A Path Analysis Model for Effective E-Commerce Transactions

Sylvanus A. Ehikioya*
Department of Computer Science, Baze University, Abuja, Nigeria
Email: ehikioya@gmail.com
&
Shenghong Lu
Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada R3T 2N2
Email: lshong@cs.umanitoba.ca

*Corresponding author

ABSTRACT
Path analysis is an effective way to understand visitors' navigation of a website. A path is a sequence of pages browsed by a user in a session and ordered by the access time. A user session can be abstracted as a full path, which is made up of subpaths with different lengths. Although partial paths, such as maximal reference sequence module, can also reveal useful information in some aspects about a user in a session, a full path contributes more to the understanding of the navigational patterns in path analysis. In this paper, we present a model for path analysis based on full paths. We use a path string to straightforwardly represent a full path in a user's session. To store all paths efficiently and display the data visually, a path tree structure and an algorithm for constructing such a path tree are developed. We also implement this model using Java. Evaluation results show that this model can provide a lot of useful information about users' navigation and a website's usage.

Keywords: *E-commerce, personalization and recommendation systems, path analysis, path tree, session, data mining, Web.*

African Journal of Computing & ICT Reference Format:

Sylvanus A. Ehikioya and Shenghong Lu (2019), A Path Analysis Model for Effective E-Commerce Transactions, *Afr. J. Comp. & ICT*, Vol.12, No. 2, pp. 55 - 71.

I. INTRODUCTION

With the rapid growth of the Web, many organisations rely on the Web to conduct business. Customers are a key element of all businesses. When businesses lack adequate knowledge of their customers, such businesses cannot develop their marketing activities efficiently. In e-commerce, users' navigational behaviour indicates their steps through the shopping process [1]. Therefore, analyzing tracked navigational data is rapidly becoming one of the most important activities for any business on the Web.

Path analysis is a technique, based on the analysis of users' navigational patterns, to understand how users navigate through a website. It also sheds light into the site's structure, and helps locate trouble spots of the site [2] and identify the website traffic patterns. All these are important for designing a more efficient and user-friendly e-commerce site on which cross marketing strategies across multiple products and effective promotional campaigns can be developed.

Unlike a regular website where most users navigate following random patterns, e-commerce sites share a lot of common features in their site structures so that shoppers can easily view product catalogue, add or remove products from shopping cart, and make payment. For example, most sites have catalogue pages (for showing products), shopping-cart or basket operation pages (for adding or removing products), check-out page (for payment), and payment confirmation page (for payment confirmation). Therefore, for most e-commerce sites, each of them has at least one expected typical path which leads its users to navigate through the site and achieve a purchasing process with a high level of effectiveness, which means the ease and ability to provide users with impulse purchase opportunities during browsing and helps attract and retain more customers.

A practical business utility example of path analysis is to understand why some shoppers abandoned the buying process and exit before making a purchase. Focusing on the entry-URL and exit-URL, we can find or assume the answers to the above question. Normally, the expected entry-URL is the first page of the site such as index.html, home.html, etc.; the exit-URL is the payment confirmation page. A single shopper's data cannot reflect something very meaningful. But if many of the shoppers leave before going to make payment, this may suggest that they do not like what they put into their shopping carts, or perhaps some pages on their way to make payment are running too slowly [2]. An online merchant would be interested in a variety of issues regarding both the general performance and effectiveness of its website in order to shape its marketing strategies. The main information which can be discovered directly from the basic analysis includes:

- the users that browse the site starting from the page */company/index.html*;
- the users that end their browsing from the page /*company/comfirmation.html*;
- the users that finish more than 5 pages on this site;
- the users that make a purchase.

Basic path analysis is based on checking both referrers and site topology to support user identification [3].

Path analysis can be used with some data mining approaches such as association rules [4]. One purpose of this approach is to find association rules between a set of co-purchased products. In other words, it is to discover "association between two sets of products such that the presence of some products in a set is also present in the same transaction" [4]. Some authors [1, 3, 5, 6, 7, 8] have proposed to extract Web usage patterns from Web logs for the purpose of developing marketing strategies. When this approach is used with path analysis, information about the users that browse the page */company/product1* and also go to the page */company/products* can be obtained.

Generally, path analysis is based on session data. Path analysis can also be based on the individual users if the users can be identified. When users' personal data[2] become available from other sources, such as user inputs, user tracking tools, etc., more advanced information can be obtained by combining these users' data with path analysis. Such information may include users who buy */company/product1* are male; majority of them come to the order page for *product1* from the page */company/index.html*; and 70% of them do their shopping in the afternoon.

While several analyses use sequential pattern discovery (i.e., path analysis) techniques [9, 10, 11, 12] to discover frequent path patterns, some authors use advanced path analysis to achieve more complex tasks such as serving as a basis of personalization [12] or recommendation systems [4, 13, 14, 15, 16]. Recommendation systems and personalization are two related and popular research areas in Web data mining. They both apply statistical and knowledge discovery techniques to achieve serving / selling of more products, thereby enhancing the profitability of e-commerce sites [4]. In a recommendation system, a new user is matched against a pre-built database, which stores consumers' preferences for products. If some neighbours, who are customers already in the database and have the same taste as the new user, are found, products favoured by those neighbours are recommended to the new user. One example to using path analysis for recommendation systems is to predict HTTP requests [17], which is based on path profiles and recommends an URL with a high probability to the user before the user makes such a request.

This paper focuses on the development of a path analysis model suitable for effective management of e-commerce transactions. We abstract a user's session as a full path, which is made up of sub-paths with different length. Although partial paths, such as maximal reference sequence module, can also reveal useful information in some aspects about a user in a session, a full path contributes more to the understanding of the navigational patterns in path analysis. Also, we present and implement in Java programming language a model for path analysis based on full paths. We use a path string to straightforwardly represent a full path in a user's session. To store all paths efficiently and display the data visually, a path tree structure and an algorithm for constructing such a path tree are developed. Evaluation results show that this model can provide a lot of useful information about users' navigation and a website's usage. In addition, the work reported in this paper can be used as a base model to develop web recommendation and personalization systems.

The remainder of this paper is structured as follows: Section 2 briefly examines some of the key related research work in the literature. Section 3 describes our path analysis model while Section 4 shows the implementation results of the model. Finally, Section 5 concludes the paper and offers directions for further research.

II. INTERNET OF THINGS

Recall, in the e-commerce environment / domain, users' navigational behaviour indicates their steps through the shopping process. Thus, analyzing tracked navigational data is critical to the success of any online business.

Web analytics is a technique for understanding users' online experience for improvement of the overall quality of experience of the users. In other words, web analytics is a technique used to collect, measure, report, and analyze website data in order to assess the performance of the website and optimize its usage with the ultimate goal of increasing the return on investment. Web analytics provides a tactical approach to track key metrics and analyze visitors' activity and traffic flow and generate reports [18, 19, 20]. Thus, it is an indispensable technique for e-commerce merchants.

