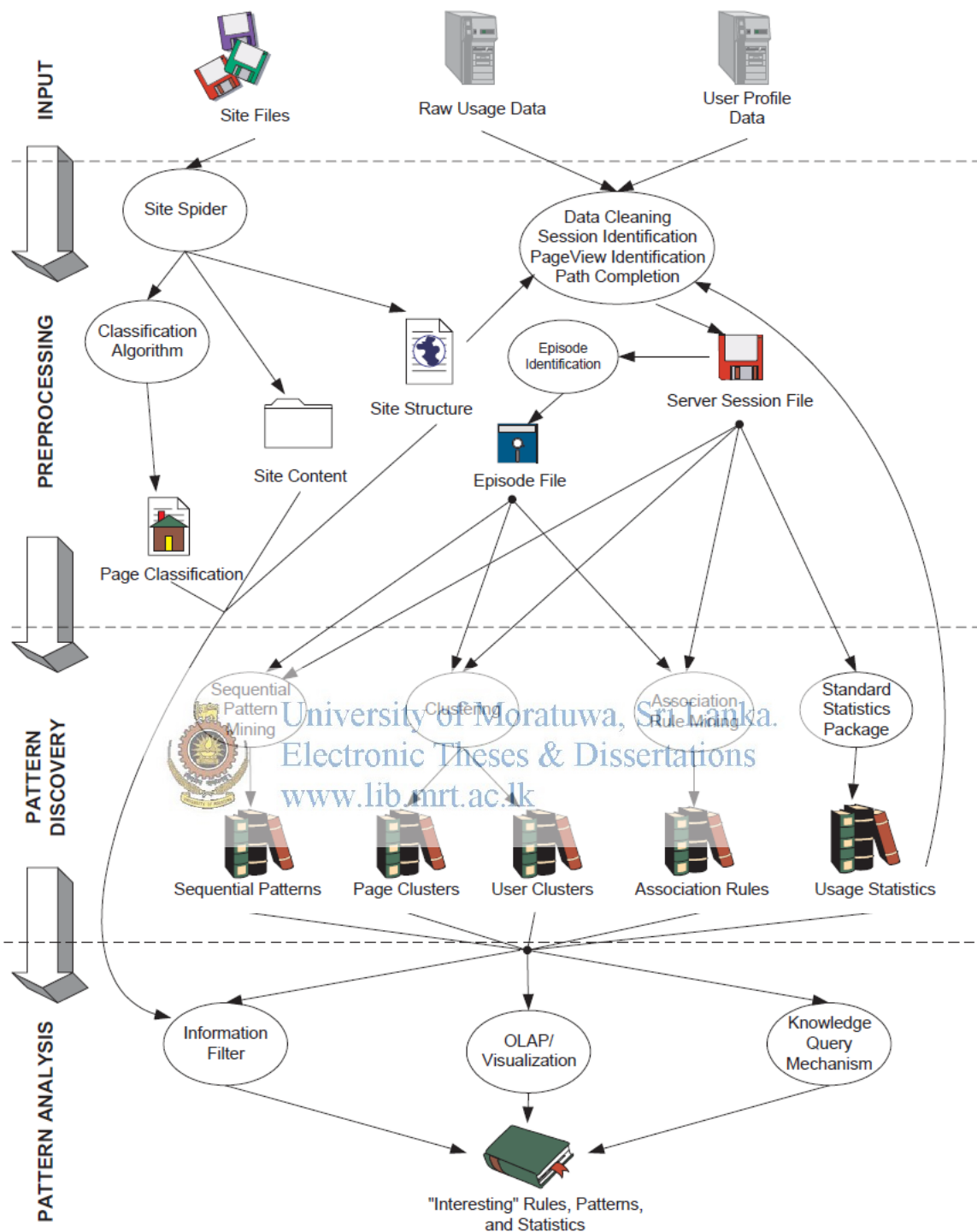


Figure 2.9: Detailed web usage mining process (source [1])

The same author explained more about the episode definition in theoretical form. The page classification to the auxiliary and media pages is based on the assumption that the user browses a website until he finds the relevant page (the media page). The path to this media

page is assembled from the auxiliary (or navigational) pages. Generally, there are two types of episodes; "Auxiliary-Media" and "Media only", as shown in Figure 2.10.

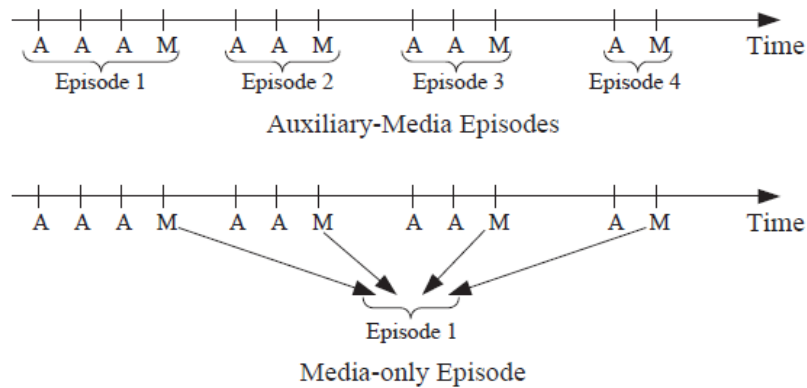


Figure 2.10: Two types of episodes (Source [29])

According to Figure 2.10, episodes can be further divided into two types [29]. With the concept of auxiliary and media page references, A and M represents auxiliary page access and media page access respectively. In the first method, an auxiliary-media episode is defined as a sequence of auxiliary page accesses ending with a media page access. The second method is to define the set of all the media page accesses as media-only episodes [11].

Another episode definition is the subset of page views that deal with the "new user registration" process. The second definition is interesting as it detects episodes only from page navigation URLs and not from page content [29].

A session is a collection of episodes. An episode is a sequence of access pages that has semantic meaning [29]. For example, in a commercial website, an episode would refer to payment. This involves pages {payment page, enter credit card details, confirmation}. Similarly, the login episode contains pages {home page, enter credentials, user account} and purchase episode contains pages {shopping cart, add items to cart, checkout, confirm} (page accesses occur in the given order).

2.8. Suffix Array to Locate the Substring Pattern

Suffix array is a well-known data structure used in full text indices and bioinformatics to find the longest pattern [108]. The suffix array is used as an index to locate the substring pattern very quickly. The suffix array (SA) of any string, T, is an array of length |T| such that SA[i] = j,

where $T[j : |T|]$ is the i -th lexicographically smallest suffix of T . The suffix array for any string of length $|T|$ can be constructed in $O(|T| \log |T|)$ time [109].

Suffix array is a popular data structure in genetic algorithms to find the longest pattern [64]. A suffix data model is used for mining frequent traversal path patterns from very large web logs. During the data cleaning phase in web usage mining, the suffix style is used to remove files and URL parameters [111]. An efficient web user pattern mining approach based on suffix array for frequent reference path generation was presented by Jing [112]. Microsoft research presented the suffix-array algorithm to efficiently derive all possible substrings and their frequencies in URL mining and is used in the RegEx generator [113].

2.9. Regular Expressions

A regular expression is a sequence of symbols and characters expressing a pattern, mainly for the use in pattern matching with strings [114]. The simplest regular expression is a single literal character, except for the following special meta characters: `*` `+` `?` `()` `|`. The use of regular expressions have been presented as a sequential pattern mining process [52]. However, regular expressions have not been used in web session pattern detection.

2.9.1. Regular Expression Engines

There are regular expression engines in operating systems, text editors and word processing applications. The next few paragraphs describe existing tools and libraries along with their pros and cons. Regular expression engines check for the existence of patterns in the data set. RE2 (Regular Expression Engine) is a fast, safe, thread-friendly regular expression engine with backtracking in Perl and Python [115]. It identifies the matches in text or content after giving the regular expression. RE2 is not capable of finding patterns in text content. It generates regular expressions from the content. The PCRE (Perl Compatible Regular Expressions) library develops regular expression pattern matching using the syntax and semantics for Perl 5 [116]. It is also a pattern matching library for a given regular expression, not capable of pattern recognition.

Regular expression builders such as RegExr and RegxBuddy are very popular. RegExr is a tool for learning, writing and testing regular expressions [117]. Although RegExr includes features such as real time results, code hinting, detailed results and a built-in regex guide, it does not build regular expressions from the context. RegxBuddy is another tool used to create and edit regular expressions [118]. After the creation of regular expressions, it is tested

on sample data, stored for later usage and sent to the application. The regular expression can be exported as well. It matches the character strings with the regular expressions. However, regular expression builders are not built for pattern recognition. Regular expression engines have pattern matching, and regular expression builders have to pass the interesting pattern or event or stream to generate the regex.

Cho and Rajagopalan addressed scale and performance issues when they matched regular expressions against a large corpus [119]. Suffix-tree was introduced to speed up the regular expression indexing engine.

2.10. Discussion

Section 2.5.2 highlighted two problems in clustering techniques that have a big impact on web usage mining. The first issue is that a lot of clustering algorithms require parameters such as cluster count before the algorithm can be executed. This is not feasible in web usage mining as the cluster count is mapped to user behavior group count, which is unknown. Later, researchers used density based clustering algorithms as explained in section 2.5.2.2. However, these require minimum density for a cluster region. The expectation maximization clustering algorithm can be executed without passing parameters and was used in web usage mining recently. Using these techniques, identification of common behavior has been the subject of most of the previous researches. Comparatively, anomaly detection using clustering techniques has received less focus.

In most clustering algorithms, the access patterns with slight deviations from each other are clustered together based on their similarity. Identifying these slight changes in access patterns is important to prevent attackers and hackers. Anomaly detection clustering techniques explained in section 2.6.1 also do not detect these minor changes within clusters.

3. USING HYBRID CLUSTERING TO IDENTIFY WEBSITE USER ACCESS PATTERNS

3.1. Overview

In the previous chapter, past and present research on web usage mining and its applications were discussed. To run clustering algorithms, parameters such as cluster count and minimum count for cluster density region are necessary. It is a major problem in web usage mining using clustering, as highlighted in section 2.5.2.

We also discussed approaches and achievements in anomaly detection. We also highlighted that many clustering algorithms are not optimized to find anomalies and need labeled data to train the classification and associative rule mining techniques. When anomalies are highly frequent, these techniques fail to detect them. There are not many achievements on detecting slight changes in patterns in web sessions.

In this research, two approaches are presented to solve these problems. The steps for those two approaches are shown in Figure 3.1 and Figure 3.2. Chapter 3 and chapter 4 describe them respectively. The first approach solves the cluster parameter issue by combining two clustering algorithms. Detecting slight differences between web user sessions is achieved by an episode-based approach. The difference in the hybrid clustering algorithm approach is that it considers the page sequence after pattern discovery by clustering.



Figure 3.1: Hybrid clustering algorithm approach



Figure 3.2: Episode based approach

Section 3.2 explains the data model and terms that are used in our approach. The next section explains web log preprocessing engine design and implementation. Many data mining techniques used in pattern discovery support only numeric values as inputs. However, log files contain text and numbers and it is important to have log data transformed from a mixed data type into a numbers only representation. This transformation is achieved at the data preprocessing phase. URLs are converted to a numerical representation. Referee and browser agent fields are used to identify the user session. After session identification, there is only numerical representation of user sessions.

The hybrid clustering technique, which is used to solve the first problem in section 2.5.2, is described in section 3.4. It includes the justification for using the hybrid clustering approach as well. The final section covers the signature module, which is responsible for finding the unique patterns in clusters of web sessions.

3.2. Terminology and the Data Model

The key step in a Knowledge Discovery System (KDS) is to pick the correct and suitable data set for the data mining task. There are proxy level, server level, client level, firewall level and site level logs. Each type of data collection differs not only in terms of the location of the data source, but also in the kind of data. The client level log contains a single user in multi-site behavior, the server level logs depict multiple users in a single server, proxy logs track multiple users in a multi-site usage environment and site level logs carry information of multiple users in a single site. Since we are interested in user access patterns, we collect site level logs, which are also called web access logs.

Web access logs contain web resource usage against the website user IP address, along with the timestamp. In other terms, the web access log maintains a history of page and web resource requests. The W3C maintains a standard format (the common log format) for web server log files, but other proprietary formats also exist. The common log format, also recognized as the NCSA common log format, is a standardized text file format by National Center for Supercomputing Applications [35]. It is referred to as the access log since it contains only basic HTTP access information. An access log is a collection of web requests where the web request includes the client IP address, requested date/time, page requested, HTTP code, bytes served, user agent, and referrer. Figure 3.3 is an example of a single web record in a web access log. Each element in the log record is explained in bullet points respectively. A "-" in a field indicates missing data.

```
220.247.227.134 user-identifier [10/Jan/2016:13:55:36 -0700] "GET /public_img.png HTTP/1.0" 200
4567 "-" "Mozilla/5.0 (Windows NT 6.1) Firefox/35.0"
```

Figure 3.3: Log record in access log file

The web log record contains the following details in order.

- Remote host - Remote hostname (or IP number if DNS hostname is not available).
- User identifier - The remote log name of the user or user identification.
- Date - Date and time of the request in box brackets.
- Request - The request line exactly as it came from the client, in double quotes.
- HTTP Status - The HTTP status code returned to the client.
- Bytes - The content-length of the document transferred, measured in bytes.

This data can be combined into a single file or separated into distinct logs, such as an access log, error log, or referrer log. Anyway, access logs typically do not collect user-specific information such as user names, user ID, user password, etc. Log files are not accessible to general internet users, but are accessible only to the webmaster or other website administrative persons.

As mentioned briefly in chapter 2, the W3C has defined several data abstractions for web usage. A user is defined as a single individual that is accessing files or web resources through a web browser using a digital device. A digital device can be a computer, tablet or mobile device, and is able to send web requests. Also, several users may use the same machine and browser.

A page view consists of the set of files that contribute to the display on a user's browser at a given time. A page view is usually associated with a single user action such as a mouse click or navigation from the website. When discussing and analyzing user behavior, it is really the aggregated page view that is important. The user normally requests a web page and does not explicitly ask for a particular image or frame or graph or database record to load into his or her browser. The set of page views in a user session for a particular website is referred to as a server session or a web user session, and is also known as a visit. A set of web user sessions are fed to web usage mining tools. Tracking these sessions using server level data is very difficult as they are recorded with respect to time, not from user sessions. Before using these access logs for analysis, the web sessions need to be identified.

3.3. Preprocessing Engine

The main tasks of the preprocessing engine are to convert web access logs into a web session matrix, and user identification. The data preprocessing engine needs to be effective and efficient since it is the starting point in the system and would affect the whole process. Multiple log formats are needed to be handled by the engine. The logs should be given in a generalized format. Session and user identification are configurable using the parameters, web agent, status, IP and time stamps. The above are requirements of the data preprocessing engine. The next section discusses the theoretical aspects in accordance with the literature review in section 2.4.2. Much research has been done on data preprocessing and their work were compared and contrasted in the section 2.4.3 and Table 2.1.

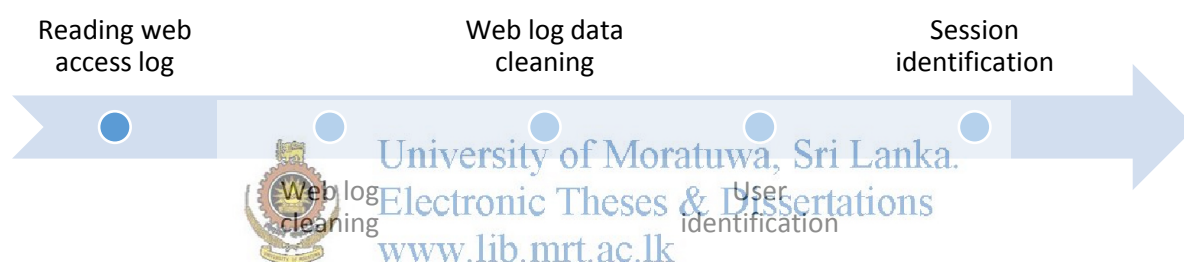


Figure 3.4: Functionality of the data preprocessing engine

After acquiring the access log, data cleaning is done in the next stage. Duplicates and formatting issues are removed from the data at this stage. User identification is done using the IP address whereas session identification is done using agent time and IP session. The data preprocessing engine functions are listed in Figure 3.4.

3.3.1. Implementation of Preprocessing Engine

Firstly, we need to preprocess the web server access logs and index them. We read the server access logs and filter the log records of a specific site and remove duplicates that can be in different formats.

