Application of Characteristics Based Incremental Clustering in Mining of Server Log Messages

[1]Garima Chouksey, [2]Prateek Gupta
[1,2]Dept. of CSE, SRIST Jabalpur, MP, India

Abstract
Information Extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most of the cases this activity concerns processing human language texts by means of natural Language Processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as information extraction.
It has been found that application of incremental clustering has not been done very much in message extraction in the field of processing of event logs of any machine. Since event logs are the data sets which grow by the time and require that the clusters must be updated in real time with the growth of the event log.
Since event logs are the huge set of datasets and grow rapidly as the system processing goes on, therefore, for extraction of data from it is variable and should grow by the time. In this work, it is being proposed to apply the incremental clustering to extract the data from the event log as per the characteristics provided by the users of the system.
Incremental Clustering requires initial clusters to be decided in advance i.e. they must pre exist for processing. If the initial clusters are to be fixed, then there are several ways it can be achieved. The algorithm being proposed is a dynamic and novice algorithm for deciding the initial clusters dynamically.

Keywords
Incremental Clustering, Data Mining, System Log, Clustering, Message Type Mining

I. Introduction

Message type or message cluster extraction is an important task in the analysis of server's system logs in computer networks. Defining these message types automatically facilitates the automatic analysis of server's system logs. When the message types that exist in a log file are represented explicitly, they can form the basis for carrying out other automatic application log analysis tasks.
Due to the difficulty of the problem, current approaches to IE focus on narrowly restricted domains. An example is the extraction from news wire reports of corporate mergers, such as denoted by the formal relation:
MergerBetween(company1, company2, date)
From an online news sentence such as:
"Yesterday, New-York based Foo Inc. announced their acquisition of Bar Corp."
A broad goal of IE is to allow computation to be done on the previously unstructured data. A more specific goal is to allow logical reasoning to draw inferences based on the logical content of the input data. Structured data is semantically well-defined data from a chosen target domain, interpreted with respect to category and context.

A. Present Significance

The present significance of IE pertains to the growing amount of information available in unstructured form. Tim Berners-Lee, inventor of the world wide web, refers to the existing Internet as the web of documents and advocates that more of the content be made available as a web of data. Until this transpires, the web largely consists of unstructured documents lacking semantic metadata. Knowledge contained within these documents can be made more accessible for machine processing by means of transformation into relational form, or by marking-up with XML tags. Tasks and subtasks
Applying information extraction on text, is linked to the problem of text simplification in order to create a structured view of the information present in free text. The overall goal is being to create a more easily machine-readable text to process the sentences.

B. Message Extraction from Event Logs

Event logs generated by applications that run on a system consist of independent lines of text data, which contain information that pertains to events that occur within a system. This makes them an important source of information to system administrators in fault management and for intrusion detection and prevention. With regard to autonomic systems, these two tasks are important cornerstones for self-healing and self-protection, respectively. Therefore, as we move toward the goal of building systems that are capable of self-healing and self-protection, an important step would be to build systems that are capable of automatically analyzing the contents of their log files, in addition to measured system metrics [2-3], to provide useful information to the system administrators.
Message type descriptions are the templates on which the individual unstructured messages in any event log are built. Message types, once found, are useful in several ways:
Compression. Message types can abstract the contents of server's system logs. We can therefore use them to obtain more concise and compact representations of log entries. This leads to memory and space savings.
Indexing. Each unique message type can be assigned an Identifier Index (ID), which in turn can be used to index historical server's system logs leading to faster searches. In [8], the authors demonstrated how message types can be used for log size reduction and indexing of the contents of event logs.
Model building. The building of computational models on the log data, which usually requires the input of structured data, can be facilitated by the initial extraction of message type information. Message types are used to impose structure on the unstructured messages in the log data before they are used as input into the model building algorithm. In [9-10], the authors demonstrate how message types can be used to extract measured metrics used for building computational models from event logs. The authors were able to use their computed models to detect faults and execution anomalies using the contents of server's system logs.
Visualization. Visualization is an important component of the analysis of large data sets. Visualization of the contents of systems logs can be made more meaningful to a human observer by using message types as a feature of the visualization. For the visualization to be meaningful to a human observer, the message

types must be interpretable. This fact provides a strong incentive for the production of message types that have meaning to a human observer.

