Development of Decision Tree Algorithm for Mining Web Data Stream

Sheetal Sharma
Research Scholar
Sanghvi Innovative Academy,
Indore

Swati Singh Lodhi
Sanghvi Innovative Academy
Indore

ABSTRACT
World Wide Web presents challenging aspects or task for mining web data stream. Currently processing of useful data from web data stream is getting complex because when we considering the large volume of web log data it does not provide well-structured data. Two major challenge involved in web usage mining are processing the raw data to provide a (very close to the truth or true number) picture of how site is being used, and filtering the result of different data mining set of computer instructions in order to present only rules and patterns. In this work we develop decision tree algorithm, which is efficient mining method to mine log files and extract knowledge from web data stream and generated training rules and Pattern which are helpful to find out different information related to log file**.**

Keywords
Web Usage Mining, Decision Tree, Temporal Rule Mining...

1. INTRODUCTION
Web mining may defined as revelation and analysis of subsidiary information from the World Wide Web. Predicated on the different accentuation and different ways to obtain information, web mining can be divided into two major components: Web Contents Mining and Web Utilization Mining. Web Contents Mining can be defined as the automatic search and retrieval of information and subsidiary things/valuable supplies available from millions of sites and on-line (computer files full of information) though search engines / web spiders. Whereas; Web Utilization Mining can be described as the revelation and analysis of utilize access patterns, through the mining of log files and connected data from a particular Web site.

Web usage mining, from the data mining aspect, is the task of applying data mining techniques to discover usage patterns from Web data in order to understand and better serve the needs of users navigating on the Web every data mining task, the process of Web usage mining also consists of three main steps: (i) preprocessing, (ii) pattern discovery and (iii) pattern analysis. In this work pattern discovery means applying the introduced frequent pattern discovery methods to the log data. For this reason the data have to be converted in the preprocessing phase such that the output of the conversion can be used as the input of the algorithms. Pattern analysis means understanding the results obtained by the algorithms and drawing conclusions**.**

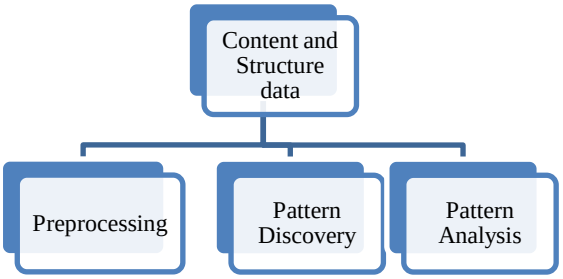


Figure 1 Web Usage Mining Process

Figure 1 demonstrates the procedure of Web utilization mining acknowledged as a contextual analysis in this work. As can be seen, the information of the procedure is the log information. The information must be preprocessed with a specific end goal to have the proper data for the mining calculations. The diverse routines need distinctive information positions, subsequently the preprocessing stage can give three sorts of yield information. The successive examples disclosure stage needs just the Web pages went by a given client. For this situation the arrangements of the pages are immaterial. Additionally the copies of the same pages are discarded, and the pages are requested in a predefined request.

The entire literature survey will be mainly focused on Web Usage Mining and Pattern Discovery in Web data streams.

The entire literature survey will be mainly focused on Web Usage Mining and Pattern Discovery in Web data streams.

To perform any site assessment, web guest's data assumes an essential part, keeping in mind the end goal to help this, numerous devices are accessible. L,Zhang and C. Also, Zhang [8] communicated that Web Mining is a prominent method for dissecting site guest's behavioral examples in e-administration frameworks. Chungsheng Zhang and Liyan Zhuang [6] found that Web Log Mining aides in removing intriguing and helpful examples from the Log File of the separate. Lee Tan [10] recommended that HTML archives contain more number of pictures on the WWW. Such reports' containing important pictures guarantees a rich wellspring of pictures bunch for which inquiry can be produced.

The reports which are exceptionally required by clients can be set close to the landing page of the site R. G. Tiwari [2] recommended that the advancement of web mining strategies, for example, web measurements and estimations, web administration enhancement, procedure mining and so forth and will empower the force of WWW to be figured it out. Jungie Chen and Wei Liu [12] found that shortcoming of both recurrence and utility can be overcome by General Utility Mining Model Archana N.Mahanta [5] uncovered that the structure of connected pages has unequivocal effect figure on the convenience Govardhan et al. [4] recommended that the quantity of pages at a specific level, the quantity of forward

connections and the quantity of in reverse connections to a specific site page mirror the conduct of guests to a particular page in the website.