There are access logs with different sizes that are created daily or weekly and sample access log file is shown in Figure 3.5. A normal website generates 100 MB to 1 GB volume of access log files weekly. It can vary depending on time periods such as days, months and

years, as web user trends vary from time to time. It is not practical to open the whole log file or read the whole file at once as it contains huge amounts of data to read and process. It is good practice to open while streaming and process it, as this will avoid memory overflows. Therefore, we used stream reading rather than file APIs or file open and read mechanism.

Log files are chunked with time periods or volume of the files without considering user sessions in real world servers. Since a log file is chunked in to different log files while it is being created in the server, web user sessions can be logged in two or more log files. Reading log files needs to be done according to the order in which log files are created. This avoids web session interruption. It is a good practice to follow when log files are generated in a daily or hourly basis. In our system, log files are filtered and structured log records are appended to one file called ‘log’.

```

21220 web.mywebsite.lk 123.231.24.74 - - [18/Jan/2015:14:15:49 +0530] "GET
/media/system/js/core.js HTTP/1.1" 200 4784 "http://www.mywebsite.lk/" "Mozilla/5.0
(Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.1 (KHTML, like Gecko)
Chrome/6.0.437.3 Safari/534.1"
21221 Jan 18 14:15:49 web httpd: web.mywebsite.lk 123.231.24.74 - - [18/Jan/2015:14:15:49
+0530] "GET /media/system/js/core.js HTTP/1.1" 200 4784 "http://www.mywebsite.lk/"
"Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.1 (KHTML, like Gecko)
Chrome/6.0.437.3 Safari/534.1"
21222 Jan 18 14:15:49 web httpd: web.mywebsite.lk 123.231.24.74 - - [18/Jan/2015:14:15:49
+0530] "GET /media/system/js/core.js HTTP/1.1" 200 4784 "http://www.mywebsite.lk/"
"Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.1 (KHTML, like Gecko)
Chrome/6.0.437.3 Safari/534.1"
21223 web.mywebsite.lk 123.231.24.74 - - [18/Jan/2015:14:15:48 +0530] "GET
/media/system/js/mootools-core.js HTTP/1.1" 200 96362 "http://www.mywebsite.lk/"
"Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.1 (KHTML, like Gecko)
Chrome/6.0.437.3 Safari/534.1"
21224 Jan 18 14:15:51 web httpd: web.mywebsite.lk 123.231.24.74 - - [18/Jan/2015:14:15:48
+0530] "GET /media/system/js/mootools-core.js HTTP/1.1" 200 96362 "
http://www.mywebsite.lk/" "Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US)
AppleWebKit/534.1 (KHTML, like Gecko) Chrome/6.0.437.3 Safari/534.1"
21225 web.mywebsite.lk 123.231.24.74 - - [18/Jan/2015:14:15:52 +0530] "GET

```

Figure 3.5: sample access log file

Log data was cleaned using three steps as proposed in [34]. Logs contain textual data. Site URLs, agents and similar parameters in the log record are duplicated. In log cleaning, those URLs in the website are indexed in order to reduce the volume of data to be processed as integers represent the URLs. Filtering is done over HTTP status and web resource types. This filtering is configurable in the implemented system by defining web resource extensions such as .js, .png and .css. By defining those file extensions in the system configuration; it excludes log records that contain records related to those files.

In computer science and networking, there are many types of sessions such as shell sessions, TCP sessions (Transmission Control Protocol sessions), login sessions, desktop sessions, browser sessions and server sessions. In web usage mining, a browser session is also called a web session. A web session is a semi-permanent interactive information interchange between the server (website) and the user and it allows associating information with individual visitors [20]. The log file consists of web requests from all users and is identified with these web sessions. Sessions are generated by grouping 'IP' or 'IP' and 'Agent' [14]. Those web user sessions are included in a session file. Each URL is given a unique number (integer). The mapping file contains the mapping of these integer numbers to the URLs. Figure 3.6 contains a snapshot of the session file and mapping file.



Figure 3.6: Session file and Mapping file

The mapping file is sorted and similarities between the URLs are found by looking at the argument length and argument name in the URLs. The same page can be referred with different sets of URL arguments. Once these arguments are removed, only one entry per URL is left in the mapping file. After removing URL arguments, the mapping file is updated with the new version and the session file is updated with the new mapping file.

The most requested pages for a given time period are retrieved by counting the number of web page requests. The new mapping file for most common pages (page occurrence count can be adjusted) is built. An NxM matrix with page ID and session number (default size is the total size of the session file and page count) respectively, is built to be used by clustering algorithms [10, 60]. Many clustering algorithms can be executed on the matrix since it has numerical elements. Figure 3.7 shows the components in the data preprocessing engine.

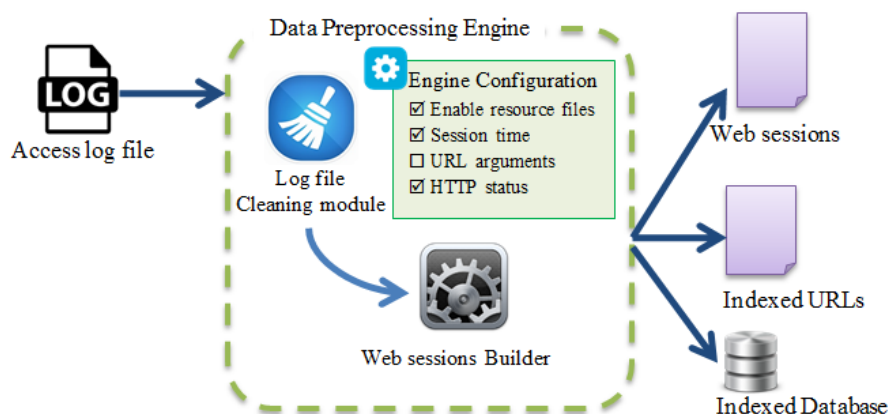


Figure 3.7: Component architecture of the data preprocessing engine

At the end of the preprocessing stage, web records are grouped and indexed by the web session sequence and all the sessions have a unique ID. The unique ID points to a list of web log records. All the log parameters such as HTTP status, web referrer and web agent are contained in a web record. The web records can be retrieved from the session ID. The advantage of an indexed database is that, in the case of an interesting session, the web records can be retrieved in an efficient manner.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

3.4. Hybrid Clustering for Web Usage Mining

A key feature of EM+DBSCAN is the integration of EM with DBSCAN clustering. In this section, we provide the background with advantages and limitations of EM and DBSCAN. Justification for the combination is discussed in section 3.4.1. Section 3.4.2 discussed the EM+DBSCAN implementation.

3.4.1. Justification for EM+DBSCAN

The parameters in a clustering technique have massive influence on the accuracy of the algorithm, as discussed in section 2.5.2.5. In web usage mining, it is hard to know the exact values or predict values for these parameters. Changing user patterns is unpredictable and there are many different website users. Updating the site and changing user trends also have a great impact on those parameters. Therefore, the values for parameters fluctuate from time to time and it is hard to handle the algorithm in an effective manner, as those algorithms are bound to those parameter values.

Modern websites range from simple to complex as it is a drastically growing field. As Table 3.1 shows, there are some facts that affect user access patterns. Some sites contain only a single page model and the URL parameter is responsible for page content to load using ajax web requests. Modern user interface components such as menus, tree and breadcrumbs let user navigate as they wish. Users can access any page with different page sequences and multiple accesses to the same page. Therefore, there can be unlimited page combinations that the user can access.

Table 3.1: Factors that affect user navigation in a website

Factors That Affect Navigation	Existing Techniques
Site design	Single Page
	Multiple pages
	Unlimited scrolling
Page load type	Hard code pages
	Dynamic pages
Navigation UI component	Hyperlinks
	Menu bars
	UI Trees
	Breadcrumbs

User patterns vary by many factors, as explained in the above paragraph and Table 3.1. There are many page access combinations and user web access requirements that also vary with the user's intentions. There can be many access patterns that are not expected beforehand so counting categories and grouping these access patterns is difficult. Since pattern count is hard to determine, finding the cluster count is not possible. Domain experts are also not able to give the correct cluster count. Many clustering algorithms such as k-means require cluster count to be known beforehand.

EM can be executed with or without passing the cluster count whereas k-means needs the cluster count in order to execute. When EM is run without the cluster count, the cluster result accuracy is low. EM contains a useful feature where it can execute with or without the cluster count. This feature is more useful in web usage mining and works to our advantage.

DBSCAN does not require the number of clusters to be specified prior to running the algorithm, but needs a minimum number of points required to form a dense region. DBSCAN

is able to find arbitrarily shaped clusters and it is robust to outliers. There exists work that has combined k-means and EM but not EM and DBSCAN. By combining the EM and DBSCAN algorithms, it is possible to eliminate the individual disadvantages of the algorithms such as prior knowledge on cluster count and accuracy degradation.

3.4.2. EM+DBSCAN Algorithm Implementation

The use of EM+DBSCAN algorithms is described below and is graphically represented in Figure 3.8. First, we execute the EM clustering algorithm passing -1 since we do not know the cluster count. The EM clustering algorithm is sufficient in finding the cluster count. The algorithm outputs a suitable value for the cluster count depending on the data set given. The count of the data set is divided by the cluster count and by a Gaussian function, and the minimum and maximum number of web sessions in a cluster are calculated. These values are input to the DBSCAN algorithm and the cluster count is the output from the system. Inter-cluster distances, intra-cluster distances and completeness values are compared (highest intra-cluster distance, lowest inter-cluster distance and highest completeness the better) to find the best value. Then the cluster count given by this best value is input to the EM algorithm again and the results from each algorithm are interchanged to get an optimal value. By examining the plot of inter-cluster distances and intra-cluster distances, it can be said that the values have reached an optimal value. The plotted graph should reach a steady value at the optimum cluster count. The final cluster result is the best result that was generated during the steady period.

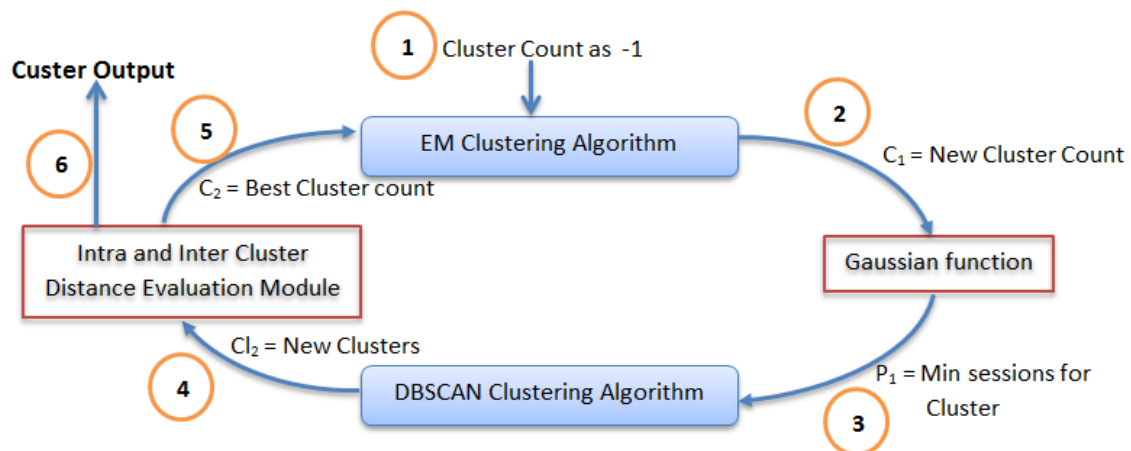


Figure 3.8: EM +DBSCAN algorithm

3.5. The Signature Module

A set of clusters are built after clustering. Each cluster contains its own features. In the initial phase, domain expert knowledge is used to evaluate the clusters. Experts can label them if the clustering is correct. Unique features are then found for each cluster and these features correspond to different user behaviors. A cluster is identified by this unique signature called the 'finger print'. Frequent pattern mining and sequence pattern mining were used to find unique signatures.

Clusters were built on the hypothesis that each cluster can be distinguished by the occurrence of a particular page or a set of pages. However, in some cases we could not find a unique page occurrence in a cluster, therefore, we needed to find a set of pages that uniquely identified that cluster. Then, the page access order also has to be considered to find the unique feature of the cluster. Association rule mining is a method of discovering interesting relations between variables or items in a large dataset [105]. Association rule mining typically does not consider the order of items either within a transaction, or across transactions in contrast to sequence pattern mining, but it is faster than sequence mining. Therefore, we perform association rule mining to get confidence in the signature. If the obtained signature is not unique, then we proceed to sequence pattern mining.

Some clusters do not have a unique page occurrence, but some do. As an example, consider the cluster 1 page occurrence matrix in

Figure 3.9. Sessions 1, 6 and 7 belong to cluster 1. All the above sessions in cluster 1 contain pages P1, P6 and P10. P6 only occurred in cluster 1 where P1 and P10 can be seen in other clusters such as cluster 2 and cluster 3 (sessions 2 and 4 respectively). Therefore, P6 is a unique signature for cluster 1. There are some clusters where it is difficult to find a unique page occurrence. Consider matrix sessions 2, 3, and 5 in

Figure 3.9. They all belong to the same cluster (cluster 2). P1 and P2 cannot be considered as the signature of this cluster since P2 does not occur in session 5. Since a unique signature could not be given to cluster 2, sequence pattern mining was used to find the signatures.

Session No	Web page occurrence matrix													Cluster No	
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13		
Session 01	1	0	0	0	0	1	0	0	0	0	1	0	0	0	Cluster 01
Session 02	1	1	0	0	0	0	0	0	0	0	0	1	1	1	Cluster 02
Session 03	1	1	0	0	0	0	0	0	0	0	0	0	0	1	Cluster 02
Session 04	1	0	0	0	0	0	1	1	1	0	0	0	0	1	Cluster 03
Session 05	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Cluster 02
Session 06	1	0	0	0	0	1	0	0	0	0	1	0	0	0	Cluster 01
Session 07	1	0	0	0	0	1	0	0	0	0	1	0	1	1	Cluster 01

Figure 3.9: Cluster matrix with session numbers and page occurrences

Apriori is used to find the signatures in clusters where the support level is one or closer to one (support level 1 indicates the occurrence of a page or pages in all the sessions in the cluster). A simple string search over the page occurrence matrix confirms the uniqueness of a signature.

The signatures are optimized by checking the length and the positions of pages in web sessions. For example, a cluster can have multiple signatures but it is worth picking a shorter length for the signature and position of the signature that can appear in the user session. When the signature is short, it improves the detection time. The first appearance of the signature is important so that the system is able to label the sessions in the initial phase.

Figure 3.10 shows the cluster signature module implementation.

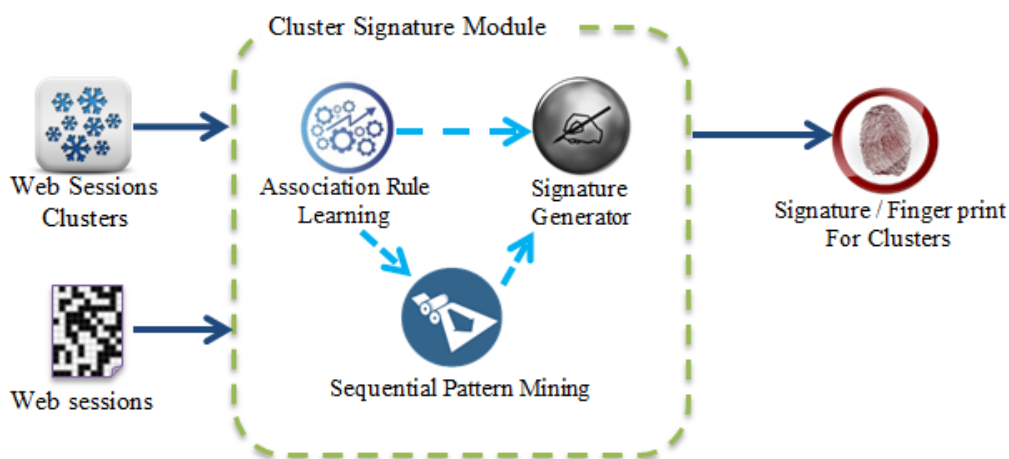


Figure 3.10: Cluster signature module

4. EPISODE BASED APPROACH

4.1. Overview

The EM+DBSCAN clustering algorithm finds interesting access patterns and groups the users, but cannot identify the slight differences between accesses patterns included in individual clusters. In reality, these could refer to important information about attacks.

This chapter introduces a methodology to identify these access patterns at a much lower level than what is provided by traditional clustering techniques, such as nearest neighbor based techniques and classification techniques. This technique makes use of the concept of episodes to represent web sessions. These episodes are expressed in the form of regular expressions. To the best of our knowledge, this is the first time that the concept of regular expressions is applied to identify user access patterns in web server log data.

4.2. Detecting Slight Changes

To capture the slight changes, the data model needs to be generated or pre-processed as lossless.