"Connection from 192.168.10.8 port 21"

These four log entries would form a cluster (group) or event type in the event log and can be represented by the message type description (or line format):

"Connection from * port *"

The wildcards "*" represent message variables. We will adopt this representation in the rest of our work. Determining what constitutes a message type might not always be as simple as this example might suggest. Consider the following messages produced by the same print statement. "Link 1 is up," "Link 1 is down," "Link 3 is down," "Link 4 is up." The most logical message type description here is "Link * is *," however from an analysis standpoint having two descriptions "Link * is up" and "Link * is down" maybe preferable.

The goal of message type extraction is to find the representations of the message types that exist in a log file. This problem is well attested to in the literature but there is as yet no standard approach to the problem [9].

One drawback of the partitional clustering is the difficulty in determining the optimal number of clusters. Incremental clustering is an efficient method and runs in linear time to the size of input data set. In most related studies, the dissimilarity between two clusters is defined as the distance between their centroid or the distance between two closest data points. Hierarchical clustering algorithms create a hierarchical decomposition of data set based on some criterion

Hierarchical clustering algorithms can differ in their operation. Agglomerative clustering methods start with each object in a distinct cluster and successively merge them to larger clusters until a stopping criterion is satisfied. Alternatively, divisive algorithms begin with all objects in a single cluster and perform splitting until a stopping criterion is met. This problem is a cause for inaccuracy in clustering, especially for poorly separated data sets.

II. Hierarchical Clustering Algorithms

As its name implies, a hierarchical clustering algorithm establishes a hierarchical structure as the clustering result. Owing to their good quality of clustering results, hierarchical algorithms are widely used especially in document clustering and classification. The outline of a general hierarchical clustering algorithm is given below:

Hierarchical Clustering Algorithm:

1. Initially, each data point forms a cluster by itself.
2. Repetitively merge the two closest clusters.
3. Output the hierarchical structure that is constructed.

Most existing hierarchical clustering algorithms are variations of the single-link and complete-link algorithms. Both algorithms require time complexity of O (n2 log n).

Where n is the size of the input data set. These algorithms differ in the way they characterize the similarity between a pair of clusters. A single-link clustering algorithm differs from a complete-link clustering algorithm in the inter-cluster distance measure. The single-link algorithm uses the distance between the two closest points of the two clusters as the inter-cluster distance.

In either case, two clusters are merged to form a larger cluster based on minimum distance criteria. The complete-link algorithm produces tightly bound or compact clusters. The single-link algorithm, by contrast, suffers from a chaining effect. It has a tendency to produce clusters that are elongated. The clusters obtained by the complete-link algorithm are more compact than those obtained by the single-link algorithm. From a pragmatic viewpoint, it has been observed that the complete-link algorithm produces more useful hierarchies in many applications than the single-link algorithm.

We focus here on clustering as an unsupervised learning process, which attempts to represent the partitions of data of unknown class origin without feedback or information beyond what is inherently gained from the data. Its purpose is to _nd underlying groups, or clusters, which ideally share similar features. Most commonly, the purpose of unsupervised clustering is to autonomously learn how to best discretize a space with the intent of classifying unseen data samples into one of several clusters with the assumption that commonly classed samples share a common features. As class labels are unnecessary for training or adjusting parameters, learning here implies a presentation of a set of samples that need not be segmented into training, test, and validation subsets. In this paper, we introduce a new approach to incremental or online clustering that puts strict restraints on centroid estimation and modi_cation in an attempt to handle the stability/plasticity dilemma that plagues all data-driven systems. As will be detailed in the following section, our approach augments traditional competitive learning clustering algorithms [1].

A. Existing System

Currently, incremental document clustering is one the most effective techniques to organize documents in an unsupervised manner for many Web applications. This paper summarizes the research actuality and new progress in incremental clustering algorithm in recent years. First, some representative algorithms are analyzed and generalized from such aspects as algorithm thinking, key technique, advantage and disadvantage. Secondly, we select four typical clustering algorithms and carry out simulation experiments to compare their clustering quality from both accuracy and efficiency. The work in this paper can give a valuable reference for incremental clustering research.

This work is using incremental clustering application in extraction of messages from the event logs of any system for clustering purposes and retrieving relevant information from it. The works of the several authors have been studied and lot of knowledge has been got to propose this work.

