

# Business Intelligence from Web Usage Mining

Ajith Abraham

Department of Computer Science, Oklahoma State University,  
700 N Greenwood Avenue, Tulsa, Oklahoma 74106-0700, USA

[ajith.abraham@ieee.org](mailto:ajith.abraham@ieee.org)

<http://ajith.softcomputing.net>

**Abstract.** The rapid e-commerce growth has made both business community and customers face a new situation. Due to intense competition on the one hand and the customer's option to choose from several alternatives, the business community has realized the necessity of intelligent marketing strategies and relationship management. Web usage mining attempts to discover useful knowledge from the secondary data obtained from the interactions of the users with the Web. Web usage mining has become very critical for effective Web site management, creating adaptive Web sites, business and support services, personalization, network traffic flow analysis and so on. This paper presents the important concepts of Web usage mining and its various practical applications. Further a novel approach called "intelligent-miner" (*i-Miner*) is presented. *i-Miner* could optimize the concurrent architecture of a fuzzy clustering algorithm (to discover web data clusters) and a fuzzy inference system to analyze the Web site visitor trends. A hybrid evolutionary fuzzy clustering algorithm is proposed to optimally segregate similar user interests. The clustered data is then used to analyze the trends using a Takagi-Sugeno fuzzy inference system learned using a combination of evolutionary algorithm and neural network learning. Proposed approach is compared with self-organizing maps (to discover patterns) and several function approximation techniques like neural networks, linear genetic programming and Takagi-Sugeno fuzzy inference system (to analyze the clusters). The results are graphically illustrated and the practical significance is discussed in detail. Empirical results clearly show that the proposed Web usage-mining framework is efficient.

**Keywords:** Web mining; knowledge discovery; business intelligence; hybrid soft computing; neuro-fuzzy-genetic system.

## 1. Introduction Related Research

The WWW continues to grow at an amazing rate as an information gateway and as a medium for conducting business. Web mining is the extraction of interesting and useful knowledge and implicit information from artefacts or activity related to the WWW (Cooley, 2000; Kosala & Blockeel, 2000). Based on several research studies we can broadly classify Web mining into three do-

main: content, structure and usage mining (Chakrabarti, 2003; Chang *et al.*, 2001). This paper is concerned with Web usage mining. Web servers record and accumulate data about user interactions whenever requests for resources are received. Analyzing the Web access logs can help understand the user behaviour and the web structure. From the business and applications point of view, knowledge obtained from the Web usage patterns could be directly applied to efficiently manage activities related to e-business, e-services, e-education and so on (Chen & Kuo, 2000; Cheung *et al.*, 1997). Accurate Web usage information could help to attract new customers, retain current customers, improve cross marketing/sales, effectiveness of promotional campaigns, track leaving customers and find the most effective logical structure for their Web space (Heer & Chi, 2001; Jespersen *et al.*, 2002). User profiles could be built by combining users' navigation paths with other data features, such as page viewing time, hyperlink structure, and page content (Cho *et al.*, 2003). What makes the discovered knowledge interesting had been addressed by several works (Hay *et al.* 2003; Heinrichs & Lim, 2003; Runkler & Bezdek, 2003). Results previously known are very often considered as not interesting. So the key concept to make the discovered knowledge interesting will be its novelty or unexpected appearance (Aggarwal *et al.*, 1999; Agrawal & Srikant, 1994; Coenen *et al.*, 2000; Nanopoulos *et al.*, 2002).

A typical Web log format is depicted in Fig. 1. Whenever a visitor accesses the server, it leaves the IP, authenticated user ID, time/date, request mode, status, bytes, referrer, agent and so on. The available data fields are specified by the HTTP protocol. There are several commercial software that could provide Web usage statistics (Analog, 2003; ClickTracks, 2003; Hitbox, 2003; LogRover, 2003; Website Tracker, 2003; WebStat, 2003). These stats could be useful for Web administrators to get a sense of the actual load on the server. However, the

```

64.68.82.66 -- [17/May/2003:03:41:23 -0500] "GET /marcin HTTP/1.0" 404 318
192.114.47.54 -- [17/May/2003:03:41:33 -0500] "GET /~aa/isda2002/isda2002.html HTTP/1.1" 404 350
216.239.37.5 -- [17/May/2003:03:41:43 -0500] "GET /~ijcr/Vols/vol10no1.html HTTP/1.0" 200 4568
218.244.111.106 -- [17/May/2003:03:41:51 -0500] "GET /~aa/his/ HTTP/1.1" 404 332
64.68.82.18 -- [17/May/2003:03:42:15 -0500] "GET /~pdcp/cfp/cfpBookReviews.html HTTP/1.0" 304 -
212.98.136.62 -- [17/May/2003:03:43:21 -0500] "GET /cs3373/programs/pgm03.dat HTTP/1.1" 200 498
212.98.136.62 -- [17/May/2003:03:43:26 -0500] "GET /cs3373/programs/pgm04.html HTTP/1.1" 200 55722
212.98.136.62 -- [17/May/2003:03:43:38 -0500] "GET /cs3373/images/WaTor.gif HTTP/1.1" 200 39021
212.29.232.2 -- [17/May/2003:03:43:40 -0500] "GET /welcome.html HTTP/1.0" 200 5253

```

Fig. 1. Sample entries from a Web server access log.

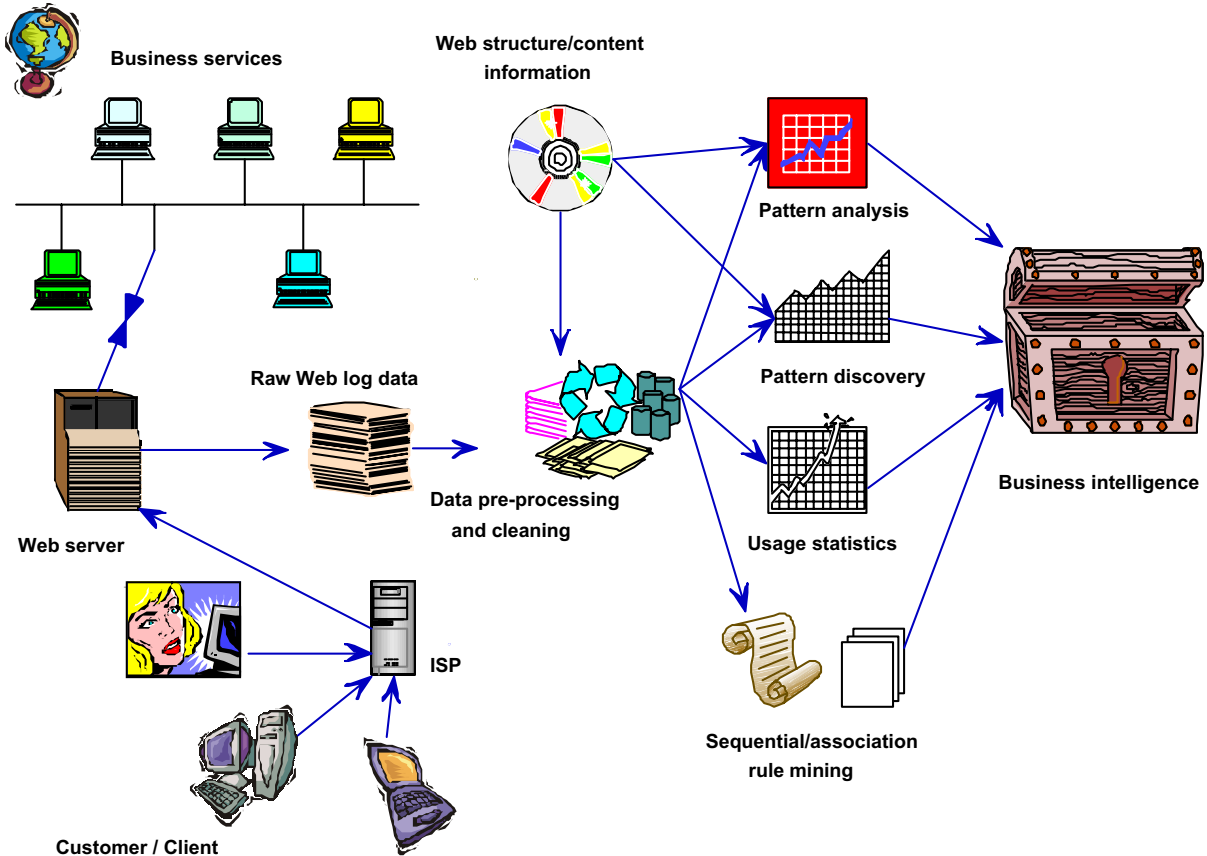


Fig. 2. Web usage mining framework.

statistical data available from the normal Web log data files or even the information provided by Web trackers could only provide the information explicitly because of the nature and limitations of the methodology itself. Generally, one could say that the analysis relies on three general sets of information given a current focus of attention: (1) past usage patterns; (2) degree of shared content; and (3) inter-memory associative link structures. After browsing through some of the features of the best trackers available it is easy to conclude that rather than generating statistical data and texts they really do not help to find much meaningful information.