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

4.2.1. Data Models for Detecting Slight Changes

There are four types of data models as shown in Figure 4.1. These are the possible ways to represent web user session data.

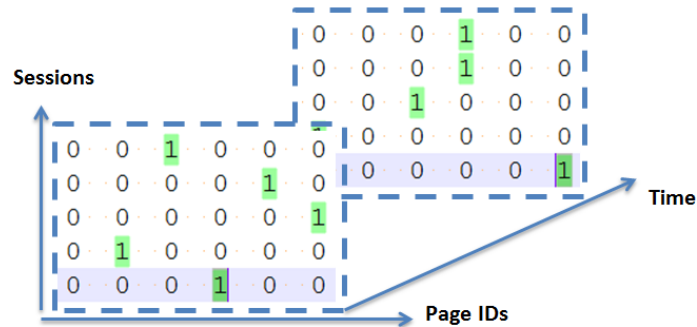
- Page occurrence matrix
- Page frequency matrix
- Data cube
- Session page list with suffix array

	P1	P2	P3	P4	P5	P6
Session No 11	0	0	1	1	0	1
Session No 12	0	0	0	1	1	1
Session No 13	0	0	1	1	0	1
Session No 14	1	1	0	0	0	0
Session No 15	0	0	0	1	0	1

(a) Page occurrence matrix

		P1	P2	P3	P4	P5	P6
Session No 11		0	0	1	4	0	1
Session No 12		0	0	0	5	5	1
Session No 13		0	0	1	10	0	3
Session No 14		3	4	0	0	0	0
Session No 15		0	0	0	7	0	1

(b) Page frequency matrix



(c) Data cube

Session No 11	P4	P3	P6	P4
Session No 12	P4	P5	P5	P5
Session No 13	P4	P4	P4	P3
Session No 14	P1	P2	P1	P2
Session No 15	P4	P4		

(d) Session page list

Figure 4.1: Data model types in web usage mining

Page occurrence matrix is an $N \times M$ matrix with elements 1 or 0. It horizontally represents page IDs and vertically represents session IDs. The value 1 or 0 denotes whether or not the page has occurred in a session.

Page frequency matrix is an $N \times M$ matrix with elements between 0 – n. The horizontal axis and vertical axis represent the page IDs and session IDs respectively. This is same as the page occurrence matrix. The value of an element indicates the frequency that the page has occurred in the web session. The frequency matrix represents the page frequency of the sessions, which is not included in the page occurrence matrix.

Data cube is a 3D structure with page ID represented along the horizontal axis and session ID represented along the vertical axis. The third dimension represents the time axis.

The session page list represents the list of page IDs with the session. It is improved with the suffixed array discussed in section 2.8. Suffix array is a well-known data structure used in full text indices and bioinformatics to find the longest pattern. The suffix array is used as an index to locate the substring pattern very quickly.

The page occurrence matrix and page frequency matrix are not used for capturing slight changes in user access patterns since they do not contain sequence related information. Therefore, data is lost in these data models. Data cube is not selected for capturing and storing web session information as it consumes a large amount of data volume. Also, the processing of the data structure is complex due to drill-down and roll-up functions. Suffix array is more light-weight than data cube. Since the sequence is stored in the suffix array and the subsequences are considered, we used it in our design as a basic data structure.

4.2.2. Slight Changes between Web User Sessions

After representing web sessions as episodes, it is easier to distinguish the slight changes between user patterns with respect to normal user behavior and anomalous behavior. The example shown in Figure 4.2 depicts the normal user access pattern and the attacker access pattern. The sub-pattern that is common to both is called episode 1. The slight changes are easily highlighted in the access patterns when sub-patterns are replaced with episodes.

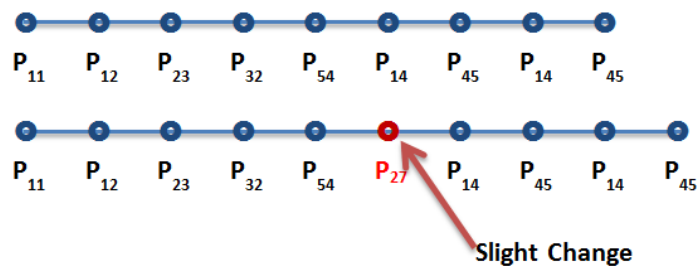


Figure 4.2: An example of a slight change in web session

4.2.3. Design

Our system contains three modules. First, the web log data cleaning module, which is responsible for purifying the web access log data by removing unwanted parameters. Web sessions are built from the same cleaning module described in section 3.3.

The second module generates regular expressions by processing web sessions. These regular expressions are indexed with the results count.

The final module groups the regular expressions by looking at their similarity and the count of their occurrences in a web session. These regular expressions are indexed and stored in suffix arrays. Suffix array used in this work is an improved version of the original suffix array, where we introduced an occurrence count. Therefore, the most common regex can be found easily. The groups represent episodes that have semantic meaning in order to understand the website user activities. Sessions are updated with episodes. Figure 4.3 shows the modular architecture of the implemented system.

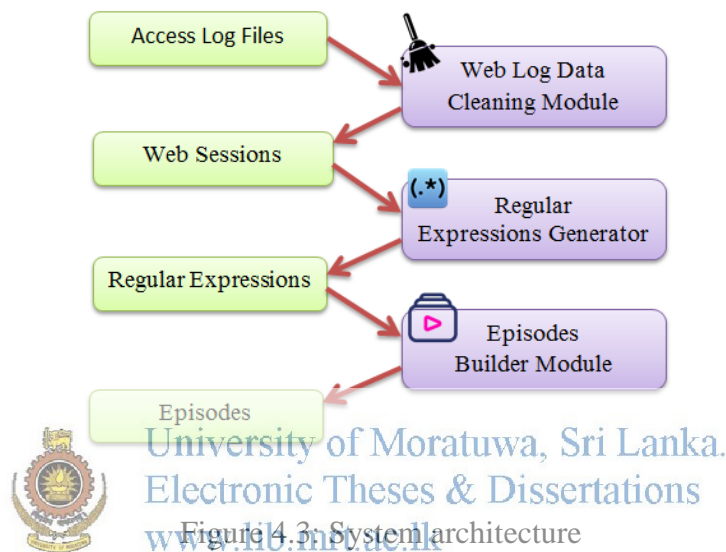


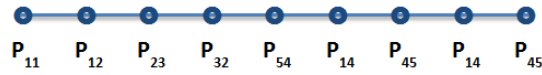
Figure 4.3 System architecture

4.3. Episode

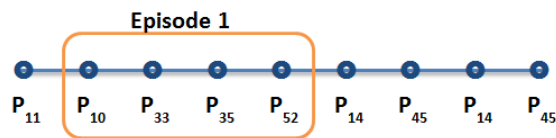
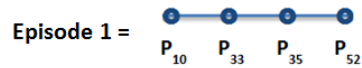
An episode is seen either as part of a session or the entire session. A set of episodes is seen on a user session as given in the example below in Figure 4.4. When a user accesses a website, he accesses the homepage, reads an article and gives feedback. The corresponding episodes are accessing home page, reading article and giving feedback. The same set of episodes can be present in another session in a different order.

When the occurrences of episodes are considered, the following structures are seen. There is the overlapping of two episodes with multiple pages in the overlapping area. There can be non-overlapping areas as well as overlapping areas in web sessions. Plain web session and web session with single episode are shown in Figure 4.4 (a) and (b), respectively. The next structure is two episodes with a page access in between. The two episodes are separated by a single page access (Figure 4.4 (c)). The other structure (Figure 4.4 (d)) is adjacent to two episodes. The episodes are non-overlapping and are next to each other. The next structures, Figure 4.4 (e) and (f) are two episodes overlapping each other. The overlapping area contains

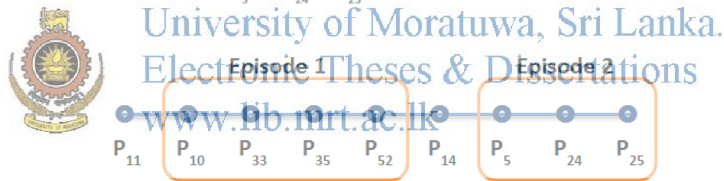
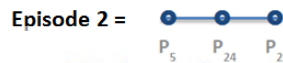
a single page access. The next structure is multiple episodes inside a single episode. There are one or more episodes inside the episode as shown in Figure 4.4 (g).



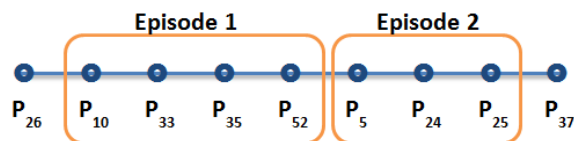
(a) Plain web session



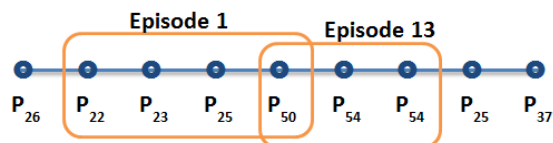
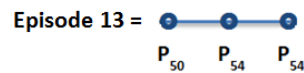
(b) Web session with one episode



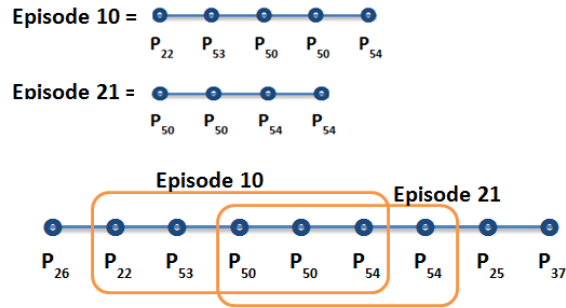
(c) Web session with episodes separated from a single page



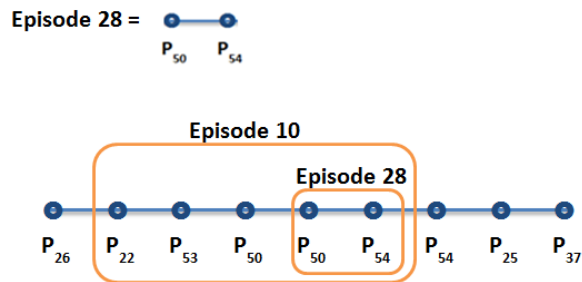
(d) Web session with adjacent episodes



(e) Episode overlapping with a single page



(f) Episode overlapping with multiple pages



(g) Child Episode

Figure 4.4: Episode structures

4.4. Regular Expressions to Represent Episodes



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

When an episode is represented as a regular expression, it reduces the volume of session data. When the episodes are represented by regular expressions, slight changes to major changes can be easily identified. Slight changes can be detected through the repetitions and alternations of individual pages. Major changes can be detected by repetitions and alternation of page groups.

The regular expressions generator in Figure 4.5 contains three components. They are; session subsequence builder, suffix array builder and regex builder.

The output of the session subsequence builder, i.e. sub-sequences (page sequences) is stored in a list similar to the improved suffix array. The suffix array builder outputs an improved suffix array of sub-sequences and the regex builder outputs a list of regular expressions.

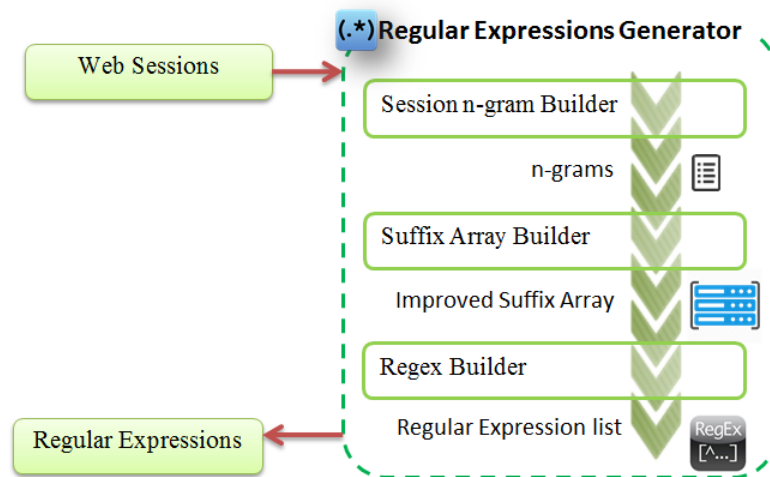


Figure 4.5: Regular expressions generator

As explained in section 2.5 and section 2.9, suffix arrays have been commonly used to identify the longest pattern in genetic algorithms [120] and on-line string searches [121]. When using suffix arrays for web sessions, the limitation of the suffix array approach is that it does not uniquely identify page sequences that are situated in the middle section of a session. Therefore, the session sub-sequence builder generates sub-sequences [122] from sessions. Lengths of sub-sequences are configured manually, or in default mode, have a fixed range from 2 to the length of the sessions.

The sub-sequences are stored with a suffix. We introduce a new feature column for suffix array called count that represents the occurrence count. When there is a suffix count for suffix array, it reduces the suffix array length and enables to find the sub-sequence distribution as well. It is a data structure used, among others, in full text indices, data compression algorithms and within the field of bioinformatics [123]. Here, we use this technique for page sequence compression and to increase the performance of our system.

Table 4.1 represents a sample session in a normal suffix array. In a suffix array, the table header, i, represents the string charter index and here, it is the session page index. If we used a sorted suffix array for sessions as in Table 4.2, we will not be able to find all the patterns in sessions as some page sequences such as {34,34,23}, {34,12,11} are missed from the middle part of the session string.

Session no 101: 12, 34, 34, 23, 11, 45

Table 4.1: Suffix Array on a sample user session

Suffix	i
12, 34, 34, 23, 11, 45 \$	1
34, 34, 23, 11, 45 \$	2
34, 23, 11, 45 \$	3
23, 11, 45 \$	4
11, 45 \$	5
45 \$	6
\$	7

Table 4.2: Sorted Suffix Array

Suffix	i
\$	7
11, 45 \$	5
12, 34, 34, 23, 11, 45 \$	1
23, 11, 45 \$	4
34, 23, 11, 45 \$	3
34, 34, 23, 11, 45 \$	2
45 \$	6



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk

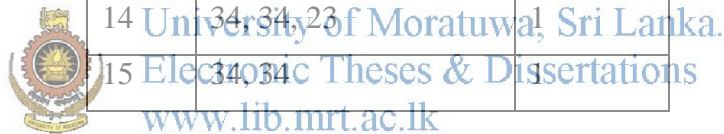
A suffix is part of a user page access sequence in sessions. Table 4.4 shows an improved sorted suffix array with index. A unique number for the suffix is assigned and it is built from n-grams in Table 4.3. This index can be used as a reference rather than using the suffix in any process or task that is not comparing suffix values. The suffix count is an important part of our system as it represents the commonness of user accesses. Using the count, we can get the most common and uncommon user access patterns.

Table 4.3: n-grams of the user session

N	n-grams
2	(12, 34), (34, 34), (34, 23), (23, 11), (11, 45)
3	(12, 34, 34), (34, 34, 23), (34, 23, 11), (23, 11, 45)
4	(12, 34, 34, 23), (34, 34, 23, 11), (34, 23, 11, 45)
5	(12, 34, 34, 23, 11), (34, 34, 23, 11, 45)
6	(12, 34, 34, 23, 11, 45)

Table 4.4: Sorted suffix array from n-gram

Index	Suffix	Count
1	11, 45	1
2	12, 34	1
3	12, 34, 34	1
4	12, 34, 34, 23	1
5	12, 34, 34, 23, 11	1
6	12, 34, 34, 23, 11, 45	1
7	23, 11	1
8	23, 11, 45	1
9	34, 23, 11, 45	1
10	34, 23, 11	1
11	34, 23	1
12	34, 34, 23, 11, 45	1
13	34, 34, 23, 11	1
14	34, 34, 23	1
15	34, 34	1



When a new n-gram is picked from a session, it is added to the sorted suffix array if it does not exist there. But if it exists, the count of suffix is incremented by one. The final improved n-gram suffix array is shorter than the regular n-gram suffix array because of the introduced count figure.

The suffix array contains substrings (sequences of pages) of suffixes. It can be confirmed by the count. If the count is the same in the substring and the string, they are from the same session. Therefore, we remove the substrings and it also reduces the array length. The suffixes (page sequences) are processed in the regular expression engine and regular expressions are its output.

We have used page sequences as strings. There are a few syntax patterns for regex. Here, we used traditional UNIX egrep regular expression syntax [124].

4.5. Regular Expression-Based Episode Representation

Let web page P be denoted as P and Session S as $S=P[1]P[2]...P[n]$ and let $P[i,j]$ denote an episode of S ranging from i to j . A regular expression can be mapped to any page sequence and it can be called an episode.