Adetokunbo Makanju et al. [1] have specified that Message type or message cluster extraction is an important task in the analysis of server's system logs in computer networks. Defining these message types automatically facilitates the automatic analysis of server's system logs. IPLoM, which stands for Iterative Partitioning Log Mining, works through a 4-step process. The first three steps hierarchically partition the event log into groups of event log messages or event clusters. In its fourth and final stage, IPLoM produces a message type description or line format for each of the message clusters. IPLoM is able to find clusters in data irrespective of the frequency of its instances in the data, it scales gracefully in the case of long message type patterns and produces message type descriptions at a level of abstraction, which is preferred by a human observer. Evaluations show that IPLoM outperforms similar algorithms statistically significantly.

A specialized algorithm such as IPLoM can significantly improve the abstraction level of the unstructured message types extracted from the data. Message types are fundamental units in any application log file. Determining what message types can be produced by an application accurately and efficiently is therefore a fundamental step in the automatic analysis of log files.

Message types, once determined, not only provide groupings for categorizing and summarizing log data, which simplifies further processing steps like visualization or mathematical modeling, but also a way of labeling the individual terms (distinct word and position pairs) in the data.

Rui Xu et al. [2] have researched on Swarm intelligence has emerged as a worthwhile class of clustering methods due to its convenient implementation, parallel capability, ability to avoid local minima, and other advantages. In such applications, clustering validity indices usually operate as fitness functions to evaluate the qualities of the obtained clusters. However, as the validity indices are usually data dependent and are designed to address certain types of data, the selection of different indices as the fitness functions may critically affect cluster quality. Here, we compare the performances of eight well-known and widely used clustering validity indices, namely, the Cali ´nski–Harabasz index, the CS index, the Davies–Bouldin index, the Dunn index with two of its generalized versions, the I index, and the silhouette statistic index, on both synthetic and real data sets in the framework of Differential-Evolution–Particle-Swarm-Optimization (DEPSO) - based clustering. DEPSO is a hybrid evolutionary algorithm of the stochastic optimization approach (differential evolution) and the swarm intelligence method (particle swarm optimization) that further increases the search capability and achieves higher flexibility in exploring the problem space.

This paper introduces a new hybrid cluster validity method based on particle swarm optimization, for successfully solving one of the most popular clustering/classifying complex datasets problems. The proposed method for the solution of the clustering/classifying problem, designated as PSORS index method, combines a particle swarm optimization (PSO) algorithm, Rough Set (RS) theory and a modified form of the Huang index function. In contrast to the Huang index method which simply assigns a constant number of clusters to each attribute, this method could cluster the values of the individual attributes within the dataset and achieves both the optimal number of clusters and the optimal classification accuracy. The validity of the proposed approach is investigated by comparing the classification results obtained for a real-world dataset with those obtained by pseudo-supervised classification BPNN, decision-tree and Huang index methods [3].

A predictive model is required to be accurate and comprehensible in order to inspire confidence in a business setting. Both aspects have been assessed in a software effort estimation setting by previous studies. However, no univocal conclusion as to which technique is the most suited has been reached. The results are subjected to rigorous statistical testing and indicate that ordinary least squares regression in combination with a logarithmic transformation performs best. Another key finding is that by selecting a subset of highly predictive attributes such as project size, development, and environment related attributes, typically a significant increase in estimation accuracy can be obtained [4].

High impact event represents the information which is frequently used. The frequently used information is maintained in different clusters such that it can be accessed quickly without involving much searching time. Clustering methods are one of the key steps that lead to the transformation of data to knowledge. Clustering algorithms aims at partitioning an initial set of objects into disjoint groups (clusters) such that objects in the same subset are more similar to each other than objects in different groups [5].

In this paper we present a generalization of the k-Windows clustering algorithm in metric spaces by following a selective Repeat ARQ protocol having fixed window size for accurate information transmission. The original algorithm was designed to work on data with numerical values. The proposed generalization does not assume anything about the nature of the data, but only considers the distance function over the data set. The efficiency of the proposed approach is demonstrated on msnbc data sets [5].

III. Proposed System

Since event logs are the huge set of datasets and grow rapidly as the system processing goes on, therefore, for extraction of data from it is variable and should grow by the time. In this work, it is being proposed to apply the incremental clustering to extract the data from the event log as per the characteristics provided by the users of the system.