Web analytics, although a relatively new field of study, emerging only within the last three decades [21][1], is well researched and documented in the literature [18, 21, 22, 23, 24, 25, 26]. For example, a detailed examination of the rationale for web analytics is available in [18, 21, 22, 25, 27, 28, 29] while [24, 26, 30] examine web analytics and web metrics tools and their characteristics, functionalities and types, and data acquisition approaches and the selection of web tools for particular business models. Further, Clifton [25] examines available methodologies and their accuracy.

The data for analysis come from two major categories [31]:

- user-centric data, i.e., data collected based on individual users, which include all browsing behaviour of a user on all websites; typically collected by Internet Service Provider (ISP). This permits the creation of a user's profile of all internet usage across multiple channels.
- site-centric data, i.e., data collected from a single website, which represent the activities and behaviours of visitors on the website. This permits focused data mining and understanding of the context of the website.

Our approach in this paper focuses on site-centric data in performing website usage characterization by identifying patterns and regularities in the way users access and use web resources.

Digging deep into a visitor's behaviour and customer's purchasing habits on a website through specific engagement metrics data provides critical insights into the performance of product pages, and optimization and improvement of the effectiveness of the ecommerce solution. Ezzedin [32] examines the top engagement metrics for each step of the purchasing cycle and show how to analyze the data collected for the different users' segments using Google Analytics [23, 25, 33] measurement platform.

Booth and Jasen [30] provide an overview of methodologies for analyzing websites for increasing revenue and customer satisfaction through careful analysis of visitor interaction with a website. They discussed how basic visitor information, such as number of visitors and visit duration, can be collected using log files and page tagging by including a "tracking code on every page of your website, and then access reports to view the data that is collected." [33].

Usually each user of a website creates a visitor path. A visitor path is the route a visitor uses to navigate through a website [30]. Each visitor creates a path of page views and actions while on a website. By studying these paths, one can identify usage characterization of the website and any challenges a user has in using the website.

Nguyen et al. [20] use web usage mining process to uncover interesting patterns in web server access log file gathered from Ho Chi Minh City University of Technology (HCMUT) in Vietnam. By incorporating attribute construction (or feature construction), one of strategies of data transformation of data pre-processing technique, they had wide knowledge about users access patterns for every country, province and ISP. Such knowledge is useful for optimizing system performance (such as deciding reasonable caching policies for web proxies) as well as enhancing personalization.

Clickstream data offers consumer's online behaviour analysis, and the effectiveness of marketing actions implemented online, due to its ability to provide information concerning the sequence of pages viewed and actions taken by consumers as they navigate a website [34]. The sequence of viewed pages and actions taken are commonly referred to as "paths", and the clickstream data collected provide valuable insight into how the website is used by its users. However, as Clark et al. [35] note, clickstream data does not reveal the true intentions of the user on the website, or other possible activities that the user engaged in during the use of the website.

Ellonen, et al. [36] analyse consumer behavioural patterns on a magazine website using a unique dataset of real-life clickstream data from 295 magazine website visitors. They found interesting behavioural patterns that 86% of all sessions only visit the blogs hosted by the magazine. Similarly, Ribeiro [37] examines the navigational patterns of users on the website of Shifter, an online media company, for a 3 months period using Microsoft Excel tool to obtain a context for each piece of content produced and published. Analysis of Shifter's data resulted in recommendations for rethink, and the redesign, of the editorial content of the business to respond to different community's needs.

Linden [38] similarly examines behavioural patterns of web users on an online magazine website with a view to first find and visualize user paths within the data generated during collection, and then identify some generic behavioural typologies of user behaviour using cluster analysis and sequential path analysis. He used a dataset of clickstream data generated from the real-life clicks of 250 randomly selected website visitors over a period of six weeks using Microsoft Excel to visualize user paths and analyze descriptive studies based on the clickstream data. The analytical process focuses on a combined methodology of cluster analysis and swim-lane diagrams. Similarly, Jain et al. [39] and Pani et al. [40] provide an analysis of lnternet browsing and site usage behaviour using sequential access pattern mining, while Siddiqui and Aljahdali [41] discuss Web mining tree structure. Also, Jokar et al. [42] examine Web mining and Web usage techniques while presenting an efficient framework for Web personalization based on sequential and non-sequential patterns, and analyse the structure of the web pages using compression of tree structure method.

E-commerce transaction systems execute in open network infrastructure and they are, therefore, not completely fraud proof because some points of a system are vulnerable in the real world due to the openness of the network, which attackers and fraudsters can exploit. Wang et al. [43] examine vulnerable points analysis for e-commerce transaction system with a known attack using a Petri nets based model, called Vulnerable E-commerce Transaction net (VET-net). They define and espouse the concepts of vulnerable points and vulnerable levels in order to describe the cause and levels of vulnerability, and show the effectiveness and rationality of the concepts and method.

Tian et al. [44] propose a systematic structural analysis framework model that reveals the hidden mechanism of e-commerce market structure and Internet social system. This model relies on ecosystem theory and network science. The model uses element identification, relationship analysis, and formation mechanism analysis as the core steps to explain e-commerce market structure. They show several illustrative applications based on the analysis model.

Lewis and White [45] present a method for web usage mining based on a linear ordering of the age transition matrix created from web server access logs. This ordering facilitates the categorization of web pages into different classes (such as origins, hubs, or destinations) based on position in the linear order; thus providing a measure of the orderliness of website traffic. They applied this technique to website traffic of a university over time by comparing the website traffic immediately after a major change to the website design and the traffic two years later since changes in website organization could also dramatically change visitors flow. The results show the traffic is more ordered. Similarly, Asha and Rajkumar

[46] discuss web usage mining techniques for enhanced quality of experience of customers shopping online on websites and also discuss web mining techniques to find dishonest recommenders in open social networks. They propose a recommendation system that uses semantic web mining process integrated with domain ontology which can be used to extract interesting patterns from complex and heterogeneous data.

Clickstream data provide information about the sequence of pages or the path viewed by users as they navigate a website. Montgomery et al. [47] show how path information can be categorized and modeled using a dynamic multinomial probit model of Web browsing (i.e., clickstream data) using data from a major online bookseller. Their results suggest that paths may reflect a user's goals, which could be helpful in predicting future movements at a website. One potential application of their model is to predict purchase conversion. This technique is useful in the personalization of Web designs and product offerings based upon a user's path.

Noreika and Drąsutis [48] propose website activity data analysis model based on a composition of website traffic and structure analysis models with intelligent methods. The measurement of visitors' website activities relies highly on data mining techniques. This approach enables theoretical predictions on how and what factor changes in website structure affect a visitor's click paths and overall website activity. Their model relies on the main principle of dividing the website analysis into two parts; namely website structure analysis model and website traffic analysis model. They construct and formalize these models separately and then establish a relation function between them based on intelligent methods. One of the limitations of their work is that they only describe the models construction leaving out the key intelligent methods as a black box, which leaves too many unknowns.