When we map episodes to sessions, there can be alternations, repetitions and concatenations. For example, suppose e_1 and e_2 are two regexes from sub-sequences n of session s , and n_1 and n_2 are sub-sequences of session s . If e_1 matches n_1 and e_2 matches n_2 , then $e_1|e_2$ matches n_1 or n_2 , and e_1e_2 matches $n_1.n_2$.

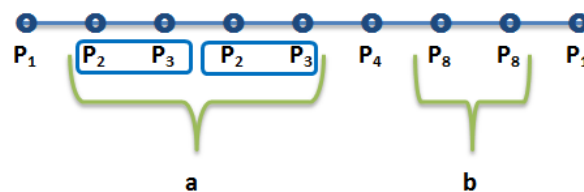


Figure 4.6: Sample session with page sequence

Consider the example given in Figure 4.6. It shows a session with page order accessed by a user. There are page repeating patterns, as shown in sections (a) and (b). Regex for a is $[P_2, P_3](2)$ and for b is $P_8[2]$. Session s is updated as in Figure 4.7. There are no overlapping episodes in the session in Figure 4.7. The regexes in the middle are also considered.

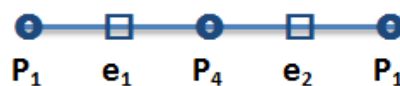


Figure 4.7: Session with episode representation

4.6. Episode Clustering

For clustering, the page and episode occurrence matrix in sessions is used. 1 or 0 in the matrix represents the occurrence of pages and episodes. Frequency matrix is another possible candidate but it does not give semantic meaningful results. Therefore, the occurrence matrix is used. The sequence of page occurrences is not considered in this approach at the cluster level but the page access sequence is included in the episode.

Episodes are created considering the sequence. The page IDs in sessions are replaced by the episode IDs. Some page IDs remain in the matrix without getting replaced by the episode IDs. The remaining page IDs and episode IDs are clustered.

4.7. Summary

In web usage mining, there are many methodologies to identify user access patterns such as clustering, Apriori algorithms, web access pattern tree (WAP-tree) and mining frequent patterns. However, in these methods it is difficult to determine how each session deviates from other sessions within the same group or cluster. It is important to find these slight differences between sessions. These seemingly insignificant changes could be the most important for a domain expert as they may resemble anomalies such as an attack. One of the limitations in k-means [96, 103] and other clustering mechanisms is that they remove noise from the data [63]. Here, important data or patterns might be lost. For example, an attacker's access sequence might be discarded as noise since it may be considered as an outlier.

To solve the above problem, the concept of an episode was used in web usage mining where an episode is a sequence of access pages that has semantic meaning. A session contains zero or more episodes. An episode is seen either as part of a session or the entire session. The slight changes are easily highlighted in the access patterns when sub-patterns are replaced with episodes.

The episode based web usage mining system was fully developed and the results were evaluated. The results are presented in the next chapter. In addition to identifying frequent patterns, we demonstrate that this technique is able to identify access patterns that occur rarely, which would have been simply treated as noise in traditional clustering mechanisms.

5. EVALUATION AND DEMONSTRATION

5.1. Overview

The purpose of this chapter is to demonstrate and evaluate the work so far described in this thesis, in order to justify how it meets the objectives of the thesis described in section 1.2, and to provide evidence on the thesis contributions (section 1.3).

In this study, we presented two approaches; normal proceeding in web usage mining with EM+DBSCAN and the episode based approach, which were explained in chapter 3 and chapter 4 respectively. For this demonstration and evaluation, access logs from different types of domains are used and the data set is explained in section 5.2. A university website, a financial institution website and a non-profit organization website are considered in our demonstration and evaluations in both approaches. These websites are denoted by U, F and N respectively, from here onwards. Here, we present quantitative and qualitative experiments done during the system evaluation.

Section 5.3 demonstrates the experiments and results of evaluating the EM+DBSCAN approach. There are four sub sections; comparing clustering algorithms with EM+DBSCAN, evaluating cluster signatures uniqueness using coverage and discrimination, detecting the temporal website changes between EM+DBSCAN and popular, often-used clustering algorithms, and demonstrating social media impact on website access patterns and network attack detection.

Evaluation on the episode based approach is described in section 5.4. It detected all the unique signatures recognized by the first approach in section 5.3.2. Section 5.4.3 describes identifying attacker sessions on a website, which was not detected in the first approach.

5.2. Data Set for Evaluation

Access logs from several websites (U, F and N) are collected and the size and time durations of the server logs for each data set are shown in Table 5.1. Access logs of each website include the entire web requests for all the website resources.

Table 5.1: Dataset for evaluation

Website	Size of log file (MB)	Duration	Web Request count	Web Session count
Website N	4781	13 Months	19 million	984,081
Website U	582	3 Months	1.75 million	32,823
Website F	274	3 Months	0.82 million	13,421

5.3. Evaluation of the EM+DBSCAN Approach

Two quantitative and two qualitative experiments are done during the system evaluation. U, F and N websites are considered in our evaluations. The four evaluations pave the way to a better understanding of the system.

The first evaluation of the quantitative analysis compares the performance of k-means, EM, DBSCAN and our EM +DBSCAN clustering algorithm. Next, we evaluate the accuracy of automated signature generation for clusters. The third experiment evaluates how the website changes with time and analyzes the temporal effects on clusters and sessions. Fourthly, we demonstrate how the anomalies and attacks are detected by the system and the impact of social media on websites.

5.3.1. Evaluating Clustering Algorithms

The quantitative analysis consists of evaluating four clustering algorithms. They are: k-means, EM, DBSCAN and EM+DBSCAN algorithms. These algorithms are used to cluster the web sessions. For the k-means algorithm, the number for the cluster count is the input to the system. However, the actual cluster count is not known beforehand in a typical scenario. Even the domain expert will have difficulty in determining this number. Table 5.2 depicts the number of clusters identified by two domain experts in four different months in 2015 for the non-profit organizational website. The experts were interrogated to discover how many behavioral model counts (clusters) they would estimate. The two domain experts gave different estimates since their levels of expertise differ.

Table 5.2: User behavior model count by domain experts and the system

	User Behavior Model Count			
Data Collected Months	Jan-Feb	Feb-March	Apr - May	June-July
First Domain Expert	5	9	13	13
Second Domain Expert	4	6	15	13
System	8	12	13	14

An incorrect estimation would not give a good clustering over the log data. Other problems arise since the experts do not know how the users really act in the system. These facts can be clearly seen in Table 6.1. There are many methods to carry out the same task, therefore the experts do not have sufficient information on which paths the users take. In January and February the domain expert did not know the actual user behavior model count, and when comparing their cluster count with that of the system, the resulting error was between 5% and 6%. In March, the website was improved and the experts were expecting a higher user group count increment, but the actual user group count increment was more than they expected. The error was between 6% and 7.5%. In April, a new user behavior model was introduced.

Figure 5.1 compares k-means, EM, DBSCAN and our combined algorithm EM+DBSCAN over non-profit organization web sessions. For this comparison, 30,000 sessions were considered. V-measure, intra-cluster and nearest cluster distances are used to evaluate the clusters. High values are preferred for the v-measure and inter-cluster distances, while a lower value is preferred for intra-cluster distances. EM+DBSCAN gives a better result for intra-cluster distance and v-measure against the other three algorithms as shown in Figure 5.1. Nearest-cluster distances represent an average result (second to k-means). Figure 5.2 and Figure 5.3 evaluate the clusters which are generated from educational website data and financial website data.

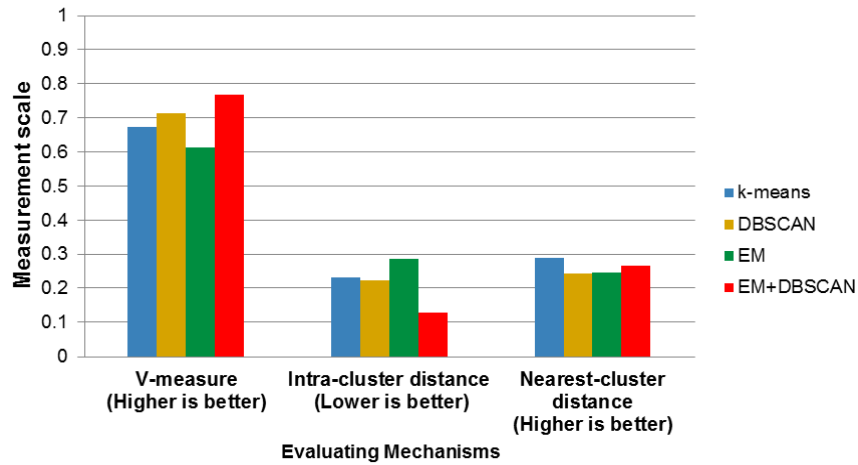


Figure 5.1: Evaluating cluster mechanisms and EM+DBSCAN for website N

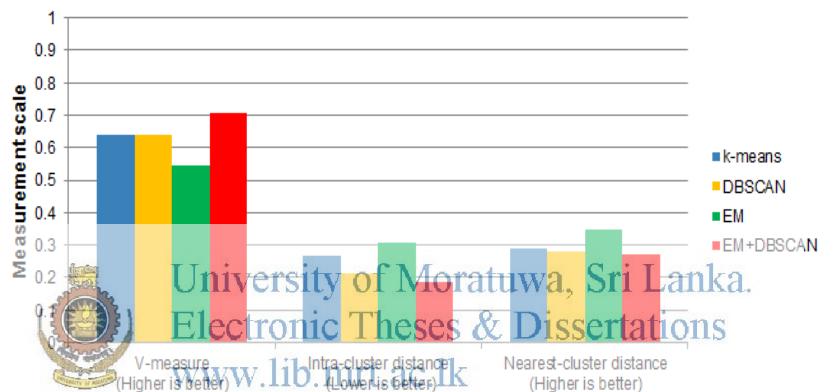


Figure 5.2: Evaluating cluster mechanisms and EM+DBSCAN for website U

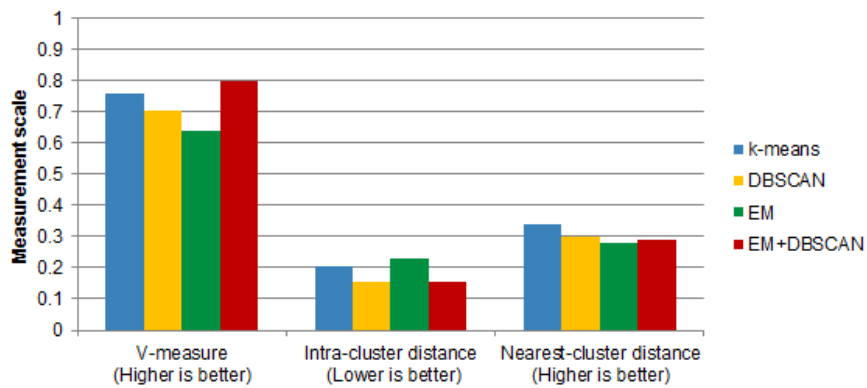


Figure 5.3: Evaluating cluster mechanisms and EM+DBSCAN for website F


5.3.2. Evaluating Cluster Signature Uniqueness

After clustering 30,000 sessions in website N, 15 clusters were identified. By using Frequent Pattern Mining (FPM), signatures were generated for 11 clusters and the remaining 4 clusters

used Sequence Pattern Mining (SPM) to generate their signatures, as shown in Table 5.3 for website N.

Table 5.3 uses standard regular expression notation and the repetitive count is given within square brackets. The two values within the square brackets represent the minimum and maximum values for the repetitive count. Page IDs are given within brackets. Further, coverage and discrimination values were used to describe the clusters. Coverage represents the percentage of sessions having the signature in a cluster and discrimination represents the percentage of sessions in other clusters not having the signature of the cluster. For a better signature, both coverage and discrimination values need to be high. In Table 5.3, except for the coverage value in cluster 10 and the discrimination value in cluster 8, others give a higher value. This implies that the signatures are unique.

Table 5.3: Cluster distribution and signature uniqueness

Cluster	Session %	Signature (FPM)	Signature (SPM)	Coverage	Discrimination
Cluster1	2.34%		('8302', '8356', '8298')	98%	92%
Cluster2	30.80%	( '14043', '14043')[1:4]		100%	82%
Cluster3	18.88%	7826', '1', '7826'		100%	87%
Cluster4	8.25%		('1' '14043')[2], '6686'	95%	97%
Cluster5	0.47%	7726'[2:4]		100%	95%
Cluster6	1.40%		('10447', '10639')	100%	100%
Cluster7	2.87%	8013', '14043'		100%	98%
Cluster8	18.91%	12344'[2:3] OR '12511' [3:5]		100%	69%
Cluster9	1.83%	1', '13571'		83%	100%
Cluster10	1.66%		6686, 17182,4944	74%	100%
Cluster11	2.66%		('8302', '8356', '8298')	93%	93%
Cluster12	4.79%	1', '8298', '1'		88%	100%
Cluster13	0.09%	10561', '8394'		100%	99%
Cluster14	1.04%	12796'[2:4]	('1' '14043')[2], '6686'	100%	89%
Cluster15	4.00%	17182'[3:5]		92%	97%

5.3.3. Evaluation of Effects of Temporal Website Changes

With time, all the websites change and evolve. The website administration needs to monitor and control user behavior on the system in a proactive manner. Here, by examining clusters and their signatures, the system is able to evaluate how far user behavior is affected by website changes. For example, training materials for school students were introduced into the non-profit organization website. The administration expects a change in the user behavior model because they expect more visitors to their website. Thereby, the administration can evaluate the success rate of the change done to the system by comparing the change in user behavior models, as shown in Figure 5.4. It represents a new cluster generation.

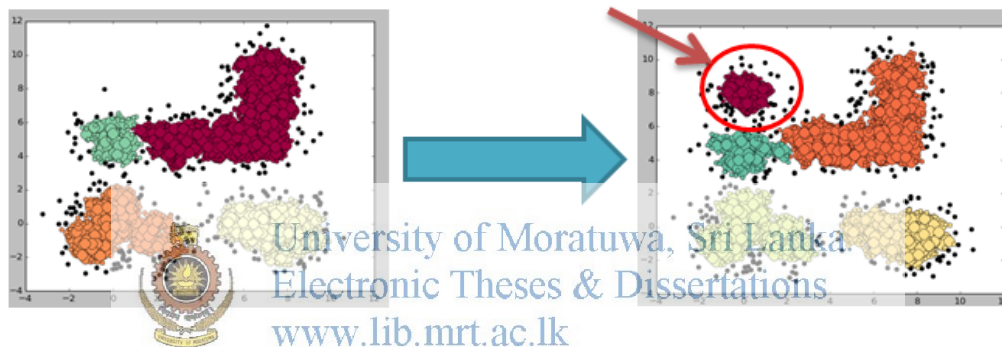


Figure 5.4: Effect of training session addition to the non-profit organization site

Other than the new cluster creation, splitting and merging behavior in the model is seen within the clusters. Those can happen due to changes in the user navigation structure such as the addition or deletion of a web page or a change in a UI component.

Figure 5.5 is generated using 30,000 sessions of website N web logs collected over a four month period. Sessions are plotted with the x-axis representing time in months and the y-axis representing clusters. There were 3 major navigation changes during the four month period, according to the website administrator. Using EM+DBSCAN all the 3 changes were clearly identified as shown in Figure 5.5. Numbers 1, 2, and 3 in Figure 8 represent the site menu change, hyperlink change and the new feature addition respectively.