Incremental Clustering requires initial clusters to be decided in advance i.e. they must pre exist for processing. If the initial clusters are to be fixed, then there are several ways it can be achieved. The algorithm being proposed is a dynamic and novice algorithm for deciding the initial clusters dynamically. In this work I am offering to create clusters dynamically after

Initially the user will choose a few characteristics to decide the initial clusters.

Data will be incrementally clustered into the user defined clusters

User can add any more clusters dynamically in the system by adding additional characteristics.

Now data will be clustered in increased number of clusters incrementally.

As the data will grow, user can restart the clustering mechanism to add the increased data in particular clusters.

The algorithm can be viewed as per the following flow chart:

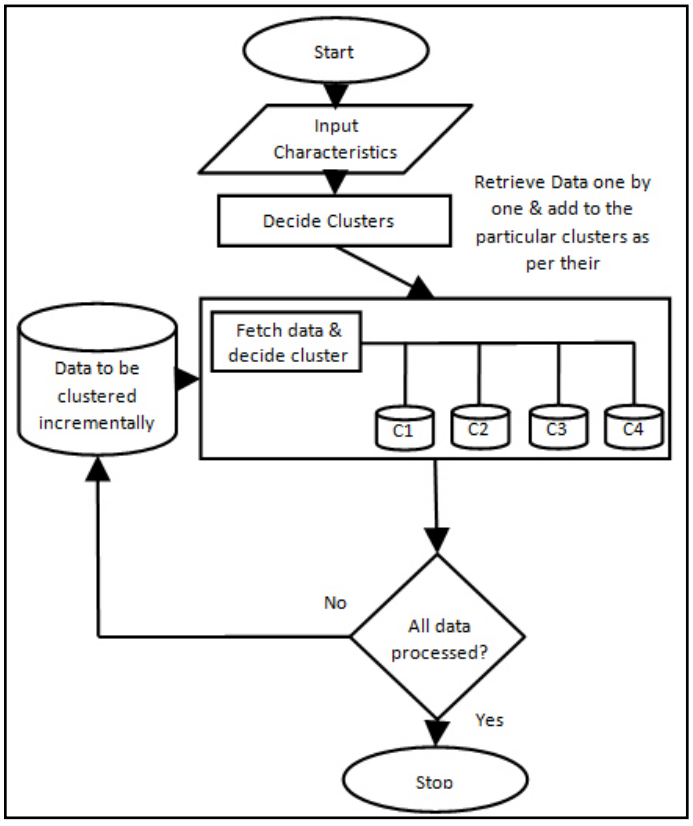

Fig. 1: Flow Chart Indicating the Flow of Processing to be Done for Implementation of the Proposed Algorithm

IV. Results

An implementation of the proposed algorithm has been done and dataset of a college server system log has been used to draw the

result and readings and the graphs drawn from the results are as follows:

Table 1: Readings Taken By Running the Proposed Algorithm

S. NO	CLUSTER	TOTAL	TP	FP	FN	PRECISION	RECALL	F-Measure
1	SBCore	3667	2115	1552	603	0.576765748568312	0.778145695364238	0.662490211433046
2	Print	101	101	0	43	1	0.701388888888889	0.824489795918367
3	Service Control Manager	3631	3067	564	1357	0.844670889562104	0.693264014466546	0.761514587212911
4	TermServLicensing	1919	1241	678	0	0.646690984887963	1	0.785443037974684
5	SMTPSVC	764	603	161	0	0.789267015706806	1	0.88222384784199
6	AppletTalk	1918	597	1321	1	0.31126173096976	0.998327759197324	0.47456279809221
7	IPSec	1511	1203	308	0	0.796161482461946	1	0.886514369933677
8	E100	2505	1287	1218	0	0.51377245508982	1	0.67879746835443
9	LPDSVC	599	599	0	0	1	1	1
10	AeLookupSvc	2519	599	1920	0	0.237792774910679	1	0.384220654265555

Table 2: Readings of the Various Algorithms Alongwith Proposed Work

SNO	ALGORITHM	RECALL	PRECISION	F-Measure
1	Loghound	0.11	0.05	0.07
2	Loghound-2	0.56	0.36	0.34
3	SLCT	0.32	0.26	0.19
4	IPLoM	0.66	0.60	0.63
5	Proposed	0.67	0.92	0.73

Table 3: Percentage of Records Available Through Various Algorithms

SNO	Instance Frequency Ranges	SLCT	Loghound	Loghound-2	IPLoM	Proposed Work
1	1-150	16.67	18.75	50.69	47.92	00.50
2	151-1000	20.59	23.53	63.24	72.06	07.12
3	>1000	34.00	38.00	74.00	82.00	92.34

Following results have been drawn using the various readings taken as above.

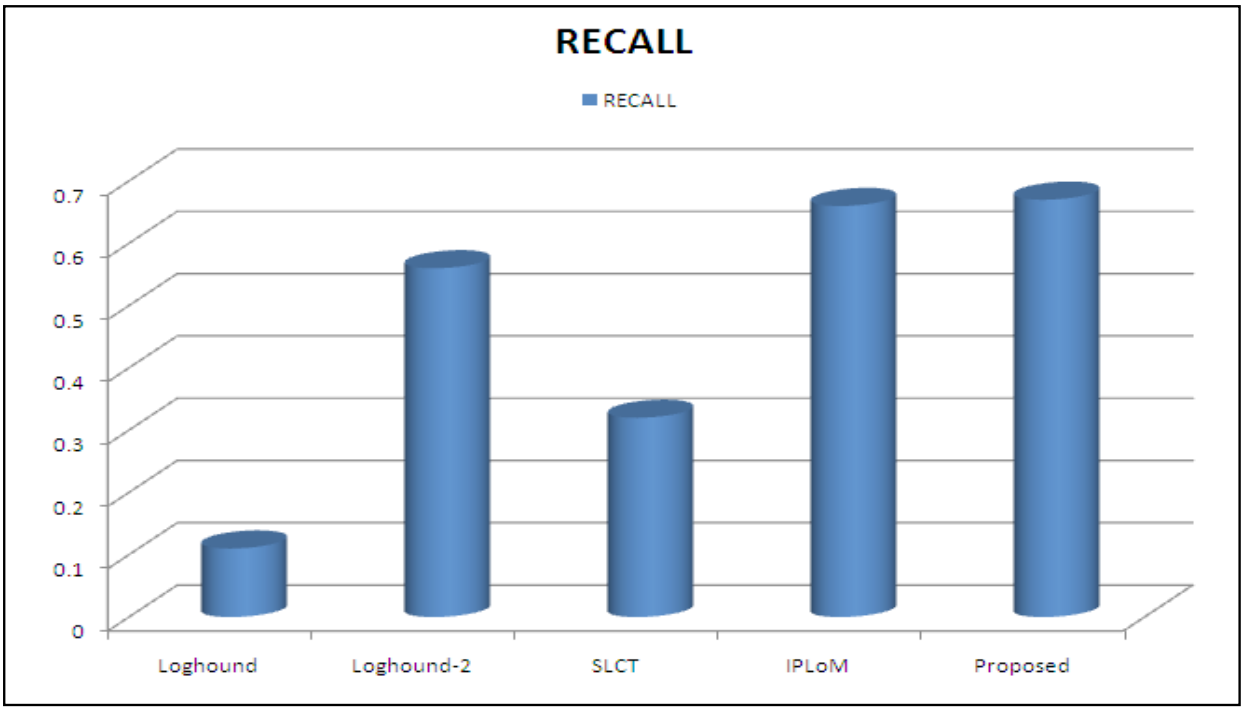

Fig. 2: Value of Recall for Various Algorithms

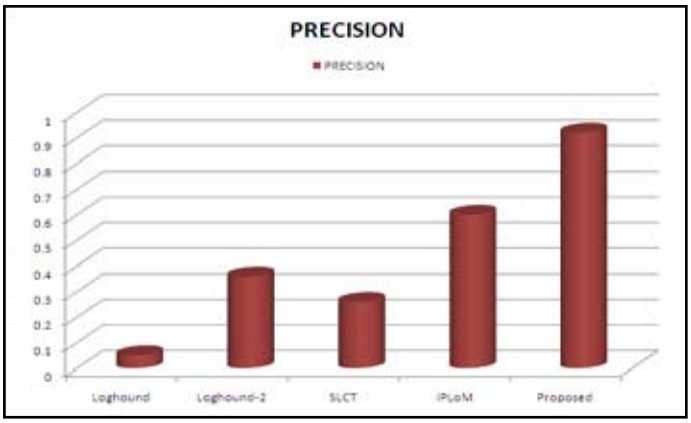

Fig. 3: Value of Precision for Various Algorithms

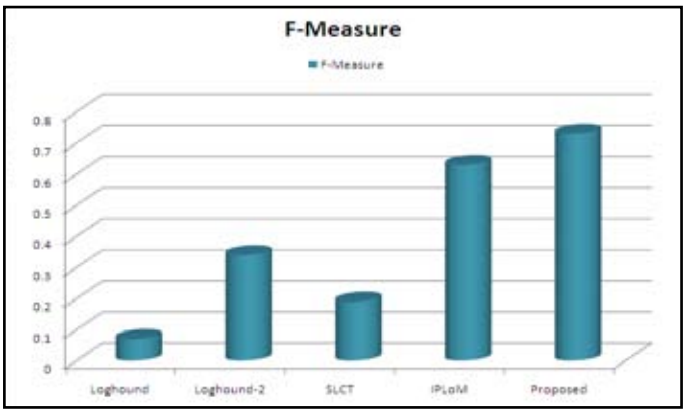

Fig. 4: Value of F-Measure for Various Algorithms

Value of the recall, precision & F-Measure for the various algorithm alongwith proposed work in this paper depicts that the proposed work is having high values in all three graphs and hence the proposed work is concluded to be more accurate in all of these terms.

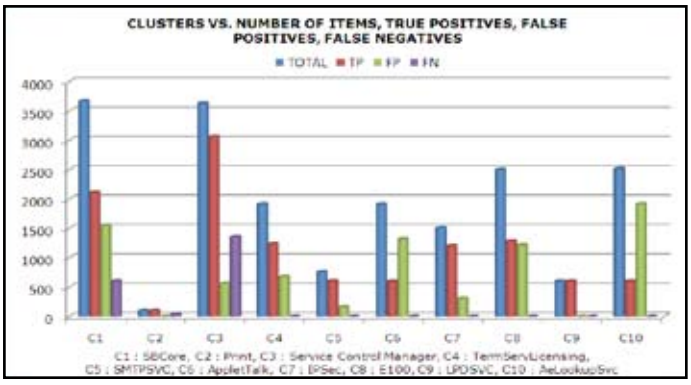

Fig. 5: Frequency of Items in Each Clusters

References

[1] Adetokunbo Makanju, A. Nur Zincir-Heywood, Evangelos E. Milios,"A Lightweight Algorithm for Message Type Extraction in System Application Logs", IEEE transactions on knowledge and data engineering, Vol. 24, No. 11, November 2012, 1041-4347/ 2012 IEEE

[2] Rui Xu, Jie Xu, Donald C. Wunsch, II,"A Comparison Study of Validity Indices on Swarm-Intelligence-Based Clustering", IEEE transactions on systems, man, and cybernetics—part b: cybernetics, Vol. 42, No. 4, August 2012, 1083-4419/ 2012 IEEE

[3] Kuang Yu Huang,"A hybrid particle swarm optimization approach for clustering and classification of datasets", Department of Information Management, Ling Tung University, Ling Tung Road, Taichung City 408, Taiwan Knowledge-Based Systems 24 (2011) pp. 420–426. knosys.2010.12.003

[4] Karel Dejaeger, Wouter Verbeke, David Martens, Bart Baesens,"Data Mining Techniques for Software Effort Estimation: A Comparative Study", IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, Vol. 38, No. 2, March/ April 2012, IEEE.

[5] Purna Chandra Sethi, Chinmay Dash,"High Impact Event Processing using Incremetal Clustering in Unsupervised Feature Space through Genetic algorithm by Selective Repeat ARQ protocol", International Conference on Computer & Communication Technology (ICCCT)-2011, IEEE.

[6] J.O. Kephart, D.M. Chess,"The Vision of Autonomic Computing", Computer, Vol. 36, No. 1, pp. 41-50, Jan. 2003.

