Open Access

IJCER ONLINE

ISSN Online: 2250–3005

Impact Factor: 1.145

IJCER - International Journal of Computational Engineering Research

Volume 05 – Issue 10, (October 2015)

www.ijceronline.com– Email–ijceronline@gmail.com

Editorial Board

Editor-In-Chief

Editorial Committees

Contents:

ISSN (e): 2250 – 3005 || Volume, 05 || Issue, 10 ||October – 2015 ||
International Journal of Computational Engineering Research (IJCER)

Modeling and Analysis of Flexible Manufacturing System with FlexSim

B.Santhosh Kumar [1], Dr.V.Mahesh[2], B.Satish Kumar[3]
[1] *ME Student, S.R Engineering College, Warangal, India*
[2] *Professor & Dean (Research), Dept. of ME, S.R Engineering College, Warangal, India,*
[3] *Professor, Dept. of ME, S.R Engineering College, Warangal, India,*

ABSTRACT
Flexible manufacturing system (FMS) is a highly integrated manufacturing system. The relation between its components is very complex. The mathematical programming approaches are very difficult to solve for very complex system so the simulation of FMS is widely used to analyze its performance measures. Also the FMS components are very sophisticated and costly. If FMS has to be implemented then it is better to analyze its results using simulation which involves no loss of money, resource and labour time. As a typical discrete event system FMS have been studied in such aspects as modeling and performance analysis. In this paper, a concept and implementation of the Flexsim for measuring and analysis of performance measures of FMS is applied. The other well defined mathematical technique, i.e. bottleneck technique has also been applied for the purpose of comparison and verification of the simulation results. An example FMS has been taken into consideration and its flexsim model and mathematical model has been constructed. Several performance measures have been used to evaluate system performance. And it has been found that the simulation techniques are easy to analyze the complex fexible manufacturing system.

Keywords: *Bottleneck, Flexible Manufacturing System (FMS), FlexSim, Simulation.*

I. INTRODUCTION

In the present market scenario, the customer demand and specification of any product changes very rapidly so it is very important for a manufacturing system to accommodate these changes as quickly as possible to be able to compete in the market. This evolution induces often a conflict for a manufacturing system because as the variety is increased the productivity decreases. So the flexible manufacturing system (FMS) is a good combination between variety and productivity. In this system, the main focus is on flexibility rather than the system efficiencies. A competitive FMS is expected to be flexible enough to respond to small batches of customer demand and due to the fact that the construction of any new production line is a large investment so the current production line is reconfigured to keep up with the increased frequency of new product design.

The optimal design of FMS is a critical issue and it is a complex problem. There are various modeling techniques for FMS; the most common one are based on mathematical programming. FMS is a highly integrated manufacturing system and the inter-relationships between its various components are not well understood for a very complex system. Due to this complexity, it is difficult to accurately calculate the performance measures of the FMS which leads to its design through mathematical techniques. Therefore, computer simulation is an extensively used numeric modeling technique for the analysis of highly complex flexible manufacturing systems.

Modeling and simulation of FMS is a field of research for many people now days. However, they all share a common goal; to search for solutions to achieve higher speeds and more flexibility and thus increase manufacturing productivity. FlexSim is a discrete event manufacturing simulation software developed by FlexSim Software Products, Inc. The FlexSim family currently includes the basic FlexSim simulation software and FlexSim Healthcare Simulation (FlexSim HC). It uses an OpenGL environment to realize real-time 3D rendering.

In this research work, FMS is modeled with the help of Flexsim to analyze its performance measures. In addition, the bottleneck technique has been applied to compare and verify the results obtained from the simulation techniques.

II. Literature survey

Browne et al., 1984 defines FMS as an integrated computer controlled system with automated material handling devices and CNC machine-tools and which can be used to simultaneously process a medium-sized volume of a variety of parts.

Bennett et al. (1992) identifies the factors crucial to the development of efficient flexible production systems, namely: effective integration of subsystems, development of appropriate controls and performance measures, compatibility between
production system design and organization structure, and argues that the flexibility cannot be potentially exploited if its objectives are not defined and considered at design stage.

Delgadillo and Llano (2006) introduced a Petri net-based integrated approach, for simultaneously modeling and scheduling
manufacturing systems. A prototype that simulates the execution of the production plan, and implements priority dispatching
rules to solve the eventual conflicts, is presented. Such an application was tested in a complex flexible job shop-type system. Scheduling is a difficult task in most manufacturing settings due to the complexity of the system. Hence there is a requirement of powerful tools that can handle both modeling and optimization.

Shnits et al. (2004) used simulation of operating system as a decision support tool for controlling the flexible system to exploit flexibility.

Tu¨ ysu¨ z and Kahraman (2009) presented an approach for modeling and analysis of time critical, dynamic and complex
systems using stochastic Petri nets together with fuzzy sets.

Nandkeolyar and Christy (1989) interfaced a computer simulation model of an FMS with the Hooke–Jeeves algorithm to
search an optimum design without full factorial experimentation. Some modifications of the HJ algorithm are carried out to accommodate the stochastic nature of computer simulation. The inter-relationships between FMS components are not well understood. Consequently, it has not been possible to develop closed form analytic models of FMSs. So, computer simulation has been extensively applied to study their performance.

After reviewing the above set of research papers it can be said that the design and modeling of the complex FMS is a difficult task using mathematical techniques, so the computer simulation seems to be a better option. Therefore, to check the accuracy of the results obtained from simulation techniques this research work has been carried out.

III. Flexible manufacturing system

Flexible manufacturing system (FMS) is a class of manufacturing system that can be quickly configured to produce variety of products. Over the last few decades, the modeling and the analysis of FMSs has been closely studied by control theorists and engineers. An FMS is a production system where a discrete number of raw parts are processed and assembled by controlled machines, computers and/or robots. It generally consists of a number of CNC machine tools, robots, material handling, automated storage and retrieval system, and computers or workstations. A typical FMS can fully process the members of one or more part families on a continuing basis without human intervention and is flexible enough to suit changing market conditions and product types without buying other equipment (the concept ‘‘flexible’’ can refer to machines, processes, products, routings, volume, or productions). The concept of FMS is credited to David Williamson, a British engineer employed by Molins during the mid 1960s. Molins applied for a patent for the invention that was granted in 1965. The concept was called System 24 then because it was believed that the group of machine tools comprising the system could operate 24 h a day. One of the first FMS installed in US was a machining system at Ingersoll-Rand Company. (Fig. 1). There are three capabilities that a manufacturing system must possess in order to be flexible:

(1) The ability to identify and distinguish among different incoming part or product styles processed by the system.

(2) Quick changeover of operating instructions.

(3) Quick changeover of physical setup.

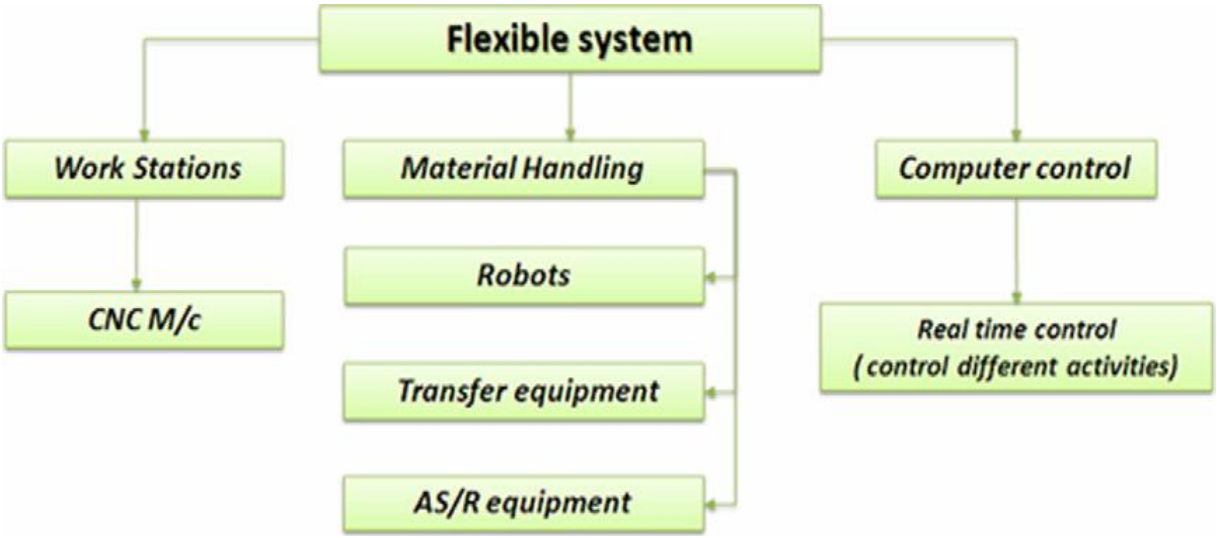


Figure 1 Flexible manufacturing system configuration**.**

IV. FlexSim

FlexSim is a powerful analysis tool that helps engineers and planners make intelligent decisions in the design and operation of a system. With FlexSim, you can build a 3-dimensional computer model of a real-life system, then study that system in a shorter time frame and for less cost than with the actual system. As a "what-if" analysis tool, FlexSim provides quantitative feedback on a number of proposed solutions to help you quickly narrow in on the optimum solution. With FlexSim's realistic graphical animation and extensive performance reports, you can identify problems and evaluate alternative solutions in a short amount of time. By using FlexSim to model a system before it is built, or to test operating policies before they are actually implemented, you will avoid many of the pitfalls that are often encountered in the startup of a new system. Improvements that previously took you months or years of trial-and-error experimentation to achieve can now be attained in a matter of days and hours using FlexSim.

V. Bottleneck technique

Important aspects of the FMS performance can be mathematically described by a deterministic model called the bottleneck model developed by Solberg (1981). The bottleneck model is simple and intuitive but it has a limitation of a deterministic approach. It can be used to provide starting estimates of FMS design parameters such as production rate, number of work stations, etc. The term bottleneck refers to the fact that an output of the production system has an upper limit, given that the product mix flowing through the system is fixed. This model can be applied to any production system that possesses possess this bottleneck feature.

Terminology and symbols

Part mix, a mix of the various parts or product styles produced by the system is defined by pi. The value of pi must sum to unity

$$\sum P_i = 1.0$$

The FMS has a number of distinctly different workstations n and s_i is the number of servers at the i^{th} workstation. Operation frequency is defined as the expected number of times a given operation in the process routing is performed for each work unit.

f_{ijk}= operation frequency

For each part or product, the process routing defines the sequence of operations, the workstations where operations are performed, and the associated processing time.

t_{ijk} = processing time for operation

The average workload, Wl_i

$$WL_i = \sum_j \sum_k t_{ijk} f_{ijk} p_i$$

The average of transport required completing the processing of a work part, n_t

$$n_t = \sum_i \sum_j \sum_k f_{ijk} p_i - 1$$

The workload of handling system, WL_{n+1}

$$WL_{n+1} = n_t t_{n+1}$$

where t_{n+1} = Mean Transport time per move, min.