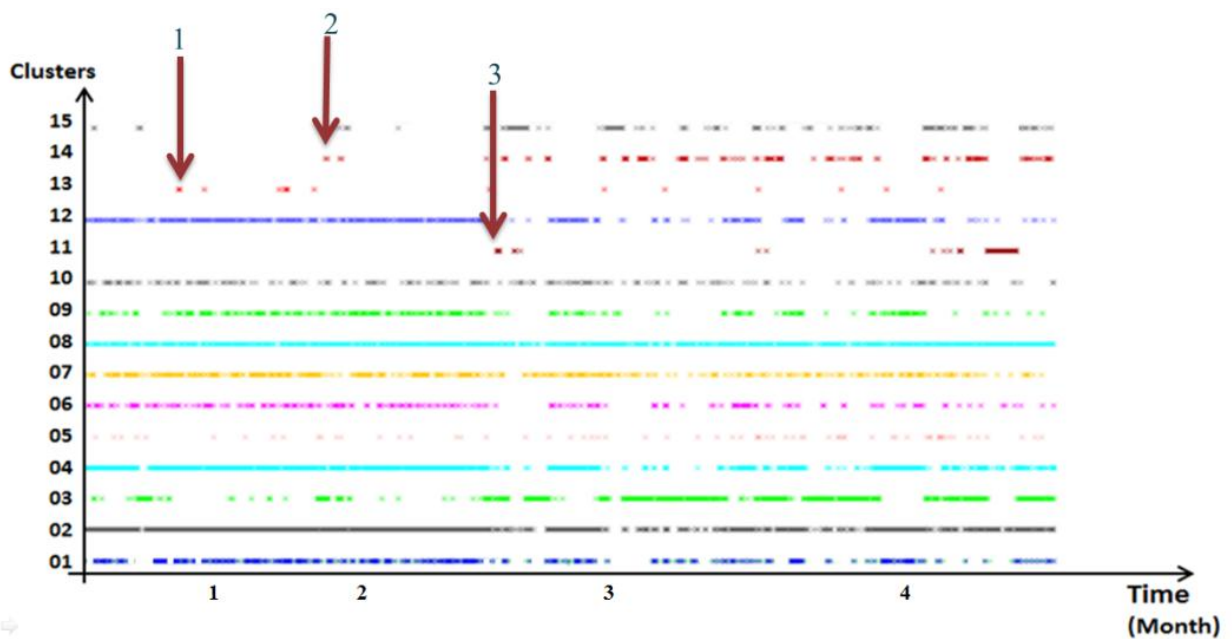


Figure 5.5: Detecting major changes in the website using EM+DBSCAN clustering

The clustering results using EM, DBSCAN, k-means (with the cluster count given by a domain expert and after finding the cluster count from EM+DBSCAN) are shown from Figure 5.6 - Figure 5.9.

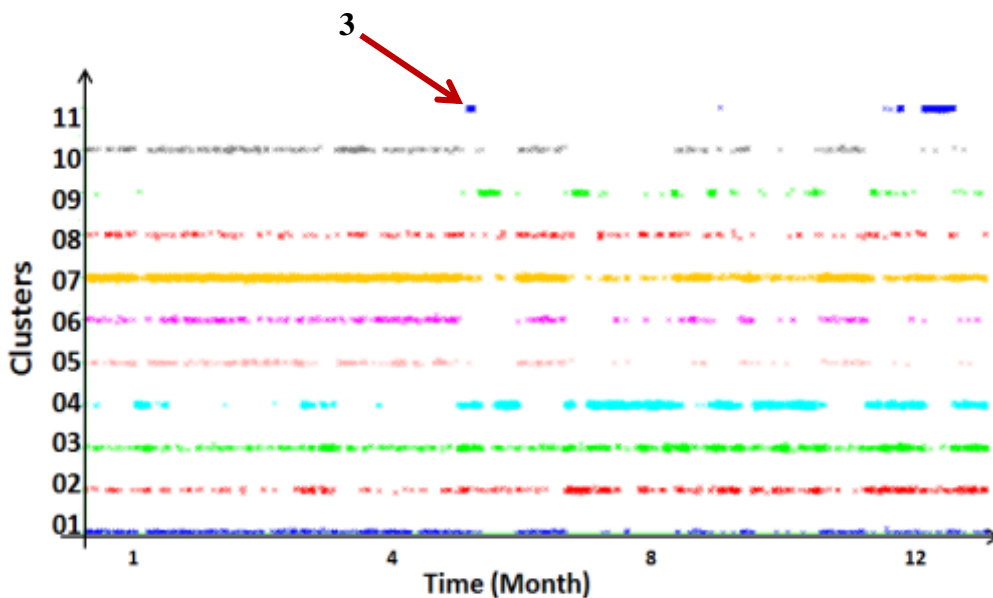


Figure 5.6: Detecting major changes in the website using DBSCAN

Figure 5.6 shows the results using DBSCAN on the aforementioned dataset. It only recognizes one change (cluster 11), out of three.

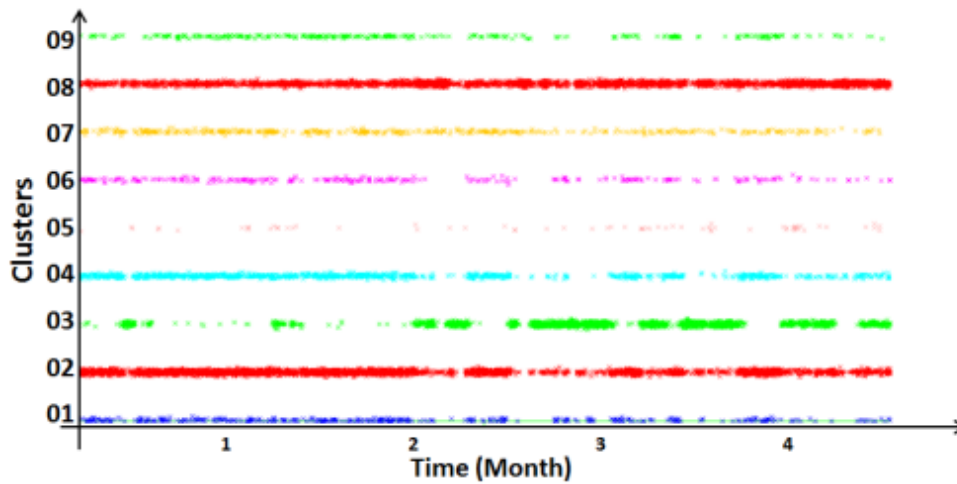


Figure 5.7: Detecting major changes in the website using k-means with domain expert

The output of k-means when run with domain expert given cluster count (9 clusters) is shown in Figure 5.7. It is unable to find any changes. Since these experiments were done for three sites we notice some domain experts also do not have accurate user group counts but later are able to identify better cluster counts (user group counts) for their websites by looking our cluster outputs.

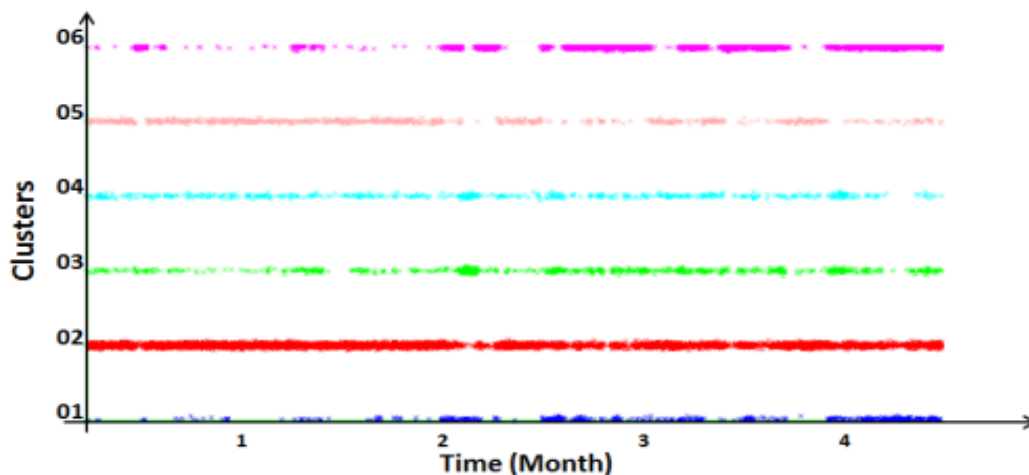


Figure 5.8: Detecting major changes in the website using EM

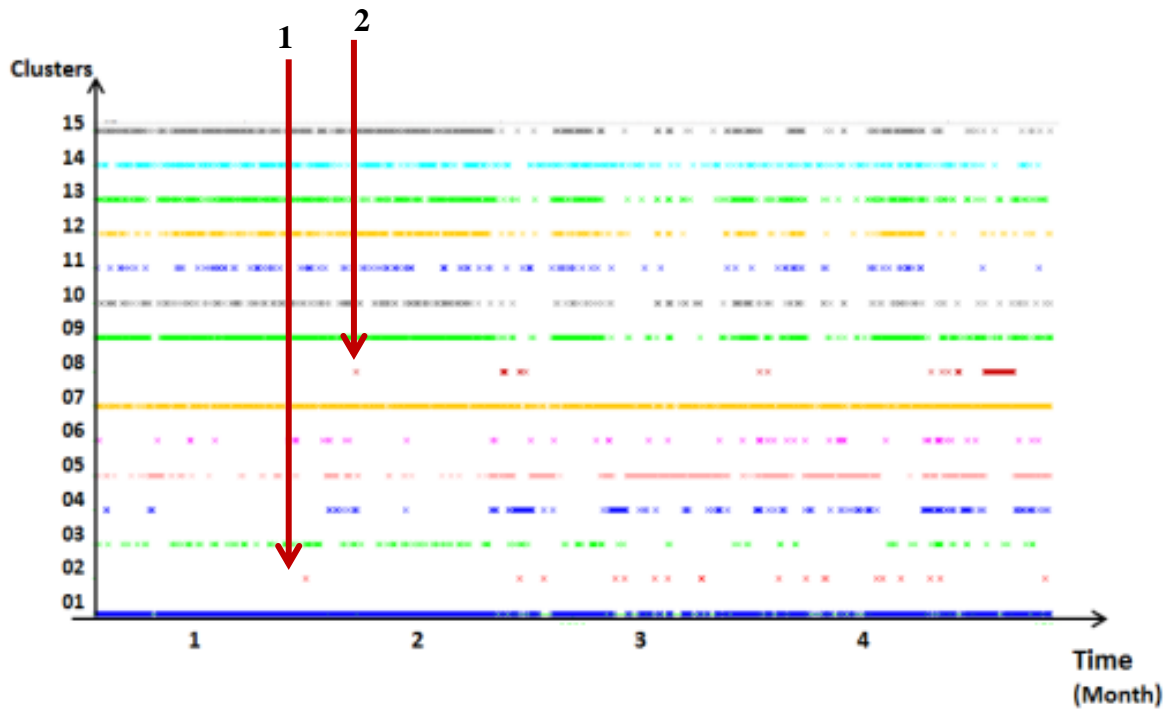


Figure 5.9: Detecting major changes in the website using K-means with a cluster count of 15



EM is unable to find any changes as shown in Figure 5.8. Figure 5.9 shows the output of k-means with the cluster count using EM+DBSCAN. It identifies two major changes (in clusters 2 and 8) out of three.

5.3.4. Demonstrating Social Media Impact on Site Access

A qualitative analysis of the impact of social media and attacks of the system is discussed below. A new cluster getting generated without any changes in the website is an indication of an anomaly or a social media impact. After clustering using EM+DBSCAN, social media impact, anomalies and attacks were distinguishably clustered together. In order to identify whether a particular anomaly or attack was caused by a malicious user, manual involvement is needed. Index database is used here to retrieve the log records for the sessions and clusters. By going through the log records (HTTP referrer and agent), it can be determined if the user is non-malicious or the request is coming from malware software. After verifying the HTTP referrer of the particular sessions in the newly generated clusters, malicious activities and malware are identified.

In Figure 5.10, cluster 3 clearly shows the density changes over time. The index database is queried for the log records corresponding to these sessions. The HTTP referrer confirmed that these accesses were initiated from a social media site. Therefore, it was identified that the high interaction on social media was a result of the website's social media page sharing the particular content on social media.

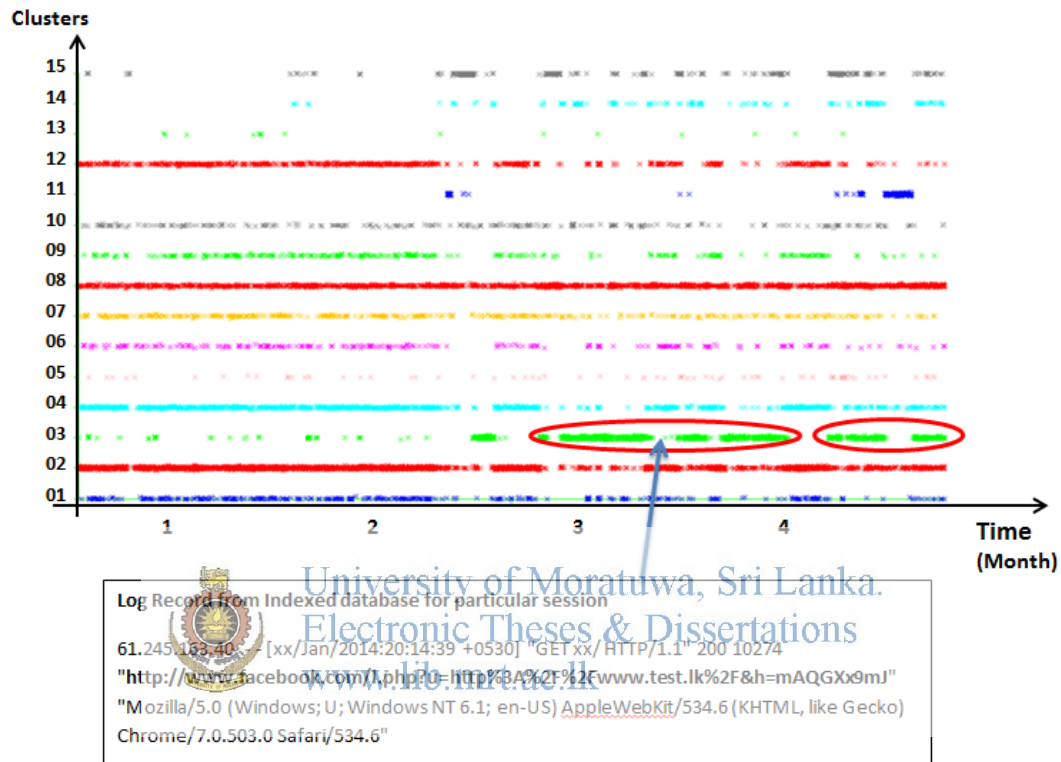


Figure 5.10: Impact of social media on the user behavior model

Figure 5.10 corresponds to the same dataset used to generate Figure 5.5. The effects of social media on the website accessing patterns were not clearly identified by the use of the DBSCAN algorithm and EM algorithm. By using the k-means algorithm with domain expert knowledge, the impact of social media on user behavior is identified in Figure 5.7 cluster 3. The effect is not seen when using the k-means algorithm, which used the cluster count from EM+DBSCAN.

5.3.5. Attack Detection

Clusters are built using the page occurrence matrix, which is a binary matrix. It represents a particular page existence in a user session. For experiments, we also used a frequency matrix that represented the number of times a page occurs in a session. The frequency matrix is built for each cluster. The clusters generated using the frequency matrices are shown in Figure

5.11. A new cluster is generated where there was no website change. Therefore, it is an interesting cluster and the corresponding web sessions were manually explored, which showed evidence that this cluster corresponds to an attack. The attack was confirmed manually using the index database. A drastic increase of the page occurrence count for one page was identified, which deviates from the normal user page access count. Time duration for a page access was less when compared with the same for a normal page access. The name of the HTTP agent, 'bit-torrent', also indicates that this is an attack.

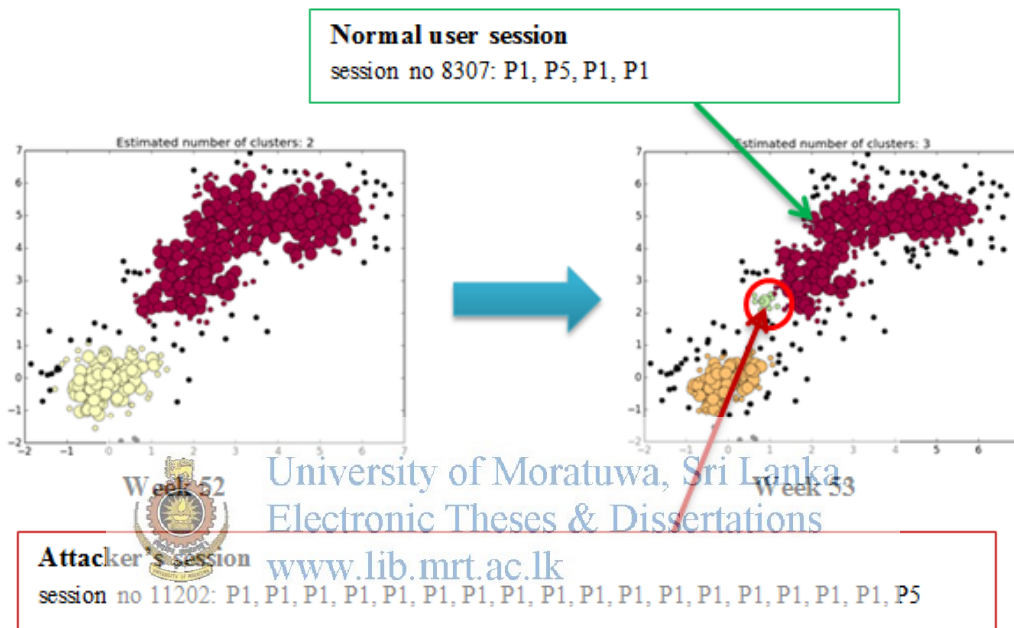


Figure 5.11: Generated new cluster that represents the sessions of an attack

5.4. Evaluation of the Episode based approach

Firstly, two quantitative experiments are discussed and secondly, a qualitative experiment is described in section 5.4.3.