However Govardhan [4] called attention to that the quantity of hit tallies ascertained from Log File is an inconsistent pointer of page prevalence. Geeta& others [10] recommended that the topology of the site assumes a vital part notwithstanding log record measurements to help clients to have snappy reaction. Jungie Chen and others [12] found that Web Usage Mining aides in finding web navigational examples for the most part to foresee route and enhance site administration. Lin and others [10] demonstrated that the web behavioral examples can be utilized to enhance the configuration of the site. These examples additionally could help in enhancing the business knowledge.

2. PROBLEM IDENTIFICATION

Web usage mining is application of data mining ways of doing things to web click stream data in order to extract usage data. As website continue to growth in size and complex difficulty, the result of web usage mining have become critical for some application such as web site design, Two major challenge involved in web usage mining are processing the raw data to provide a (very close to the truth or true number) picture of how site is being used, and filtering the result of different data mining set of computer instructions in order to present only rules and patterns. For the mining data from weblog files an effective and efficient algorithm is required that works with high performance. Moreover it required to authenticate the algorithm for that purposes we use a traditional algorithm for mining sequential pattern from web log data.

In this work author develop decision tree algorithm, which is efficient mining method to mine log files and extract knowledge from web data stream and generated training rules and Pattern which are helpful to find out different information related to log file. Author increase accuracy of generating non redundant association rule for both nominal and numerical data with less time complexity and memory space. In this method Author use N-fold cross validation technique for performance evaluation and for classification of data set author is using decision learning algorithm with some modification in decision tree algorithm.

To resolve the need of effective and efficient algorithm author propose solution based on following facts:

1. Search a most frequently used sequential web log mining algorithm
2. Implement the found algorithm
3. Find the performance study of that algorithm
4. Compare the performance parameters for comparative analysis

A decision tree is basically a stream graph of inquiries or information directs that at last leads toward a decision [6]. For instance, an auto purchasing decision tree may begin by asking whether you need a 1999 or 2000 model year auto, then solicit what sort from auto, then ask whether you incline toward force or economy, et cetera. At last it can figure out what may be the best auto for you.

Decision trees frameworks are joined in item choice frameworks offered by numerous merchants. They are incredible for circumstances in which a guest goes to a Web website with a specific need [7-8]. However, once the choice has been made, the responses to the inquiries contribute little to focusing on or personalization of that guest later on.

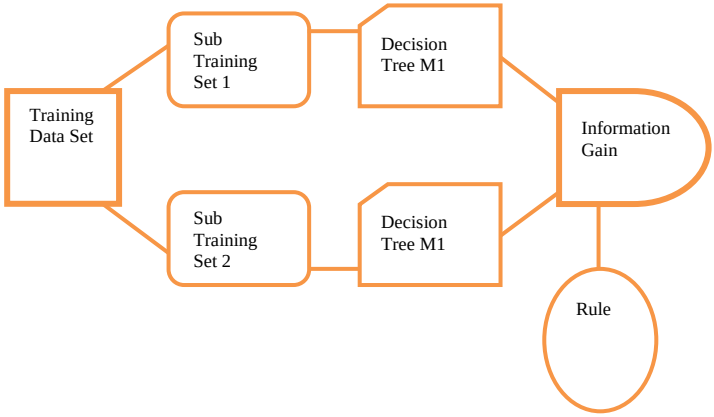


Figure 2 Decision Tree Classifiers

3. DEVELOPMENT OF ALGORITHM

This project is designed with the main aim to mine log files and extract knowledge from the experimental web log and after training rules are generated. these rules are helpful to find out different information related to log file. For that purpose author propose architecture to generate the rules from the experimental data set. This is done in these phases

1. Data collection
2. Data processing using selected model
3. Model building and model evaluation
4. Pattern Discovery

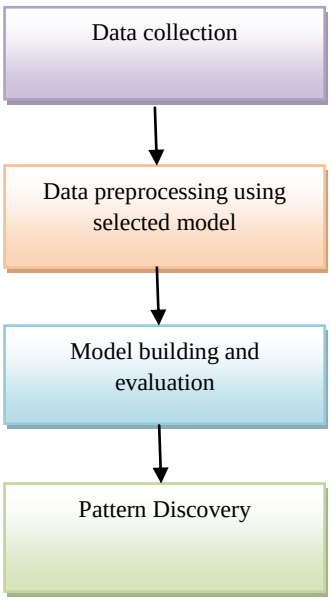


Figure 3 shows the basic structure of our proposed model

The proposed work and thesis follows the following steps:

1. **Experimental data selection:** in this phase required to input log files in to the system for analysis the input log files are in w3c format
2. **Data processing:** in this phase system clean the data and separate them and arrange them.

3. **Model building and evaluation:** in this phase of system processing using the supplied data is converted in to data model using the selection of algorithm in other words selected data model is used to prepare a navigational model for queries of user.
4. **Performance study:** in this phase author calculate the performance parameters for results analysis.

Proposed System Architecture

Below given diagram shows the system architecture of desired system. In this diagram author show the different sub systems of the complete system. These sub systems are work together and form the complete system. To describe complete systems working we describe each stage of processing one by one.

- Experimental data selection: using this module author supply input to the system and using this data author prepare navigational model in next phase.
- Algorithm selection: here required to select an appropriate data model to work with.
- After selection of algorithm there are two different algorithms are implemented and using the data author generate data model according to the supplied data.
- Model generation: selected algorithm here works over the supplied data and generates rules for prediction.
- Result evaluation: after model generation here author check the authenticity of models and evaluate performance parameters.

Algorithm used

1. select data set D
2. find list of all attributes in data set
3. check attributes data types
4. if all attributes = numerical data type
   a. get average of each attributes mark as threshold value
   b. compare with all selected attributes
      i. if attribute value <= threshold then
      ii. mark as 0
      iii. else
      iv. mark as 1
      v. end if
   c. find distance from all instance of data set
   d. arrange according to distance
5. if all attributes = nominal data type then
   a. find all unique attributes to attributes list
   b. get threshold for each attributes using the given formula
   c. threshold = (total unique values/ total count)$\log_n$ (total unique values/ total count)
   d. calculate the index of each unique value using the given formula=(no of values in list/ total values)$\log_n$(no of values in list/ total values)
   e. Assign label index to the values and compare with threshold
   f. Find distance for all instance
6. end if
7. return classes

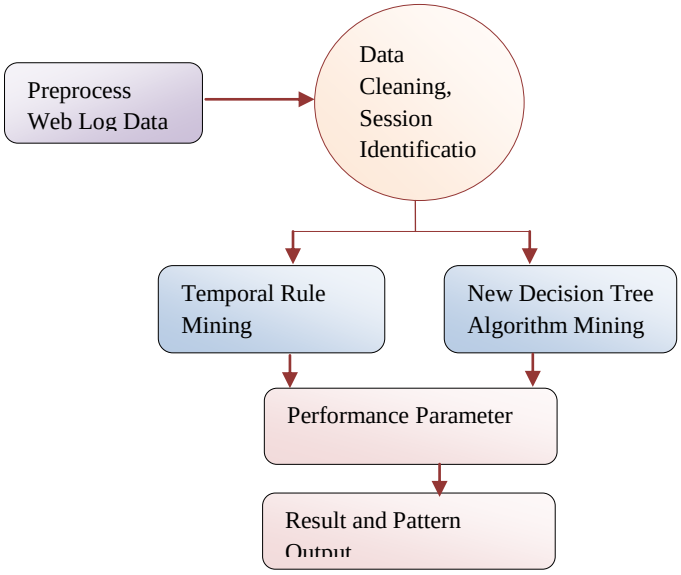


Figure 4 Proposed System's Architecture

Example

Table 1 Input data set

S.No	IP address	Method	URL	Agent
1	151.48.123.70	GET	http://www.smsync.com	Mozilla/4.0
2	151.48.123.70	GET	http://www.smsync.com	Mozilla/4.0
3	200.88.101.168	HEAD	http://www.123loganalyzer.com	Mozilla/5.0
4	200.88.101.168	GET	http://www.smsync.com	Mozilla/5.0
5	86.132.136.211	GET	http://www.123loganalyzer.com	Mozilla/4.0
6	151.48.123.70	HEAD	http://www.google.com/source	Mozilla/4.0

Unique values of IP address is =3

Unique values of Method is =2
Unique values of URL is =3

Unique values of IP address is =3

Unique values of Agent =2

If there is assume target value is agent.

Threshold of Input data set is

$S = - (4/6) \log_2 (4/6) - (2/6) \log_2 (2/6)$

$=0.39+.52$

$=0.91$

Notice threshold is 0 if all members of S belong to the same class (the data is perfectly classified). The range of entropy is 0 ("perfectly classified") to 1 ("totally

Output of data

Relation Name: Server Log File

Number of Instances: 24

Attributes:

Method

Requested_Value

Requested_Value = /images/download.gif

Method = GET: http://www.123loganalyzer.com/

Method = HEAD: http://www.123loganalyzer.com/

Method = POST: null

Requested_Value = /images/samle.gif

Method = GET: http://www.123loganalyzer.com/

Method = HEAD: http://www.123loganalyzer.com/

Method = POST: null

Requested_Value = /images/contact.gif

Method = GET: http://www.123loganalyzer.com/

Method = HEAD: null

Method = POST: http://www.123loganalyzer.com

Use case analysis:

Here we provide the user case analysis of the system. Below given table describe the use case details of the system

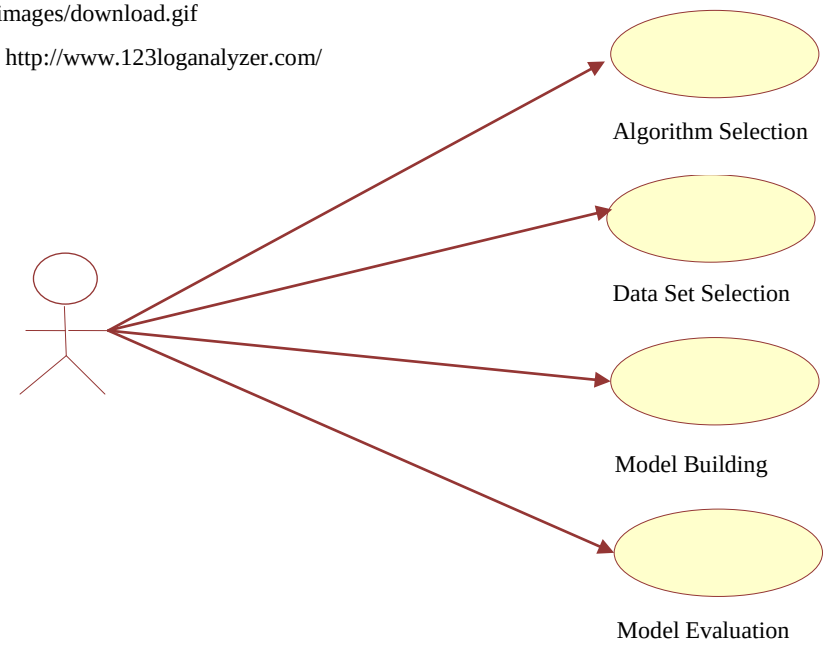


Figure 5 Use case diagram

Table 2 Description of Use case diagram

1.Use Case	Algorithm Selection
Description	here user select decision tree for evaluation
Actors	Software User
Assumptions	User should know about the user interface and also required the knowledge about data mining algorithms
Steps	Select decision tree algorithm for test
Variations	NON
2. Use Case	Data set selection

Description	user select test data set to construct tree
Actors	Software User
Assumptions	User should know about the user interface and also required the knowledge about data set
Steps	select test data set
Variations	NON
3. Use Case	Model Building
Description	to build model using the second step data user follow the step build model
Actors	Software User
Assumptions	User should know about the user interface and also required the knowledge about data set
Steps	Start Building model
Variations	NON
4. Use Case	Model Evaluation
Description	Build model is a training phase of algorithm for tree and for evaluation required to test the build model, after testing randomly selected data from data set supplied we discover the following factors accuracy, error rate, time to built model, time for search and memory required.
Actors	Software User
Assumptions	User should know about the user interface and factors calculated
Steps	End of Function
Variations	NON

System classes and library

Table 3 Description of System classes and library

S.No	Class name	Description
1	java.io.File	An abstract representation of file and directory pathnames. User interfaces and operating systems use system-dependent pathname strings to name files and directories. This class presents an abstract, system-independent view of hierarchical pathnames.
2	java.util	Contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string tokenizer, a random number generator, and a bit array).
3	javax.swing	Provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms.
4	java.awt.event	Provides interfaces and classes for dealing with different types of events fired by AWT components.
5	Java.sql.*	Provides the API for accessing and processing data stored in a data source (usually a relational database) using the JavaTM programming language. This API includes a framework whereby different drivers can be installed dynamically to access different data sources.

User defined classes

Table 4 Description of User defined classes

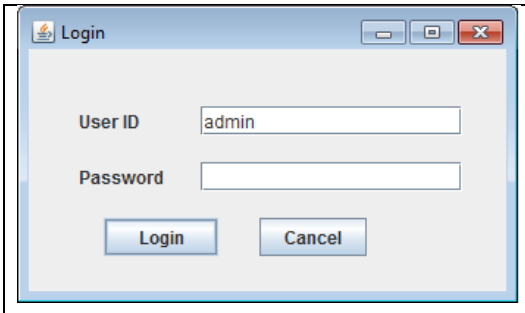
S.No.	Class Name	Description
1	ALGO1_Calculation	This class is responsible for containing the old algorithm member function and their complete algorithm implementation
2	Main Form	It is MDI form contains J Desktop Plane to organize all file related to project
3	Temporal rule mining	Using this GUI user can evaluate data and results using temporal mining algorithm.
4	My Data Set	Using this GUI user can import log file data and preprocess the complete data
5	N Cross Validation	This is a simple java class implementation to evaluate the performance parameters of both algorithm
6	String Tokenizer	This is a simple java class implemented to convert strings into small tokens
7	Attribute	This is class help to define data tokens as the attributes
8	Decision tree algo	This class is complete implementation of our proposed algorithm for the purpose of finding pattern of data from log file

Method Signature

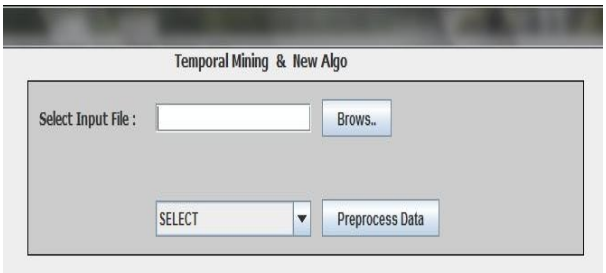
Table 5 Signature of method

S.No	Method Name	Signature
1	Load_DataSet()	It is a user defined method created to connect with data set used to build model
2	DoClassification()	To perform classification using our proposed algorithm this method is called after loading the data set
3	EvaluateModel()	To perform model evaluation we use this function
4	getDiff()	To extract the Execution time we can use this method
5	GetSampleError()	To find error in this build model we use
6	getRuntime().totalMemory()	To get memory consumption using the particular algorithm execution we can use this function

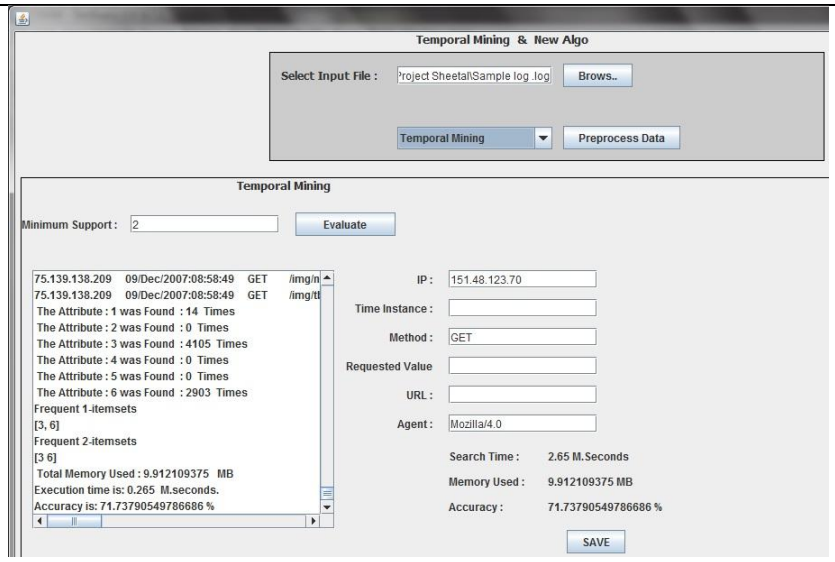
Screen shots of Research Work:



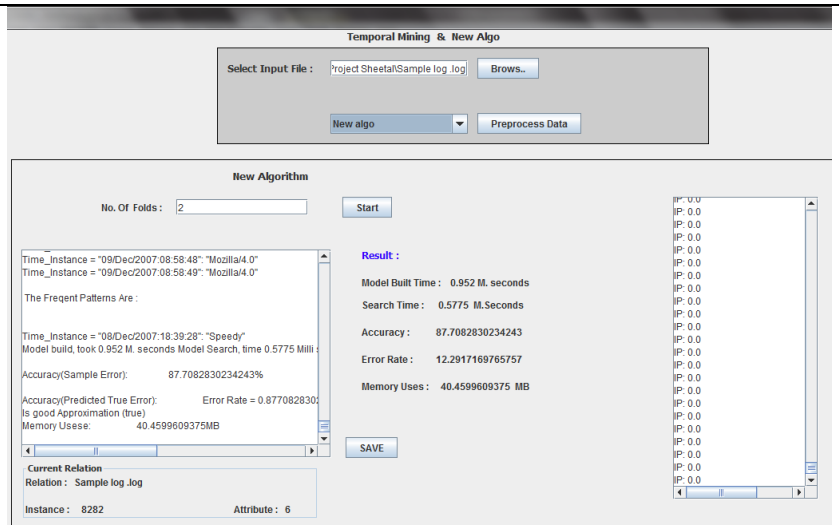
Login form



Main Form



Execution of Temporal Mining algorithm



Execution of New algorithm

4. RESULT AND DISCUSSION

Accuracy: accuracy of the system is defined by the actually predicted values verses wrong values predicted. The accuracy of system is calculated using the cross validation in this method we calculate the values using given formula

$$Accuracy = \frac{total\ values - wrong\ values}{total\ values}\ X\ 100$$

Accuracy of the system is derived using above formula and here we include the results obtained by the system in first five experiments

Table 6 Comparison of Accuracy of both Temporal Mining and New Algorithm

No. of Execution	Temporal Mining	New Algorithm	No. of attribute taken
1	71.50 (support=2)	87.7%(No.of fold=2)	3
2	83.45%(support=3)	98.77%(No.of fold=3)	3
3	71.24%(support=4)	86.81%(No.of fold=4)	3
4	71.26%(support=5)	93.25%(No.of fold=5)	3
5	71.26%(support=5)	99.91%(No.of fold=5)	3

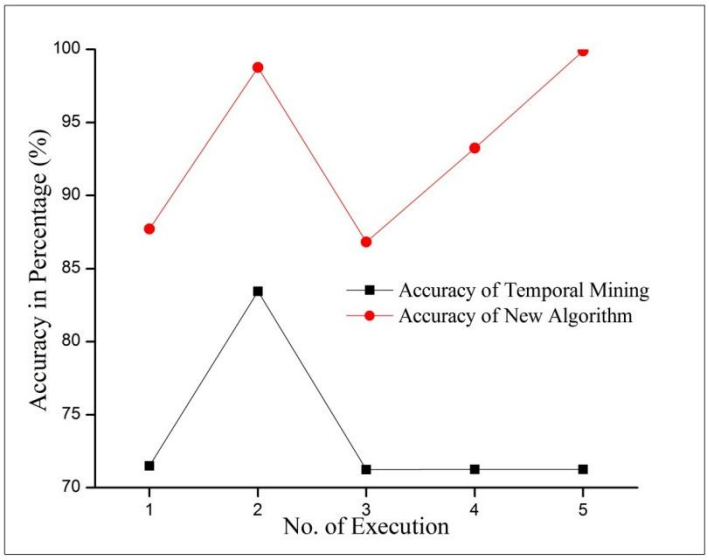


Figure 7.1 Graphical representation of Accuracy

The evaluation of results is performed for Temporal Mining by minimizing the support and increase the parameter after applies such condition we found that as we minimize the support and increase the parameters accuracy of system decreases.

Moreover it proposed method include all parameters and thus simulate better results for the evaluation of such kind of data.

Execution Time: to find the execution time we calculate the time required to build model results evaluation time included and we found that below given results.

Table 7 Comparison of Execution of both Temporal Mining and New Algorithm

No.of Execution	Temporal Mining	New Algorithm	No. of attribute taken
1	0.77 (support=2)	0.521 (No. of fold=2)	3
2	1.53 (support=3)	1.063 (No. of fold=3)	3
3	1.36 (support=4)	1.08(No. of fold=4)	3
4	1.25 (support=5)	1.10 (No. of fold=5)	3
5	2.17 (support=5)	1.94 (No. of fold=5)	3

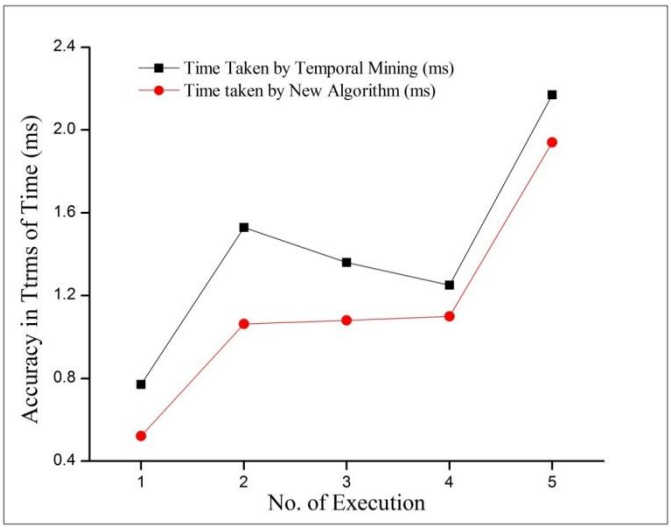


Figure 7.2 Graphical representation of Execution time

According to our analysis we found that execution time simulate is our algorithm is better than Temporal Mining Because the Temporal Mining time consumption graph is more uneven than proposed algorithm. And it is also considered that most of the time our model is much efficient then Temporal Mining.

Memory uses: requirement of main memory to execute the algorithm is defined as memory uses. The results simulate the memory used in terms of MB.

Table 8 Comparison of Memory Consumption of both Temporal Mining and New Algorithm

No. of Execution	Temporal Mining	New Algorithm	No. of attribute taken
1	20.051(support=2)	81.49(No. of fold=2)	3
2	85.74(support=3)	104.79(No. of fold=3)	3
3	55.41(support=4)	51.49(No. of fold=4)	3
4	16.82(support=5)	47.18(No. of fold=5)	3
5	98.52(support=5)	57.50(No. of fold=5)	3

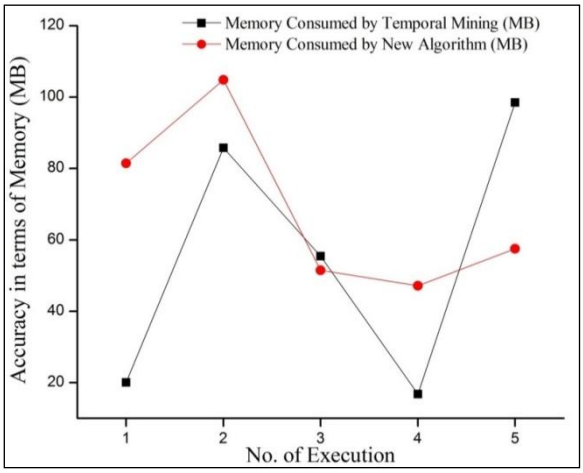


Figure 7.3 Graphical representation of Memory Consumption

From the results on evaluation of Using above results we can clearly see that temporal rule algorithm consumes less memory then our proposed algorithm

5. CONCLUSION

After evaluation of the obtained results, it was observed that the proposed work withstand with all the supplied input parameters. However the temporal Mining calculation resulted to work with selected parameters. Also, it was observed that the developed new algorithm performs better precise results than temporal mining although it was achieved by a fewer compromise of Memory employments. Consequently we can synopses the accompanying truths about our work.

1. Accuracy of proposed algorithm 75%-85% is better than Temporal Mining algorithm
2. Memory uses of proposed algorithm were found to be higher than Apriori.
3. Time required to execute model is 85%-95% less than Temporal Mining algorithm
4. Proposed algorithm is good algorithm but when where required less resource it is fail to work with low configuration system.
5. Memory uses of proposed algorithm is 80%-85% is higher than Temporal Mining algorithm
6. Temporal mining performs better where the need of resources are less and supplied parameters are less.

