Abstract - In the Web of Things (WoT) environment, Web activity logs contain important data of how individuals cooperate with shrewd gadgets and Web servers. Web traffic logs are composed of massive HTTP requests with information of corresponding responses. Data cleaning method which is used in the traditional system is not effective. It poses many technical challenges that emerge from the huge volume and low nature of information. Thus HTTP requests are collected as primary and secondary requests. Based on the count of these request, the Request dependency graph (RDG) are generated which models the dependency relationship among the requests. RDG will enhance the quality of web usage mining and it improves network and web server performance.

Index terms: Request dependency graph (RDG), Web of Things (WoT), Web usage mining.

1. Introduction

Information Mining is the computational procedure of finding examples in the vast information sets including Artificial Intelligence, Machine Learning, Statistics and Database Systems. It's fundamental point being "to concentrate data from an information set change it into a justifiable structure for further utilization". When all is said in done information mining is the way toward breaking down information from alternate points of view and outlining it into helpful data. It by and large results in the revelation of new examples in huge sets. It encourages clients to investigate information from various measurements/points of view, sort it and rundown the connections. It can likewise be begun as sorting through information to distinguish designs and set up connections. Organizations utilize this method to change crude information into helpful data. It makes utilization of methods, for example, Artificial Intelligence, Neural Networks, and Advanced measurable devices to uncover patterns, example and connections.
2. Related Surveys

Recent developments in the field of embedded devices have led to smart things increasingly populating our daily life. We define smart things as digitally enhanced physical objects and devices that have communication capabilities contributions materialize into an ecosystem of building-blocks for the Web of Things: a world-wide and interoperable network of smart things on which applications can be easily built, one step closer to bridging the gap between the virtual and physical worlds.

Being responsible for more than half of the total traffic volume in the Internet, HTTP is a popular subject for traffic analysis. From the experiences with HTTP traffic analysis it is identified as a number of pitfalls which can render a carefully executed study flawed. Often these pitfalls can be avoided easily. Based on passive traffic measurements of 20,000 European residential broadband customers, we [3] quantify the potential error of three issues: Non-consideration of persistent or pipelined HTTP requests, mismatches between the Content-Type header field and the actual content, and mismatches between the Content-Length header and the actual transmitted volume. Also found that 60% (30%) of all HTTP requests (bytes) are persistent (i.e., not the first in a TCP connection) and 4% are pipelined. Moreover, a Content-Type mismatch for 35% of the total HTTP volume is observed [3].

As Web sites move from relatively static displays of simple pages to rich media applications with heavy client-side interaction, the nature of the resulting Web traffic changes as well. Understanding this change is necessary in order to improve response time, evaluate caching effectiveness, and design intermediary systems, such as firewalls, security analyzers, and reporting /management systems. Unfortunately, we have little understanding of the underlying nature of today’s Web traffic. A new Web page analysis algorithm is introduced [6] that is better suited for modern Web page interactions by grouping requests into streams and exploiting the structure of the pages. Using this algorithm, various aspects of page-level changes are analyzed. Finally, the redundancy of this traffic is investigated, using both traditional object-level caching as well as content-based approaches. Identifying user clicks from a large number of measured HTTP requests is the fundamental task for web usage mining, which is important for web administrators and developers [7]. Nowadays, the prevalent parallel web browsing behavior caused by multi-tab web browsers renders accurate user click identification from massive a request which is a great challenge. Thus, a dependency graph model is proposed [7] to describe the complicated web browsing behavior. Based on this model, two algorithms are developed to establish the dependency graph for measured requests, and identify user clicks by comparing their probabilities of being primary requests with a self-learned threshold. Web prefetching is one of the techniques proposed to reduce user’s perceived latencies in the World Wide Web. The spatial locality shown by user’s accesses makes it possible to predict future accesses based on the previous ones. The existing prediction algorithms achieved an acceptable performance when they were proposed but the high increase in the amount of embedded objects per page has reduced their effectiveness in the current Web. Thus, they predict the accesses to the embedded objects in an HTML after reading the HTML itself. For this reason, the prediction advance is not enough to prefetch the objects and therefore there is no latency reduction. As a result of a wide analysis of the behavior of the most commonly used algorithms, the DDG algorithm is presented that distinguishes between container objects (HTML) and embedded objects to create a new prediction model according to the structure of the current Web. Results show that, for the same amount of extra requests to the server, DDG always outperforms the existing algorithms by reducing the perceived latency between 15% and 150% more without increasing the computing complexity [8].