5.4.1. Improving Clustering with Episodes

In the first experiment, we cluster the three sites U, F and N using DBSCAN and k-means. First, we run the algorithms with web page occurrences in the session. Then the algorithms are run with episode based sessions. Completeness is a standard way to verify the effectiveness of clustering mechanisms [125]. A clustering result satisfies the completeness if all the data points that are members of a given class are elements of the same cluster. This test was conducted only for website N for 1 month of sessions. Figure 5.12 shows the results

in a bar chart. We can see that our episode-based approach gives a better performance with respect to completeness.

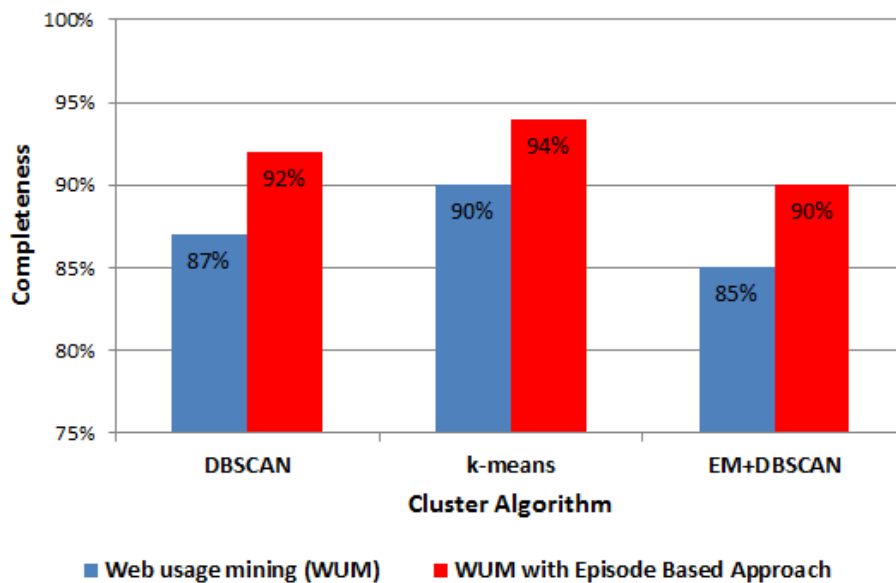


Figure 5.12: Comparing completeness of clustering algorithms with episodes

User sessions are built from log data of websites U, F and N but for this test, only website N was used as website F and U did not have a labeled dataset. Domain experts labeled the clusters and data records for this experiment purpose for website N. We ran the k-means and DBSCAN clustering several times with different parameters and got results with completeness. Figure 5.12 presents the best count for completeness for each cluster technique. A labeled data set is used to evaluate the system for completeness. First, clustering algorithms are run on a normal web usage mining dataset. Completeness score for DBSCAN, k-means and EM+DBSCAN was above 80%. Secondly, we pass a dataset that we gained from the episode based system and scores were above 90%. All algorithms results were improved as show in Figure 5.12.

Figure 5.13 denotes the variation of intra-cluster distances where a lower value means better clustering against clustering algorithms k-means, DBSACN, EM and EM+DBSCAN. General web sessions are represented in blue and web sessions with episodes are represented in red. The k-means algorithm gives a better result for web sessions with episodes with a lower value for intra-cluster distance over general web sessions. With episode session data, k-means, DBSCAN and EM+DBSCAN give lower values for inter-cluster distances, which is better. EM has not been improved as the cluster count is not fed to the algorithm. A slight

improvement is shown for the EM algorithm when it is run with a cluster count as shown in Figure 5.13.

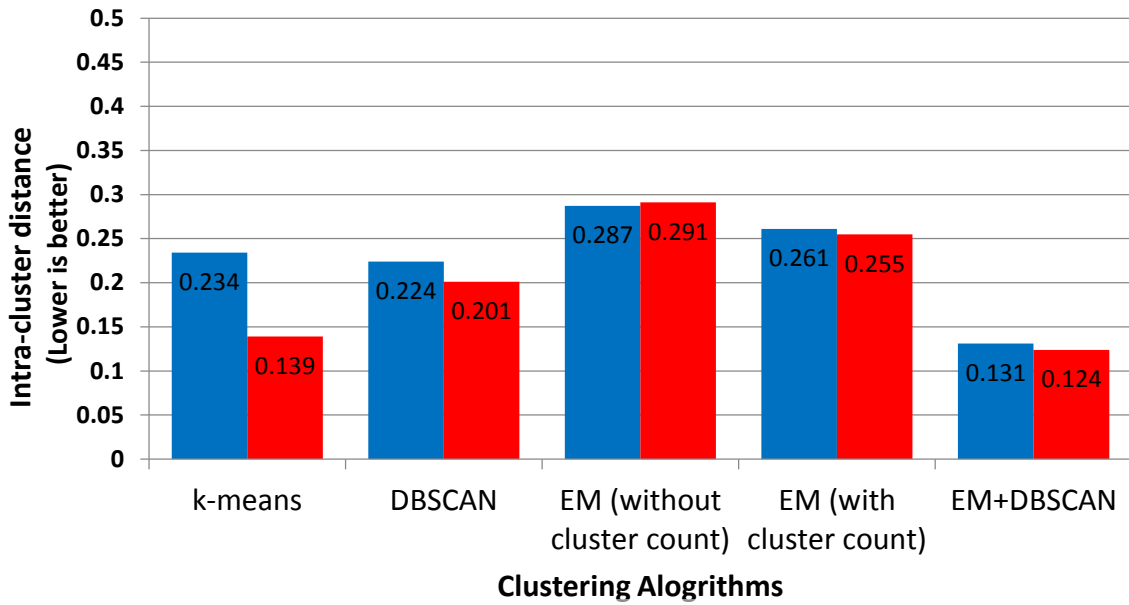
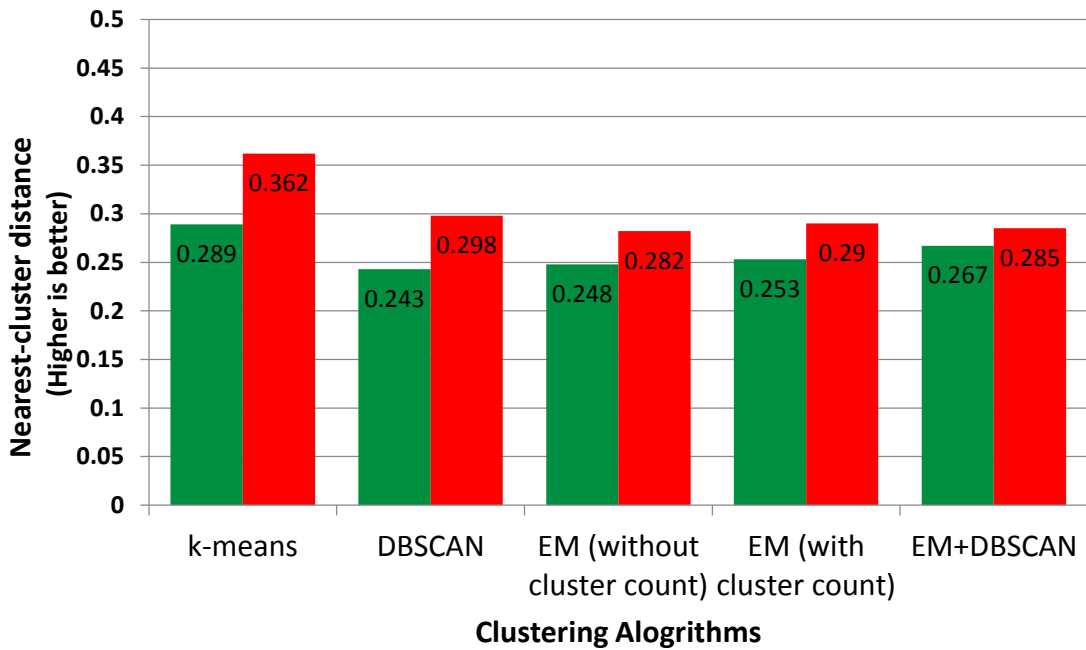


Figure 5.13 Intra-cluster distance of clusters (website N)



Legend: ■ General Web sessions ■ Web sessions with Episodes

Figure 5.14: Nearest-cluster distance of clusters (website N)

In Figure 5.14, the x-axis represents the clustering algorithms k-means, DBSCAN, EM and EM+DBSCAN and the y-axis represents the nearest-cluster distance where a higher value means better clustering. General web sessions are represented in green and web sessions with episodes are represented in red. The k-means algorithm gives a higher value for nearest-cluster distances for web sessions with episodes. The improvement of clustering using web sessions with episodes in descending order, respectively DBSCAN and EM, with EM+DBSCAN. EM+DBSCAN gives the lowest improvement of clustering on web sessions with episodes. When EM is executed without passing a cluster count, it is not that accurate, as explained in section 2.5.2.3. The aforementioned is the reason for the performance degradation in EM+DBSCAN.

5.4.2. Evaluation of Memory Usage

In the second experiment, we improved the suffix array by introducing the suffix count. This improvement reduced the length of the array. This is due to the array only containing unique suffixes with the session count. So we can identify the most common suffixes of all the sessions.

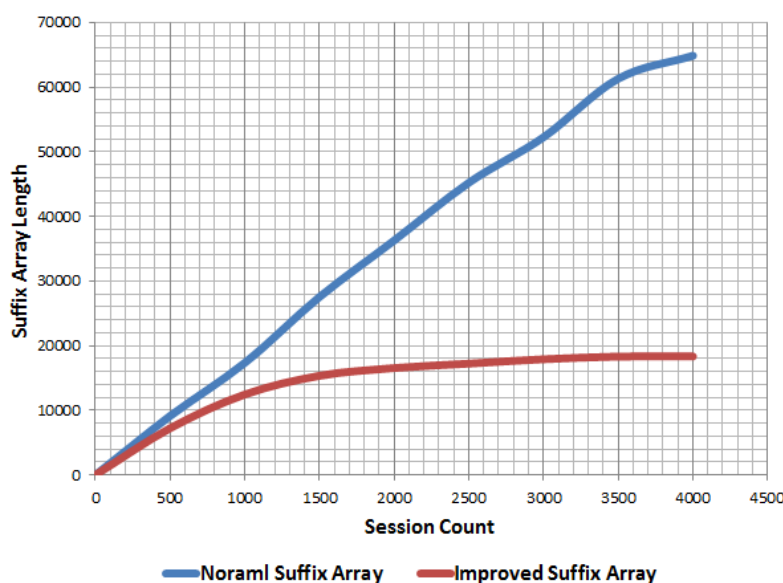


Figure 5.15: Suffix array length growth for the two suffix array versions in website N

As shown in Figure 5.15, the normal suffix array represents a linear relationship with session count and suffix array length. The improved suffix array comes to a steady state when the session count is increased. This reduces the machine memory usage as it contains less data respective to the normal suffix array.

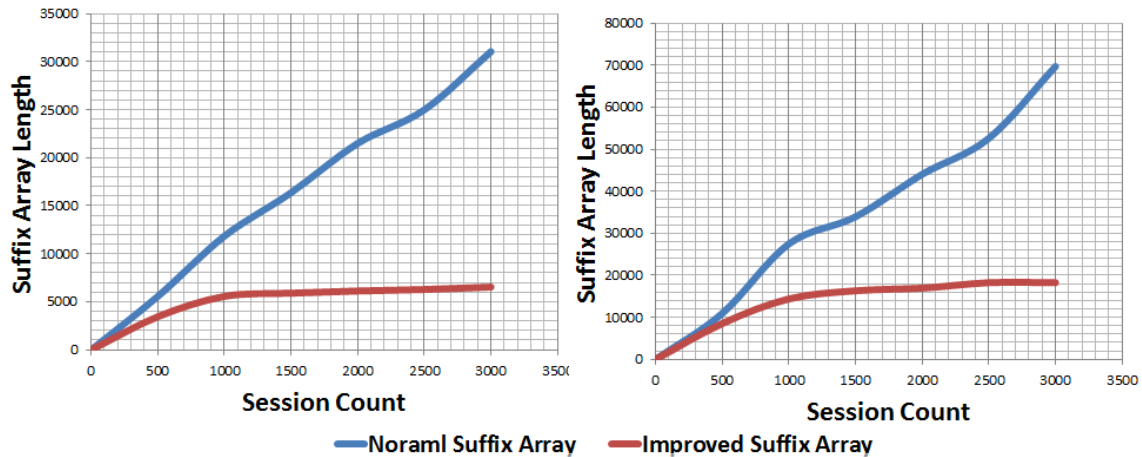


Figure 5.16: Suffix array length growth for the two suffix array versions in websites U (left) and F (right)

Figure 5.16 presents suffix length reductions by the improved version of suffix array. It improves system performance pattern retrieval with less memory usage.

5.4.3. Identifying Attacks on a Website

In the qualitative experiment, we demonstrate how we found two attacks and some other interesting user patterns in website N. Here, two detected website attacks are demonstrated.

We have a list of episodes with regex representations and each episode has the occurrence count stored in the improved suffix array. The most common episodes can be identified from the count. We can also find the most similar episodes for any given episode.

By looking at the count distribution, we can find interesting patterns. If a particular episode has a less count value in the suffix array, and the similarity between this episode and others is less, then this episode could most probably be an attack. If regex similarity is high, count is low and the episode repeating count is very high, this could also be an attack. After inspecting the particular session and log record references (where the user was previously browsing) we can confirm whether this is an attack or not. The time gap between web

requests sent can be used to confirm whether the interesting pattern belongs to a human user or a malicious tool.

Using the approach just discussed above, we detected some attacks, which are explained below. An attack was detected and the attacker's web session sequence was similar to a normal user access pattern. In Figure 5.17(a), a normal user access pattern contains 1 to 4 episodes repeating and in (b) the attacker pattern also repeats the same episode (page sequences) but the repeating count is very high (from 30 to 60). This attack was also detected in our first approach as explained in Section 5.3.5

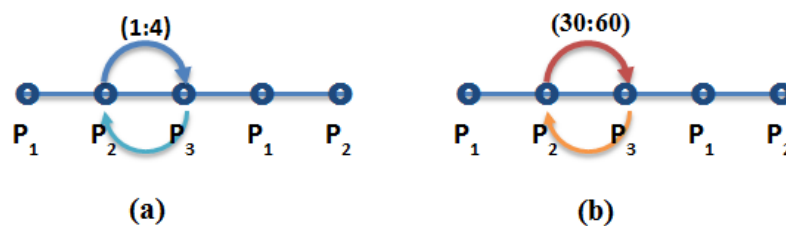


Figure 5.17: (a) Normal user pattern (b) Attacker pattern



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

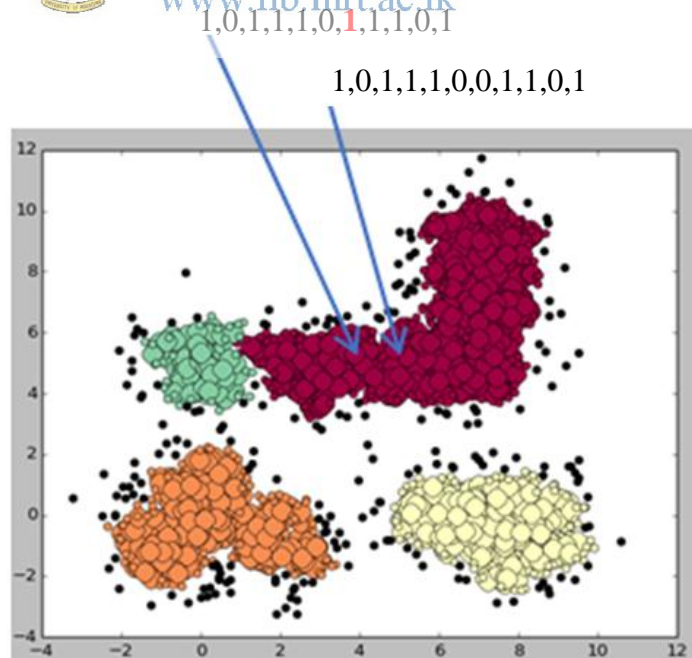


Figure 5.18: Slight change in cluster

Figure 5.18 represents two sessions that have very similar page occurrences but with a slight difference. Clustering techniques assigned these two in to one cluster. However, the change

is interesting. This can be an attacker pretending to be a normal user, which can be more harmful and harder to detect. This was not detected in our first approach.