The FMS maximum production rate of all part, R^*_p, Pc/min

$$R_p = S^*/WL^*$$

Where WL^* is workload min/Pc and S^* = Number of machines at the bottle-neck station.
The part (*j*) maximum production rate, R_{pi}, Pc/min.

$$R^*_{Pi} = P_i(R^*_{Pi}) = Pi\frac{S*}{WL*}$$

Mean utilization of a station (i), U_i

$$U_i = \frac{WLi}{Si}(R^*_p) = \frac{WLi}{Si} \times \frac{s*}{WL*}$$

Average Utilization of FMS including Transport system

$$\overline{U} = \frac{\sum_{i-1}^{n+1} U_i}{n+1}$$

Overall FMS utilization

$$\overline{U_s} = \frac{\sum_{i-1}^{n} S_i U_i}{\sum_{i-1}^{n} S_i}$$

VI. Case study

A flexible manufacturing system consists of two machining workstations and a load/unload stations. Station 1 is the load/unload station. Station 2 performs milling operations and consists of two servers (two identical CNC milling machines). Station 3 has one server that performs drilling(one CNC drill press). The stations are connected by a part handling system that has four work carriers. The mean transport time is 3.0min. The FMS produces two parts A and B. The part mix fractions and process routings f_{ijk}=1.0 for all operations.

Table 1 List of operations and process time on different machining centers.

Part j	Part Mix P_j	Operation K	Description	Station i	Process time t_{ijk} (min)
A	0.4	1	Load	1	4
		2	Mill	2	30
		3	Drill	3	10
		4	Unload	1	2
B	0.6	1	Load	1	4
		2	Mill	2	40
		3	Drill	3	15
		4	Unload	1	2

6.1 Solution methodology

Two types of techniques are applied to find the parameters of the given FMS; the one is simulation techniques and another one is mathematical technique. The simulation technique is Flexsim. The mathematical one is the bottleneck technique. The system is modeled in simulation technique and then the results are compared with the mathematical technique.

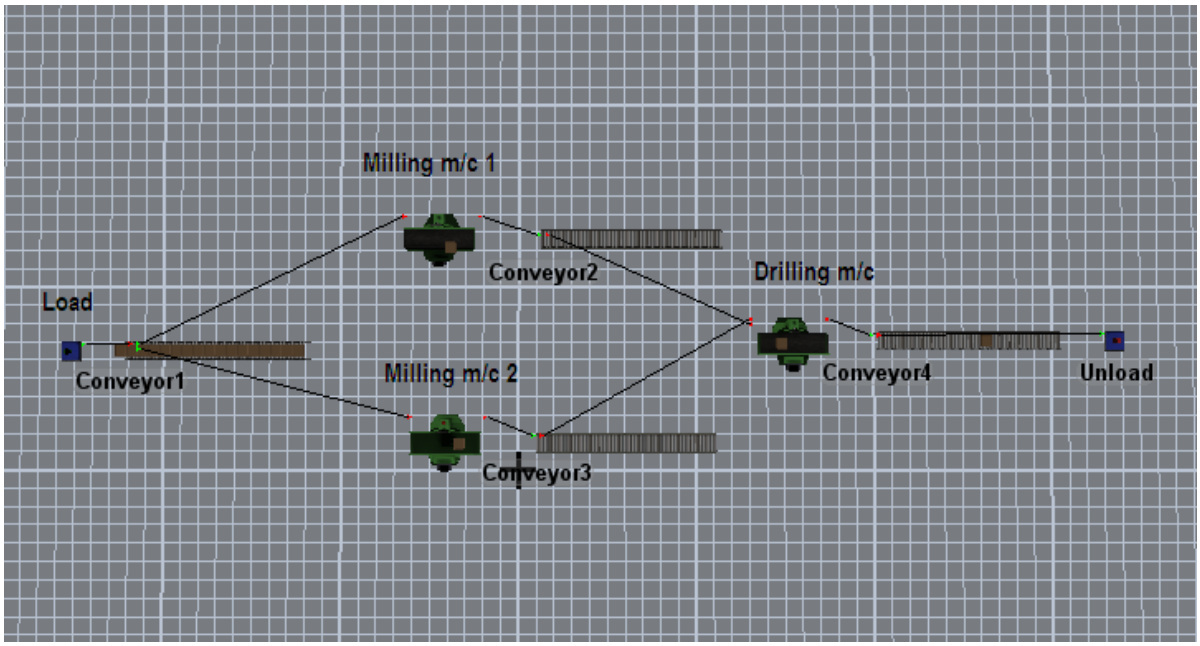


Figure 2: Flexsim model of the FMS of case study

6.1.1 Modeling in flexsim

The system consists of loading and unloading station, three process stations, four work carriers. To start a cycle, raw parts and the work carriers must be available. Then only firing will take place. The conveyor carries a raw part from loading station to the process station according to the given sequence for the different parts. After the completion of an operation in one station the part is again carried by another conveyor to its next required station. At the end the part is carried to the unloading station.

Place representing the stations have tokens according to the number of machines they have. The Flexsim model is simulated to get the overall productivity of the given system. The Flexsim model is shown in Fig 2.

6.1.2 Bottleneck technique

A C program has been developed to get the performance measures of the FMS system using bottleneck technique.

6.1.3 Results

By the two different techniques the obtained results for case study are as follows:

Solution Techniques	Utilization	
Operations	Flexsim	Bottle-neck model
Milling	0.9865	0.9989
Drilling	0.7162	0.7225

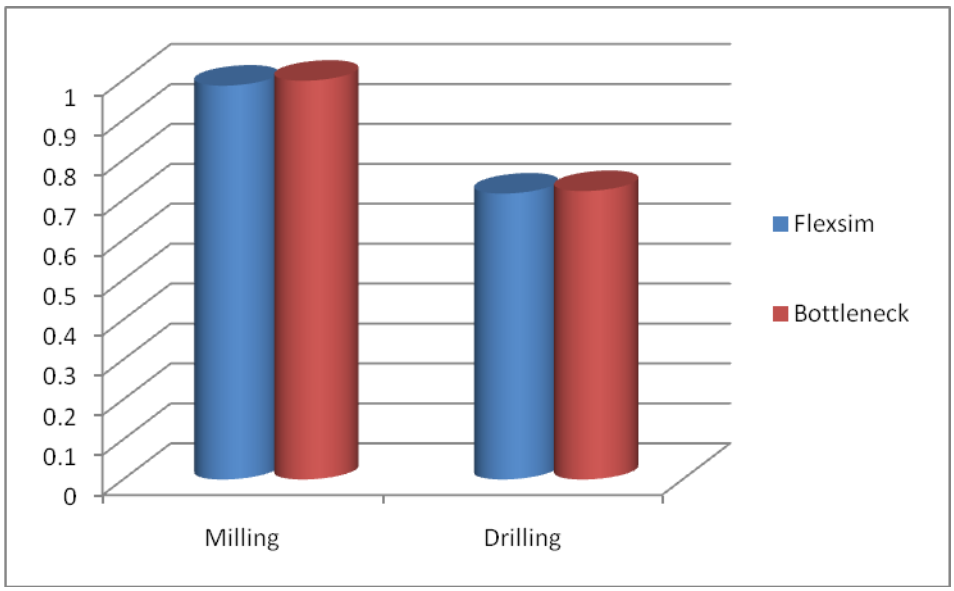


Fig 3. Comparison of utilization from different techniques

VII. Conclusions

In this research, a concept and implementation of the Flexsim for measuring and analysis of performance measures of FMS is applied. The other well defined mathematical technique, i.e. bottleneck technique has also been applied for the purpose of comparison and verification of the simulation results. An example FMS has been taken into consideration and its flexsim model and mathematical model has been constructed. Several performance measures have been used to evaluate system performance. Then finally the utilization of two techniques are approximately same. And it has been done that the simulation techniques are easy to analyze the complex fexible manufacturing system. FMS has to be implemented then it is better to analyze its results using simulation which involves no loss of money, resource and labour time

References

[1] Bennett, D., Forrester, P., Hassard, J., 1992. Market-driven strategies and the design of flexible production systems: evidence from the electronics industry. Int. J. Oper. Prod. Manag. 12 (2), 25–43.

[2] Browne, J., Dubois, D., Rathmill, K., Sethi, P., Steke, KE., 1984. Classification of flexible manufacturing systems. FMS Mag., 14–27.

[3] Bruccoleri, M., Sergio, NL., Perrone, G., 2003. An object-oriented approach for flexible manufacturing controls systems analysis and design using the unified modeling language. Int. J. Flexible Manuf. Syst. 15 (3), 195–216.
[4] Cheng, T.C.E., 1985. Simulation of flexible manufacturing system. Simulation 45 (6), 299–302.
[5] Delgadillo, G.M., Llano, S.B., 2006. Scheduling application using petri nets: a case study: intergra´ ficas s.a.'. In: Proceedings of 19th international conference on production research, Valparaiso, Chile.
[6] Gupta, D., Buzacott, JA., 1989. A framework for understanding flexibility of manufacturing systems. J. Manuf. Syst. 8 (2), 89–97.
[7] Kumar, R., Tiwari, MK., Shankar, R., 2003. Scheduling of flexible manufacturing systems: an ant colony optimization approach. Proc. Inst. Mech. Eng. 217 (10), 1443–1453.
[8] Nandkeolyar, U., Christy, D.P., (1989). Using computer simulation to optimize flexible manufacturing system design. In: Proceedings of the 1989 Winter Simulation Conference.
[9] Narakari, Y., Viswanadham, N., 1985. A Petri net approach to modeling and analysis of flexible manufacturing system. Annu. Oper. Res. 3, 449–472..
[10] Ruiz, C.M., Cazorla, D., Cuartero, F., Macia, H., 2009. Improving performance in flexible manufacturing systems. J. Logic Algebra Program. 78, 260–273.
[11] Santarek, K., Buseif, I.M., 1998. Modelling and design of flexible manufacturing systems using SADT and Petri nets tools. J. Mater. Process. Technol. 76, 212–218..
[12] Shnits, B., Rubinovittz, J., Sinreich, D., 2004. Multi-criteria dynamic scheduling methodology for controlling a flexible manufacturing system. Int. J. Prod. Res. 42, 3457–3472.
[13] Tavana, M., 2008. Dynamic process modelling using Petri nets with applications to nuclear power plant emergency management. Int. J.Simul. Process Mod. 4 (2).
[14] Tu¨ ysu¨ z, F., Kahraman, C., 2009. Modeling a flexible manufacturing cell using stochastic Petri nets with fuzzy parameters. Expert Syst. Appl.
[15] Wang, FK., Yen, PY., 2001. Simulation analysis of dispatching rules for an automated interbay material handling system in wafer fab. Int. J. Prod. Res. 39, 1221–1238.