There have been recent interests in studying the “goal” behind a user’s Web query, so that this goal can be used to improve the quality of a search engine’s results. Previous studies have mainly focused on using manual query-log investigation to identify Web query goals. Later, results are presented from a human subject study that strongly indicates the feasibility of automatic query-goal identification. Here, two types of features are proposed for the goal-identification task: user-click behavior and anchor-link distribution[10].

2.1 Preliminaries and Problem definition

Web traffic mining can be categorized into three types as per the locations of acquired traffic logs: client-side, server-side, and network-side traffic mining. An active area of research in recent years has been mining the web traffic at the client-side and server-side. On the basis of statistical and structural properties of complete web environment, web logs at server side are analyzed. There are moderately not many studies that consider the HTTP (the standard protocol underlying the web) requests as a set of associated records to passage the interests and the preferences of individuals at
the network-side. Also, the RDG are developed based on the requests collected from the web search history. Thus, RDG are also generated for the same number of requests and are not distinguished which is time consuming and makes the graph looks complicated. Thus the problems identified in the existing system are

I. Web traffic logs, which are composed of massive HTTP requests with information of corresponding responses, poses many technical challenges that arise from the large volume and low quality of data.

II. Web traffic analysis at network-side has been based on the statistical and structural properties of the overall web environment which are not analyzed fully in the previous surveys.

III. Web traffic in WoT is increasingly critical for network administrators for operational and security purposes.

2.2 Proposed work
Web traffic logs carries precious information of how people interact with smart devices and web servers in Web of Things environment. Mining the wealth of information present in the web access logs has theoretical and practical importance. Mining web logs involves modeling the relationships between HTTP requests. It provides a graphical representation of behavior of web clients. We introduce the concept of the request dependency graph (RDG) based on the Primary and Secondary request counts identified, which models the dependency relationships among HTTP requests to analyze the behavioral characteristics of Web traffic, such as interaction structures of Web objects and browsing patterns of web clients. Also, similar requests are combined as one using this request count method which saves time and increases the performance. We demonstrate the value of the RDG by proposing a new effective primary requests identification algorithm. We conduct extensive experiments based on real-world datasets to illustrate the structural characteristics of the graph and validate the primary requests identification algorithm. Thus, it enhances the quality of Web usage mining and achieves higher accuracy than the traditional data cleaning method.

2.3 Architecture diagram

3. Modules Description
3.1 Data collection from browsers
Data Collection to be done for analyzing the primary and. For this Analysis the Data has to be collected from the Browsers based on the history. We will Collect the collect the from Chrome and other browsers. the Web client like a Web browser or an embedded Web application will send an initial HTTP request containing the URL of this page to the Web server. Responded page content of this initial request usually contains many hyperlinks of the embedded objects. After parsing these hyperlinks, the Web client on the device produces a set of requests to retrieve embedded objects from Web servers in a multithread manner.

3.2 Identifying the Primary Request
The Primary request to be identified based the Data we collected from the browser. If the request is the Root Request it will be considered as a Primary Request. We also identify the number of same primary request in order to reduce the traffic. The primary requests are identified by comparing their probabilities of being the primary request with a self-learned threshold. Experiments on real-world data demonstrate that our method can achieve higher accuracy in comparison with traditional data cleaning (DC) method.

3.3 Identifying the Secondary Request
Commonly, a Web page contains two types of Web objects: 1) primary object and 2) secondary object.
The primary object is the first object requested to retrieve the page by the devices, such as the objects requested by r1, r4, and r6. Each Web page has only one primary object. Other objects in the Webpage are the secondary objects, such as the objects requested by r2, r5, and r7. The requests of retrieving the primary object and the secondary object are called primary request and secondary request, respectively. Example of the RDG is triggered by devices accessing a URL via Web application programming interface (API) or opening a hyperlink in a Web page. We call this kind of requests as primary requests, which are the key information source to reveal devices’ behaviors. Other requests are defined as secondary requests. A secondary request is the successor of a primary request on the temporal dimension. So, the major part of the algorithm is to identify the predecessor and successor relationships between the HTTP requests. We can see that the number of edges increases when the look ahead window increases because more requests are treated as the secondary requests that connect to the predecessor primary requests when the look ahead window gets larger.

3.4 Graph Generation

Based on the web browsing process and basic concepts we generate dependency graph model to sketch the dynamic web browsing. The RDG is initially empty and is established through a learning process, which is summarized in Algorithm. Each sequence is made up of requests from the same device, and ordered by the accessing time. For each successor request, a directed edge is added from the current predecessor request to this request, and the weight of this edge is incremented by one. In the DG, a node represents the accessed object and the occurrence count. The RDG can be applied to analyze the network traffic of WoT elements.

3.5 Traffic Reduction on WoT

An extensive analysis of a large graph derived from the traffic log containing millions of requests. Several interesting characteristics of the RDG have emerged from the analysis. In particular, the graph appears to be weakly connected, decentralized, heterogeneous, and a number of its measures are governed by power laws. Our method achieves higher accuracy as compared with the widely used DC method. Browsing behavior modeling and primary requests identification is fundamentally critical for subsequent Web usage mining. Our work will enhance the quality of Web usage mining, and benefit the analysis of user behaviors and interests to improve network and Web server performance.

4. Algorithm

The algorithm used in the project is to project is known as Establish the Request Dependency Graph. Here requests are identified from the web users. Based on the identification of primary and secondary request, a request dependency graph is generated. The request dependency graph (RDG) models the dependency relationships among HTTP requests to analyze the behavioral characteristics of Web traffic, such as interaction structures of Web objects and browsing patterns of Web clients. The RDG is initially empty and is established through a learning process, which is summarized in this algorithm. Conceptually, a directed link from A to B in the graph means that the accessing of Web object B is caused by the accessing of A, i.e., B depends on A. The input data of the algorithm is a set of HTTP requests R. Each request r_i maintains information including the device identification u_i, the accessing time t_i, and the URL of the accessed object o_i. They are sorted in ascending order of the accessing time. The output of the algorithm is the RDG G, which has a set of nodes O with occurrence counts S and a set of edges with weights W.

5. Results and Discussions

In this paper, we have proposed a RDG based on the identified primary and secondary requests count to show the confused or complicated Web-browsing behavior in the WoT environment. We have built up a philosophy to build up the RDG by preparing the arrangement of HTTP requests. We have introduced a broad investigation of a huge graph got from the activity log containing a large number of requests. A several interesting qualities of the RDG have risen up out of the examination. In specific, the chart has all the earmarks of being pitifully associated, decentralized, heterogeneous, and some of its measures are administered by power laws. At that point, we have demonstrated a key application, essential demands ID, in the Web use mining that can be adequately handled by the RDG. We have built up a essential requests ID.
calculation from enormous HTTP demands by a self-learning process in light of the graph model. Trial comes about have substantiated that our technique accomplishes higher exactness as contrasted and the generally utilized DC technique. We expect our work will improve the quality of Web usage mining, and benefit the examination of user behaviors furthermore, interests to enhance system and Web server performance.

6. Conclusion

Thus Request dependency graph generated based on the information retrieved in this project will enhance the quality of web usage mining and benefit the analysis of user behaviors and interests to improve network and web server performance. Also it achieves higher accuracy and significant for many applications like network optimization and security management. The future work is based on finding a way to decompose and visualize the large and complex RDG built from massive traffic logs. Also exploring more applications based on the RDG.

7. References