For small web servers, the usage statistics provided by conventional Web site trackers may be adequate to

analyze the usage pattern and trends. However as the size and complexity of the data increases, the statistics provided by existing Web log file analysis tools may prove inadequate and more intelligent knowledge mining techniques will be necessary (Piramuthu, 2003; Roussinov & Zhao, 2003; Yang & Zhang, 2003; Abraham, 2003; Zhang & Dong, 2003).

A generic Web usage mining framework is depicted in Fig. 2. In the case of Web mining, data could be collected at the server level, client level, proxy level or some consolidated data. These data could differ in terms of content and the way it is collected etc. The usage data collected at different sources represent the navigation patterns of different segments of the overall Web traffic, ranging from

single user, single site browsing behaviour to multi-user, multi-site access patterns. Web server log does not accurately contain sufficient information for inferring the behaviour at the client side as they relate to the pages served by the Web server. Pre-processed and cleaned data could be used for pattern discovery, pattern analysis, Web usage statistics and generating association/ sequential rules (Kitsuregawa *et al.*, 2002; Massegia *et al.*, 1999; Pal *et al.*, 2002). Much work has been performed on extracting various pattern information from Web logs and the application of the discovered knowledge range from improving the design and structure of a Web site to enabling business organizations to function more efficiently (Paliouras *et al.*, 2000; Pazzani & Billsus, 1997; Perkowski & Etzioni, 1998; Pirolli *et al.*, 1996; Spiliopoulou & Faulstich, 1999).

Jespersen *et al.* (2002) proposed an hybrid approach for analyzing the visitor click sequences. A combination

of hypertext probabilistic grammar and click fact table approach is used to mine Web logs which could be also used for general sequence mining tasks. Mobasher *et al.* (1999) proposed the Web personalization system which consists of offline tasks related to the mining of usage data and online process of automatic Web page customization based on the knowledge discovered. LOGSOM proposed by Smith *et al.* (2003), utilizes a self-organizing map to organize web pages into a two-dimensional map based solely on the users' navigation behavior, rather than the content of the web pages. LumberJack proposed by Chi *et al.* (2002) builds up user profiles by combining both user session clustering and traditional statistical traffic analysis using K-means algorithm. Joshi *et al.* (1999) used relational online analytical processing approach for creating a Web log warehouse using access logs and mined logs (association rules and clusters). A

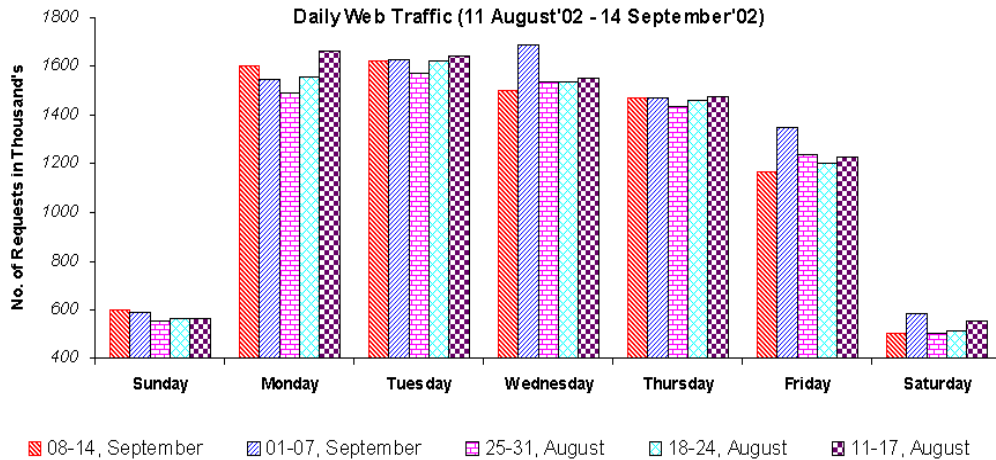


Fig. 3. University's daily Web traffic pattern for 5 weeks.

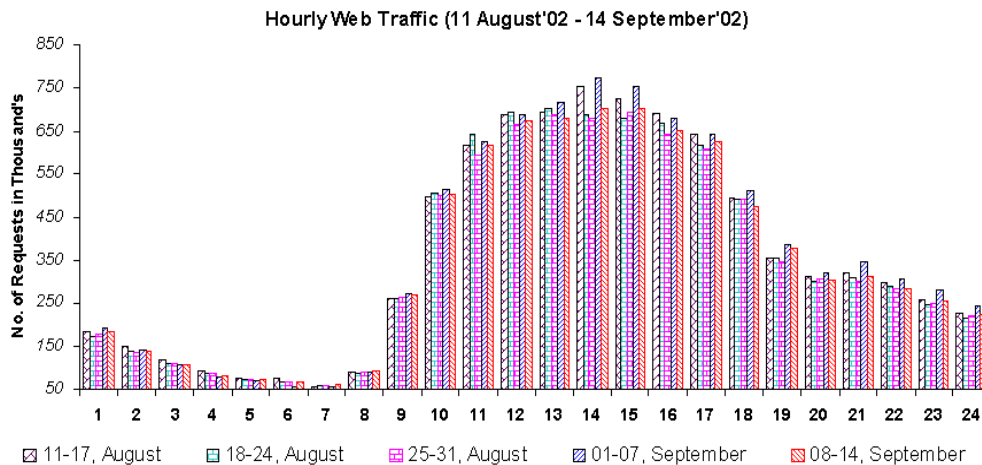


Fig. 4. Average hourly Web traffic patterns for 5 weeks.

comprehensive overview of Web usage mining research is found in (Cooley, 2000; Kosala & Blockeel, 2000; Srivastava *et al.*, 2000).

To demonstrate the efficiency of the proposed frameworks, Web access log data at the Monash University's Web site (Monash, 2003) were used for experimentations. The University's central web server receives over 7 million hits in a week and therefore it is a real challenge to find and extract hidden usage pattern information. To illustrate the University's Web usage patterns, average daily and hourly access patterns for 5 weeks (11 August '02–14 September '02) are shown in Figs. 3 and 4 respectively. The average daily and hourly patterns nevertheless tend to follow a similar trend (as evident from the figures) the differences tend to increase during high traffic days (Monday–Friday) and during the peak hours (11:00–17:00 Hrs). Due to the enormous traffic volume and chaotic access behavior, the prediction of the user access patterns becomes more difficult and complex.

Self organizing maps and fuzzy c-means algorithm could be used to segregate the user access records and computational intelligence paradigms to analyze the user access trends. Abraham (2003) and Wang *et al.* (2002) have clearly shown the importance of the clustering algorithm to analyze the user access trends.

In the subsequent section the *i-Miner* framework is presented and some theoretical concepts of clustering algorithms and computational intelligence paradigms are discussed. Experiment results are provided in Sec. 3 and some conclusions are provided towards the end.

## 2. Mining Framework Using Intelligent Miner (i-Miner)

The *i-Miner* hybrid framework optimizes a fuzzy clustering algorithm using an evolutionary algorithm and a Takagi–Sugeno fuzzy inference system using a combination of evolutionary algorithm and neural network learning. The raw data from the log files are cleaned and pre-processed and a fuzzy C means algorithm is used to identify the number of clusters. The architecture is illustrated in Fig. 5. The developed clusters of data are fed to a Takagi–Sugeno fuzzy inference system to analyze the trend patterns. The *if-then* rule structures are learned using an iterative learning procedure (Cordón *et al.*, 2001) by an evolutionary algorithm and the rule parameters are fine-tuned using a backpropagation algorithm.

The hierarchical distribution of the *i-Miner* is depicted in Fig. 6. The arrow direction depicts the speed of the evolutionary search. The optimization of

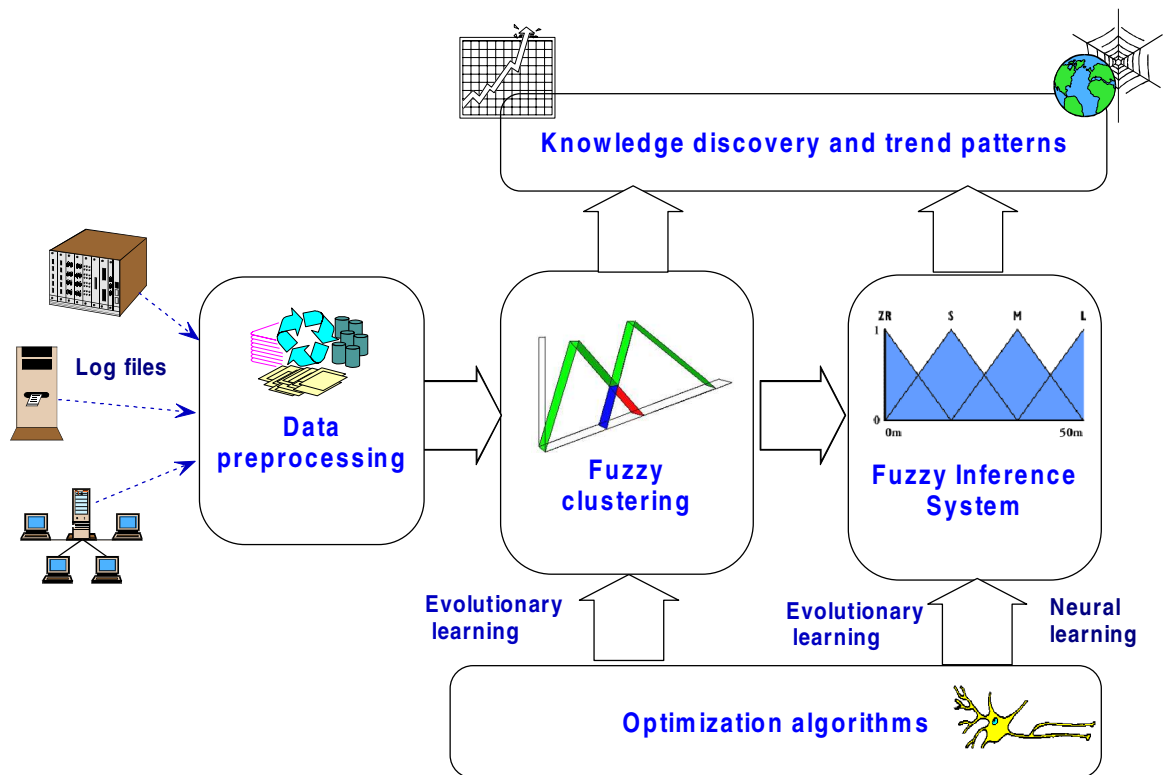


Fig. 5. *i-Miner* framework.

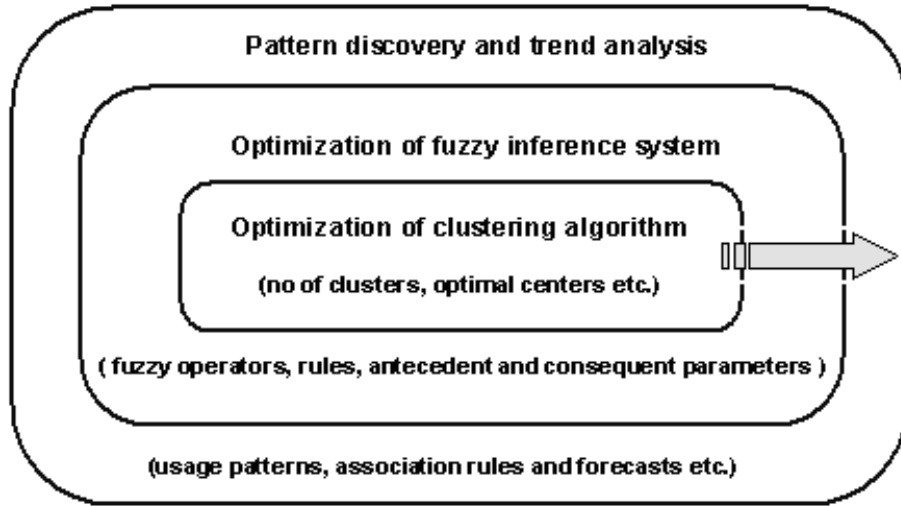


Fig. 6. Hierarchical architecture of *i-Miner*.

clustering algorithm progresses at a faster time scale in an environment decided by the inference method and the problem environment. More technical details are provided in the subsequent sections.

## 2.1. Clustering algorithms

### 2.1.1. Fuzzy clustering algorithm

One of the widely used clustering methods is the fuzzy *c*-means (FCM) algorithm developed by Bezdek (1981). FCM partitions a collection of  $n$  vectors  $x_i$ ,  $i = 1, 2, \dots, n$  into  $c$  fuzzy groups and finds a cluster center in each group such that a cost function of dissimilarity measure is minimized. To accommodate the introduction of fuzzy partitioning, the membership matrix  $U$  is allowed to have elements with values between 0 and 1. The FCM objective function takes the form

$$J(U, c_1, \dots, c_c) = \sum_{i=1}^c J_i = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d_{ij}^2 \quad (1)$$

where  $u_{ij}$  is a numerical value between  $[0, 1]$ ;  $c_i$  is the cluster center of fuzzy group  $i$ ;  $d_{ij} = \|c_i - x_j\|$  is the Euclidian distance between  $i$ th cluster center and  $j$ th data point; and  $m$  is called the exponential weight which influences the degree of fuzziness of the membership (partition) matrix.

### 2.1.2. Optimization of fuzzy clustering algorithm

Usually a number of cluster centers are randomly initialized and the FCM algorithm provides an iterative approach to approximate the minimum of the objective function starting from a given position and leads to any

of its local minima. No guarantee ensures that FCM converges to an optimum solution (can be trapped by local extrema in the process of optimizing the clustering criterion). The performance is very sensitive to initialization of the cluster centers. An evolutionary algorithm is used to decide the optimal number of clusters and their cluster centers. The algorithm is initialized by constraining the initial values to be within the space defined by the vectors to be clustered. A very similar approach is given in (Hall *et al.*, 2001). In the *i-Miner* approach, the fuzzy clustering algorithm is optimized jointly with the trend analysis algorithm (fuzzy inference system) in a single global search.

### 2.1.3. Self organizing map (SOM)

A self organizing map was used to cluster the user access records. The SOM is an algorithm used to visualize and interpret large high-dimensional data sets. The map consists of a regular grid of processing units, "neurons". A model of some multidimensional observation, eventually a vector consisting of features, is associated with each unit. The map attempts to represent all the available observations with optimal accuracy using a restricted set of models. At the same time the models become ordered on the grid so that similar models are close to each other and dissimilar models far from each other. Fitting of the model vectors is usually carried out by a sequential regression process, where  $t = 1, 2, \dots$  is the step index: For each sample  $x(t)$ , first the winner index  $c$  (best match) is identified by the condition

$$\forall_i, \|x(t) - m_c(t)\| \leq \|x(t) - m_i(t)\|. \quad (2)$$

After that, all model vectors or a subset of them that belong to nodes centered around node  $c = c(x)$  are updated

as

$$m_i(t+1) = m_i(t) + h_{c(x),i}(x(t) - m_i(t)). \quad (3)$$

Here  $h_{c(x),i}$  is the neighborhood function, a decreasing function of the distance between the  $i$ th and  $c$ th nodes on the map grid. This regression is usually reiterated over the available samples.

## 2.2. Computational intelligence (CI) based algorithms for trend analysis

CI substitutes intensive computation for insight into how complicated systems work. Artificial neural networks, fuzzy inference systems, probabilistic computing, evolutionary computation etc are some of the main components of CI. CI provides an excellent framework unifying them and even by incorporating other revolutionary methods.

### 2.2.1. Artificial neural network (ANN)

ANNs were designed to mimic the characteristics of the biological neurons in the human brain and nervous system. Learning typically occurs by example through training, where the training algorithm iteratively adjusts the connection weights (synapses). Backpropagation (BP) is one of the most famous training algorithms for multilayer perceptrons. BP is a gradient descent technique to minimize the error  $E$  for a particular training pattern. For adjusting the weight ( $w_{ij}$ ) from the  $i^{th}$  input unit to the  $j$ th output, in the batched mode variant the descent is based on the gradient  $\nabla E(\frac{\delta E}{\delta w_{ij}})$  for the total training set

$$\Delta w_{ij}(n) = -\varepsilon^* \frac{\delta E}{\delta w_{ij}} + \alpha^* \Delta w_{ij}(n-1). \quad (4)$$

The gradient gives the direction of error  $E$ . The parameters  $\varepsilon$  and  $\alpha$  are the learning rate and momentum respectively.

### 2.2.2. Linear genetic programming (LGP)

Linear genetic programming proposed by Banzhaf *et al.* (1998) is a variant of the GP technique that acts on linear genomes. Its main characteristics in comparison to tree-based GP lies in that the evolvable units are not the expressions of a functional programming language (like LISP), but the programs of an imperative language (like C/C++). An alternate approach is to evolve a computer program at the machine code level, using lower level representations for the individuals. This can tremendously hasten up the evolution process as, no matter how an individual is initially represented, finally it always has to be represented as a piece of machine code, as fitness evaluation requires physical execution of the individuals.

The basic unit of evolution here is a native machine code instruction that runs on the floating-point processor unit (FPU). Since different instructions may have different sizes, here instructions are clubbed up together to form instruction blocks of 32 bits each. The instruction blocks hold one or more native machine code instructions, depending on the sizes of the instructions. A crossover point can occur only between instructions and is prohibited from occurring within an instruction. However the mutation operation does not have any such restriction.

### 2.2.3. Fuzzy inference systems (FIS)

Fuzzy logic provides a framework to model uncertainty, human way of thinking, reasoning and the perception process. Fuzzy *if-then* rules and fuzzy reasoning are the backbone of fuzzy inference systems, which are the most important modelling tools based on fuzzy set theory. A Takagi Sugeno fuzzy inference scheme was made used of in which the conclusion of a fuzzy rule is constituted by a weighted linear combination of the crisp inputs rather than a fuzzy set (Sugeno, 1985). The Adaptive Network Based Fuzzy Inference System (ANFIS) proposed by Jang (1992) was used in the simulation, which implements a Takagi Sugeno fuzzy inference system learned using neural network learning.

### 2.2.4. Optimization of fuzzy inference system

The EvoNF framework proposed by Abraham (2002) was used to optimize the fuzzy inference method, which is an integrated computational framework to optimize fuzzy inference system using neural network learning and evolutionary computation. Solving multi-objective scientific and engineering problems is, generally, a very difficult goal. In these particular optimization problems, the objectives often conflict across a high-dimension problem space and may also require extensive computational resources. The hierarchical evolutionary search framework could adapt the membership functions (shape and quantity), rule base (architecture), fuzzy inference mechanism (T-norm and T-conorm operators) and the learning parameters of neural network learning algorithm. In addition to the evolutionary learning (global search) neural network learning could be considered as a local search technique to optimize the parameters of the rule antecedent/consequent parameters and the parameterized fuzzy operators. The hierarchical search could be formulated as follows:

For every fuzzy inference system, there exist a global search of neural network learning algorithm parameters, parameters of the fuzzy operators, *if-then* rules and

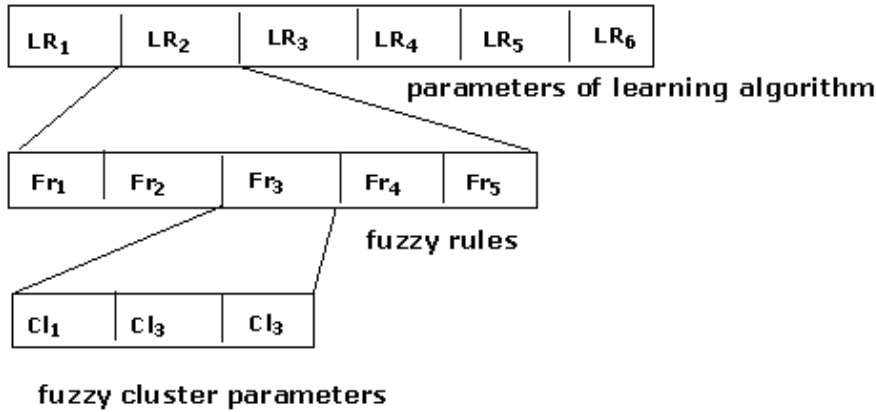


Fig. 7. Chromosome structure of the *i-Miner*.

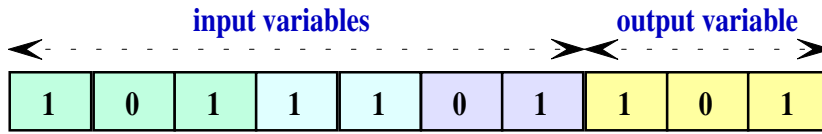


Fig. 8. Chromosome representing an individual fuzzy rule (3 input variables and 1 output variable).

membership functions in an environment decided by the problem. The evolution of the fuzzy inference system will evolve at the slowest time scale while the evolution of the quantity and type of membership functions will evolve at the fastest rate. The function of the other layers could be derived similarly. Hierarchy of the different adaptation layers (procedures) will rely on the prior knowledge (this will also help to reduce the search space). For example, if one knows certain fuzzy operators will work well for a problem then it is better to implement the search of fuzzy operators at a higher level. For fine-tuning the fuzzy inference system all the node functions are to be parameterized. For example, the Schweizer and Sklar's T-norm operator can be expressed as:

$$T(a, b, p) = [\max\{0, (a^{-p} + b^{-p} - 1)\}]^{-\frac{1}{p}}. \quad (5)$$

It is observed that

$$\begin{aligned} \lim_{p \rightarrow 0} T(a, b, p) &= ab \\ \lim_{p \rightarrow \infty} T(a, b, p) &= \min\{a, b\} \end{aligned} \quad (6)$$

which correspond to two of the most frequently used T-norms in combining the membership values on the premise part of a fuzzy *if-then* rule.

### 2.2.5. *i-Miner*: Chromosome modeling and representation

Hierarchical evolutionary search process has to be represented in a chromosome for successful modeling of the

*i-Miner* framework. A typical chromosome of the *i-Miner* would appear as shown in Fig. 7 and the detailed modeling process is as follows.

**Layer 1.** The optimal number of clusters and initial cluster centers is represented in this layer.

**Layer 2.** This layer is responsible for the optimization of the rule base. This includes deciding the total number of rules, representation of the antecedent and consequent parts. The number of rules grows rapidly with an increasing number of variables and fuzzy sets. The grid-partitioning algorithm was used to generate the initial set of rules. An iterative learning method is then adopted to optimize the rules. The existing rules are mutated and new rules are introduced. The fitness of a rule is given by its contribution (strength) to the actual output. To represent a single rule a position dependent code with as many elements as the number of variables of the system is used. Each element is a binary string with a bit per fuzzy set in the fuzzy partition of the variable, meaning the absence or presence of the corresponding linguistic label in the rule. For a three input and one output variable, with fuzzy partitions composed of 3, 2, 2 fuzzy sets for input variables and 3 fuzzy sets for output variable, the fuzzy rule will have a representation as shown in Fig. 8.

**Layer 3.** This layer is responsible for the selection of optimal learning parameters. Performance of the gradient descent algorithm directly depends on the learning rate according to the error surface. The optimal learning

parameters decided by this layer will be used to tune the parameterized rule antecedents/consequents and the fuzzy operators.

The rule antecedent/consequent parameters and the fuzzy operators are fine tuned using a gradient descent algorithm to minimize the output error

$$E = \sum_{k=1}^N (d_k - x_k)^2 \quad (7)$$

where  $d_k$  is the  $k$ th component of the  $r$ th desired output vector and  $x_k$  is the  $k$ th component of the actual output vector by presenting the  $r$ th input vector to the network. All the gradients of the parameters to be optimized, namely the consequent parameters  $\frac{\partial E}{\partial P_n}$  for all rules  $R_n$  and the premise parameters  $\frac{\partial E}{\partial \sigma_i}$  and  $\frac{\partial E}{\partial c_i}$  for all fuzzy sets  $F_i$  ( $\sigma$  and  $c$  represents the MF width and center of a Gaussian MF).

Once the three layers are represented in a chromosome  $C$ , and then the learning procedure could be initiated as follows:

- a. Generate an initial population of  $N$  numbers of  $C$  chromosomes. Evaluate the fitness of each chromosome depending on the output error.
- b. Depending on the fitness and using suitable selection methods reproduce a number of children for each individual in the current generation.
- c. Apply genetic operators to each child individual generated above and obtain the next generation.
- d. Check whether the current model has achieved the required error rate or the specified number of generations has been reached. Go to Step b.
- e. End

### 3. Experiment Setup-Training and Performance Evaluation

In this research, the statistical/text data generated by the log file analyzer from 01 January 2002 to 07 July 2002 were used. Selecting useful data is an important task

in the data pre-processing block. After some preliminary analysis, the statistical data comprising of domain byte requests, hourly page requests and daily page requests were selected as focus of the cluster models for finding Web users' usage patterns. It is also important to remove irrelevant and noisy data in order to build a precise model. The additional input "index number" was also included to distinguish the time sequence of the data. The most recently accessed data were indexed higher while the least recently accessed data were placed at the bottom. Besides the inputs "volume of requests" and "volume of pages (bytes)" and "index number", the "cluster information" provided by the clustering algorithm was also used as an additional input variable. The data was re-indexed based on the cluster information. The task is to predict (few time steps ahead) the Web traffic volume on a hourly and daily basis. The data from 17 February 2002 to 30 June 2002 for training and the data from 01 July 2002 to 06 July 2002 were used for testing and validation purposes.

The initial populations were randomly created based on the parameters shown in Table 1. A special mutation operator was used, which decreases the mutation rate as the algorithm greedily proceeds in the search space. If the allelic value  $x_i$  of the  $i$ th gene ranges over the domain  $a_i$  and  $b_i$  the mutated gene  $x'_i$  is drawn randomly uniformly from the interval  $[a_i, b_i]$ .

$$x'_i = \begin{cases} x_i + \Delta(t, b_i - x_i), & \text{if } \omega = 0 \\ x_i + \Delta(t, x_i - a_i), & \text{if } \omega = 1 \end{cases} \quad (8)$$

where  $\omega$  represents an unbiased coin flip  $p(\omega = 0) = p(\omega = 1) = 0.5$ , and

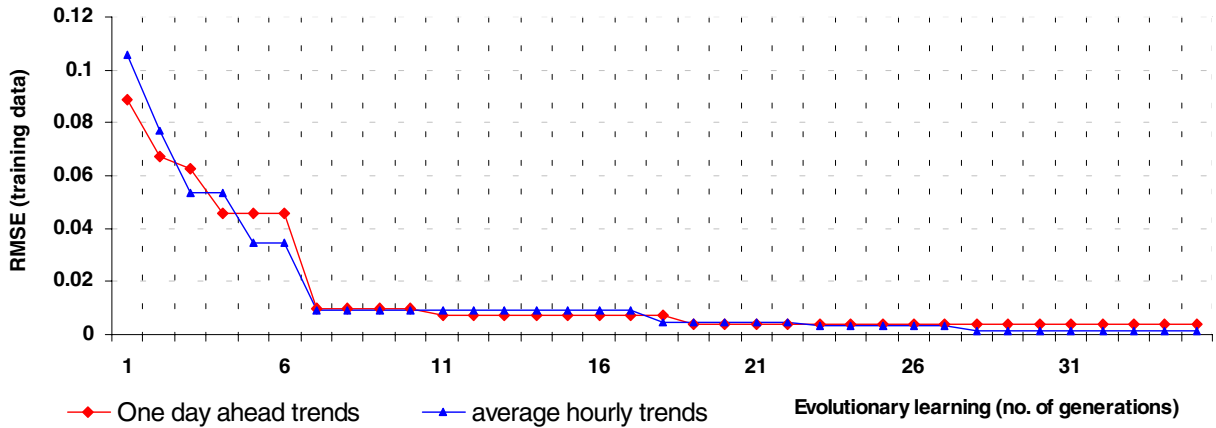
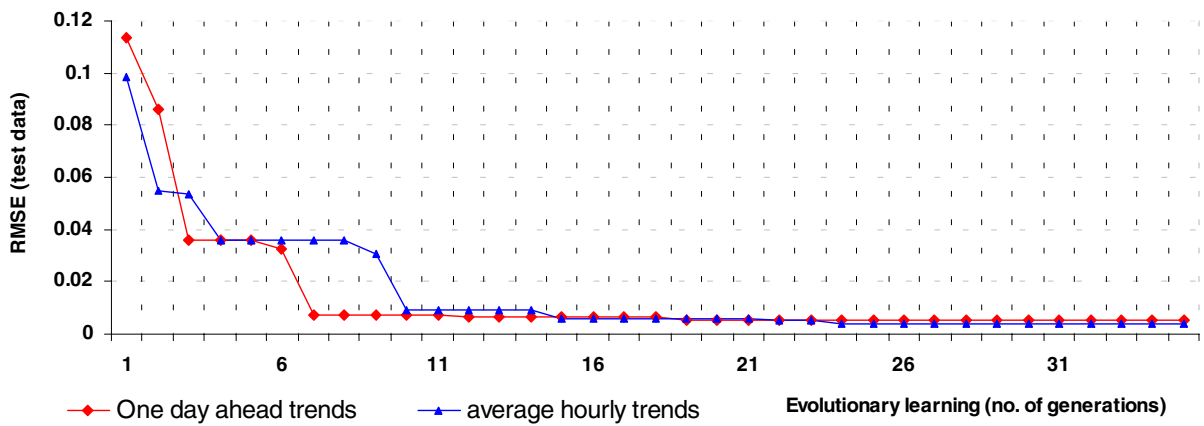
$$\Delta(t, x) = x \left( 1 - \gamma^{(1 - \frac{t}{t_{\max}})^b} \right) \quad (9)$$

defines the mutation step, where  $\gamma$  is the random number from the interval  $[0, 1]$  and  $t$  is the current generation and  $t_{\max}$  is the maximum number of generations. The function  $\Delta$  computes a value in the range  $[0, x]$  such that the probability of returning a number close to zero increases as the algorithm proceeds with the search. The parameter  $b$  determines the impact of time on the probability

Table 1. Parameter settings of *i-Miner*.

Population size	30
Maximum no of generations	35
Fuzzy inference system	Takagi Sugeno
Rule antecedent membership functions	3 membership functions per input variable (parameterized Gaussian)
Rule consequent parameters	linear parameters
Gradient descent learning	10 epochs
Ranked based selection	0.50
Elitism	5%
Starting mutation rate	0.50



***i*-Miner training performance**Fig. 9. Meta-learning performance (training) of *i*-Miner.***i*-Miner test performance**Fig. 10. Meta-learning performance (testing) of *i*-Miner.

distribution  $\Delta$  over  $[0, x]$ . Large values of  $b$  decrease the likelihood of large mutations in a small number of generations. The parameters mentioned in Table 1 were decided after a few trial and error approaches. Experiments were repeated 3 times and the average performance measures are reported. Figures 9 and 10 illustrate the meta-learning approach combining evolutionary learning and gradient descent technique during the 35 generations.

Table 2 summarizes the performance of the developed *i*-Miner for training and test data. Performance is compared with the previous results reported by Wang *et al.* (2002) wherein the trends were analyzed using a Takagi-Sugeno Fuzzy Inference System (ANFIS), Artificial Neural Network (ANN) and Linear Genetic Programming (LGP). The Correlation Coefficient (CC) for the test

data set is also given in Table 2. The 35 generations of meta-learning approach created 62 *if-then* Takagi-Sugeno type fuzzy rules (daily traffic trends) and 64 rules (hourly traffic trends) compared to the 81 rules reported by Wang *et al.* (2002).

Figures 11 and 12 illustrate the actual and predicted trends for the test data set. A trend line is also plotted using a least squares fit (6th order polynomial). As evident, the trend prediction by *i*-Miner outperformed other conventional function approximation techniques.

FCM approach created 7 data clusters (Figs. 13 and 14) for hourly traffic according to the input features compared to 9 data clusters (Figs. 15 and 16) for the daily requests. The dark dots represent the cluster centers formed by the evolutionary fuzzy clustering algorithm. Several

Table 2. Performance of the different paradigms.

Method	Period						
	Daily (1 day ahead)			Daily (1 day ahead)			
	RMSE			RMSE			
Train	Test	CC	Train	Test	CC		
i-Miner	0.0044	0.0053	0.9967	0.0012	0.0041	0.9981	
TKFIS	0.0176	0.0402	0.9953	0.0433	0.0433	0.9841	
ANN	0.0345	0.0481	0.9292	0.0546	0.0639	0.9493	
LGP	0.0543	0.0749	0.9315	0.0654	0.0516	0.9446	

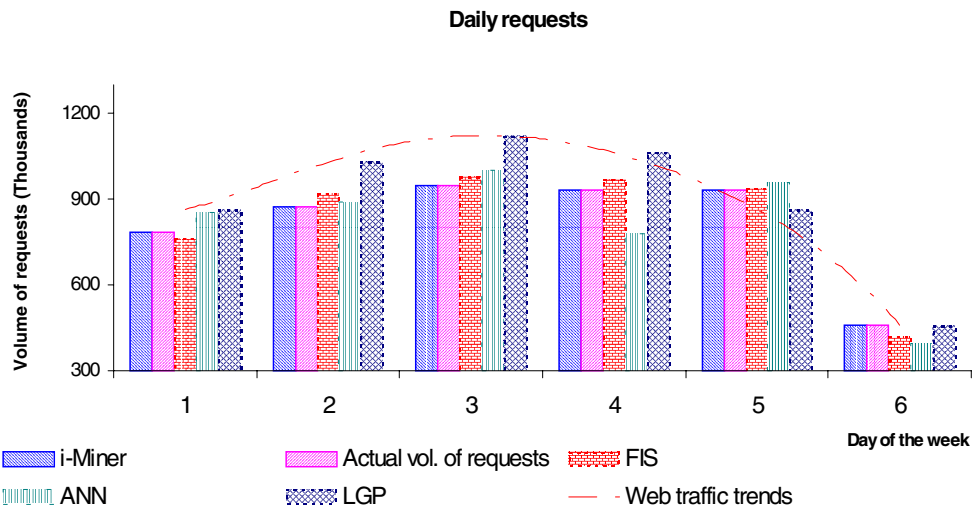


Fig. 11. Test results of the daily trends for 6 days.