ISSN (e): 2250 – 3005 || Volume, 05 || Issue, 10 ||October – 2015 ||
International Journal of Computational Engineering Research (IJCER)

Efficient Resource Allocation to Virtual Machine in Cloud Computing Using an Advance Algorithm

Rajeev Kumar [1], Aditya Sharma[2]
[1] Deptt. of C.S.E.,Arni University, Kangra,India
[2] Deptt. of C.S.E, Arni University, Kangra, India

ABSTRACT:
The focus of the paper is to generate an advance algorithm of resource allocation and load balancing that can deduced and avoid the dead lock while allocating the processes to virtual machine. In VM while processes are allocate they executes in queue , the first process get resources , other remains in waiting state .As rest of VM remains idle . To utilize the resources, we have analyze the algorithm with the help of First-Come, First-Served (FCFS) Scheduling, Shortest-Job-First (SJR) Scheduling, Priority Scheduling, Round Robin (RR) and CloudSIM Simulator.

KEYWORDS: *VM(Virtual machine)*

I. INTRODUCTION

Cloud computing has attracted attention as an important platform for software deployment, with perceived benefits such as elasticity to fluctuating load, and reduced operational costs compared to running in enterprise data centers. While some software is written from scratch especially for the cloud, many organizations also wish to migrate existing applications to a cloud platform. A cloud environment is one of the most shareable environments where multiple clients are connected to the common environment to access the services and the products. A cloud environment can be public or the private cloud. In such environment, all the resources are available on an integrated environment where multiple users can perform the request at same time. In such case , some approach is required to perform the effective scheduling and the resource allocation.

II. RESOURCE ALLOCATION

There are different algorithm that defines the load balancing to provide resources on the criteria as following

2.1 Token Routing: The main objective of the algorithm is to minimize the system cost by moving the tokens around the system. But in a scalable cloud system agents cannot have the enough information of distributing the work load due to communication bottleneck. So the workload distribution among the agents is not fixed. The drawback of the token routing algorithm can be removed with the help of heuristic approach of token based load balancing. This algorithm provides the fast and efficient routing decision. In this algorithm agent does not need to have an idea of the complete knowledge of their global state and neighbor's working load. To make their decision where to pass the token they actually build their own knowledge base. This knowledge base is actually derived from the previously received tokens. So in this approach no communication overhead is generated.

2.2 Round Robin: In this algorithm, the processes are divided between all processors. Each process is assigned to the processor in a round robin order. The process allocation order is maintained locally independent of the allocations from remote processors. Though the work load distributions between processors are equal but the job processing times for different processes are not same. So at any point of time some nodes may be heavily loaded and others remain idle. This algorithm is mostly used in web servers where Http requests are of similar nature and distributed equally.

2.3 Randomized: Randomized algorithm is of type static in nature. In this algorithm process can be handled by a particular node n with a probability p. The process allocation order is maintained for each processor independent of allocation from remote processor.

This algorithm works well in case of processes are of equal loaded. [10]. However, problem arises when loads are of different computational complexities. Randomized algorithm does not maintain deterministic approach. It works well when Round Robin algorithms generate solver head for process queue.

2.4 Central queuing: This algorithm works on the principal of dynamic distribution. Each new activity arriving at the queue manager is inserted into the queue. When request for an activity is received by the queue manager it removes the first activity from the queue and sends it to the requester. If no ready activity is present in the queue the request is buffered, until a new activity is available. But in case new activity comes to the queue while there are unanswered requests in the queue the first such request is removed from the queue and new activity is assigned to it. When a processor load falls under the threshold then the local load manager sends a request for the new activity to the central load manager.

2.6 Connection mechanism: Load balancing algorithm can also be based on least connection mechanism which is a part of dynamic scheduling algorithm. It needs to count the number of connections for each server dynamically to estimate the load. The load balancer records the connection number of each server. The number of connection increases when a new connection is dispatched to it, and decreases the number when connection finishes or timeout happens.

III. ALGORITHM

The algorithm provide parallel processes to each virtual machine rather than serial processes one by one.

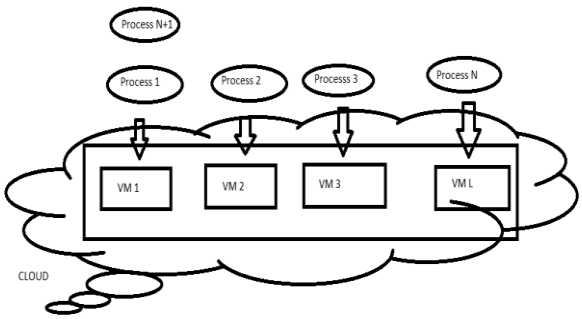


Figure 1. VM Function

*a. *Input the M number of Clouds with L1, L2, L3……., Ln (n tends to no. of last virtual machine))number of Virtual Machines associated with each cloud.
*b. *Define the available memory and load for each virtual machine.
*c. *Assign the priority to each cloud.
*d. *Input n number of user process request with some parameters specifications like arrival time, process time, required memory etc.
*e. *Arrange the process requests in order of memory requirement
*f. *For i=1 to n
*g. *{
*h. *Identify the priority Cloud and Associated VM having Available Memory(L1,L2,L3……Ln)>Required Memory(i)
*i. *Perform the initial allocation of process to that particular VM and the Cloud
*j. *}
*k. *For i=1 to n
*l. *{
*m. *Identify the Free Time slot on priority cloud to perform the allocation. As the free slot identify, record the start time, process time, turnaround time and the deadline of the process.
*n. *}
*o. *fori=1 to n
*p. *(
q. start queue Q1.
*r. *{
*s. Process i1allocate to VM L1.
*t. *Print "Migration Done"
*u. Process i2, i3……in allocate to VM L2, L3……,Ln respectively.
w. Q1, i1, p1 ends till i(n+1) allots to L1 again.
*X. * start new Queue Q2 [i (n+1)],
Q3 {I (2n+1)},……. Respectively.
*y. *}
*z. *}

IV. EXPERIMENTAL REVIEW

Larger waiting time and Response time

In round robin architecture the time the process spends in the ready queue waiting for the processor to get executed is known as waiting time and the time [13] the process completes its

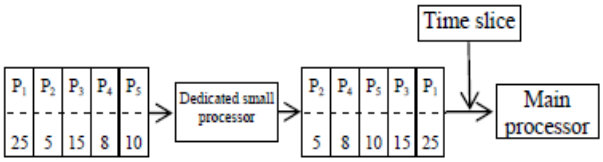


Figure: Process scheduling in shortest round robin

Intelligent time slice generation

A new way of intelligent time slice calculation has been proposed which allocates the frame exclusively for each task based on priority, shortest CPU burst time and context switch avoidance time.

Let the original time slice (OTS) is the time slice to be given to any process if it deserves no special consideration

Intelligent time slice = Original Time Slice(OTS)+ Priority Component (PC)+ Shortness Component for CPU burst time (SC)+ Context Switch Component(CSC)

The intelligent time slice of process P1 is same as the original time slice of four milliseconds and time slice of four milliseconds is assigned to process P1. After the execution of four milliseconds time slice the CPU is allocated to process P2. Since the CPU burst of process P2 is lesser than the assumed CPU burst (ATS), one milliseconds of SC has been included. The process P3 has the highest priority, so priority component is added and the total of five milliseconds is allocated to process P3. The Balanced CPU burst for process P4 is leaser than OTS, context switch component is added and a total of eight millisecond time slice is given to process P4. Process P5 is given a total of five milliseconds with one millisecond of priority component is added to original time slice. After executing a cycle the processor will again be allocated to process P1 for the next cycle and continuously schedules in the same manner.

$$\sum_{i=1}^{n} \frac{wt_i}{n}$$

wt → waiting time of process.

n → No. of process.

$$\sum_{i=1}^{n} \frac{tt_i}{n}$$

Steps for scheduling are as follows

Step 1:

Master system (VMM) receives information regarding virtual machine from slave (VM-1….n). If the master node capability doesn't catch the data, it will determine the virtual machine to be dead. This study proposed by parameter W.

- If W=0 is set up, it will define the virtual machine to be working and still alive now.
- If W=1 then node is dead.
- If W=2 then node is in previous state.

Step 2: If Master node receives the data from slave, then it gets the information's regarding data (memory used, CPU time etc...)

Step 3: Then Master node builds the weighted table containing the details which is collected from step 2.

Step 4: Then the master node sorts (Round-robin method) all the virtual machines according to their performance. Which is 1≦i≦N.Where N is the number of the virtual machines.

Step 5: The scheduling capability generates the weighted table.

Step 6: The virtual machine control capability receives the weighted table from the Step 5, and distributes the task to the virtual machines according to the weighted value.

V. RESULT

The proposed algorithm and existing round robin algorithm implemented like graphical simulation. Java language is used for implementing VM load balancing algorithm. Assuming the application is deployed in one data centre having virtual machines (with 2048Mb of memory in each VM running on physical processors capable of speeds of 100 MIPS) and Parameter Values are as under Table discuss the Parameter value's which are used for Experiment.

Parameter	Value
Data Center OS	Linux
VM Memory	2048mb
Data Center Architecture	X86
Service Broker Policy	Optimize Response Time
VM Bandwidth	1000

Table 1. Parameter values used for Experiment

These experimental results shows that weighted round robin method improves the performance by consuming less time for scheduling virtual machines.