The second attacker sends few requests to common pages such as the home page and contact page denoted by p_1 and p_2 as shown in Figure 5.19. Then he sends requests to admin pages (p_{23} , p_{24}) and login pages (p_{25}), which a normal user does not request in this sequence. Web pages p_{26} and p_{27} are also admin pages that normal website users do not access. He repeats this process again many times. Figure 5.19 shows the page request sequences.

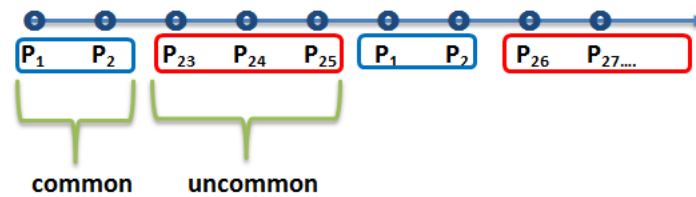


Figure 5.19: Sample web session attack

Therefore, we can see that the attackers camouflage themselves as common users but examining their user patterns can expose their malicious behavior.

5.4.4. Common User Patterns

Common user patterns are nicely highlighted and the access patterns are presented using the regex feature repetition and alternation.

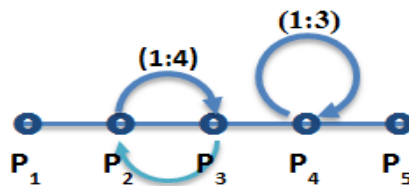


Figure 5.20: Search user access pattern

Figure 5.20 shows the search user access pattern where, P_1 = home page, P_2 = feature list page, P_3 = search page, P_4 = Result page, P_5 = result detail page. The user comes to P_1 (home page) and then moves to the feature list page, P_2 where it lists down website features. He picks the search feature from the feature list page and navigates to the search page, P_3 . He loops 1 to 4 times between pages P_2 and P_3 as shown in Figure 6.19. Afterwards he finds the interesting results, he sorts and filters the results in the result page, P_4 . The page P_4 is looped 1 to 3 times. He then finds the correct result and goes to the detail view of the result item P_5 . Figure 5.20 represents the regular expression by $P_1, (P_2, P_3)\{1:4\}, P_4\{1:3\}, P_5$

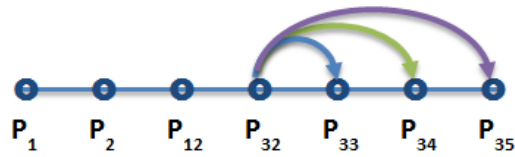


Figure 5.21: Article readers access patterns in website N

Figure 5.21 shows the access pattern of alternation where, P_{32} = article list, and P_{33} , P_{34} and P_{35} are article pages. He navigates through P_1 , P_2 , P_{12} and P_{32} . P_{32} is a page that lists articles. The user can select from the alternatives P_{33} , P_{34} and P_{35} , the article he prefers. $P_1, P_2, P_{12}, P_{32}(P_{33}|P_{34}|P_{35})$ is the regex of Figure 5.21.

From the experiments, we can conclude that for the non-profit organization's website, most users use only about 5% of the website functionality and some users loop through few of the web pages over and over again. This leads to a better understanding of human interaction and interfaces of a website. It brings about better designs in HCI and resolves navigational and other access issues in a website.



6. Discussion

This chapter discusses the contributions of this thesis, and possible improvements. A discussion of the implemented framework with respect to its accuracy, usability, and scalability is also provided.

6.1. Contributions

For web usage mining, modelling website user behavior from web access logs is a basic and important requirement. This research has been able to develop novel mechanisms based on regular expression and episodes. The system identifies the slight variations between user web sessions, which is one of the tough tasks in web usage mining and anomaly detection. The hybrid algorithm, EM+DBSCAN, solves the problem of cluster parameters and produces good results with respect to accuracy.

The contributions of this thesis are:

- An approach to detect website user common and uncommon access patterns from web access logs
- Implemented an episode based approach for web usage mining
- Introduces EM+DBSCAN to overcome the limitations in clustering algorithms
- The concept of regular expressions for web usage mining
- Improvement of web log preprocessing stage

The following paragraphs discuss each of these contributions.

This thesis presents two approaches to detect website user access patterns from web access logs, as discussed in Chapter 3 and Chapter 4. This system clusters the common access patterns in web session data. The system identifies unique signature for each cluster that groups the sessions into the same cluster. Signature coverage and discrimination values, which are presented in Table 5.3, show that signature uniqueness is high.

We implemented an episode based web usage mining system. Regular expression is used for the first time in web usage mining. Section 5.4 demonstrates results from the episode based approach and evaluates the results between normal web usage mining against episode based web usage mining. The figures in section 5.4.1 shows improved nearest-cluster distances and

intra-cluster distances. The preprocessing stage is also improved by the episode approach as it reduces session length.

EM+DBSCAN executed in web usage mining without passing parameters for clusters solves the cluster parameter issue described in section 2.5.2. This hybrid clustering technique identifies temporal effects on the website such as web page updates and social media impacts. It also detects web attacker sessions, as described in section 5.3.5. EM+DBSCAN is evaluated against existing clustering algorithms with real data in different domains with regard to web usage mining as described in section 5.3.1 and it shows that the accuracy and clustering has improved.

Regular expressions are used to represent an episode, which is a novel technique in web usage mining. Repetitions and alternations are used in regular expressions as explained in section 4.5 and web sessions with episodes with regular expressions are demonstrated in section 5.4.4.

Section 5.4.2 explains the improvement in the web log preprocessing module with the improved suffix array and regular expressions as it reduces the volume of the data used in web usage mining.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

6.2. Usability

The system is implemented in Python scripts and is a Python module. It can be installed in any machine that has Python installed. It is tested with an i3 (3rd generation) processor with 4 GB RAM. The preprocessing unit, which is explained in section 3.3, handles access logs by reading the directory configured in the system. The system generates output files with clustering results that is in a human readable format (CSV). The system does not contain a user interface, which could degrade the usability. This could be added as a future improvement.

6.3. Scalability

The Python module is developed in standard web usage mining architecture, as explained in section 2.3. It has followed the standard Python development style and all custom parameters are in the configuration file. Therefore, it can be deployed into a new server or PC easily. The system can be configured to execute just one module of web usage mining such as preprocessing only. EM+DBSCAN contains an iterating process and takes more time to

execute than other clustering algorithms. Improving the performances of EM+DBSCAN is something that can be done in the future.

6.4. Accuracy

As explained in Chapter 5, EM+DBSCAN has produced better results when compared to EM, DBSCAN and k-means. It handles the effects of temporal website changes compared to other clustering algorithms such as EM, DBSCAN and k-means, as explained in section 5.3.3. Section 5.3.4 demonstrated interesting findings such as social media impact on website access. The system detects slight changes between web user sessions and leads to detecting web attack sessions that are not detected in normal web usage mining clustering techniques, as explained in section 5.4.3. No system is perfect; therefore, there can be more interesting patterns that are not identified by the current system. There is room for improvement in web usage mining with respect to accuracy.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

7. CONCLUSION & FUTURE WORK

Many clustering algorithms have been used in web usage mining, and many of these clustering algorithms face the parameter problem. Most cluster algorithms need the number of clusters in the data set or minimum instance count for density region. All those parameters have a massive influence on each algorithm and its results. In web usage mining, it is hard to know the exact values or predict values for those parameters and data is presented as user navigation patterns. Values for parameters fluctuate with time as websites are updated and it is hard to handle the clustering algorithm in an effective manner since those algorithms are bound to those parameter values.

The second problem in web usage mining is that there are many methodologies to identify user access patterns. These include clustering, Apriori algorithms, web access pattern tree (WAP-tree) and mining frequent patterns. However, with these methods, it is difficult to determine how each session deviates from other sessions within the same group or cluster. It is important to find these slight differences between sessions. These seemingly insignificant changes could be most important for a domain expert as they may resemble anomalies such as an attack.



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The first approach solves the web usage mining problem with regard to the clustering techniques, by combining two well-known clustering algorithms to overcome their drawbacks and to come up with an optimal result for web log clustering. This hybrid clustering algorithm stands ahead of k-means in clustering the web logs. To execute the k-means, we need to know the cluster count. The EM+DBSCAN is a novel method of mining web logs and it gives an accuracy improvement over k-means.

User access patterns and user behavior are the main outputs of the system. After every site update, the administrators can verify if the system has achieved its intended goals by looking at the latest trends of the user navigation models. Therefore, the method can be used to improve the HCI and navigations across the website. It can also give an indication of how far the core features of the site are used. Detecting attacks, identifying the normal user model and social media impact evaluation are some of the other uses of the system.

As future work, a performance improvement is required. Sampling the dataset and eliminating the iterations are some of the possible approaches for performance improvement.

Weblog cleaning, clustering and signature generation are fully automated. However, the attack detection process is a semi-manual system with index database queries. Therefore, the attack detection should also be automated.

The second approach is called episode-based approach. This thesis presented a regular expression based approach to identify website access patterns of users, which is a novel mechanism. Although regular expressions have been commonly used in many other domains to identify string patterns, this is the first time they are applied in identifying website access patterns. The use of regular expressions not only identifies the common access patterns, but also the rare access patterns, which could refer to anomalies such as attacks. We demonstrated that the completeness, intra-cluster and nearest-cluster of clusters with this episode based approach are considerably higher in some popular clustering algorithms (k-means, DBSCAN, EM and EM+DBSCAN).

Currently, we have not considered the ‘not’ notation in regex. Therefore, we cannot include them in our representations. In future, we are going to implement the ‘not’ notation in our solution and cover all meta characters in regex such as [^], ? and \.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

REFERENCES


- [1] J. Srivastava and R. Cooley, "Web usage mining: Discovery and applications of usage patterns from web data," *ACM SIGKDD*, pp. 12–23, 2000.
- [2] P. Chwalinski, "Detection of Unsolicited Web Browsing with Clustering and Statistical Analysis," University of Middlesex, London, 2013.
- [3] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, "Discovery and evaluation of aggregate usage profiles for Web personalization," *Data Min. Knowl. Discov.*, vol. 6, no. 1, pp. 61–82, 2002.
- [4] Sunena and Kamaljit kaur, "Web Usage Mining-Current Trends and Future Challenges," in *International Conference on Electrical, Electronics, and Optimization Techniques 2016*, 2016.
- [5] F. M. Facca and P. L. Lanzi, "Mining interesting knowledge from weblogs: A survey," *Data Knowl. Eng.*, vol. 53, no. 3, pp. 225–241, 2005.
- [6] R. Kosala and H. Blockeel, "Web Mining Research: A Survey," *ACM Sigkdd Explor. Newsl. 2.1*, vol. 2, no. 1, 2000.
- [7] C. Ghezzi, M. Pezzè, M. Sama, and G. Tamburrelli, "Mining behavior models from user-intensive web applications," *Proc. 36th Int. Conf. Softw. Eng. - ICSE 2014*, pp. 277–287, 2014.
- [8] N. M. Khairudin, A. Mustapha, and M. H. Ahmad, "Effect of Temporal Relationships in Associative Rule Mining for Web Log Data," *Sci. World J.*, vol. 2014, 2014.
- [9] J. Pei, J. Han, B. Mortazaviasl, and H. Zhu, "Mining Access Patterns Efficiently from Web," *Knowl. Discov. Data Mining. Curr. Issues New Appl.*, 1999.
- [10] K. S. Y. Fu, M.Y. Shih, "Clustering of Web Users Based on Access Patterns," *KDD Work. Web Mining. San Diego, CA. Springer-Verlag*, 1999.
- [11] Lukas Cenovsky, "Web Usage Mining on is.muni.cz," Masaryk University, 2003.
- [12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, p. 74, 2009.
- [13] B. Aaron and D. E. Tamir, "Dynamic Incremental K-means Clustering," in *Computational Science and Computational Intelligence (CSCI)*, 2014, pp. 308–313.
- [14] W. Wang and O. R. Zaiane, "Clustering Web sessions by sequence alignment," *13th Int. Work. Database Expert Syst. Appl. 2002. Proc.*, pp. 394–398, 2002.
- [15] R. Cooley, B. Mobasher, and J. Srivastava, "Web mining: Information and pattern discovery on the world wide web," *Tools with Artif. Intell. 1997. Proceedings., Ninth IEEE Int. Conf.*, no. DECEMBER 1997, pp. 558–567, 1997.
- [16] M. Etzioni and M. Perkowit, "Adaptive Web Sites: Automatically Synthesizing Web Pages," *Proc. AAAI*, 1998.

- [17] D. Ngu and X. Wu, "Sitehelper: A localized agent that helps incremental exploration of the world wide web," *Comput. Networks ISDN Syst.*, 1997.
- [18] T. Joachims, D. Freitag, and T. Mitchell, "Webwatcher: A tour guide for the world wide web," *IJCAI (1)*, 1997.
- [19] P. Kumar, R. Bapi, and P. Krishna, "SeqPAM: a sequence clustering algorithm for Web personalization," *International J. Data Warehous. Min.*, 2007.
- [20] M. Eirinaki and M. Vazirgiannis, "Web mining for web personalization," *ACM Trans. Inter. Tech.*, vol. 3, no. 1, pp. 101–127, 2003.
- [21] Y. Xie and S. Tang, "Online Anomaly Detection Based on Web Usage Mining," in *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, 2012, pp. 1177–1182.
- [22] H. Liu, Z. Huang, H. Zhang, and J. Lv, "A Security Analysis of Web Users' Behavior Based on Web Log Mining," pp. 89–92, Nov. 2013.
- [23] P. Wang, X. Ma, and J. Yu, "An Effective Network Security Log Mining Algorithm based on Fuzzy Clustering," *Appl. Math. Inf. Sci.*, vol. 1, no. 1, pp. 307–315, 2016.
- [24] M. Rai and V. Mishra, "Detection of UDP and HTTP Anomalies on Real Time Traffic Based on NIDS using OURMON Tool," *Int. J. Sci. Res. Sci. Eng. Technol.*, 2015.
- [25] A. Tang, S. Sethumadhavan, and S. Stolfo, "Unsupervised anomaly-based malware detection using hardware features," *Res. Attacks, Intrusions Defenses. Springer Int. Publ.*, pp. 109–129, 2014.
- [26] A. Juan-Verdejo and H. Baars, "Decision support for partially moving applications to the cloud: the example of business intelligence," in *Proceedings of the 2013 international workshop on Hot topics in cloud services. ACM*, 2013.
- [27] A. Abraham, "i-miner: A web usage mining framework using hierarchical intelligent systems," in *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, 2003.
- [28] D. Martin, "Abstract for the Invited Talk: The IBM SurfAid Project: Transactive Analysis and Prediction," 1999.
- [29] R. W. Cooley, "Web Usage Mining: Discovery and Application of Interesting Patterns from web Data," University of Minnesota, 2000.
- [30] V. Losarwar and M. Joshi, "Data Preprocessing in Web Usage Mining," *Int. Conf. Artif. Intell. Embed. Syst.*, 2012.
- [31] T. Aye, "Web log cleaning for mining of web usage patterns," *Comput. Res. Dev. (ICCRD), 2011 3rd Int. Conf. on. IEEE*, vol. 2, 2011.
- [32] S. Alsbaugh, A. Ganapathi, M. Hearst, and R. Katz, "Better Logging to Improve Interactive Data Analysis Tools," *ACM SIGKDD Work. Interact. Data Explor. Anal.*, 2014.