[7] I. Cohen, S. Zhang, M. Goldszmidt, J. Symons, T. Kelly, and A. Fox,"Capturing, Indexing, Clustering, and Retrieving System History", Proc. 20th ACM Symp. Operating Systems Principles, pp. 105-118, 2005

[8] M. Jiang, M.A. Munawar, T. Reidemeister, P.A. Ward, "Dependency-Aware Fault Diagnosis with Metric-Correlation Models in Enterprise Software Systems", Proc. Sixth Int'l Conf. Network and Service Management, pp. 137-141, 2010

[9] M. Klemettinen,"A Knowledge Discovery Methodology for Telecommunications Network Alarm Databases", Ph.D dissertation, Univ. of Helsinki, 1999

[10] S. Ma., J. Hellerstein,"Mining Partially Periodic Event Patterns with Unknown Periods", Proc. 16th Int'l Conf. Data Eng., pp. 205- 214, 2000

[11] Q. Zheng, K. Xu, W. Lv, S. Ma,"Intelligent Search for Correlated Alarm from Database Containing Noise Data", Proc. Eighth IEEE/IFIP Network Operations and Management Symp., pp. 405-419, 2002

[12] J. Stearley,"Towards Informatic Analysis of Syslogs", Proc. IEEE Int'l Conf. Cluster Computing, pp. 309-318, 2004

[13] A. Makanju, A.N. Zincir-Heywood, E.E. Milios,"Storage and Retrieval of System Log Events Using a Structured Schema Based on Message Type Transformation", Proc. 26th ACM Symp. Applied Computing (SAC), pp. 525-531, Mar. 2011.

[14] Yongli Liu, Qianqian Guo, Lishen Yang, Yingying Li, "Research on Incremental Clustering", 2012 IEEE

[15] K. M. Hammouda, M. S. Kamel,"Efficient Phrase-Based Document Indexing for Web Document Clustering", IEEE Transactions on Knowledge and Data Engineering, 2004, 16(10), pp. 1279-1296.

[16] Y. Liu, Y. Ouyang, Z. Xiong, "Incremental Clustering using Information Bottleneck Theory", International Journal of Pattern Recognition and Artificial Intelligence, 2011, 25(5), pp. 695-712.

[17] [Online] Available: http://www.en.wikipedia.org/wiki/User-generated_content.

[18] X. Wan,"A novel document similarity measure based on earth mover's distance", Information Sciences, 2007, 177(18), pp. 3718-3730.

[19] K.M. Hammouda, M. S. Kamel, "Incremental document clustering using cluster similarity histograms", In Proc. of Int. Conf. on Web Intelligence, 2003, pp. 597-601.

[20] O. Zamir, O. Etzioni,"Web document clustering: A feasibility demonstration", In Proc. of the 21st Annual Int. ACM SIGIR

Conf., 1998, pp. 46-54.
[21] W. Wong, A. Fu,"Incremental document clustering for Web page classification", In Proc. 2000 Int. Conf. Information Soc. In the 21st Century: Emerging Technologies and New Challenges (IS2000), 2000.
[22] S. Noam, T. Naftali,"Document clustering using word clusters via the information bottleneck method", In Proc. 23rd Ann. Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 2000, pp. 208-215.
[23] [Online] Available: http://www.en.wikipedia.org/wiki/K-nearest_neighbor_ algorithm.
[24] S. Branson, A. Greenberg, "Clustering Web Search Results Using Suffix Tree Methods", Stanford University, unpublished.
[25] M. Ester, H. Kriegel, J. Sander, X. Xu , "A density-based algorithm for discovering clusters in large spatial databases with noise", in Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining, 1996, pp. 226-231.
[26] Tu. Nguyen-Hoang, K. Hoang, D. Bui-Thi, A. Nguyen, "Incremental Document Clustering Based on Graph Model", Advanced Data Mining and Applications, 2009, pp. 569-576.
[27] S. Noam, F. Nir, T. Naftali,"Unsupervised document classification using sequential information maximization", In Proc. of the 25th Ann. Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 2002, pp. 129-136

Garima Chouksey received her B.E. degree in computer science from Gyan Ganga College of Technology, Jabalpur, M.P., India, in 2011. She received the Certificate on Java, C, C++, Asp.NET and SQL from HCL. She participated in Race in GGCT in 2008 and stood 2nd in 400 Meter race and 3rd in 200 Meter race. Her research interests include digital signal processing, data mining techniques, networking and data structure technique.

.