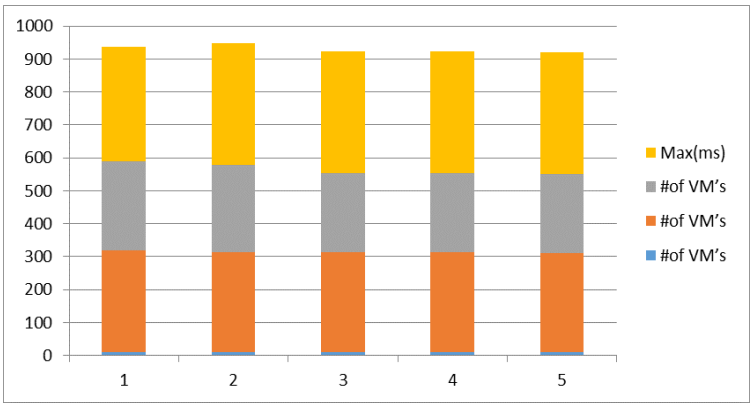


Figure 2. The results based on Round robin algorithm

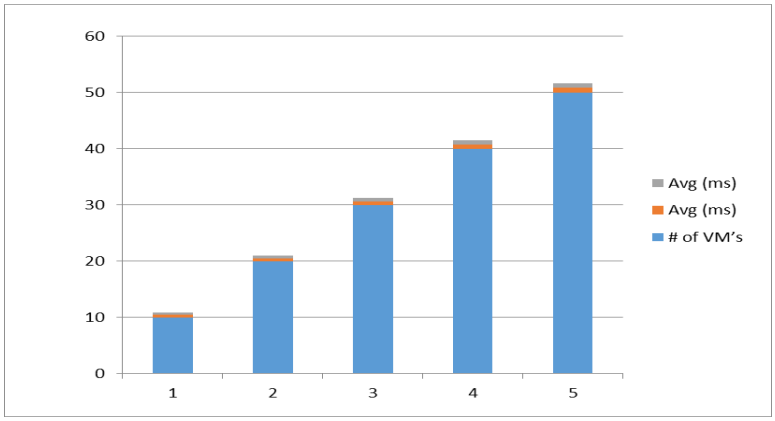


Figure 3. The Data Centre for Round robin algorithm

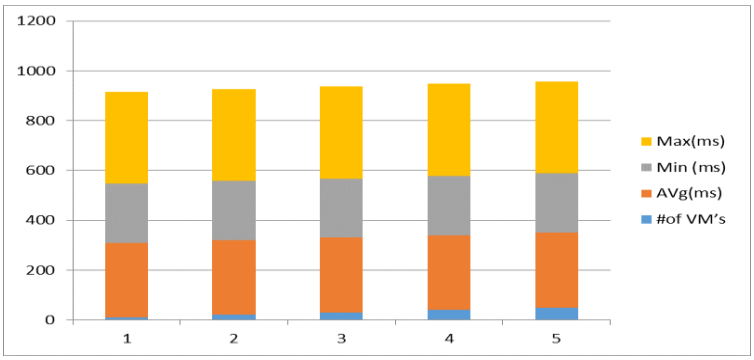


Figure 4. The results based on Weighted Round robin table

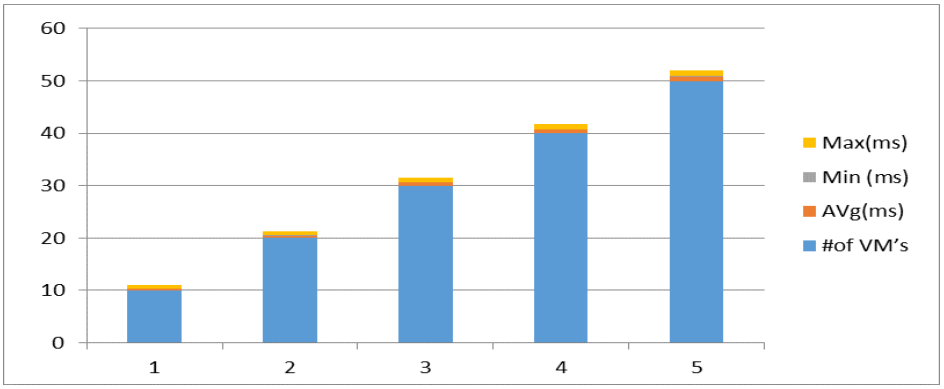


Figure 5. The Data Centre for Weighted Round robin table

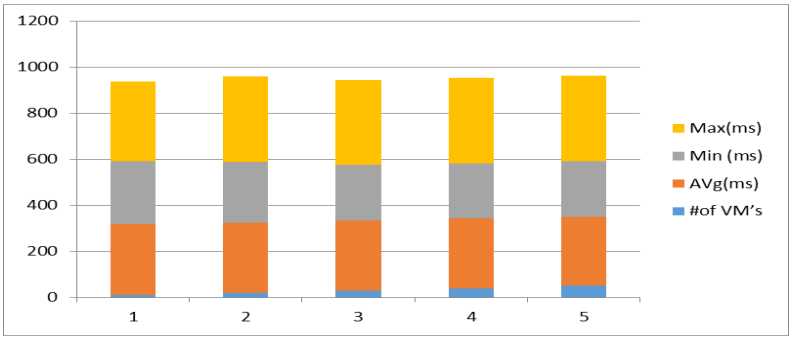


Figure 6. Comparison of results between Round Robin and weighted round robin For Overall response time

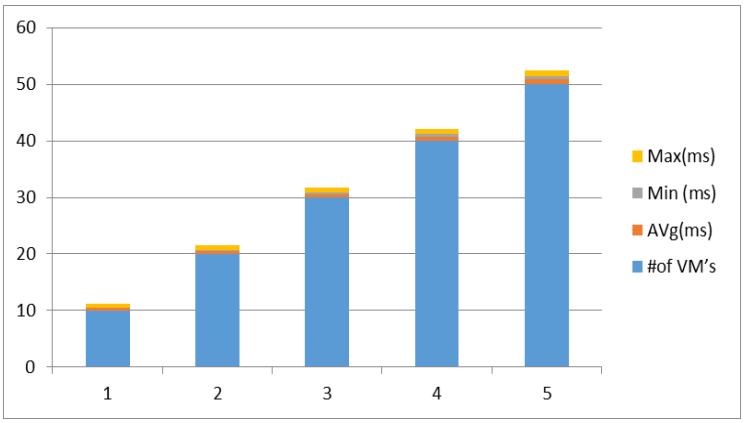


Figure 7. Comparison of results between Round Robin and Weighted Round Robin for Data Center processing time.

VI. CONCLUSION

The above algorithm distributes the process allocation in such a way that process does not concede each other and the waiting state time for process is very much less .as well as all recourses (VMs, memory) are using efficiently. That means the dead lock accruing chances are very much lees. .if the processes may allocate to virtual machine at time. Processes may execute fast and chances of deadlock accruing is less. So we need an algorithm that can describe how process execution can be done on virtual machine fast. A comparative study of round robin architecture shortest round robin and intelligent time slice for round robin architecture is made. It is concluded that the proposed architectures are superior as it has less waiting, response times, usually less preemption and context switching thereby reducing the overhead and saving of memory space. Future work can be based on these architectures modified and implemented for hard real time system where hard deadline systems require partial outputs to prevent catastrophic events.

REFFERENCES:

[1] Anupama Prasanth, "Cloud Computing Services: A Survey*", International Journal of Computer Applications,* Vol. 46, No.3, May 2012, PP.0975 – 8887.

[2] Flavio Lombardi a, RobertoDiPietro, "Secure Virtualization For Cloud Computing", *Secure virtualization for cloud computing,* Journal of Network.

[3] Amazon cloud computing, virtualization and virtual machine, 2002.

[4] Thomas Weigold, Thorsten Kramp and Peter Buhler, *"*ePVM- An Embeddable Process Virtual Machine*"Annual International Computer Software and Applications Conference(COMPSAC 2007),* IEEE.

[5] Rajeev Kumar, Rajiv Ranjan " virtual Machine Scheduling To Avoid Deadlocks", International Journal of Computer Science and Information Technology Research ,Vol.2, Issue 2, pp:(369-372),June,2014.

[6] Robert Blazer, "Process Virtual Machine", *IEEE, 1993,* PP.37-40.

[7] Loris Degioanni, Mario Baldi, Diego Buffa, FulvioRisso, Federico Stirano, GianlucaVarenni, "Network Virtual Machine (NETVM): A New Architecture for Efficient and Portable Packet Processing Applications", 8th *International Conference on Telecommunications,* Zagreb, Croatia, June 15 - 17, 2005,PP.163-168.

[8] DongyaoWu,JunWei,ChushuGao,WenshenDou**, "**A Highly Concurrent Process Virtual Machine Based on Event-driven Process Execution Model"**,** *2012 Ninth IEEE International Conference on e-Business Engineering,*PP.61 – 69.

[9] Vue Hu, YueDongWang, "Process-Level Virtual Machine Embedded Chain", *2011 International Conference on Computer Science and Network Technology,* 1EEE, December 24-26, 2011,PP.302 – 305.

[10] Yosuke Kuno, Kenichi Nii, SaneyasuYamaguchi, "A Study on Performance of Processes in Migrating Virtual Machines", *2011 Tenth International Symposium on Autonomous Decentralized Systems,*IEEE,2011, PP.568-572.

[11] GeetaJangra, PardeepVashist, ArtiDhounchak, " Effective Scheduling in Cloud Computing is a Risk? ",*IJARCSSE,*Volume 3, Issue 8, August 2013,PP.148 – 152.

[12] Ghao G, Liu J, Tang Y, Sun W, Zhang F, Ye X, Tang N (2009) Cloud Computing: A Statistics Aspect of Users. *In:First International Conference on Cloud Computing (CloudCom),* Beijing, China. Heidelberg: Springer Berlin. PP.347-358.

[13] Zhang S, Zhang S, Chen X, Huo X (2010) Cloud Computing Research and Development Trend. *In: Second International Conference on Future Networks (ICFN'10),* Sanya, and Hainan, China. Washington, DC, USA: IEEE Computer Society. PP.93-97

[14] Cloud Security Alliance (2011) Security guidance for critical areas of focus in Cloud ComputingV3.0.Available:https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdfweb cite

[15] MarinosA, Briscoe G (2009) Community Cloud Computing. *In: 1st International Conference on Cloud Computing (CloudCom),* Beijing, China. Heidelberg: Springer-Verlag Berlin.

[16] Khalid A (2010) Cloud Computing: applying issues in Small Business. *International Conference on Signal Acquisition and Processing (ICSAP'10)*PP. 278 – 281.

ISSN (e): 2250 – 3005 || Volume, 05 || Issue, 10 ||October – 2015 ||
International Journal of Computational Engineering Research (IJCER)

Web Content Mining Based on Dom Intersection and Visual Features Concept

Shaikh Phiroj Chhaware[1], Dr.Mohammad Atique[2], Dr. Latesh. G. Malik[3]
[1] esearch Scholar, G.H. Raisoni College of Engineering, Nagpur 440019 (MS),
[2] Associate Professor, Dept. of Computer Science & Engineering,
S.G.B. Amravati University, Amravati (MS)
[3] Professor, Department of Computer Science & Engineering G.H. Raisoni College of Engineering,
Nagpur 440019 (MS)

ABSTRACT
Structured Data extraction from deep Web pages is a challenging task due to the underlying complex structures of such pages. Also website developer generally follows different web page design technique. Data extraction from webpage is highly useful to build our own database from number applications. A large number of techniques have been proposed to address this problem, but all of them have inherent limitations because they present different limitations and constraints for extracting data from such webpages. This paper presents two different approaches to get structured data extraction. The first approach is non-generic solution which is based on template detection using intersection of Document Object Model Tree of various webpages from the same website. This approach is giving better result in terms of efficiency and accurately locating the main data at the particular webpage. The second approach is based on partial tree alignment mechanism based on using important visual features such as length, size, and position of web table available on the webpages. This approach is a generic solution as it does not depend on one particular website and its webpage template. It is perfectly locating the multiple data regions, data records and data items within a given web page. We have compared our work's result with existing mechanism and found our result much better for number webpage.

Keywords: *Document Object Model, Web Data Extraction, Visual Features, Template Detection, Webpage Intersection, Data Regions, Data Records.*

I. Introduction

The web contains the large amount of structured data and served as a good interface for databases over the Internet. A large amount of web content is generated from web databases in response to the user queries. Often the retrieved information which is a user query results are enwrapped in a dynamically generated web page in the form of data records. Such special web pages are called as deep web page and online databases are treated as deep web databases. Various web databases have reached 25 million according to a recent survey [1].

The data records and data items which are available on these deep web pages need to be convert in machine processable form. This machine processable data is required in many post data processing applications such as opinion mining, deep web crawling, meta-searching. And hence the structured data needs to be extracted efficiently considering the enormous amount data available on internet which is very rapidly growing day by day. Templates of the deep web pages are generally similar and spread over the other web pages of the same website [2].

Many works can be found in the literature for deep web structured data extraction and mostly they are relying on the web page segmentation either visual clues segmentation or DOM tree based web page segmentation [3]. But in all such techniques it is observed that they require much machine time to process the single web page as it is web browser dependent for gathering necessary visual information for the various objects displayed on the web page [4]. World Wide Web has millions searchable information sources. The retrieved information is enwrapped in web pages in the form of data records. These special Web pages are generated dynamically and are hard to index by traditional crawler-based search engines, such as Google and Yahoo.

Figure 1 shows such deep web pages from www.yahoo.com. In this example figure we can see that the actual interesting information contained within the page occupies only 1/3 of the entire web page while other information which are generally treated as unwanted information or simple "noise" information occupies rest of the web page i.e about 2/3 of web page space. This noise information generally possesses advertisement of product, navigation link, hyperlink, contact information, web page menu bar and items, links to other web pages of the web sites etc. These types of noisy information are very harmful for the data extraction process. They generally hamper the efficiency of web data extraction algorithm employed because

Figure 1. An example deep Web page from Yahoo.com

algorithm needs to be processed this information also. So actual efficiency of algorithm cannot be achieved and if we are trying to process very large numbers of web pages for data extraction, noisy information consume too much of the processing time.

This webpage contains information regarding books which are presented in the form of data records and each data record has some data items such as author, title etc. To make the data records and data items in them machine processable, which is needed in many applications such as deep Web crawling and meta searching, the structured data need to be extracted from the deep Web pages. Extraction of web data is a critical task this data can be very useful in various scientific tools and in a wide range of application domains.

II. Literature Review

Many works has been carried out for the data extraction from the deep web pages. The basis for categorizing them are generally based how much amount of human efforts and interference required for data extraction process. These categories are as follows:

This paper present the actual work carried out to target the problem of structured data extraction from the deep web pages. This paper is organized as follows. Part II contains the literature review of existing system and methodologies used to solve this problem. Part III focuses first approach that we have used that is based on webpage template detection mechanism based on DOM intersection method. Part IV gives the details of second approach which is generic mechanism to carry out the task of structured data extraction based on DOM tree parsing and using visual features to locate the main data on the page. The part V gives the experimental setup for the methods, result obtained and comparisons of result with existing system. Part VI presents conclusions and future works.

A. Supervised Approach

The idea present in paper [5] is one such approach where a manual intervention is required during data extraction from deep web pages. Other system is MINERVA which is again utilizes the same mechanism for this problem

B. Semi-Supervised Approach

Paper [6] and paper [7] has built up the system for data extraction and they are employing the semi-supervised approach for it. At some of data extraction process the user intervention is required specially during actual data population to the local database while rest of the part extraction process is automatic.

C. Unsupervised Approach

This is a novel approach for web data extraction. Now a day all the research focus in this area is diverted on this kind of mechanism. [1] gives such a kind of idea which is based on visual features. But it also has severe limitation as it does not mine the data from multiple data regions. Also the process given for the data extraction is very much complex as multiple visual are needs to be examined.

III. Data Extraction Based On Dom Intersection Method

This section presents a novel approach for website template detection based on intersection of document object model (DOM) tree of a test page with the another web page from the same website. The result of this mechanism is stored in a table which has common nodes in the DOM tree. Then for the next web page processing, the node information from this table is applied on the incoming web page's DOM tree instead of DOM tree processing for that web page. The table information then can be very much useful for forming the rules for automatic data extraction I,e for automatic wrapper generation method.

A. The Proposed Framework

Our website template detection method is based on the webpage DOM tree intersection concept. Here we consider the most common intersection node information of the two web pages from the same website.

Whenever any new web page is available, it passes though the four steps:

1) DOM Tree building
2) Template Detection
3) Template Intersection
4) Wrapper Table Generation.

For the next web page from the same website only steps 1 and 3 are required. Wrapper table will get updated automatically whenever there is a change in DOM tree of the both web pages. Here the changes in the DOM tree of first web pages is learn. The step 4 is essential for wrapper generation.

B. Document Object Model Tree Building

This is the first step towards actual data extraction. Here we build a DOM tree or tag tree of a HTML web page.

- Most HTML tags works in pairs. The nesting of the tag-pair can be possible to have the parent child relationship among other nodes of the tree.
- The DOM tree is build up from the HTML code of a web page.

The DOM tree node presents the pair of tags and each pair of tags can be nested within which children node may be available.

C. Template Detection

The web page templates are just fragments of the HTML code included in the collection of HTML documents. Automatic generation of templates is generally done and they are replicated to other web page development. This ensures uniformity and speedy development of the web site. For this purpose, we are using simple tree matching algorithm which is given below:

Algorithm: SimpleTreeMatching(A, B)

Input: DOM Tree A and B

1. If the roots of the two trees A and B contain distinct symbols then return 0.
2. Else m = the number of first-level subtrees of A;
n = the number of first-level subtrees of B;
Initialization M[i,0] = 0 for i = 0, . . ., m;
M[0,j] = 0 for j = 0, . . ., n;
3. For i = 1 to m do
4. For j = 1 to n do
5. M[i,j] = max(M(i,j-1),M(i-1, j),M(i-1,j-1)+W(i,j)
where W(i,j) = SimpleTreeMatching(A_i,B_j)
6. Return (M(m,n)+1)

The working of the Simple Tree matching algorithm is illustrated by means the figure 2. It takes two labels ordered rooted tree and map the nodes in each tree. At a glance, a generalized mapping can be possible between same nodes among the two different trees. Replacements of nodes are also allowed, e.g., node C in tree A and node G in tree B. Simple Tree Matching algorithm is top-down algorithm and it evaluated the similarity between two trees by producing the maximum matching.

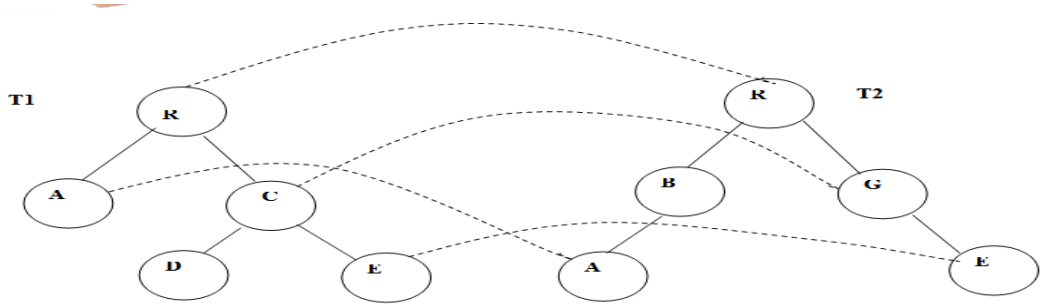


Figure 2. Matching between two labels ordered rooted trees

D. Template Intersection

The template intersection phase is applied when there is new DOM tree appears for the processing. This phase simply eliminates non-matching nodes of first tree with another one and hence here we get the appropriate template of that deep web page. This allows the extraction of main content from the deep web page at later stage.

E. Wrapper Table Generation

The wrapper table gives the information about which nodes are common in both trees. The first column presents the tree, e.g., T1 and T2 and further columns presents the node tag. As shown in figure 1, tree T1 has nodes R, A, C, D, and E. So in the wrapper table numeric number 1 will be marked in T1 row which presents that these nodes are the part of tree T1. Such kind of processing is done for all appearing tree. The wrapper table is very for automatic data extraction from other pages of the website. We can form wrapper rules from this table.

IV. Data Extraction Based On Dom Tree Parsing & Visual Features

This is our second approach which is a generic solution for extracting main structured data from deep webpages. This uses DOM Tree parsing process and some visual features of the table such area, size and location.

This approach uses HTML Agility Pack parser to process the DOM tree. This parser parses the document in the form of DOM tree. DOM tree is used to traverse the web document by using XPath method this method is traverse the web document in two ways i.e. from root node to leaf node and vice versa. The following figure 3 shows the proposed system. This shows how to extract the main data from deep web pages and how we take query from user and what is the use of HTML Agility Pack for parsing the document. It shows the extracted data is stored in database and it displays the extracted data on the user's window.

As we know the web page is consists of group of blocks means each block has specific height and specific width. To get the main data which is available in the page we have find out area of each block which web page has and the find the largest block from the page that has the main data which we required to extract. Here we used algorithm to calculate the largest block. This algorithm is applied on the DOM tree and it searches for the largest table on the web page by comparing all the tables of the same web page. This is iterative procedure and once the largest table is found, it is send to the local database.