- [33] A. Nanopoulos, Y. Manolopoulos, M. Zakrzewicz, and T. Morzy, "Indexing web access-logs for pattern queries," *Proc. fourth Int. Work. Web Inf. data Manag. - WIDM '02*, p. 63, 2002.
- [34] R. Cooley, B. Mobasher, and J. Srivastava, "Data Preparation for Mining World Wide Web Browsing Patterns," *Knowl. Inf. Syst.*, vol. 1, no. 1, pp. 5–32, Jul. 1999.
- [35] P. Sharma, S. Yadav, and B. Bohra, "A review study of server log formats for efficient web mining," in *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on. IEEE*, 2015, pp. 1373–1377.
- [36] P. Nithya and P. Sumathi, "An enhanced pre-processing technique for web log mining by removing web robots," in *2012 IEEE International Conference on Computational Intelligence and Computing Research*, 2012, pp. 1–4.
- [37] N. Algiriyage, S. Jayasena, G. Dias, A. Perera, and K. Dayananda, "Identification and characterization of crawlers through analysis of web logs," in *2013 IEEE 8th International Conference on Industrial and Information Systems*, 2013, pp. 150–155.
- [38] L. Zheng, H. Gui, and F. Li, "Optimized data preprocessing technology for web log mining," *Comput. Des. Appl. (ICCD), 2010 Int. Conf. on. IEEE*, vol. 1, 2010.
- [39] J. Chang-bin and C. Li, "Web Log Data Preprocessing Based On Collaborative Filtering," *Educ. Technol. Comput. Sci.*, vol. 2, 2010.
- [40] Y. Li, B. Feng, and Q. Mao, "Research on path completion technique in web usage mining," *Comput. Sci. Comput. Technol. 2008. ISCSCIT 08. Int. Symp. on. IEEE*, vol. 1, 2008.
- [41] T. Hussain, S. Asghar, and N. Masood, "Hierarchical sessionization at preprocessing level of WUM based on swarm intelligence," *Emerg. Technol. (ICET), 6th Int. Conf. on. IEEE*, 2010.
- [42] D. Tanasa and B. Trousse, "Advanced data preprocessing for intersites web usage mining," *Intell. Syst. IEEE*, 2004.
- [43] Y. Fang and Z. Huang, "A session identification algorithm based on frame page and pagethreshold," *Proc. - 2010 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol. ICCSIT 2010*, vol. 6, pp. 645–647, 2010.
- [44] H. Sha, T. Liu, P. Qin, Y. Sun, and Q. Liu, "EPLogCleaner : Improving Data Quality of Enterprise Proxy Logs for Efficient Web Usage Mining," vol. 17, pp. 812–818, 2013.
- [45] J. Vellingiri and S. C. Pandian, "A Novel Technique for Web Log mining with Better Data Cleaning and Transaction Identification," vol. 7, no. 5, pp. 683–689, 2011.
- [46] B. Thakur, S. Abbas, M. Beg, and S. Rizvi, "Preprocessing of web usage data for log analysis," *Int. J. Sci. Eng. Res.*, pp. 1773–1779, 2015.
- [47] M. Spiliopoulou, "A framework for the evaluation of session reconstruction heuristics in web-usage analysis," *Inform. J. Comput.* 15.2, pp. 171–190, 2003.

- [48] M. Baglioni, U. Ferrara, A. Romei, S. Ruggieri, and F. Turini, "Preprocessing and Mining Web Log Data for Web Personalization," *AI* IA 2003 Adv. Artif. Intell. Springer Berlin Heidelberg.*, pp. 237–249, 2003.
- [49] P. Langley, "User modeling in adaptive interfaces," *Proc. Seventh Int. Conf. User Model.*, 1999.
- [50] N. Singh, A. Jain, and R. Raw, "Comparison analysis of web usage mining using pattern recognition techniques," *Int. J. Data Min. Knowl. Manag. Process 3.4*, pp. 137–158, 2013.
- [51] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Rec.*, vol. 22, no. May, pp. 207–216, 1993.
- [52] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: current status and future directions," *Data Min. Knowl. Discov.*, vol. 15, no. 1, pp. 55–86, Jan. 2007.
- [53] X. Huang, A. An, and N. Cercone, "Comparison of interestingness functions for learning web usage patterns," *Int. Conf. Inf. Knowl. Manag. Proc.*, no. March, pp. 617–620, 2002.
- [54] B. Mortazavi-Asl, "Discovering and mining user web-page traversal patterns," Simon Fraser University, 2001.
- [55] R. Srikant and R. Agrawal, "Mining sequential patterns: generalizations and performance improvements," *Proc. 5th Int. Conf. Extending Database Technol.*, pp. 3–17, 1996.
- [56] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic personalization based on web usage mining," *Commun. ACM*, 2000.
- [57] D. Stevanovic, N. Vlajic, and A. An, "Unsupervised clustering of web sessions to detect malicious and non-malicious website users," *Procedia Comput. Sci.*, vol. 5, pp. 123–131, 2011.
- [58] M. Ester, H. Kriegel, X. Xu, and D. Miinchen, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Kdd. Vol. 96*, 1996.
- [59] S. Kisilevich, F. Mansmann, and D. Keim, "P-DBSCAN: A density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos," *Proc. 1st Int. Conf. Exhib. Comput. Geospatial Res. Appl. - COM.Geo '10*, p. 1, 2010.
- [60] A. Denton, "Kernel-density-based clustering of time series subsequences using a continuous random-walk noise model," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, pp. 122–129, 2005.
- [61] M. Parimala, D. Lopez, and N. Senthilkumar, "A survey on density based clustering algorithms for mining large spatial databases," *Int. J. Adv. Sci. Technol. 31.1*, vol. 31, pp. 59–66, 2011.
- [62] C. Fraley and A. E. Raftery, "How Many Clusters? Which Clustering Method?"

- Answers Via Model-Based Cluster Analysis,” *Comput. J.*, vol. 41, no. 8, pp. 578–588, 1998.
- [63] N. Mustapha, M. Jalali, A. Bozorgniya, and M. Jalali, “Navigation Patterns Mining Approach based on Expectation Maximization Algorithm,” *Eur. J. Sci. Res.*, vol. 3, no. 2, pp. 855–859, 2009.
- [64] R. C. Barros, M. P. Basgalupp, A. C. P. L. F. de Carvalho, and M. G. Quiles, “Clus-DTI: Improving decision-tree classification with a clustering-based decision-tree induction algorithm,” *J. Brazilian Comput. Soc.*, vol. 18, no. 4, pp. 351–362, 2012.
- [65] L. Bolelli, S. Ertekin, D. Zhou, and C. L. Giles, “K-SVMMeans: A Hybrid Clustering Algorithm for Multi-Type Interrelated Datasets,” *IEEE/WIC/ACM Int. Conf. Web Intell.*, pp. 198–204, Nov. 2007.
- [66] H. Frigui, “SyMP : An Efficient Clustering Approach to Identify Clusters of Arbitrary Shapes in Large Data Sets,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM*, 2002.
- [67] Y. Xie and V. V. Phoha, “Web user clustering from access log using belief function,” *Proc. Int. Conf. Knowl. capture - K-CAP 2001*, no. JANUARY 2001, p. 202, 2001.
- [68] A. Banerjee and J. Ghosh, “Clickstream clustering using weighted longest common subsequences,” *Proc Work. Web Min. SIAM Conf. Data Min.*, pp. 33–40, 2001.
- [69] Y. Chen, “Improving user profiles for e-commerce by genetic algorithms,” *Stud. Fuzziness Soft.*, vol. 105, pp. 5–6, 2002.
- [70] S. Oyanagi, K. Kubota, and A. Nakase, “Application of Matrix Clustering to Web Log Analysis and Access Prediction,” *Proc. WEBKDD.*, vol. Vol. 1, 2001.
- [71] S. Samangoei, J. Hare, D. Dupplaw, M. Niranjana, N. Gibbins, and P. Lewis, “Social event detection via sparse multi-modal feature selection and incremental density based clustering,” *CEUR Workshop Proc.*, vol. 1043, pp. 18–19, 2013.
- [72] L. Chaofeng, “Research on Web Session Clustering,” *J. Softw. 4.5*, vol. 4, no. 5, pp. 460–468, 2009.
- [73] J. S. Cooley, Robert, Pang-Ning Tan, “Discovery of interesting usage patterns from web data,” *Web Usage Anal. User Profiling. Springer Berlin Heidelb.*, pp. 163–182, 1999.
- [74] S. Chakraborty and N. K. Nagwani, “Analysis and Study of Incremental DBSCAN Clustering Algorithm,” *Int. J. Enterp. Comput. Bus. Syst.*, vol. 1, no. 2, 2011.
- [75] K. Mumtaz, “A Novel Density based improved k-means Clustering Algorithm–Dbkmeans,” *Int. J.*, vol. 02, no. 02, pp. 213–218, 2010.
- [76] M. Eirinaki, C. Lampos, S. Paulakis, and M. Vazirgiannis, “Web personalization integrating content semantics and navigational patterns,” *Proc. 6th Annu. ACM Int. Work. Web Inf. data Manag. WIDM 04*, no. JANUARY, p. 72, 2004.

- [77] Z. Ansari, W. Ahmed, M. F. Azeem, and A. V. Babu, “Discovery of Web Usage Profiles Using Various Clustering Techniques,” vol. 1, no. 3, pp. 18–27, 2011.
- [78] M. Eirinaki, M. Vazirgiannis, and I. Varlamis, “SEWeP,” *Proc. ninth ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '03*, no. JANUARY, p. 99, 2003.
- [79] P. Fournier-viger, C. Wu, V. S. Tseng, L. Cao, and R. Nkambou, “Mining Partially-Ordered Sequential Rules Common to Multiple Sequences,” vol. 27, no. 8, pp. 2203–2216, 2015.
- [80] R. Agrawal and R. Srikant, “Mining sequential patterns,” in *Proceedings of the Eleventh International Conference on Data Engineering*, 1995.
- [81] A. I. V. Mannila, Heikki, Hannu Toivonen, “Discovering frequent episodes in sequences Extended abstract,” in *The first Conference on Knowledge Discovery and Data Mining*, 1995.
- [82] J. Pei, J. Han, Q. Chen, M.-C. Hsu, B. Mortazavi-Asl, H. Pinto, and U. Dayal, “PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth,” *Int. Conf. Data Eng.*, pp. 215 – 224, 2001.
- [83] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M. C. Hsu, “Mining sequential patterns by pattern-growth: The prefixspan approach,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 11, pp. 1424–1440, 2004.
- [84] E. Menasalyas, S. Millán, J. M. Peña, M. Hadjimichael, and O. Marbán, “Subsessions: A granular approach to click path analysis,” *Int. J. Intell. Syst.*, vol. 19, no. 7, pp. 619–637, 2004. 
- [85] P. K. Alkan, Ozgur Kirmemis, “Assisting Web Site Navigation Through Web Usage Patterns,” *Recent Trends Appl. Artif. Intell. Springer Berlin Heidelberg, 2013.*, pp. 161–170, 2013.
- [86] K. Bommepally, T. K. Glisa, J. J. Prakash, S. R. Singh, and H. a Murthy, “Internet activity analysis through proxy log,” *2010 Natl. Conf. Commun.*, pp. 1–5, Jan. 2010.
- [87] S. Chakrabarti, “Data mining for hypertext: A tutorial survey,” *SIGKDD Explor.*, vol. 1, no. 2, pp. 1–11, 2000.
- [88] A. Jain and R. Dubes, “Algorithms for clustering data,” 1988.
- [89] S. Guha, R. Rastogi, and K. Shim, “ROCK: A robust clustering algorithm for categorical attributes,” *Data Eng. 1999. Proceedings., 15th Int. Conf. on. IEEE*, 1999.
- [90] K. a. Smith and A. Ng, “Web page clustering using a self-organizing map of user navigation patterns,” *Decis. Support Syst.*, vol. 35, no. 2, pp. 245–256, May 2003.
- [91] M. A. W. Hartigan, John A., “A K-Means Clustering Algorithm,” *Appl. Stat.*, pp. 100–108, 1979.
- [92] Z. He, X. Xu, and S. Deng, “Discovering cluster-based local outliers,” *Pattern Recognit. Lett.*, 2003.

- [93] A. Chaudhary, A. Szalay, and A. Moore, "Very Fast Outlier Detection in Large Multidimensional Data Sets.," *DMKD*, 2002.
- [94] A. Pires and C. Santos-Pereira, "Using clustering and robust estimators to detect outliers in multivariate data," *Proc. Int. Conf. Robust Stat.*, 2005.
- [95] H. Sun, Y. Bao, F. Zhao, G. Yu, and D. Wang, "CD-trees: An efficient index structure for outlier detection," *Adv. Web-Age Inf. Manag. Springer Berlin Heidelberg.*, pp. 600–609, 2004.
- [96] G. Carl, G. Kesidis, R. Brooks, and S. Rai, "Denial-of-service attack-detection techniques," *Internet Comput. IEEE*, 2006.
- [97] B. Schölkopf, J. Platt, and J. Shawe-Taylor, "Estimating the support of a high-dimensional distribution," *Neural Comput.* 13.7, pp. 1443–1471, 2001.
- [98] V. Roth, "Kernel fisher discriminants for outlier detection," *Neural Comput.*, 2006.
- [99] F. J. Anscombe and I. Guttman, "Rejection of Outliers," *Technometrics*, vol. 2, no. 2, pp. pp. 123–147, 1960.
- [100] H Motulsky, "Choosing a statistical test," in *Intuitive Biostatistics*, Oxford University Press Inc, 1995, pp. 1–5.
- [101] A. Kind, M. Stoecklin, and X. Dimitropoulos, "Histogram-based traffic anomaly detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 6, no. 2, pp. 110–121, 2009.
- [102] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.
- [103] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan, "Using artificial anomalies to detect unknown and known network intrusions," *Knowl. Inf. Syst.* 6.5, pp. 507–527, 2004.
- [104] P. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. 2006.
- [105] and G. N. Hipp, Jochen, Ulrich Güntzer, "Algorithms for Association Rule Mining – A General Survey and Comparison," *ACM sigkdd Explor. Newsl.* 2.1, pp. 58–64, 2000.
- [106] M. Qin and K. Hwang, "Frequent episode rules for internet anomaly detection," *Netw. Comput. Appl. (NCA 2004). Proceedings. Third IEEE Int. Symp. on. IEEE*, 2004.
- [107] S. Cho and S. Cha, "SAD: web session anomaly detection based on parameter estimation," *Comput. Secur.*, vol. 23, no. 4, pp. 312–319, 2004.
- [108] U. Manber and G. Myers, "Suffix arrays: A new method for on-line string searches," 1991.
- [109] T. Kasai, G. Lee, H. Arimura, S. Arikawa, and K. Park, "Computation in Suffix Arrays and Its Applications," *Comb. pattern matching. Springer Berlin Heidelberg.*, pp. 181–192, 2001.

- [110] D. E. Goldberg, *Genetic Algorithms in Search , Optimization , and Machine Learning*. Pearson Education India, 2006.
- [111] K. R. R. Makwana, Chintan H., “An Efficient Technique for Web Log Preprocessing using Microsoft Excel,” *Int. J. Comput. Appl.*, vol. 90, no. 12, pp. 25–28, 2014.
- [112] T. Jing, W.-L. Zuo, and B.-Z. Zhang, “An efficient Web traversal pattern mining algorithm based on suffix array,” in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, vol. 4, pp. 1535–1539.
- [113] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, “Spamming botnets,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, p. 171, 2008.
- [114] M. D. Tianyi Liang, Andrew Reynolds, Cesare Tinelli, Clark Barrett, “A DPLL (T) Theory Solver for a Theory of Strings and Regular Expressions,” *Comput. Aided Verif. Springer Int. Publ.*, pp. 1–22, 2014.
- [115] R. Cox, “Regular Expression Matching in the Wild,” no. March, 2010.
- [116] M. Becchi and P. Crowley, “Extending finite automata to efficiently match Perl-compatible regular expressions,” *Proc. 2008 ACM Conex. Conf. - Conex. '08*, no. March, pp. 1–12, 2008.
- [117] G. Skinner, “RegExr,” 2011. [Online]. Available: <http://regexpr.com/>. [Accessed: 12-Mar-2016].
- [118] J. Goyvaerts, “RegexBuddy,” *Just Great Software*, 2010. [Online]. Available: <https://www.regexbuddy.com/lib>. [Accessed: 12-Mar-2016].
- [119] J. Cho and S. Rajagopalan, “A fast regular expression indexing engine,” *Data Eng. 2002. Proceedings. 18th Int. Conf.*, vol. 3, pp. 419–430, 2002.
- [120] D. K. Pham, Duc, *Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks*. Springer Science & Business Media, 2012.
- [121] L. Wang, K. Huang, J. Zhang, and J. Yao, “A Complete Suffix Array-Based String Match Search Algorithm of Sliding Windows,” *2012 Fifth Int. Symp. Comput. Intell. Des.*, pp. 210–213, Oct. 2012.
- [122] G. Sidorov, F. Velasquez, E. Stamatatos, and A. Gelbukh, “Syntactic n-grams as machine learning features for natural language processing,” *Expert Syst. with Appl.* 41.3, no. Cic, pp. 853–860, 2014.
- [123] V. Arnau, R. Moreno, D. Cazorla, I. Medina, U. De Valencia, and D. De, “Acceleration of short and long DNA read mapping without loss of accuracy using suffix array,” vol. 30, no. 23, pp. 3396–3398, 2014.
- [124] V. K. P. Sidhu, Reetinder, “Regular expression matching can be simple and fast,” *Perls Dev. Conf.*, 2007.
- [125] J. H. Rosenberg, Andrew, “V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure,” *EMNLP-CoNLL. Vol. 7*, 2007.

APPENDIX A: SOURCE CODE

The source code and the libraries which were used with the software have been included in the attached compact disc. A guide on how to install the software for testing has been included with the software source code.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk