International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 2 | ISSN : 2456-3307

Fault Identification from Web Log Files by Pattern Discovery

M. S. Anbarasi*, B. Vasanthi

Department of Information Technology, Pondicherry Engineering College, Puducherry, India

ABSTRACT

In current scenario, the existing Web log files are informative based on website. However, even fault occurrence is high as the data increases tremendously. Because of fault occurrence, the browser provides fault pages with the pages to be searched pages. The manual process is not possible because of large amount of complex data. In the existing systems, identifying failure occurrence in web log file is difficult and time-consuming task. The basic reason is the large size and complexity of these systems, and the vast amount of monitoring data they generate. In existing system, fault identification technique does not provide maximum accuracy to improve the website. To overcome this problem the proposed system applying faults identification technique in efficient way using naïve sting matching algorithm with enhanced graph grammar[2] applied and then discover fault patterns from that browser find out root cause of failure.

Keywords: Web Log Files, Fault Identification, Naive String Matching Algorithm, Naïve Bayesian classifier, Pattern discovery.

I. INTRODUCTION

Information retrieval from search engine with web log files has major fault occurrence issue. Web log files are automatic log information maintained by a web server. Every "hit" to the Web site, includes each view of a HTML document, image or other objects. The format of web log file is essentially one line of text for each hit to the website.

Each line in the log file represents one request (hit). If a client requests an HTML page that contains two images represented with three lines (one for the page and two for the images). For this classification naïve Bayesian algorithm is used.

The naïve string-matching algorithm uses to identify fault from web log files. With the help of naïve Bayesian classification technique classify the faults information depends upon their fault categories in efficient way. For each condition one MB web log file data were consider for four classes at a time. Classified faults have some sub-categories.

After identification of fault, the graph grammar technique is applied. The false negative based on graph grammar technique infers the number of fault occurred in web log files.

RELATED WORK

In existing work, increasing deployment of Web services, the research on the dependability and availability of Web service composition becomes more and more active. Since unexpected faults of Web service composition may occur in different levels at runtime, log analysis as a typical data driven approach for fault diagnosis[1] is more applicable and scalable in various architectures. Considering the trend that more and more service logs are represent using XML or JSON format that has good flexibility and interoperability, fault classification problem of semi-structured logs is considered as a challenging issue in this area. Solution estimates degrees of similarity among structural elements[9] from heterogeneous log data, constructs combined Bayesian network (CBN), uses similarity based learning algorithm to compute probabilities in CBN, and classifies test log data into

most probable fault categories based on the generated CBN.

A lightweight, pattern based approach to specifying service interaction protocols. It has been incorporated into OWL-S for service developers to describe service interaction constraints. A framework for monitoring the run-time interaction behaviour of Web services and validating the behaviour against their pre-defined interaction constraints. The framework involves interception of service interactions/messages, representation of interaction constraints using finite state automata, and conformance checking of service interactions against interaction constraints. As such, the framework provides a useful tool for validating the implementation and use of services regarding their interaction behaviour.

Finding useful information from the Web becomes increasingly difficult as the volume of Web data rapidly grows. To facilitate effective Web browsing, Web designers usually display the same type of information with a consistent layout (referred to as a Web pattern). Discovering Web patterns can benefit many applications, such as extracting structured data. A Web pattern specified as a graph grammar approach, which is induced automatically through a grammar induction engine. Based on the induced pattern, matching instances are recognize from Web pages through a graph parsing process

Overcoming the above issues FIWLFPD (Identifying Fault from Web Log Files by Pattern Discovery) system is to be proposed.

II. METHODS AND MATERIAL

Searching helpful information from website without error is impossible, because the website grows rapidly day-by-day. The system errors and website errors identified through log files. The proposed system to improve the website performance is identified the faults through web log files using fault identification technique.

The web log files contain large amount of fault and it provides clue for identifying faults. The FIWLFPD system reduces the complexity of fault identification while applying naïve string matching algorithm compare to existing system. Improve the diagnosing fault from website; find out the root cause of faults from the website. The proposed system finds out maximum occurrence of the fault by using classification technique.

Naïve Bayesian classification algorithm, used for classifying faults into types depends upon their fault category. The fault related data are added to the training dataset with already identified category. To increases the quality of matching using graph grammar algorithm approachess. The proposed system improve accuracy and time by using naïve string matching algorithm[8] and enhanced graph grammar algorithm interesting patterns from log files are discovered. The architecture of this proposal is as shown in figure 1.

FIWLFPD System Architecture

In this proposal, the input is considered as a number of raw log data from different levels including, warnings, failure and error. Naive String Matching (NSM) is applied for extract fault related data from web log files. The fault records will be stored in training dataset with a dynamic environment and then log data can be classified into most probable fault categories using naïve classifier algorithm and set as different parameter.

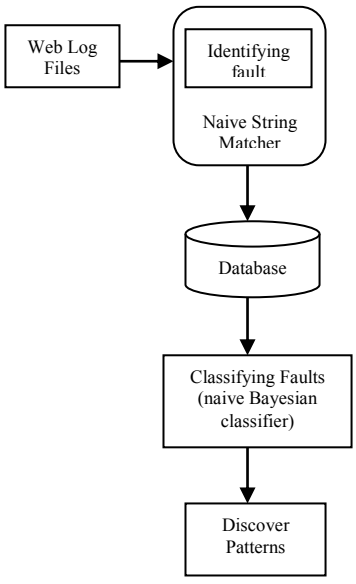

Figure 1. FIWLFPD System Architecture

The enhanced graph grammar algorithm is used to store the fault in the form of tree structure from the sample web log files. By applying graph grammar algorithm, the web pattern is automatically generated as a graph

structure and its produce maximum occurrences of fault from the website in a hierarchical structure.

This proposal recursively generates category wise graph from web log files using enhanced graph grammar algorithm. To increases the quality of pattern and to improve the efficiency of log processing time, naïve string matching algorithm is used. By applying the above-mentioned techniques in FIWLFPD system, fault diagnosis process is performed by applying below mentioned modules.

- Identifying web fault
- Naïve Bayes Classification for fault data
- Pattern Discovery

A. Identifying web fault using Naïve String Matching Algorithm

Fault analysis module, identifying faults in the website through web log files. The web log files contain information about the server and web user after that provides information about the web page fault. Some of types fault identified in the FIWLFPD system given in below

- Session Fault
- Link Fault
- Browser interaction
- Data Store Fault
- Page Fault

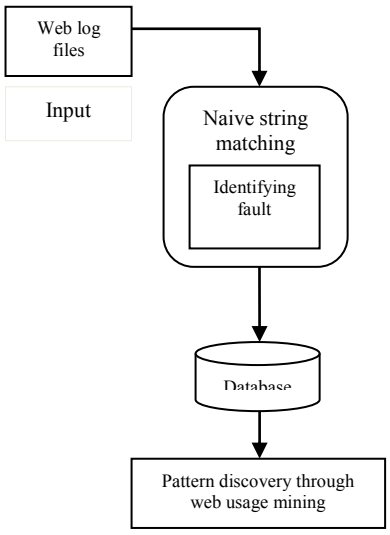

Figure 2. Fault Identification

This information is provided as the input of the Identifying web fault module. Naïve string matching algorithm is applied to remove the data which is incomplete and extract the complete, accurate fault relevant data's separately stored into the database. Naïve string matching algorithm does not require pre-processing, so it consumes processing time and improve the efficiency of fault identification.

In the fault identification module, Web log data is given as an input in structured format. Due to this a web log file may consists of some undesirable log entries, whose presence does not matters from the web usage mining point of view. In this module log files contains important fault related data, extracting fault related data using Naïve string matching algorithm and fault log will be stored in database. Web usage mining[3] is applied to discover pattern as shown in Figure 2.

B. Classifying Faults using Naïve Bayes Classifier

Naive Bayesian classifiers[5] are among the most successful known algorithms for learning to classify web log files. The labelled training data for fault classification is expensive, how to use a very large web log data as a source of unlabeled data to aid in automatic fault classification becomes a major issue. These web logs record may contains the Web user's behaviour when they search for information via a WWW.

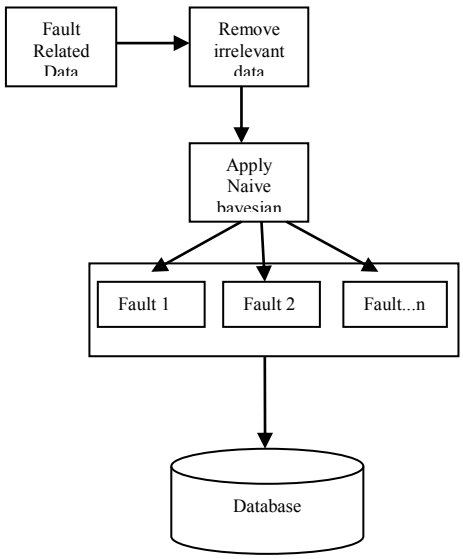

Figure 3. Naïve Bayesian Fault Classification

The fault logs have become a rich resource which contains Web user's knowledge about the World Wide Web. The web log dataset comprises of labels which indicates the class of the various observation of fault data. New fault is classified based on the training set. The entire fault[7] data is separated into various categories depends upon their categories and unknown fault data as created into new category that will be stored in the database shown in figure 3.

C. Discovery of patterns

Pattern discovery[6] model on web log files for web usage mining. The main approach of the fault identification system is to predict the web usage patterns using a sequence of steps are web log collection, fault log identification, fault classification and pattern discovery technique. Fault data is stored in different formats and structures in web servers. Web usage applications based on data collected from Apache access log. The log files stored the browser data into different formats and fields.

D. Fault Diagnosis

Fault diagnostics are probably the most common use of logs. In fault diagnosis, website admin know that a failure has occurred and try to find the root cause and the fault that caused the failure. The website developers typically use a database of sequences of events for known problems and search the database for the sequence of events that a browser submits to them. Web browser wants to identify the fault in the source code that caused the failure so fix the source code.

III. RESULTS AND DISCUSSION

The web log data used in FIWLFPD system experiments are collected from Apache web access log files[4]. The web log files have different levels of faults including error, attack, debugging, warning error, etc. The input dataset size is 50 MB with structured format and unstructured format. In Identifying web fault module is implemented to extract fault related from web log data through the regular expression pattern of websites. Each pattern in the regular expression represents unique information. While applying Naïve String Matching algorithm ignores the inaccurate and incomplete information data from raw web log data.

To simulate the data obtained from heterogeneous sources, web log data using various structured format and stored into training dataset. Classify the identified fault depends upon their category using naïve Bayesian classifier. Naïve Bayesian classifier classify the data effectively when compare to existing algorithm. And each fault category represented in graph structure using enhanced Graph Grammar Algorithm.

Experimental setup

Evaluating the throughput in FIWLFPD system defines the performance efficiency rate of web log files. Compare to the existing system the performance efficiency is better in the proposed system as shown in the graph below.

Input Web Log File

The input web log file is applied in visual studio as a common Log file or structured format file. In the input file it includes, an explicit timestamp, the message type, dynamic variable parameter information, and an integer that uniquely identifies the message type for each line in the log file. There could be a separate file with the tuple. A comma separated value format is also suggested. The authentication ID could also contain information about the location in the website from which log information originally came from.

Execution time in visual studio is a runtime during which a program is running. It is the time required by the process to execute the web log files. For the given web log file at time of execution it generates accurate fault category based on its type. The accuracy of this system is formulate as shown below

Accuracy = (Total number of classified fault web log files/ Total number of web log files)*100

A metric to evaluate the overall performance of fault diagnosis is throughput rate. It can recognize any repetitive structures within a Web log file. Those recognized records may belong to different categories. In additional evaluate the processing time and complexity of fault in the website.

Recall

The predictive accuracy of an inferred automaton is measured using the recall of the list it recommends. The recall of an information retrieval system is defined as the proportion of the totally available relevant fault. To calculate the recall, compute the percentage of the

faults in which the web log belongs in the recommended types of the fault.

Recall of some automata induced from the Web log data. The x axis represents the length of the dataset size. The graphs initial hypothesis is clearly superior over the models that were induced from a graph grammar. The graph shows that traditional grammatical inference methods can produce models with more than 94% recall in the different web log files.

Throughput Rate

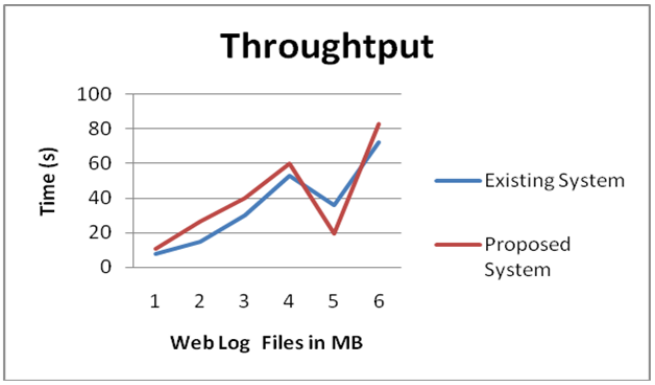

Figure 4. Throughput values of Existing system and proposed system.

Throughput time = Processing time + Identification Fault time + Waiting time + Inspection time.

Throughput indicates the number of faults occurrences in the web log files can handle, the amount of faults identified over time during a test. Lots of fault types identified from different web log files. To ensure that, load and performance testing is the solution. Also before starting a performance test it is common to have a throughput goal that the application needs to be able to handle a specific number of fault classifications per hour. Above Figure 4, represents the comparative result analysis of proposed and existing system. Throughput performance provides the efficiency of FIWLFPD system. The x axis represents the occurrence of fault from various web log files. The y axis represent the processing time of fault.

IV.CONCLUSION

Identify the fault related information from large amount of Web log data collected by Apache web servers transforming fault diagnosis problem into classification problem, we can utilize the corresponding classification methods to diagnose faults. The proposed a Naive Bayesian networks algorithm improve the accuracy of faults, when compare to constructing combined Bayesian networks, which are used as generative model to classify fault related log data. Then proposed system generates the graph for fault to optimizing the accuracy and efficiency of web patterns through fault identification for web log files.

V. REFERENCES

[1] A.Benharref, R. clitho, R. Dssouli, “A web service based architecture for detecting faults in web services”, IEEE 7803-9088-I/05, 2005.

[2] Amin Roudakia, Jun Kong and Kang Zhang (2016), “Specification and discovery of web patterns: a graph grammar approach”, Elsevier on Information Science in ScienceDirect.

[3] Cooley.R.,“Web Usage Mining: Discovery and Application of Interesting Patterns from Web data”, 2000.

[4] K. R. Suneetha et. al,"Identifying User Behavior by Analyzing Web Server Access Log File", International Journal of Computer Science and Network Security(IJCSNS), Vol. 9, pp. 327-332, 2009

[5] A. K. Santra and S. Jayasudha, “Classification of Web Log Data to Identify Interested Users Using Naïve Bayesian Classification ” IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, ISSN- 1694-0814, pp.381-387,Jan 2012.

[6] Namit Jain, Bamshad Mobasher, Eui_Hong _Sam_ Han and Jaideep Srivastava, “Web Mining Pattern Discovery from World Wide Web Transactions”, Technical Report, Department of Computer Science, University of Minnesota (1996).

[7] Marchetto, F. Ricca and P. Tonella, “Empirical Validation of a Web Fault Taxonomy and its usage for Fault Seeding”, In the IEEE Int. Symposium on Web Site Evolution, 2007.

[8] Yanhong Cui and Renkuan Guo “A Naïve String Algorithm”, Proceedings of IEEE International Workshop on Education Technology and Training,2008.

[9] Xu Han , Binyang Li, Kam-Fai Wong and Zhongzhi Shi (2016), “Exploiting structural similarity of log files in fault diagnosis for Web service composition” ,Elsevier on intelligence technology in ScienceDirect.