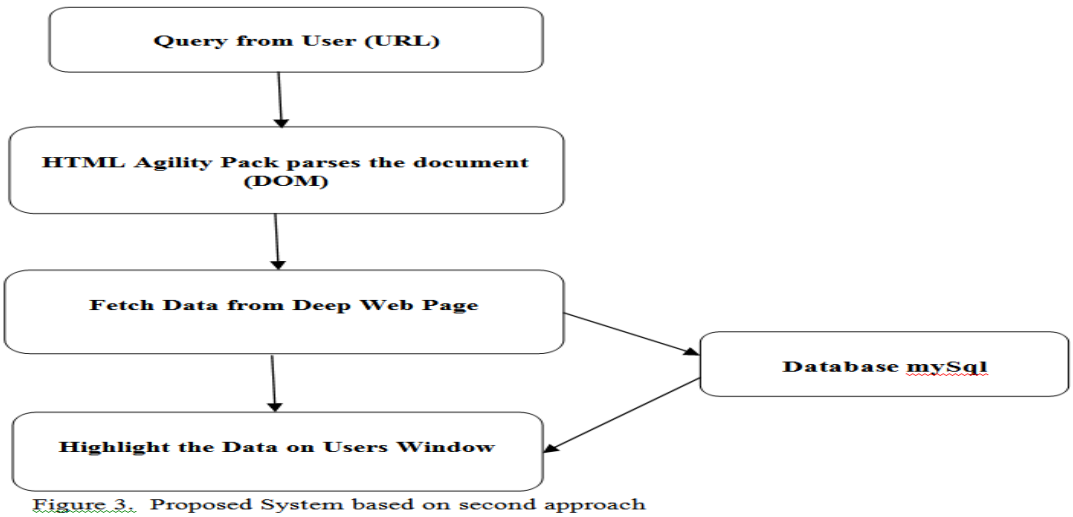


Figure 3. Proposed System based on second approach

Algorithm: Find the largest block of web page

1. Get the width of each block
2. Get the height of the each block
3. Calculate the Area of each block
4. MAXArea = 0
5. Area = Height * Width
6. Repeat step 6 compare MAXArea< Area
7. then MAxArea= Area
8. goto step 5
9. Displays the contains of MAXArea
   End.

The Deep web is normally defined as the content on the Web not accessible through a search on general search engines. This content is sometimes also considered to as the hidden or invisible web to extract. The Web is a complex that contains information from a variety of sources and includes an evolving mix of different file types and media files. It is much more than static, self-contained Web pages. Above algorithm we calculate the area for each execution and find the largest block from the web document. Suppose we execute this query for static web page sometimes it does not work because static page might have or not block wise allocation of the web page but it works for dynamic web page properly means for deep web page.

V. Experimental Setup and Result Comparisons

For the experimentation we have gathered the input web pages from various commercial websites such as www.dell.co.in, www.amozon.in, www.shopclues.com, www.snapdeal.com, www.yahoo.com, www.rediff.com etc.

Our first approach based on the template detection of DOM intersection of various web pages from same website, for this we have used multiple webpages of the same website and try to extract the data from it.

The second approach is having better result as it present the generic solution. The detail experimental setup of this approach is given as below.

A. System Requirement

The experimental results of our proposed method for vision-based deep web data extraction for deep web document are presented in this section. The proposed approach has been implemented in VB.NET. It usually ships in two types, either by itself or as part of Microsoft Visual Studio .NET. To use the lessons on this site, you must have installed either Microsoft Visual Basic .NET 2003 or Microsoft Visual Studio .NET 2003. All instructions on this site will be based on an installation of Microsoft Visual Studio .NET. From now on, unless specified otherwise, we will use the expressions "Microsoft Visual Basic" or "Visual Basic" to refer to Microsoft Visual Basic .NET 2003. Memory requirement is minimum 1GB. If we want to refer to another version, we will state it. WAMP Server for the execution of MY-SQL, HTML AGILITY PACK. It is an open source technology so it easily available and free of cost. It is used for parsing web document in DOM. The Html Agility Pack is a free, open-source library that parses an HTML document and constructs a Document Object Model (DOM) that can be traversed manually or by using XPath expressions. (To use the Html Agility Pack you must be using ASP.NET version 3.5 or later.) In a nutshell, the Html Agility Pack makes it easy to examine an HTML document for particular content, and to extract or modify that markup. Wamp Server is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. These pages are used in proposed method for removing the different noise of deep web page. The removal of noise blocks and extracting of useful content chunks are explained in this section.

B. User Window

In this system use plays very important because user request for data extraction and the data is available anywhere in the centralized database. Then user enters URL and then presses the button extract clues, and then the data is extracted as per requirement of user. For process execution following procedure is followed. Initially user opens browser for data extraction. Figure 4 shows the browser for data extraction. It has two parts first part contains input means normal web page which is available anywhere in World Wide Web and second part is output window which contains the extracted data on the basis of largest area of web page block and hide the remaining part of the web page.

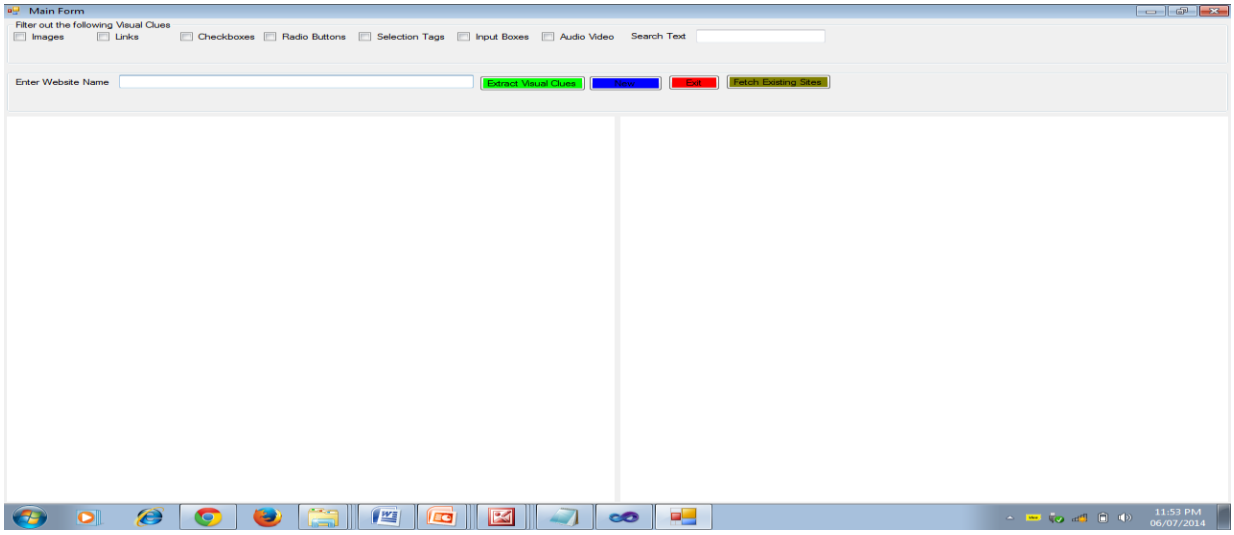


Figure 4. User Window

C. Extracted Data Window

The figure 5 shows the execution of the process in this image user enters the URL and asks for data extraction.

D. Data Fetch from Database

Sometimes user wants to visit that data which is already extracted by someone else that data is available in database (MySql). For every execution the extracted data is stored in database in tabular format. Here we use the technique to extract the largest block (area) of the web page means to calculating the area of every block of the web page. That extracted block has specific height and width with help of this parameter we calculate the area of every block. Following window shows that how user access the data from database. Here user simply clicks on the Fetch Data button and then the list will be opened user needs to select the required site and then click on the OK button. Then the required dataa is displayed on the output window. Figure 6 shows the interface for the data from database.

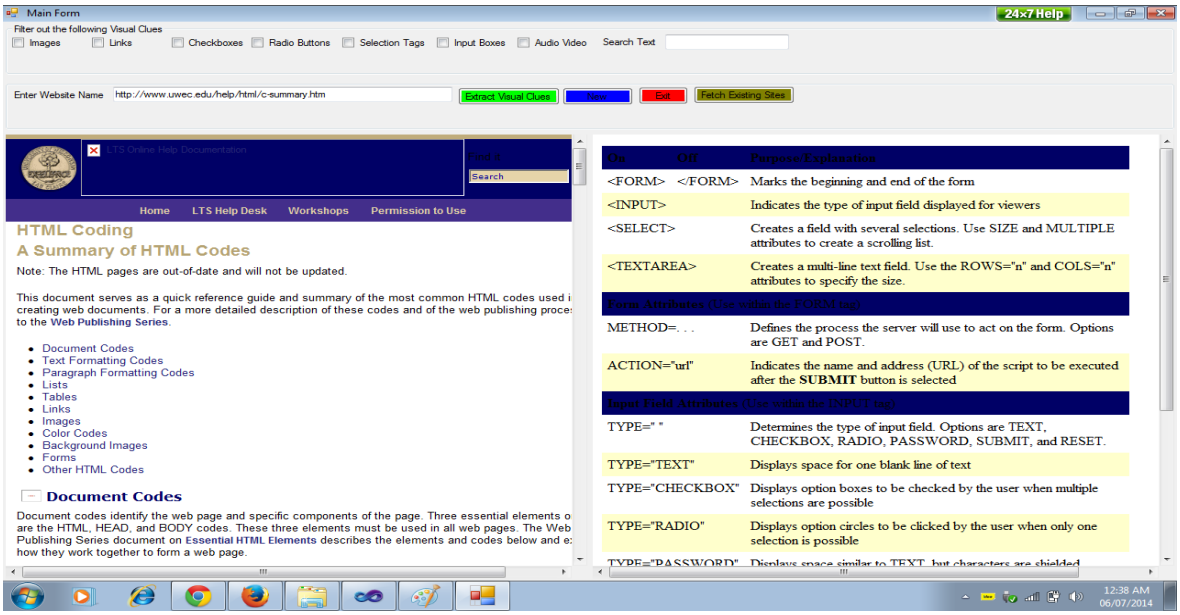


Figure 5. Extracted Data Window

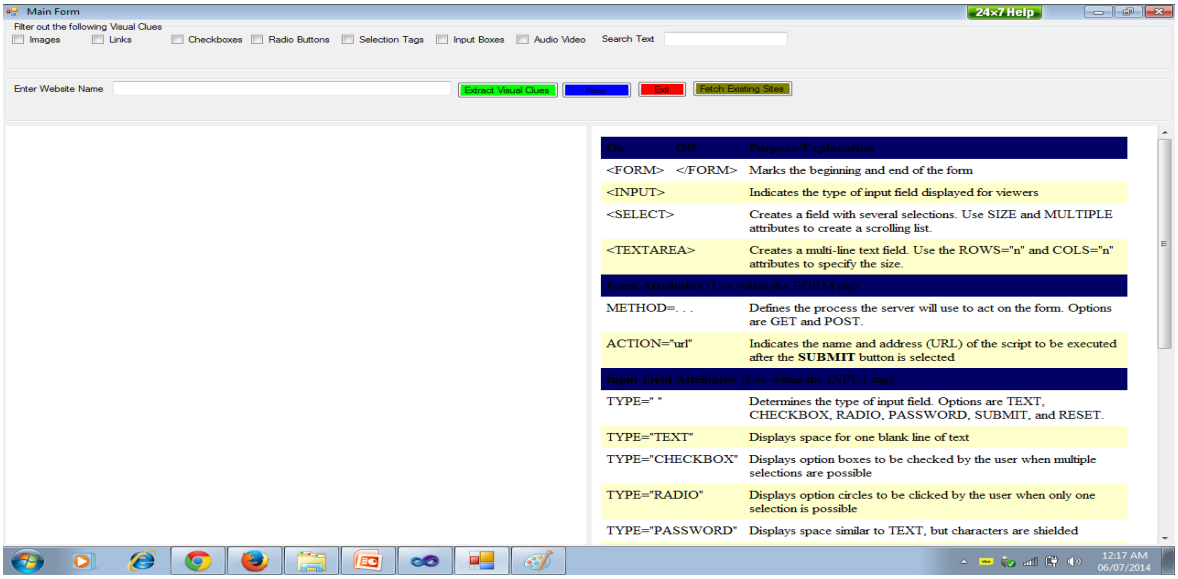


Figure 6. Fetching Data from Database

E. Database

For every execution the extracted data is stored in the database in the form of tables for future use. Suppose user wants to fetch the data from database which is available in the database then user just click on the fetch existing site button then data automatically fetched from the database as per requirement. The figure 7 shows the sample of database where our data is stored. In our approach we have extracted some deep web page data, extracted web page URL is shown inn following table. Table contains ID and URL here ID is allotted for every extracted web document.

F. Result Evaluation

Our main approach is to extract the data from deep web pages and remove all unwanted noise from the web page then find out the result in the form of extracted data and it is to be calculated by using following expressions.

Precision is the percentage of the relevant data records identified from the web page.

Precision = DRc / DRe

Recall defines the correctness of the data records identified.

Recall = DRc / DRr

Where, DRc is the total number of successfully extracted data records sample web page is subjected to the proposed approach to identify the relevant web data region. Data region having description of some products is extracted by our data extraction method after removing the noises. The filtered data region, DRe is the total number of data records on the page. DRr is the total number of data records extracted.

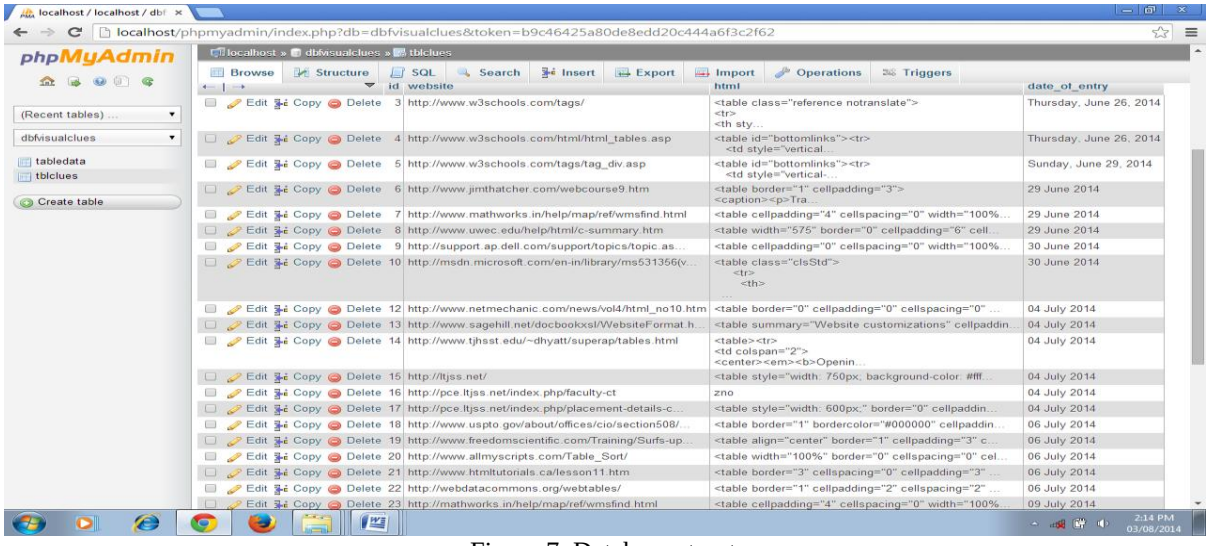


Figure 7. Database structure

Table I presents the overall result of the above two system presented in this paper and their performances are compared against the existing systems like ViDe and MDR against the parameters like recall and precision.

Table I

Result Evaluation and Comparisons

Approach	Total No. webpages	Recall	Precision
First Approach	78	96.7%	98.8%
Second Approach	121	92.8%	91.5%
ViDE	133	95.6%	93.4%
MDR	118	82.3%	78.6%

VI. Conclusion and Future Works

The desired information is embedded in the deep Web pages in the form of data records returned by Web databases when they respond to users' queries. In this paper, we have given two different approaches for the problem main data extraction from the deep web pages in structured form. The first approach is presenting a non-generic solution and is totally based on the specific website's webpage template. The second approach is generic and uses minimal number of visual features and hence the efficiency of the system enhances. Also this system is able to mine the data from the multiple data regions because it is mining the data in structured format.

There are numbers of ways available where we can improve the performance of the systems and eliminate the constraints that we have considered. There are several issues which yet to be address like in perfect data mapping with local database attributes and web data table attributes. This work eventually may lead to the use of natural language processing and text mining approach at advance level.

References

[1] Wei Liu, Xiaofeng Meng, Weiyi Meng,"ViDE: A Vision-Based Approach for Deep Web Data Extraction", *IEEE Transactions on Knowledge and Data Engineering*, vol.22, no.3, pp.447-460, 2010.

[2] Yossef, Z. B. and Rajgopalan, S., "Template Detection via Data Mining and its Applications", *Proceedings of the 11th international conference on World Wide Web*, pp. 580-591, 2002

[3] Ma, L., Goharian, N., Chowdhury, A., and Chung M., "Extracting Unstructured Data from Template Generated Web Document", *Proceedings of the 12th international conference on Information and /knowledge Management*, pp. 512-515, 2003.

[4] Chakrabarti, D., Kumar, R., and Punera, K., "Page Level Template Detection via Isotonic Smoothing", *Proceedings of the 16th international conference on World Wide Web*, pp. 61-70, 2007.

[5] G.O. Arocena and A.O. Mendelzon, "WebOQL: Restructuring Documents, Databases, and Webs", *Proc. Int'l Conf. Data Eng (ICDE)*, pp. 24-33, 1998.

[6] L. Liu, C. Pu, and W. Han, "XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources," *Proc. Int'l Conf. Data Eng. (ICDE)*, pp. 611-621, 2000.

[7] J. Hammer, J. McHugh, and H. Garcia-Molina, "Semistructured Data: The TSIMMIS Experience," *Proc. East-European Workshop Advances in Databases and Information Systems (ADBIS)*, pp. 1-8, 1997.

[8] Yi, L., Liu, B., and Li, X., "Eliminating Noisy Information in Web Pages for Data Mining.", *Proceedings of the 9th ACM SIGKDDInternational Conference on Knowledge Discovery and Data Mining*, pp. 296-305, 2003

[9] D. Cai, S. Yu, J. Wen, and W. Ma, "Extracting Content Structure for Web Pages Based on Visual Representation," *Proc. Asia Pacific Web* Conf. (APWeb), pp. 406-417, 2003.

[10] Ashraf, F.; Ozyer, T.; Alhajj, R., "Employing Clustering Techniques for Automatic Information Extraction from HTML Documents", *IEEE* Transactions on Systems, Man, and Cybernetics, Part C: Applications *and reviews*, vol.38, no.5, pp.660-673, 2008.

[11] Sandip Debnath,Prasenjit Mitra,C. Lee Giles, "Automatic Extraction of Informative Blocks from WebPages", *In Proceedings of the ACM* symposium on Applied computing, Santa Fe, New Mexico, pp. 1722 – 1726,2005.

[12] J. Madhavan, S.R. Jeffery, S. Cohen, X.L. Dong, D. Ko, C. Yu, and A. Halevy, "Web-Scale Data Integration: You Can Only Afford to Pay As ou Go,", Proc. Conf. Innovative Data Systems Research (CIDR), pp. 342-350, 2007.

ISSN (e): 2250 – 3005 || Volume, 05 || Issue, 10 ||October – 2015 ||
International Journal of Computational Engineering Research (IJCER)

Sub-Graph Finding Information over Nebula Networks

K.Eswara Rao[$1] , A.NagaBhushana Rao[$2]
[$1,$2] Asst. Professor, Dept. Of CSE, AITAM, Tekkali, SKLM, AP, INDIA.

ABSTRACT

Social and information networks have been extensively studied over years. This paper studies a new query on sub graph search on heterogeneous networks. Given an uncertain network of N objects, where each object is associated with a network to an underlying critical problem of discovering, top-k sub graphs of entities with rare and surprising associations returns k objects such that the expected matching sub graph queries efficiently involves, Compute all matching sub graphs which satisfy "Nebula computing requests" and this query is useful in ranking such results based on the rarity and the interestingness of the associations among nebula requests in the sub graphs. "In evaluating Top k-selection queries, "we compute information nebula using a global structural context similarity, and our similarity measure is independent of connection sub graphs". We need to compute the previous work on the matching problem can be harnessed for expected best for a naive ranking after matching for large graphs. Top k-selection sets and search for the optimal selection set with the large graphs; sub graphs may have enormous number of matches. In this paper, we identify several important properties of top-k selection queries, We propose novel top–K mechanisms to exploit these indexes for answering interesting sub graph queries efficiently.

Key words: *Nebula Networks, Top-k sub graph, Indexing*

I. INTRODUCTION

With the ever-increasing popularity of entity-centric applications, it becomes very important to study the interactions between entities, which are captured using edges in the entity relationship (or information) nebula networks. Entity-relationship networks with multiple types of entities are usually referred to as heterogeneous information networks. For example, bibliographic networks capture associations like 'Identification of fingerprints for the serine protease Family'. Similarly, social networks, biological protein-enzyme classification Using SVM ((support vector machine), Wikipedia entity network, etc. also capture a variety of rich associations. In these applications, it is critical to detect novel connections or associations among objects based on some subgraph queries. Two example problems are shown in Figures1 and are described as follows.