Ohta and Higuchi [49] analyse store layout that underlie supermarket store design and product display styles and then examine the interaction between shop floor layout and customer behaviour from the perspective of the supermarket owner to discover the main sections within the shop likely to attract customers into the store. The authors made a general classification between the standard layout, which accounted for approximately 90% of the survey sample, and the minority layout, used by less than 10% of the survey sample. Using the survey results, they analyzed the customer circulation rates and section drop-by rates as influenced by the store layout. They concluded that the standard layout is superior. This study is fundamental and analogous to the behaviour of online visitors to e-commerce websites.

Zheng et al. [50] propose an important way of detecting fraud in users transactions by extracting the behaviour profiles (BPs) of users based on their historical transaction records, and then verify if an incoming transaction is a fraud or not in view of their BPs. The Markov chain models [51] are popular in representing BPs of users, which is effective for those users whose transaction behaviours are relatively stable. However, with the development and popularity of e-commerce, it is more convenient for users to shop online, which diversifies the transaction behaviours of users. Therefore, Markov chain models are unsuitable for the representation of these behaviours. However, they propose use of logical graph of BP (LGBP), a total order-based model, to represent the logical relation of attributes of transaction records. Based on the LGBP and users' transaction records, one can compute a path-based transition probability from one attribute to another, and diversity coefficient to characterize users' transaction behaviours and diversity. In addition, to capture temporal features of a user's transactions they also define a state transition probability matrix. Their experiments over a real data set illustrate that the LGBP method can characterize the users transaction behaviours precisely, and abstracts and covers all different transaction records.

III. A PATH ANALYSIS MODEL

Most current path analysis use Web log files as data source, though some use the data tracked by other approaches such as packet sniffers and single-pixel images. The different approaches used to obtain the data sources also determine the easiness of the different data preparation processes, which mainly involve session identification and obtaining page views [3, 16]. Session identification groups all pages in a user's session, while page views are obtained by filtering out all noise files, such as those records created by image links and together with the requested page. Instead of repeating the data preparation process, this paper assumes that the page references in the data source are grouped by sessions and are sorted by access time in each session. Each processed entry of the data is a page view of each request made by a user.

3.1 Definitions

To build our path analysis model, it is imperative we first define the following terms.

- *Path and Path Element*

A path P is a sequence of URLs, which are ordered by access time and accessed by a single user in a session. In the sequence, same URLs can appear more than once. For example, (URL1, URL1, URL3) and (URL1, URL3) are two different paths. Each URL in a path P is called path element. For example, the path elements are URL1 and URL3 for path (URL1, URL3).

- **Sub-path**

A path M (m_1, m_2, …m_i) is said to be a sub-path of P (p_1, p_2, …p_j) if and only if:

1) $j \geq i$ and
2) There exist continuous i path elements starting at the integer index k ($j-i \geq k \geq 0$) and the following relation between each path element in M and P: $m_i = p_{k+i}$.

- *Path Length*

The path length |P| is the length of the path P. The value |P| is equal to the number of pages accessed by a single user in one session. For example, if a user visits the following pages {1, 3, 5, 6, 5, 7} in one session, |P| is 6. A session S for a user denotes all URLs ordered by access time and accessed by the user in one visit or during a predefined length of time, for example, 30 minutes.

Observation 1 Any path is made up of its total sub-paths with different lengths**.** For example, in a given session S = (S_1, S_2, ·· ·· , S_n), S_1 , S_2 ·· ···, S_n are all sub-paths of length one, (S_1, S_2), (S_2, S_3), and so on are all sub-paths of length two, etc. Therefore, there are |S| sub-paths of length one, |S|-1 sub-paths of length two, |S|-2 sub-paths of length three, and finally, only one sub-path of length |S|, in the session path. In the session S, the total sub-paths P in S are:

$$(|S|-0) + (|S|-1) + \cdots [|S|-(|S| -1)] = \frac{|S|*(|S|+1)}{2} \approx \frac{|S|^2}{2} \quad \ldots\ldots\ldots\ldots (1).$$

Some mathematical analyses in [16] show that the average length of the path in a session S is $\frac{|S|}{3}$, thus the total number of URLs that would be needed to store every path can be calculated as follows:

$$\frac{|S|^2}{2} * \frac{|S|}{3} = \frac{|S|^3}{6} \quad \ldots\ldots\ldots\ldots (2).$$

Obviously, it makes sense if the above number is an integer and more than 1. To satisfy these two requirements, we have $\frac{|S|^3}{6} \geq 1$, which leads to $|S| \geq 1.82 \approx 2$. This means that in any session where only one page is requested, the path analysis based on one page is insignificant and, thus, can be overlooked. This finding allows us to consider only those important paths in analysis and helps decrease the requirement for storage and memory in the analysis process.

Observation 2. Two paths M and N are equal if and only if the following two conditions hold:

1) |M| = |N|, and
2) All sub-paths of either can be found in the other.

- **Page String**

A page string is used to represent a URL to facilitate comparison and store process. It consists of a letter “p”, which denotes “page”, and a number, which identifies the URL uniquely. For example, the URL *http://www.yahoo.com* can be represented as “p1” in this approach.

- **Path String**

A path string is a string, which consists of all page strings in a user’s session. If a page string appears twice (for example, p1) in a session, that string (p1) appears twice too.

- **Path Node**

A path node represents a path which a user navigated the site in the session. A path node can be a root node (if it has children) or leaf. All path nodes have the same data structure.

- **Path Tree**

A path tree is a tree structure used to represent all paths in all users’ sessions. For a session S, the maximum path node for a path tree is given by $\frac{|S|*(|S|+1)}{2}$.

$$\ldots\ldots\ldots\ldots (3).$$

3.2 Data Structure for the Path Node

In the path analysis, we must know which paths exist and how many times a path occurred. Also, we wish to find the percentage of this path occurrence so as to find those "frequent" paths. To efficiently reflect these three requirements for a path, we use the record data structure, shown in Figure 1, to represent a node of the path tree.

In this data structure,

- "pathName" denotes a path in the path tree;
- "occurTime" is an integer, which represents the number of occurrences of this path;
- "pathWeight" denotes this path's percentage in all paths with the same length.

Since a path includes all sub-paths, it makes more sense to calculate the percentage of the occurrences of the path in the group of the same length than in the whole paths.

3.2.1 Construct Page Strings and Path Strings

To obtain all paths in each session of a user, we should know all the pages the user accessed in that session. This leads to a problem: should the URLs we are using to form paths contain any parameter field or only the name of the requested file?

The current version of HTTP allows a Web client to pass almost everything in the parameter field when using the GET method and the attached information forms an essential part of the URLs. The GET method is used to request data (i.e., retrieve resource representation / information) from a specified resource [52]. The GET request does not change the state of the resource, so it is a safe method. Also, "GET APIs should be idempotent, which means that making multiple identical requests must produce the same result every time until another API (POST or PUT) has changed the state of the resource on the server." [53].

The search engine "Google" is one obvious example. When searching for "hello" using "Google", after the search, the URL on the result page of the search becomes "http://www.google.ca/search?hl=en&ie=UTF-8&oe=UTF-8&q=hello&meta=". We find some parameters are attached to the original URL "http://www.google.ca". However, most of the parameters attached to the URLs serve no purpose other than tracking the users' sessions [54]. They are of less use for analyzing paths since in the path analysis, for example, we are interested only in knowing whether a client used "Google", and not what the client searched. Although no agreement about whether or not keeping parameters in URLs in data mining has been reached, in this paper, all parameters are ignored.

After processing the parameters in all URLs, in Table 1, we list 5 page strings which are 5 different URLs on Yahoo!.

Suppose we have 10 users' sessions. The URLs browsed in each session are listed in Table 2.

Following the definition of the path string, all pages in each session can be denoted as a path string based on the ordered pages in that session. For example, in session 1, the path string "p1p2p1p2p3" represents all the paths in this session. One benefit of this kind of representation is that we can easily apply string operations to differentiate and show paths in a straightforward way. For example, if we know a path exists from page 1 to page 2, we can use "p1p2" to represent it; on the other hand, a path of "p1p3p4" can be straightforwardly interpreted as the path starts from page1 to page 3, and then to page 4.

3.2.2 Full Paths in a Session

In Web data mining, some authors [55] discard all backward movements in a user's session because they think those backward movements are merely for navigational convenience. They develop some modules (such as maximal reference sequence module, reference length module, and time window module) to split each session into smaller transactions to create more meaningful clusters of references. Others [3, 10] develop different algorithms to achieve similar functions. In fact, a backward movement in a user's navigation is not just for ease of navigation since during the backward movement the user may browse the page again (not just for getting to the result page more quickly). Also, the path {1, 2, 3, 4, (3, 2), 5} is not just identical to two separate paths {1, 2, 3, 4} and {1, 5}, since the former path not only reflects the full navigational order, but also the number of times each page is browsed by the user; while the two separate paths reflect two forward sequences.

One major reason some authors discard the backward movements is that their analyses are based on the traditional Web log files, in which backward movements are not available generally. Some advanced tracking approaches like single-pixel image can track backward movements. These advanced tracking approaches provide the data source for building a full path string for each session. Table 2 lists ten (10) full paths from 10 users' session.

3.2.3 An Algorithm to Create a Path Tree

The tree structure is an efficient data structure to store non-linear data like paths. We present an algorithm to build such a tree. Figure 2 shows the pseudo-code of the algorithm.

To use this algorithm, the first task is to determine what data source should be used. Some authors (e.g., [1]) think a session with at least five page views offers a usable data source, although this choice can be made subjectively. The number of Web users is large, running into several millions. Considering data storage and memory constraints, we can only record those "important paths" into the path tree. No uniform opinion about what paths are "important" for data analysis exists. Some people prefer at least two URLs, while others would like more [17]. This algorithm provides such scalability and flexibility by allowing the user to choose the minimal path length to record into the path tree. To show how this algorithm works, we build a tree based on the data in Table 2 with the smallest path of length 2, which is described in Observation 1 of the least importance in path analysis. Later, we show another example for multiple sessions as in real life; in each session, we consider those paths with length greater than 3. This means that session 9 and session 10 will be omitted in the analysis.

3.2.4 Steps to Build a Path Tree

In this section, we apply the algorithm to record all the paths with length ≥ 2 (i.e., the minimum path length N in the algorithm equals 2) for session 1 in Table 2 into a tree. In this example, the path string is "p1p2p1p2p3" and the length of the input array is 5 (i.e., LENGTH = 5).

An empty node, shown in Figure 1, is created every time a node is needed. Therefore, to create the tree, at the beginning we create and initialize an empty node to be the root node of the tree.

To control the outer WHILE loop, a control variable I is set. The range of I, in this example, is from 0 to 4 (less than 5). The outer WHILE loop starts by setting the root node as current node and setting another index J for controlling the inner WHILE loop.

In the first outer WHILE loop, I is equal to 0. The current node is the root node. The value of the inner loop control variable J is equal to 1 (i.e., J = I + 1). A temporary node TEMP is created with the path value A [0] (i.e., p1). In this example, we set N, the minimum length of a path string which we record in the path tree to 2. The first inner WHILE loop starts with J = 1. The path of node TEMP becomes "p1p2". The length of this path is 2, which is not less than 2 (i.e., the value of N). This path string satisfies the length requirement for the path string in the path tree. The next step is to check whether there exists a node with the same path string as "p1p2" in the tree or not. If such a node exists, we simply increase the occurrence times of that path one more time and set that node as a current node; otherwise, add this node TEMP as a child node of the current node, set the occurrence time of node as 1, and make this node as the current node. Since no such node in the tree with the path value of "p1p2" exists, in this example, the node TEMP is added to the current node, i.e., root node. The new node has the path value of "p1p2" and occurTime value of 1. This new node becomes the current node. After adding this node, the inner loop runs again with J = 2. In this loop, since the variable TEMP keeps its old values (i.e., TEMP.path = "p1p2" and TEMP.occurTime = 1), after a new string A[2] (i.e., "p3") is added to the current path string, the new path value of TEMP becomes "p1p2p3". Since no such node with the same path value of "p1p2p3" exists in the tree, this node is added to the tree as a child of the current node, which we just created in the first inner loop. Also, the new set value of occurTime of this node overrides the value saved from the previous loop. This ensures that every time a new node is created, the occurTime of the new node is 1. Following the same procedures two more times (J = 3 and 4), two more nodes are added. So far, the first outer loop (I = 0) ends, and the current node goes back to the root node.

In the second outer loop, I equals 1. The inner loop starts with J = 2. Just as in the first outer loop, new paths "p2p1", "p2p1p2", and "p2p1p2p3" are added to the tree after 3 inner loops with J = 2, 3, and 4.

In the third outer loop, I is 2. The inner loop starts with J = 3. In this first inner loop, there exists a node, which is created at the beginning and whose path is "p1p2", same as that of TEMP. According to the algorithm, we cannot add TEMP to the tree as a child of the current node. Instead, we update the occurTime of that node in the tree by increasing it by 1 and set that node as the current node. The occurTime of the node whose path is equal to "p1p2" is 2. The third outer loop ends with another new node with the path value of "p1p2p3" added to the tree.

In the fourth outer loop, I equals to 3. The inner loop starts with J = 4, which is the range limit of the J. Therefore, the inner loop can run only once and a new node is added. Although the outer loop can run one more time with I = 4, since J exceeds its range limit, no mode can be added to tree. The last step is to update the path weight field for all nodes since this value can only be obtained after the whole tree is built. Figure 3 shows the results of all above steps to build the path tree.

IV. THE ENVIRONMENT WITH MULTIPLE USERS' SESSIONS

In this section, we present the implementation results of the path analysis model in an environment with multiple users' sessions and the evaluation of this model. Our implementation environment consists of Java / JavaScript programming language, Tomcat (version 4.1.27) built-in Web server, and Microsoft SQL Server 2013 database server tools.

We use an HTTP proxy and several Java servlets to implement the server-side tracking while we use client-side tracking technology (JavaScript and HTML image technologies) to obtain user data more directly and naturally. Our implementation is more effective as it can track most keyboard and mouse events of online users and it is capable of tracking any form input on the page. In addition, our approach is designed to work on both static and dynamic Web pages.

4.1 Implementation Result

The Web environment involves multiple users and sessions. Figure 4 shows a screenshot of the result of implementation of the algorithm on all the session data in Table 2. Since we considered only those paths with length ≥ 3, sessions 9 and 10 are ignored.

4.2 Evaluation of the Path Analysis Model

In our model, it is easy to find the following basic information: In 40% (4 out of 10) sessions, users started browsing with page 1; suppose page 4 is the product catalogue page, in 40% (2 out of 5) sessions in which users came to this product page, the users were from page 2; in 50% sessions, users finished more than 5 page views. Suppose page 5 is the page for finishing a purchase, we can think in 40% sessions, users finished a purchase. Also, we can find information such as: in 50% sessions, the users that went to page 3 also went to page 4.

Path tree reveals much more advanced information. The minimum path consists of 3 page views. In the total 10 sessions, there are 29 paths of length 3. Five (5) of these 29 paths have no sub-paths (leaf node), and the path "p1p2p3" was used in most sessions since this path has the highest weight of 10%. The largest path is "p1p2p5p2p3p4p2", and it occurred just once.

In data mining, statistical concepts of support and confidence [4] are usually used to evaluate association rules [56, 57], one popular approach to find similarities among objects. Applying this approach to our path analysis model, we can obtain more advanced information about users' groups.

An association rule [58] is a way to find relation between two groups. Let P = {p_1, p_2, …, p_m} denotes a collection of m items, T={t_1, t_2,…, t_n} denotes another collection of n items whose elements always occur together, and T $\subseteq$ P. An association rule between two sets of items X and Y, such that X, Y $\subseteq$ P and X $\cap$ Y= $\varnothing$, means that the presence of items in the set X in T indicates a strong probability that items from the set Y are also present in T. Such an association can be denoted as X $\Rightarrow$ Y.

Support and confidence are two key values commonly used to evaluate the quality of association rules. The support (s) measures the occurrence frequency of the pattern in the rule while the confidence (c) is the measure of the strength of that rule. For a rule X $\Rightarrow$ Y, the support s is defined as:
as:

$$s\% = \frac{\text{number of considered sets containing } X \cup Y}{\text{number of considered sets}}$$

In other words, support s means that s% of considered sets contain X $\cup$ Y. The confidence (c) indicates that c% of considered sets that contain X also contains Y.

$$c\% = \frac{\text{number of consideredsets containing X} \cup Y}{\text{number of consideredsets containing X}}$$

A high confidence value means the prediction is with a high accuracy, while a high support means the considered rules occur more frequently. Therefore, when using this approach, high confidence and support values are preferred.

To apply association rules to our analysis, we can think of our previous paths in this way: two paths P_a = "p1p2p3" and P_b = "p1p2p3p4p5" can be considered as "p1p2"→p3 and "p1p2p3"→ "p4p5", respectively.

Now, consider the 10 path strings in Table 2. Following the model, we set X = p1p2, Y = P3. We can obtain the support (s) = 70% (7 out of 10), and the confidence (c) = 85% (7 out of 8). We can translate these results as: 70% users who go to the path p1p2 will go to p3; 85% users who have been to p1p2 have been to p3.

Path analysis plays a key role in traffic analysis of e-commerce sites. One obvious benefit is that key products can be put on the frequently followed path so that most users can access them easily. Also, the site structure can be improved through finding why some paths are visited less by checking the site topology. Moreover, path analysis can lead to a basis for personalization and recommendation systems [59]. For example, if we know a path P_c is followed by another path P_d with a high support and confidence, we can recommend the pages matching the path P_d to those users who are navigating the path P_c. When path analysis is used with other data sources or data mining approaches, such as user profiles, collaborative filtering [60, 61], and clustering [62, 63], effective market campaign strategies and user-oriented promotion systems [59] can be developed.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a model for mining traversal / navigational patterns of the Web users. The information discovered by the analysis is of significant importance to the site owner, especially for e-commerce sites, to know the users and develop effective market strategies such as targeting specific audience with customized products and services. Construction of this model involves following major steps. First, we defined a simple string (i.e., page string) to represent an URL visited by users. In this way, each visited page maps to a unique page string. This approach enables an easy way to store and operate the session data later. Second, we concatenated all page strings that are the complete page views in a session and ordered by the access time as one string, i.e., path string. This string is the full path of the session for the user and reflects the user's navigational behaviour in this session in a straightforward way. To store and analyze the path statistically, we used a tree structure and developed a data structure for the tree node for the later analysis purpose. Fourth, an algorithm to build such a path tree was developed. Fifth, we implemented the model in an environment with multiple users' sessions and evaluated the experimental results.

Interesting future work involves exploring the potential usage of this model in developing Web personalization and recommendation systems, and also in user profiling process in e-commerce when this model is used with other data mining approaches such as clustering, collaborative filtering, and associated rules, etc. We note, however, that although user profiling could be possible, ethical issues and privacy laws in some countries (e.g., Canada and the United States of America) may hinder the extension of this work in this direction.

POSTSCRIPT

This research was conducted while Dr. Ehikioya was with the University of Manitoba, Winnipeg, Canada.

^{2}Interested readers should refer to the following resource: Sylvanus A. Ehikioya and Shenghong Lu**,** "Traffic Tracking Analysis for E-commerce", *Technical Report #: TR-04-06,* Department of Computer Science, University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada, March 2004. Number of Pages: 31

ACKNOWLEDGEMENT

The authors will like to acknowledge the external reviewers whose inputs have greatly improved the final overall quality of the paper.

REFERENCES

[1] C. Theusinger and K. Huber, "Analyzing the Footsteps of Your Customers", *WEB Knowledge Discovery in Databases (WEBKDD)*, Boston, MA, USA, August 2000.

[2] S. G. Eick, "eBusiness Performance Analysis", *Superiore G. Reiss Romoli (SSGRR2000)*, L'Aquila, Italy, Jul 31 - Aug 06, 2000.

[3] R. Cooley, J. Srivastava, and B. Mobasher, "Data Preparation for Mining World Wide Web Browsing Patterns", *Journal of Knowledge and Information Systems,* Vol. 1, No. 1, pp. 5-32, 1999.

[4] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Analysis of Recommendation Algorithms for E-commerce", Proceedings of the ACM Conference on E-commerce (EC00), Minneapolis, October 2000.

[5]. O. R. Zaiane, M. Xin, and J. Han, "Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs", *Advances in Digital Libraries*, Santa Barbara, 1998, pp. 19-29.

[6] M. Spiliopoulou and L. C. Faulstich, "WUM: A Web Utilization Miner", *Proceeding of International Conference on Extending Database Technology (EDBT) Workshop WebDB98*, Valencia, Spain, LNCS 1590, Springer Verlag, 1999.

[7] A. G. Buchner and M. D. Mulvenna, "Discovering Internet Marketing Intelligence Through Online Analytical Web Usage Mining", *SIGMOD Record*, (4) 27, 1999.

[8] J. Pei, J. Han, B. Mortazavi-asl, and H. Zhu, "Mining Access Patterns Efficiently from Web", *Proceedings Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'00),* Kyoto, Japan, April 2000, pp. 396-407.

[9] P. Berkhin, J. D. Becher, and D. J. Randall, "Interactive Path Analysis of Web Site Traffic", *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* San Francisco, CA, August 2001, pp. 414-419.

[10] M-S Chen, J. S. Park, and P. S. Yu, "Efficient Data Mining for Path Traversal Patterns", *IEEE Trans. Knowledge and Data Eng.*, Vol. 10, No. 2, March 1998, pp. 209-221.

[11] Mannila H., Toivonen H., and Verkamo A. I., "Discovering Frequent Episodes in Sequences", *Proceedings of the 1st Int'l Conference on Knowledge Discovery and Data Mining (KDD'95)*, Montreal, Quebec, 1995, pp. 210-215.

[12] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements", *Proceedings of the 5th International Conference on Extending Database Technology (EDBT'96): Advances in Database Technology*, Avignon, France, March 25-29, 1996, pp. 3-17.

[13] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce Recommendation Applications", *Data Mining and Knowledge Discovery*, Volume 5 Issue 1-2, January-April 2001, pp. 115-153.

[14] E. J. Glover, S. Lawrence, M. D. Gordon, W. P. Birmingham, and C. L. Giles, "Recommending Web Documents Based on User Preferences", *ACM SIGIR '99 Workshop on Recommender Systems*, University of California, Berkeley, August, 1999.

[15] R. Cooley, B. Mobasher, and J. Srivastava, "Web Mining: Information and Pattern Discovery on the World Wide Web", *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI' 97)*, 1997, pp. 558-567.

[16] B. Mobasher, H. Dai, T. Luo, Y. Sun, and J. Zhu, "Integrating Web Usage and Content Mining for More Effective Personalization", *Proceedings of the International Conference on E-commerce and Web Technologies (ECWeb2000)*, Greenwich, UK, 2000.

[17] S. Schechter, M. Krishnan, and M. D. Smith, "Using Path Profiles to Predict HTTP Requests", *Proceedings of the Seventh International World Wide Web Conference,* Brisbane, Australia, April 1998.

[18] "*Web Analytics*", Tutorials Point (I) Pvt. Ltd., 2015. (Available at: http://www.tutorialspoint.com/web_analytics_turorial.pdf). Accessed on December 12, 2018.

[19] Zheng Guangzhi and Peltsverger Svetlana , "Web Analytics Overview", in Mehdi Khosrow-Pour, *Encyclopedia of Information Science and Technology*, 3rd Edition, IGI, 2015.

[20] M. T. Nguyen, T. D. Diep, Vinh T.Hoang, T. Nakajima, N. Thoai, "Analyzing and Visualizing Web Server Access Log File", In: Dang T., Küng J., Wagner R., Thoai N., Takizawa M. (eds) *Future Data and Security Engineering (FDSE 2018) ---*

Lecture Notes in Computer Science, Vol 11251. Springer, Cham, 2018.

[21] Dykes Brent, *Web Analytics Kick Start Guide: A Primer on the Fundamentals of DIGITAL Analytics*, Adobe Press, 2014.

[22] T. Peterson Eric, *Web Analytics Demystified: A Marketer's Guide to Understanding How Your Web Site Affects Your Business*, Celilo Group Media and CafePress, 2004.

[23] Kaushik Avinash, *Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity*, Wiley Publishing, 2010.

[24] Bekavac Ivan and Garbin Praničević Daniela, "Web Analytics Tools and Web Metrics Tools: An Overview and Comparative Analysis", *Croatian Operational Research Review (CRORR)* 6 (2015), pp. 373–386.

[25] Brian Clifton, *Advanced Web Metrics with Google Analytics*, 3rd Edition, Wiley Publishing, 2012.

[26] Bernard J. (Jim) Jansen, *Understanding User – Web Interactions via Web Analytics*, Morgan & Claypool, 2009.

[27] Avinash Kaushik, *Web Analytics: An Hour a Day*, Wiley Publishing, 2007.

[28] Alistair Croll and Sean Power, *Complete Web Monitoring: Watching Your Visitors, Performance, Communities, and Competitors*, O'Reilly, 2009.

[29] Steve Jackson, *Cult of Analytics: Driving Online Marketing Strategies Using Web Analytics*, Butterworth-Heinemann, 2009.

[30] Danielle Booth and Bernard J. Jansen, "A Review of Methodologies for Analyzing Websites", in B. J. Jansen, A. Spink, and I. Taksa (Ed.), *Handbook of Research on Web Log Analysis*, IGI Global, 2010. pp. 141-162.

[31] Randolph E. Bucklin and Catarina Sismeiro, "Click Here for Internet Insight: Advances in Clickstream Data Analysis in Marketing", *Journal of Interactive Marketing*, 23 (1), 2009. pp. 35 - 48.

[32] Allaedin Ezzedin, *Tracking Product Journey from Carting to Purchasing: 15 Secrets to Perfecting Your Online Store*, E-Nor Inc., 2014. (Available at: https://www.e-nor.com/wp-content/uploads/pubs/ebooks/tracking-product-journey-from-carting-to-purchasing.pdf)

[33] Eric Fettman, *Google Analytics Universal Guide - Best Practices for Implementation and Reporting*, E-Nor Inc., March 2014. (Available at: https://www.e-nor.com/blog/ebooks/ google-analytics-universal-guide-best-practices-for-implementation-and-reporting).

[34] J. Andersen, A. Giversen, A. H. Jensen, R. S. Larsen, Torben Bach Pedersen, and J. Skyt, "Analyzing Clickstreams Using Subsessions", *Proceedings of the ACM Third International Workshop on Data Warehousing and OLAP (DOLAP00)*, Washington DC, USA, November 10, 2000.

[35] Lillian Clark, I-Hsien Ting, Chris Kimble, Peter C. Wright, and Daniel Kudenko, "Combining Ethnographic and Clickstream Data to Identify User Web Browsing Strategies", *Information Research*, 11(2), January 2006.

[36] Ellonen, Hanna-Kaisa, Wikstrom, Patrik, and Johansson, Anette, "The Role of the Website in a Magazine Business: Revisiting Old Truths", *Journal of Media Business Studies*, 12(4), 2015, pp. 238-249.

[37] João Pedro de Almeida Ribeiro, *The Use of Web Analytics on a Small Data Set in an Online Media Company: Shifter's Case Study*, Master's Thesis in Information Management, NOVA Information Management School, Instituto Superior de Estatística e Gestão de Informação, Universidade Nova de Lisboa, November 2016.

[38] Michael Lindén, *Path Analysis of Online Users Using Clickstream Data: Case Online Magazine Website*, Master's Thesis in Strategy, Innovation and Sustainability, LUT School of Business and Management, Lappeenranta University of Technology, 2016.

[39] Kumar Jain, R., Kasana, R.S. and Jain, S., "Efficient Web Log Mining using Doubly Linked Tree", *International Journal of Computer Science and Information Security (IJCSIS)*, 3(1), 2009.

[40] S. K. Pani, L. Panigrahy, V. H. Sankar, B. K. Ratha, A. K. Mandal, and S. K. Padhi, "Web Usage Mining: A Survey on Pattern Extraction from Web Logs", *International Journal of Instrumentation, Control & Automation (IJICA)*, 1(1), 2011, pp.15–23.

[41] Ahmad Tasnim Siddiqui and Sultan Aljahdali, "Web Mining Techniques in E-Commerce Applications", *International Journal of Computer Applications*, Vol. 69, No.8, 2013.

[42] Nasrin Jokar, Ali Reza Honarvar, Shima AgHamirzadeh, Khadijeh Esfandiari, "Web Mining and Web Usage Mining Techniques", *Bulletin de la Société des Sciences de Liège*, Vol. 85, January 2016, pp. 321-328.

[43] Mimi Wang, Guanjun Liu, Chungang Yan, and Changjun Jiang, "Modeling and Vulnerable Points Analysis for E-commerce Transaction System with

a Known Attack", 9th *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage (SpaCCS 2016),* Zhangjiajie, China, November 16-18, 2016, Lecture Notes in Computer Science, Vol 10066. Springer, Cham, 2016, pp. 422-436.

[44] Zhihong Tian, Zhenji Zhan, and Xiaolan Guan, "A New Structural Analysis Model for E-Commerce Ecosystem Network", *International Journal of Hybrid Information Technology*, Vol. 7, No. 1, 2014, pp. 43-56.

[45] Mark W. Lewis and Barbara Jo White, "SOLO: A Linear Ordering Approach to Path Analysis of Web Site Traffic", *INFOR: Information Systems and Operational Research*, Vol. 50, Issue 4, 2012. pp. 186-194.

[46] K. N. Asha and R. Rajkumar , "Survey on Web Mining Techniques and Challenges of E-commerce in Online Social Networks", *Indian Journal of Science and Technology,* Vol. 9 No. 13, April 2016, pp. 1-5. DOI: 10.17485/ijst/2016/v9i13/85481

[47] Alan L. Montgomery, Shibo Li, Kannan Srinivasan, and John C. Liechty, "Modeling Online Browsing and Path Analysis Using Clickstream Data", *INFORMS* Vol. 23, #4, November 2004.

[48] Algirdas Noreika and Sigitas Drąsutis, "Website Activity Analysis Model", *Information Technology and Control*, Vol.36, No.3, 2007, pp. 268-272.

[49] Masao Ohta, and Yoshiyuki Higuchi, "Study on the Design of Supermarket Store Layouts: The Principle of "Sales Magnet"", *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, Vol. 7, No. 1, 2013, pp. 209-212.

[50] Lutao Zheng, Guanjun Liu , Chungang Yan, and Changjun Jiang, "Transaction Fraud Detection Based on Total Order Relation and Behavior Diversity", *IEEE Transactions on Computational Social Systems*, August 2018.

[51] Nicolas Privault, *Understanding Markov Chains: Examples and Applications*, 2nd Edition, Springer, July 2018.

[52] "HTTP Request Methods", W3Schools.com, Refsnes Data. No Date. (Available at: https://www.w3schools.com/tags/ref_httpmethods.asp). Accessed on February 25, 2019.

[53] "HTTP Methods", *REST API Tutorial*, RESTfulAPI.net, No Date. (Available at: https://restfulapi.net/http-methods/#get). Accessed on February 25, 2019.

[54] B. Mobasher, R. Cooley, and J. Srivastava, "Automatic Personalization Based on Web Usage Mining", *Communications of the ACM*, 43(8), 2000, pp. 142-151.

[55] R. Cooley, B. Mobasher, and J. Srivastava, **"**Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns", *Proceedings of the 1997 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX-97)*, Nov. 1997.

[56] L. Shen, L. Cheng, J. Ford, F. Makedon, V. Megalooikonomou, and T. Steinberg, "Mining the Most Interesting Web Access Associations", *Proceedings of the World Conference on the WWW and Internet (WebNet)*, San Antonio, Texas, Nov. 2000, pp. 489-494.

[57] E.H. Han, G. Karypis, V. Kumar, and B. Mobasher, "Clustering Based On Association Rule Hypergraphs", *Proceedings of SIGMOD'97 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'97)*, May 1997.

[58] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Analysis of Recommendation Algorithms for E-commerce", *Proceedings of the ACM Conference on E-commerce (EC00)*, Minneapolis, October 2000.

[59] "Web Site Personalization", *IBM High-Volume Web Site Team*, January 2000.

[60] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Ridl, "GroupLens: Applying Collaborative Filtering to Usenet News", *Communications of the ACM*, 40(3), 1997, pp.77-87.

[61] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl, "An Algorithmic Framework Performing Collaborative Filtering", *Proceeding of the 1999 Conference on Research and Development in Information Retrieval*, August, 1999.

[62] C. Shahabi, A.M. Zarkesh, J. Adibi, and V. Shah, "Knowledge Discovery from Users Webpage Navigation", *Proceedings of Seventh International Workshop on Research Issues in Data Engineering*, Birmingham, England, 1997, pp. 20-29.

[63] A. Joshi and R. Krishnapuram, "Robust Fuzzy Clustering Methods to Support Web Mining", *Proceedings of Workshop in Data Mining and knowledge Discovery, SIGMOD*, 1998, pp. 15-1 - 15-8.

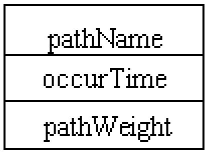


Figure 1. Data Structure for Path Tree Nodes

Table 1. Page Strings for URLs

URL Index	URLs	Page Strings
1	http://www.yahoo.com	p1
2	http://autos.yahoo.com	p2
3	http://realestate.yahoo.com/	p3
4	http://auctions.shopping.yahoo.com	p4
5	http://travel.yahoo.com/	p5

Table 2. Data Source for Sample Path Analysis

Session Index	Pages Browsed Ordered by Access Time	Path String	Length
1	p1, p2, p1, p2, p3	p1p2p1p2p3	5
2	p1, p2, p4, p3, p2, p5	p1p2p4p3p2p5	6
3	p4, p1, p2, p5, p3, p1	p4p1p2p5p3p1	6
4	p2, p4, p1, p2, p3, p5	p2p3p1p2p3p5	6
5	p2, p1, p3	p2p1p3	3
6	p2, p5, p1, p2, p4, p3	p2p5p1p2p4p3	6
7	p3, p2, p1,p2, p3, p4	p3p2p1p2p3p4	6
8	p1, p2, p5, p2, p3, p4, p2	p1p2p5p2p3p4p2	7
9	p1, p2	p1p2	2
10	p3	p3	1

```
10  INPUT a path string
20  set LENGTH = the length of the path string
30  create an empty path node T as root node with values (null, 0, 0.0)
40  set outer loop control variable I = 0
50  WHILE (I < LENGTH)
60    set T as current node
70    set inner loop control variable J = I +1
80    set N as the minimum path length required in the path tree
90    create a temp node TEMP with values (A[ I ], 0, 0.0)
100    WHILE (J < LENGTH)
110      set TEMP.path = TEMP.path + A[ J ]     // string operation
120      IF the length of TEMP.path > N
130        IF there exists a node Y with the same path string as TEMP
140          increase Y's occurTime by 1
150          set Y as current node
160        ELSE
170          add TEMP as a child of the root node T
180          set TEMP.occurTime = 1
190          set TEMP as current node
200      NEXT J
210    END WHILE
220  NEXT I
230  END WHILE
235  Update path weight of all nodes
240  END
```

Figure 2. An Algorithm to Build a Path Tree

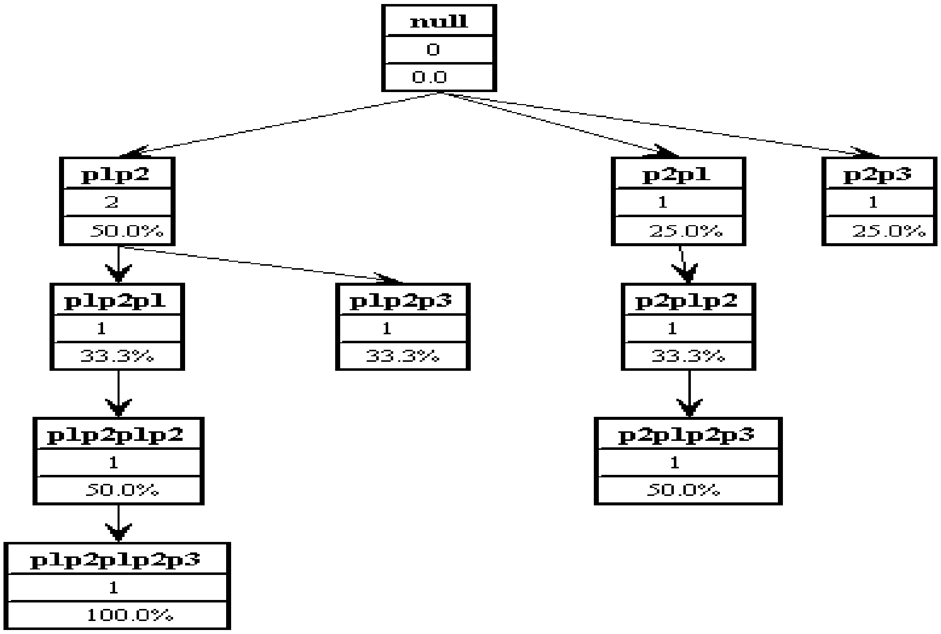
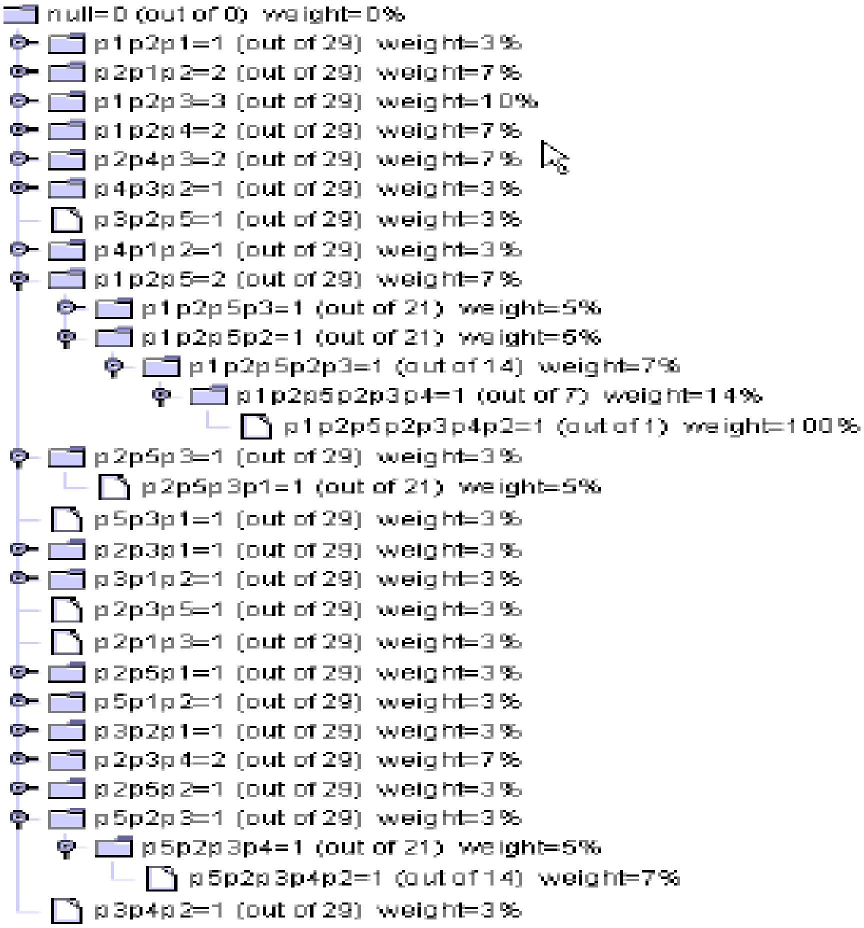


Figure 3. The Steps to Building a Path Tree



```
null=0 (out of 0)  weight=0%
  p1p2p1=1 (out of 29)  weight=3%
  p2p1p2=2 (out of 29)  weight=7%
  p1p2p3=3 (out of 29)  weight=10%
  p1p2p4=2 (out of 29)  weight=7%
  p2p4p3=2 (out of 29)  weight=7%
  p4p3p2=1 (out of 29)  weight=3%
  p3p2p5=1 (out of 29)  weight=3%
  p4p1p2=1 (out of 29)  weight=3%
  p1p2p5=2 (out of 29)  weight=7%
    p1p2p5p3=1 (out of 21)  weight=5%
    p1p2p5p2=1 (out of 21)  weight=5%
      p1p2p5p2p3=1 (out of 14)  weight=7%
        p1p2p5p2p3p4=1 (out of 7)  weight=14%
          p1p2p5p2p3p4p2=1 (out of 1)  weight=100%
  p2p5p3=1 (out of 29)  weight=3%
    p2p5p3p1=1 (out of 21)  weight=5%
  p5p3p1=1 (out of 29)  weight=3%
  p2p3p1=1 (out of 29)  weight=3%
  p3p1p2=1 (out of 29)  weight=3%
  p2p3p5=1 (out of 29)  weight=3%
  p2p1p3=1 (out of 29)  weight=3%
  p2p5p1=1 (out of 29)  weight=3%
  p5p1p2=1 (out of 29)  weight=3%
  p3p2p1=1 (out of 29)  weight=3%
  p2p3p4=2 (out of 29)  weight=7%
  p2p5p2=1 (out of 29)  weight=3%
  p5p2p3=1 (out of 29)  weight=3%
    p5p2p3p4=1 (out of 21)  weight=5%
      p5p2p3p4p2=1 (out of 14)  weight=7%
  p3p4p2=1 (out of 29)  weight=3%
```

Figure 4. Path Tree with Partially Collapsed Branch Nodes (Length ≥ 3)