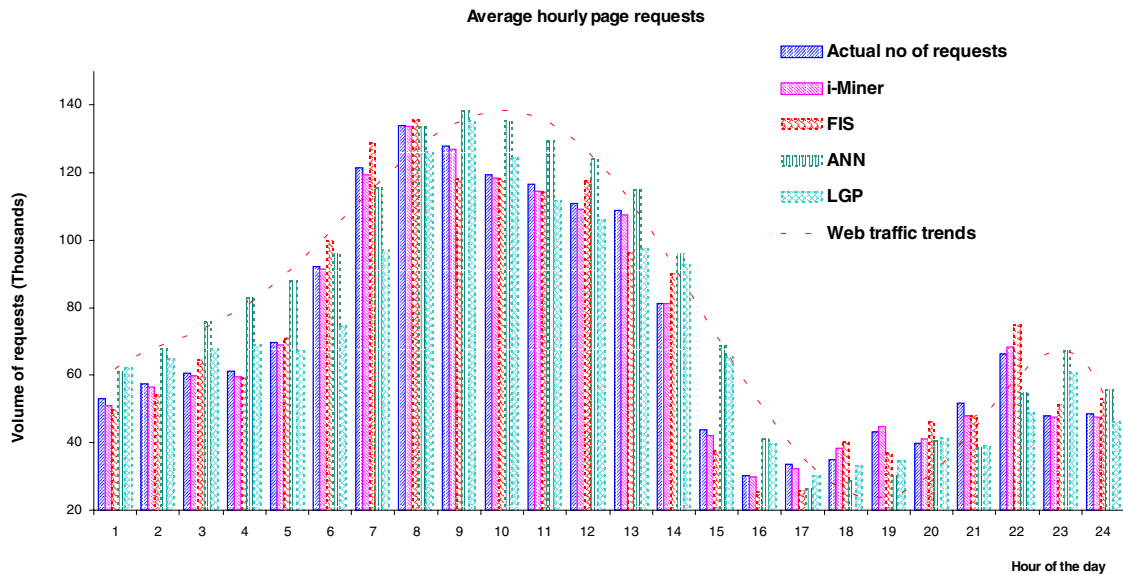


Fig. 12. Test results of the average hourly trends for 6 days.

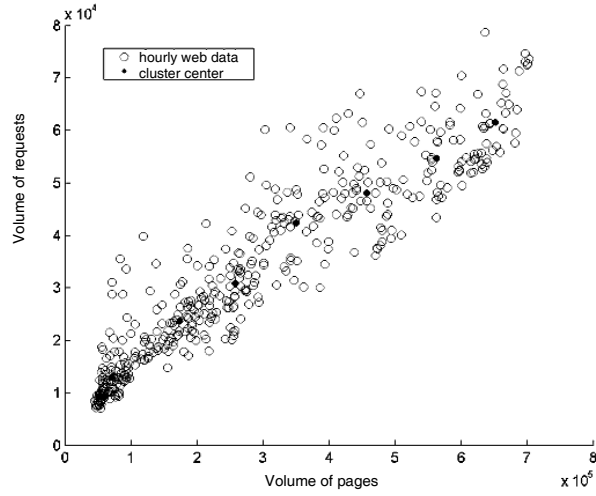


Fig. 13. FCM clustering — hourly volume of requests and pages.

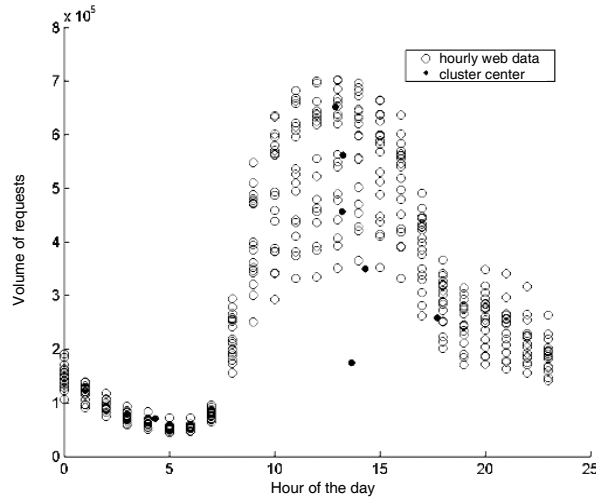


Fig. 14. FCM clustering — hour of the day and volume of requests.

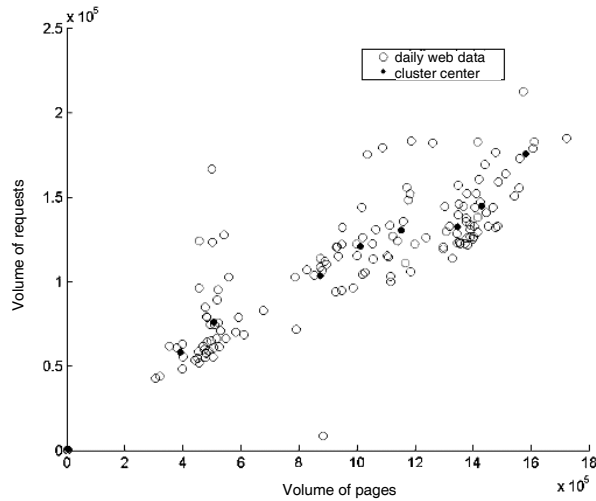


Fig. 15. FCM clustering — daily volume of requests and volume of pages.

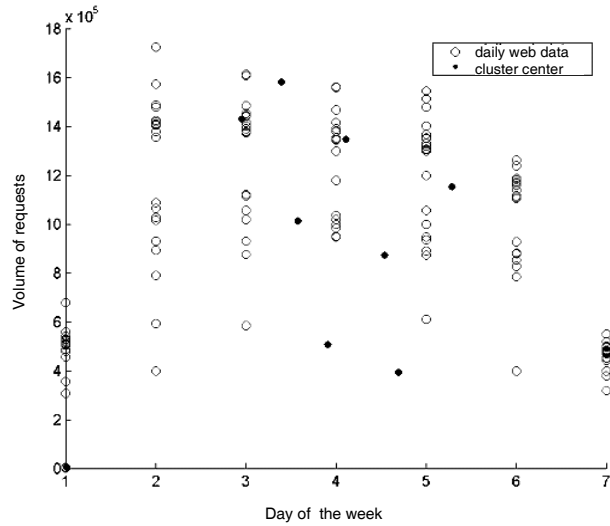


Fig. 16. FCM clustering — day of the week and volume of requests.

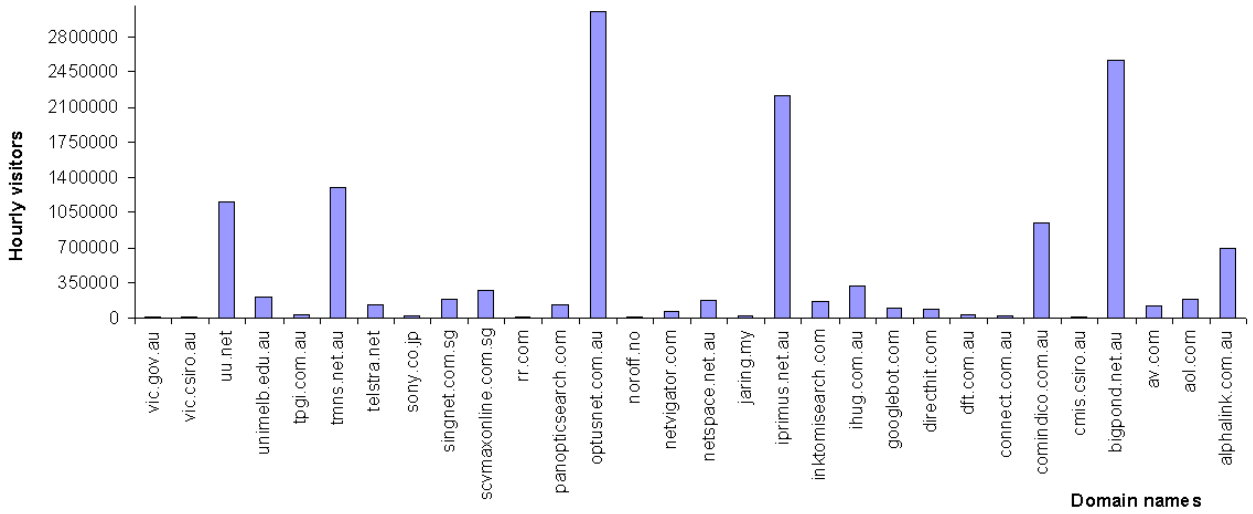


Fig. 17. Hourly visitor information according to the domain names from an FCM cluster.

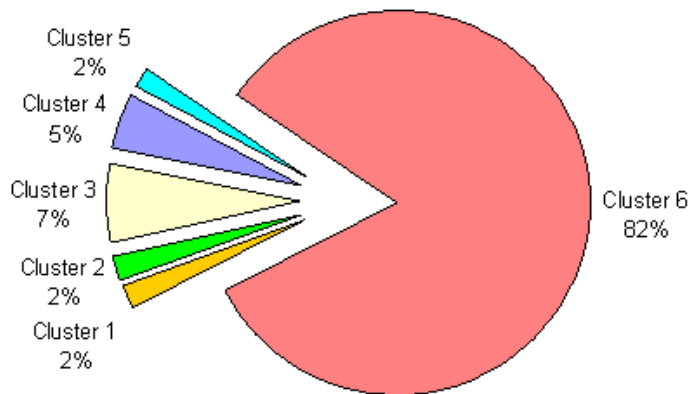


Fig. 18. Fuzzy clustering of visitors based on the day of access (weekday/weekend).

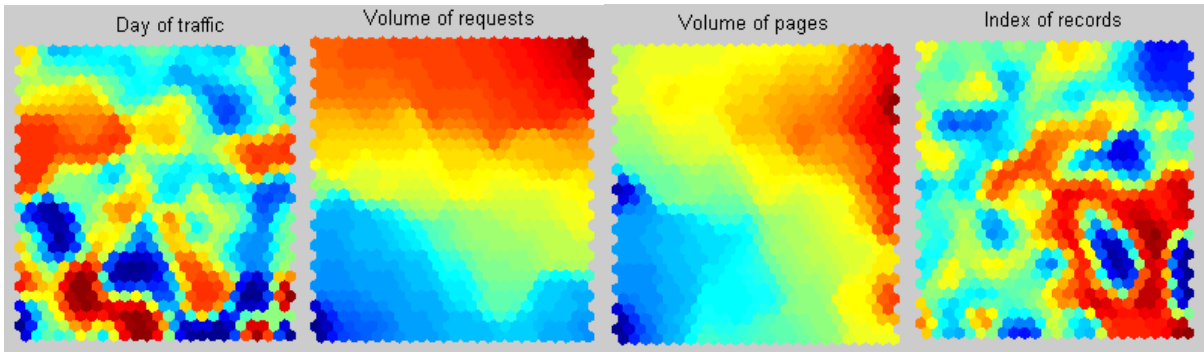


Fig. 19. Developed SOM clusters showing the different input variables (title of each map) for daily Web traffic.

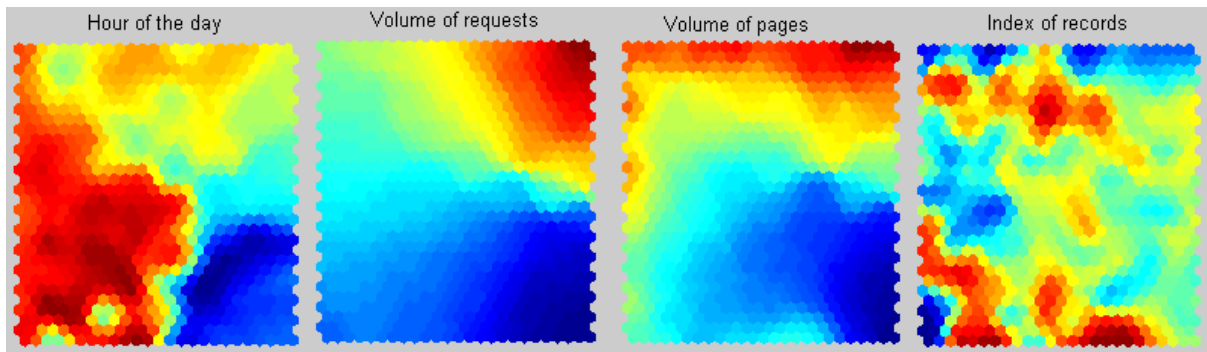


Fig. 20. Developed SOM clusters showing the different input variables for hourly Web traffic.

meaningful information could be obtained from the clustered data. Depending on the volume of requests and transfer of bytes, data clusters were formulated. Clusters based on hourly data show the visitor information at certain hour of the day. Figure 17 depicts the volume of visitors according to domain names from an FCM cluster. Figure 18 illustrates the volume of visitors in each FCM cluster according to the day of access. Some clusters accounted for the visitors according to the peak hour and certain weekday traffic and so on. Clusters developed using Self-organizing Map (SOM) for daily and hourly traffic is depicted in Figures 19 and 20 respectively. The previous study using Self-organizing Map (SOM) created 7 data clusters (daily traffic volume) and 4 data clusters (hourly traffic volume) respectively. The main advantage of SOMs comes from the easy visualization and interpretation of clusters formed by the map. As evident (Figs. 13–16), FCM approach resulted in the formation of additional data clusters.

#### 4. Conclusions

Recently Web usage mining has been gaining a lot of attention because of its potential commercial benefits. The proposed *i-Miner* framework seems to work very well for

the problem considered. The empirical results also reveal the importance of using soft computing paradigms for mining useful information. Several useful information could be discovered from the clustered data. FCM clustering resulted in more clusters compared to SOM approach. Perhaps more clusters were required to improve the accuracy of the trend analysis. The main advantage of SOMs comes from the easy visualization and interpretation of clusters formed by the map. The knowledge discovered from the developed FCM clusters and SOM could be a good comparison study and is left as a future research topic.

As illustrated in Table 2, *i-Miner* framework gave the overall best results with the lowest RMSE on test error and the highest correlation coefficient. It is interesting to note that the three considered soft computing paradigms could easily pickup the daily and hourly Web-access trend patterns. When compared to LGP, the developed neural network performed better (in terms of RMSE) for daily trends but for hourly trends LGP gave better results. An important disadvantage of *i-Miner* is the computational complexity of the algorithm. When optimal performance is required (in terms of accuracy and smaller structure) such algorithms might prove to be useful as evident from the empirical results.

So far most analysis of Web data have involved basic traffic reports that do not provide much pattern and trend analysis. The proposed *i-Miner* framework is an initial effort to patch up some of the weaknesses of the conventional web log file analyzers. By linking the Web logs with cookies and forms, it is further possible to analyze the visitor behavior and profiles which could help an e-commerce site to address several business questions. Our future research will be oriented in this direction by incorporating more data mining paradigms to improve knowledge discovery and association rules from the clustered data.

## Acknowledgements

Author is grateful to the technical reviewers for the comments, which improved the clarity and presentaton of the paper. Initial part of the research was done during the author's stay at Monash University, Australia. Author wishes to thank Ms. Xiaozhe Wang for all the discussions and contributions during the initial stages of this research.

## References

- Abraham, A (2001). Neuro-fuzzy systems: State-of-the-art modeling techniques, connectionist models of neurons, learning processes, and artificial intelligence. In *Lecture Notes in Computer Science 2084*, J Mira and A Prieto (eds.), Germany, Spain: Springer-Verlag, pp. 269–276.
- (2002). EvoNF: A framework for optimization of fuzzy inference systems using neural network learning and evolutionary computation. In *Proc. of 17th IEEE International Symposium on Intelligent Control*, pp. 327–332, IEEE Press.
- (2003). *i-Miner*: A web usage mining framework using hierarchical intelligent systems. *The IEEE International Conference on Fuzzy Systems FUZZ-IEEE'03*, pp. 1129–1134, IEEE Press.
- Aggarwal, C, JL Wolf and PS Yu (1999). Caching on the World Wide Web. *IEEE Transaction on Knowledge and Data Engineering*, Vol. 11, No. 1, pp. 94–107.
- Agrawal, R and R Srikant (1994). Fast algorithms for mining association rules. In *Proc. of the 20th International Conference on Very Large Databases*, M Kaufmann, JB Bocca, M Jarke and C Zaniolo (eds.), pp. 487–499.
- Analog Website Tracker (2003). (<http://www.analog.cx/>) [3 October 2003].
- Banzhaf, W, P Nordin, ER Keller and FD Francone (1998). *Genetic Programming: An Introduction on The Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann Publishers, Inc.
- Bezdek, JC (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press.
- Chakrabarti, S (2003). *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers.
- Chang, G, MJ Healey, JAM McHugh and JTL Wang (2001). *Web Mining, Mining the World Wide Web*. Chapter 7, pp. 93–104. Kluwer Academic Publishers.
- Chen, PM and FC Kuo (2000). An information retrieval system based on an user profile. *The Journal of Systems and Software*, 54, 3–8.
- Cheung, DW, B Kao and J Lee (1997). Discovering user access patterns on the World Wide Web. *Knowledge-Based Systems*, 10, 463–470.
- Chi, EH, A Rosien and J Heer (2002). LumberJack: Intelligent discovery and analysis of web user traffic composition. In *Proc. of ACM-SIGKDD Workshop on Web Mining for Usage Patterns and User Profiles*. Canada: ACM Press.
- Cho, YH, JK Kim and SH Kim (2003). A personalized recommender system based on web usage mining and decision tree induction, *Expert Systems with Applications*, 23(3), 329–342.
- ClickTracks, (<http://www.clicktracks.com>) [3 October 2003].
- Coenen, F, G Swinnen, K Vanhoof and G Wets (2000). A framework for self adaptive websites: Tactical versus strategic changes. In *Proc. of the Workshop on Web-mining for E-commerce: Challenges and Opportunities (KDD'00)*, pp. 75–80.
- Cooley, R (2000). Web usage mining: Discovery and application of interesting patterns from web data. PhD Thesis, Department of Computer Science, University of Minnesota.
- Cordón, O, F Herrera, F Hoffmann and L Magdalena (2001). *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. Singapore: World Scientific Publishing Company.
- Hall, LO, IB Ozyurt and JC Bezdek (1999). Clustering with a genetically optimized approach. *IEEE Transactions on Evolutionary Computation*, 3(2), 103–112.
- Hay, B, G Wets and K Vanhoof (2003). Segmentation of visiting patterns on web sites using a sequence alignment method. *Journal of Retailing and Consumer Services*, 10(3), pp. 145–153.
- Heer, J and EH Chi (2001). Identification of web user traffic composition using multi-modal clustering and information scent. In *Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining*, pp. 51–58.
- Heinrichs, JH and JS Lim (2003). Integrating web-based data mining tools with business models for knowledge management. *Decision Support Systems*, 35(1), pp. 103–112.
- Hitbox Central Web Traffic Analysis 92003). (<http://www.hitboxcentral.com/>) [3 October 2003].
- Jang, R (1992). Neuro-fuzzy modeling: Architectures, analyses and applications. PhD Thesis, University of California, Berkeley.

- Jespersen, SE, J Thorhauge and TB Pedersen (2002). A hybrid approach to web usage mining. In *Proc. of 4th International Conference Data Warehousing and Knowledge Discovery*, the (DaWaK'02), LNCS 2454, pp. 73–82. Germany: Springer Verlag.
- Joshi, KP, A Joshi, Y Yesha and R Krishnapuram (1999). Warehousing and mining web logs. In *Proc. of the 2nd ACM CIKM Workshop on Web Information and Data Management*, pp. 63–68.
- Kitsuregawa, M, M Toyoda and I Pramudiono (2002). Web community mining and web log mining: Commodity cluster based execution. In *Proc. of the Thirteenth Australasian Conference on Database Technologies*, Vol. 5, 3–10. ACM Press.
- Kosala, R and H Blockeel (2000). Web mining research: A survey. *ACM SIGKDD Explorations*, 2(1), 1–15.
- LogRover, (2003). (<http://www.logrover.com/>) [3 October 2003].
- Masseglia, F, P Poncelet and R Cicchetti (1999). An efficient algorithm for web usage mining. *Networking and Information Systems Journal (NIS)*, 2(5–6), 571–603.
- Mobasher, B, R Cooley and J Srivastava (1999). Creating adaptive web sites through usage-based clustering of URLs. In *Proc. of 1999 Workshop on Knowledge and Data Engineering Exchange*, pp. 19–25. USA.
- Monash University (2003). (<http://www.monash.edu.au>) [3 October 2003].
- Nanopoulos, A, D Katsaros and Y Manolopoulos (2002). Exploiting web log mining for web cache enhancement, WEBKDD 2001 — Mining web log data across all customers touch points. Third International Workshop, San Francisco, *Lecture Notes in Computer Science 2356*, pp. 68–87. Springer-Verlag.
- Pal, SK, V Talwar and P Mitra (2002). Web mining in soft computing framework: Relevance, state of the art and future directions. *IEEE Transactions on Neural Networks*, Vol. 13, No. 5, pp. 1163–1177.
- Paliouras, G, C Papatheodorou, V Karkaletsisi and CD Spyropoulos (2000). Clustering the users of large web sites into communities. In *Proc. of the 17th International Conference on Machine Learning (ICML'00)*, pp. 719–726. USA: Morgan Kaufmann.
- Pazzani, M and D Billsus (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27, 313–331.
- Perkowitz, M and O Etzioni (1998). Adaptive web sites: Automatically synthesizing web pages. In *Proc. of the 15th National Conference on Artificial Intelligence*, pp. 727–732.
- Piramuthu, S (2003). On learning to predict web traffic. *Decision Support Systems*, 35(2), 213–229.
- Pirolli, P, J Pitkow and R Rao (1996). Silk from a sow's ear: Extracting usable structures from the web. In *Proc. on Human Factors in Computing Systems (CHI'96)*. ACM Press.
- Roussinov, D and JL Zhao (2003). Automatic discovery of similarity relationships through web mining. *Decision Support Systems*, 35(1), 149–166.
- Runkler, TA and JC Bezdek (2003). Web mining with relational clustering. *International Journal of Approximate Reasoning*, 32(2–3), 217–236.
- Smith, KA and A Ng (2003). Web page clustering using a self-organizing map of user navigation patterns. *Decision Support Systems*, 35(2), 245–256.
- Spiliopoulou, M and LC Faulstich (1999). WUM: A web utilization miner. In *Proc. of EDBT Workshop on the Web and Data Bases (WebDB'98)*, pp. 109–115. Springer Verlag.
- Srivastava, J, R Cooley, M Deshpande and PN Tan (2000). Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2), 12–23.
- Sugeno, M (1985). *Industrial Applications of Fuzzy Control*. Elsevier Science Pub Co.
- Wang, X, A Abraham and KA Smith (2002). Soft computing paradigms for web access pattern analysis. In *Proc. of the 1st International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 631–635.
- Website Tracker (2003). (<http://www.websitetracker.com/>) [3 October 2003].
- WebSTAT Web Traffic Analyser (2003). (<http://www.webstat.com/>) [3 October 2003].
- Yang, Q and HH Zhang (2003). Web-log mining for predictive web caching. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 4, pp. 1050–1053.
- Zhang, D and Y Dong (2003). A novel web usage mining approach for search engines. *Computer Networks*, 39(3), 303–310.

---

**Ajith Abraham** is currently a faculty member of the computer science department at Oklahoma State University, USA. He received his PhD degree from Monash University, Australia. He obtained Master of Science degree in control and automation from the Nanyang Technological University (Singapore) and a first class (honours) degree in Electrical and Electronics Engineering from the Univer-

sity of Calicut. Before turning into a full time academic, he has been working with three multi-national companies on several industrial research and development projects for over eight years.

His primary research interests are in computational intelligence with a focus on hybridizing intelligent techniques involving artificial neural networks, fuzzy inference

systems, evolutionary computation and other heuristics. Application areas include several real world data mining applications like information security, Web intelligence, energy management, financial modeling (stock/forex market analysis), weather analysis, knowledge management, texture classification, fault monitoring, multi criteria decision-making, etc. He has over 100 research publications (published/ currently under review) in peer reviewed international conference proceedings and reputed journals of which three have won "best paper" awards. He is an associate editor of the International Journal of Systems Science, Taylor & Francis Group, London and the founding co-editor in chief of The International Journal of Hybrid Intelligent Systems (IJHIS). He has also reviewed papers for IEEE Intelligent Systems, IEEE Transactions on Neural Networks, IEEE Transactions on Fuzzy Systems, IEEE Transactions on Evolutionary Computation, IEEE Transactions on Systems, Man and Cybernetics, Fuzzy Sets and

Systems, IEEE Transactions on Power Engineering, Applied Soft Computing, International Journal of Neural, Parallel & Scientific Computations, International Journal of Control and Intelligent Systems, Bioinformatics, European Journal of Operational Research, ACM Computing Reviews, Informatica, etc.

He is the general chair/co-chair of the Intelligent Systems Design and Applications (ISDA) and Hybrid Intelligent Systems (HIS) series of international conferences that was started in 2001. During the last few years, he has also served the technical committee of several AI related International conferences and has also given a number of conference tutorials in the area of hybrid intelligent systems. He is a member of several professional societies like IEEE, IEEE (CS), ACM, IEE (UK), IEAust and also works closely with several academic working groups like EvoNet, EUSFLAT, WFSC, etc.