6. REFERENCES

[1] Nazli Mohd Khairudin, AidaMustapha, andMohd Hanif Ahmad (2014). Effect of Temporal Relationships in Associative Rule Mining for Web Log Data System. Hindawi Publishing Corporation Scientific World Journal.

[2] Agrawal, M. Husain, R. G. Tiwari, and S. Vishwakarma(2011), "Web information recuperation from strewn text resource systems,"International Journal of Advances in Engineering and Technology,vol. 1, no. 2, pp. 126–137

[3] Arumugam G. and Suguna S(2009),"Optimal Algorithms for Generation of User Session Sequences Using Server Side Web User Logs, ",ESRGroups, France

[4] D. Vasumathi and A. Govardhan(2009), "Efficient web usage mining based on formal concept analysis," Journal of Theoretical and Applied Information Technology, vol. 9, no. 2, pp. 99–109.

[5] Archana N.Mahanta(2008) ,"Web Mining:Application of Data Mining,",of NCKM ,

[6] Chungsheng Zhang and Liyan Zhuang(2008) , "New Path Filling Method on Data Preprocessing in Web Mining ,", Computer and Information Science Journal

[7] V. S. Tseng, K.W. Lin, and J.-C. Chang(2008), "Prediction of user navigation patterns by mining the temporal web usage evolution," Soft Computing, vol. 12, no. 2, pp. 157–163

[8] Zhuang, L., Kou, Z., & Zhang, C. (2005). Session identification based on time interval in web log mining. In Intelligent information processing II (pp. 389-396): Springer-Verlag

[9] E. Winarko and J. F. Roddick(2007), "ARMADA—an algorithm for discovering richer relative temporal association rules from interval-based data," Data and Knowledge Engineering, vol. 63, no. 1, pp. 76–90 .

[10] E. Keogh, J. Lin, S.-H. Lee(2007), and H. Van Herle, "Finding the most unusual time series subsequence: algorithms and applications," Knowledge and Information Systems, vol. 11, no. 1, pp. 1–27,

[11] Jose M. Domenech1 and Javier Lorenzo(2007), "A Tool for Web Usage Mining , " , 8th International Conference on Intelligent Data Engineering and Automated Learning.

[12] Jungie Chen and Wei Liu(2006), "Research for Web Usage Mining Model,", International Conference on Computational Intelligence for Modelling Control and Automation, IEEE.

[13] Suresh R.M. and Padmajavalli .R.(2006) ,"An Overview of Data Preprocessing in Data and Web usage Mining ," IEEE.

[14] K. Verma and O. P. Vyas(2005), "Efficient calendar based temporal association rule," SIGMOD Record, vol. 34, no. 3, pp. 63–70.

[15] Y. Li, P.Ning, X. S.Wang(2003), and S. Jajodia, "Discovering calendarbased temporal association rules," Data and Knowledge Engineering, vol. 44, no. 2, pp. 193–218.

[16] Gaul, W., & Schmidt-Thieme, L. (2001). Mining Generalized Association Rules for Sequential and Path Data. Proceedings of the 2001 IEEE International Conference on Data Mining.

[17] Wang, S., Gao, W., & Li, J. (2000). Discovering Sequence Association Rules with User Access Transaction Grammars. Proceedings of the 11th International Workshop on Database and Expert Systems Applications.

[18] Srivastava, J., Cooley, R., Deshpande, M., & Tan, P.-N. (2000). Web usage mining: discovery and applications of usage patterns from Web data. SIGKDD Explorations Newsletter, 1(2), 12-23

[19] J. M. Ale and G. H. Rossi(2000), "An approach to discovering temporal association rules," in Proceedings of the ACM Symposium on Applied Computing (SAC '00), vol. 1, pp. 294–300.

[20] Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. SIGMOD Rec., 29(2), 1-12.

[21] Cooley, R. (1997). Web Mining: Information and Pattern Discovery on the World Wide Web. Proceedings of the 9th International Conference on Tools with Artificial Intelligence.

[22] Agrawal, R., & Srikant, R. (1995). Mining sequentialpatterns. Proceedings of the Eleventh International Conference on Data Engineering, 1995