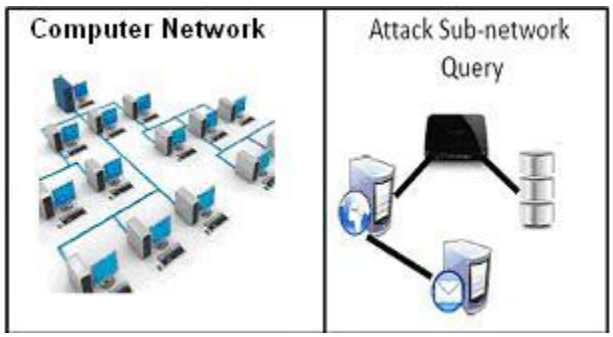


Fig.1 Attack Localization Problem

P1: query Selection over nebula: Organization Nebula networks consist of two subgraphs and object nodes where two graphs are connected if they have worked together on a successful mission in the past, and a subgraph is linked to an nebula if the person has a known expertise in using that subgraph. For example, US army network which consists of 2-3M nebulas and much more linked objects. A manager in such an organization may have a mission which can be defined by a query graph of objects and networks. For example, the right half of Figure 1 shows an organization network while the left half of the figure shows a sample mission

query graph consisting of two subgraphs. The network has edges with weights such that a high weight implies higher compatibility between the nodes connected by the edge. The manager is interested in selecting a subgraph to accomplish the mission with the network to network compatibilities as specified in the mission query. Using the historical compatibility based organization network, how can we find the best query (selection) over the nebula?

P2: Attack Localization: Consider a computer network as shown in the left part of Figure 2. It can consist of a large number of components like database servers, hubs, switches, desktops, routers, VOIP phones, etc. Consider a simple attack on multiple web servers in such a network where the attack script runs on a compromised web server. The script reads a lot of data from the database server (through the network hub) and sends out spam emails through the email server. Such an attack leads to an increase in data transfer rate along the connections in multiple "attack sub-networks" of the form as shown in the right part of the figure. Many such attacks follow the same pattern of increase in data transfer rates. How can a network administrator localize such attacks quickly and find the worst affected sub-networks given the observed data transfer rates across all the links in the network?

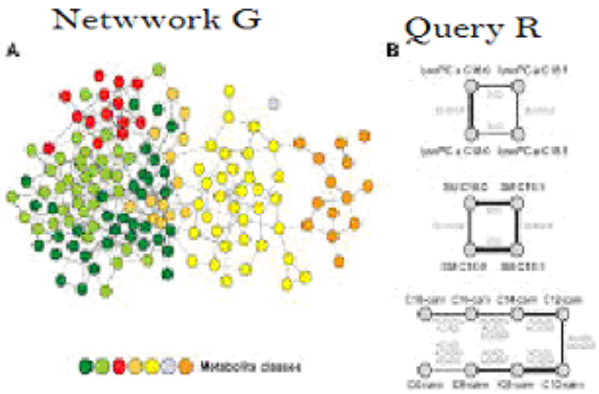


Fig.2 Example of a Network ***G*** and a Query ***Q***

Both of these problems share a common underlying problem: Given a heterogeneous network G, a heterogeneous subgraph query Q, and an edge interestingness measure I which defines the edge weight, find the top-K matching subgraphs S with the highest interestingness. The two problems can be expressed in terms of the underlying problem as follows. P1: G = organization network, Q = mission query, I = historical compatibility, S = team. P2: G = computer network, Q = an "attack sub-network" query, I = data transfer rate, S = critical sub-networks. Besides the two tasks, this proposed problem finds numerous other applications. For example, the interesting subgraph matches can be useful in network bottleneck discovery based on link bandwidth on computer networks, suspicious relationship discovery in social networks, de-noising the data by identifying noisy associations in data integration systems, etc.

Comparison with Previous Work

The proposed problem falls into the category of the subgraph matching problems. Subgraph matching has been studied in the graph query processing literature with respect to approximate matches [4], [25], [26], [30] and exact matches [18], [27], [31]. Subgraph match queries have also been proposed for RDF graphs [15], probabilistic graphs [24] and temporal graphs [1]. The proposed problem can be solved by first finding all matches for the query using the existing graph matching methods and then ranking the matches. The cost of exhaustive enumeration for all the matches can be prohibitive for large graphs. Hence, this paper proposes a more efficient solution to the top-K subgraph matching problem which exploits novel graph indexes. Many different forms of top-K queries on graphs have been studied in the literature [5], [21], [23], [28], [30]. Gou et al. [5] solve the problem only for twig queries while we solve the problem for general subgraphs. Yan et al. [21] deal with the problem of finding top-K highest aggregate values over their h-hop neighbors, in which no subgraph queries are involved. Zhu et al. [28] aim at finding top-K largest frequent patterns from a graph, which does not involve a subgraph query either. Different from existing top-K work, the proposed work deals with a novel definition of top-K general subgraph match queries, which have a large number of practical applications as discussed above.

Brief Overview of Top-K Interesting Subgraph Discovery
Given a heterogeneous network containing entities of various types, and a subgraph query, the aim is to find the top-K matching subgraphs from the network. We study the following two aspects of this problem in a tightly integrated.
way: (1) computing all possible matching subgraphs from the network, and (2) computing interestingness score for each match by aggregating the weights of each of its edges. To solve these problems, we present an efficient solution which exploits two low-cost index structures (a graph topology index and a maximum metapath weight (MMW) index) to perform top-K ranking while matching (RWM). Multiple applications of the top-K heuristic, a smart ordering of edges for query processing, quick pruning of the edge lists using the topology index and the computation of tight upper bound scores using the MMW index contribute to the efficiency of the proposed solution in answering the top-K *interesting subgraph* queries.

Summary
We make the following contributions in this paper. We propose the problem of top-K interesting subgraph discovery in information networks given a heterogeneous edge-weighted network and a heterogeneous unweighted query. To solve this problem, we propose two low-cost indexes (a graph topology index and a maximum metapath weight (MMW) index) which summarize the network topology and provide an upper bound on maximum metapath weights separately. Using these indexes, we provide a ranking while Matching (RWM) algorithm with multiple applications of the top-K heuristic to answer *interesting subgraph* queries on large graphs efficiently. Using extensive experiments on several synthetic datasets, we compare the efficiency of the proposed RWM methodology with the simple ranking after matching (RAM) baseline. We also show effectiveness of RWM on two real datasets with detailed analysis. Our paper is organized as follows. In Section II, we define the *top*-K *interesting subgraph discovery* problem. The proposed approach consists of two phases: an offline index construction phase and an online query processing phase which are detailed in Sections III and IV respectively. In Section V, we discuss various general scenarios in which the proposed approach can be applied. We present results with detailed insights on several synthetic and real datasets in Section VI. We discuss related work and summarize the paper in Sections VII and VIII respectively.

II. PROBLEM DEFINITION

In this section, we formalize the problem definition and present an overview of the proposed system. We start with an introduction to some preliminary concepts.
Definition 1 (A Heterogeneous Network). A heterogeneous network is an undirected graph
$G = < V_G, E_G, type_G, weight\ G >$ where V_G is a finite set of vertices (representing entities) and E_G is a finite set of edges each being an unordered pair of distinct vertices. $type_G$ is a function defined on the vertex set as $type_G : V_G \rightarrow T_G$ where T_G is the set of entity types and $|T_G| = T$. weight G is a function defined on the edge set as $weight_G : E_G \rightarrow R \in [0, 1]$. $Weight_G(e)$ represents the interestingness measure value associated with the edge e.

For example, Figure 3 shows a network G with three types of nodes. $T_G = \{A,B,C\}$. $|V_G|=13$, and $|E_G|=18$.

Definition 2 (Subgraph Query on a Network). *A subgraph query Q on a network G is a graph consisting of node set V_Q and edge set E_Q. Each node could be of any type from T_G.*
For example, Figure 2 shows a query Q with four nodes. $|V_Q|=4$, and $|E_Q|=3$. The network G and the query Q shown in Figure 2 will be used as a running example throughout this paper.

Definition 3 (Subgraph Isomorphism). A graph $g = <V_g, E_g, type_g >$ is subgraph isomorphic to another graph $g' = <V_{g'}, E_{g'}, type_{g'} >$ if there exists a subgraph isomorphism from g to g′. A subgraph isomorphism is an injective function $M : V_g \rightarrow V_{g'}$ such that (1) $\forall v \in Vg$, $M(v) \in V_{g'}$ and $type_g(v)=type_{g'}(M(v))$, (2) $\forall e =(u, v) \in E_g$, $e' =(M(u), M(v)) \in E_{g'}$.
Definition 4 (Match). The query graph Q can be subgraph isomorphic to multiple subgraphs of G. Each such subgraph of G is called a match or a matching subgraph of G. The query Q can be answered by returning all exact matching subgraphs from G. For example, the subgraph of G induced by vertices (8, 9, 5, 6) is a match for the query Q on network G shown in Figure 2. For sake of brevity, we will use the vertex set (tuple notation) to refer to the subgraph induced by the vertex set.

Definition 5 (Interestingness Score). The interestingness score for a match M for a query Q in a graph G is defined as the sum of its edge weights. For example, the interestingness score for the occurrence {8,9, 5, 6} is 2.1. Though we use sum as an aggregation function here, any other monotonic aggregation function could also be used.

Definition 6 (Top–*K* Interesting Subgraph Discovery Problem).
Given*: A heterogeneous information network* G*, a heterogeneous unweighted query* Q*, and an edge interestingness measure.*
Find*: Top-*K *matching subgraphs with highest interestingness scores.*

For example, (3, 4, 5, 6) and (4, 3, 2, 7) are the Top two matching sub-graphs both with the score 2.2 for the query *Q* on network *G* in Figure 2.
For example, the metapath corresponding to the path (5, 4,7) is (A,A,B). There are TD distinct metapaths of length D where T is the number of types. Each column of a topology index corresponds to a metapath.

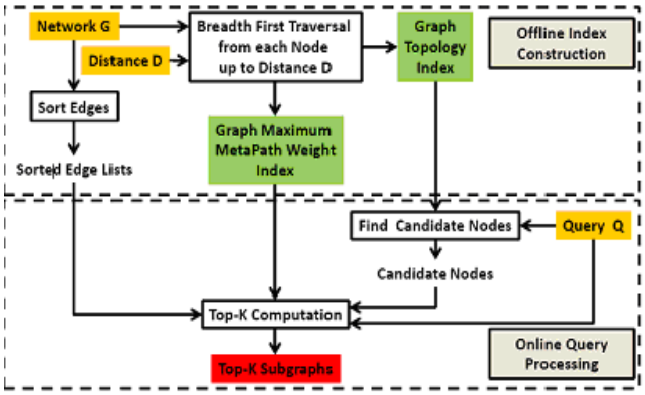


Fig-3 Top-*K* interesting sub graph Discovery System Diagram

Figure 2 shows the graph topology index for the first 4 nodes of the graph shown in Figure 2. For example, for node 2, there are two 2-hop neighbors of type A (4 and 8) reachable via the metapath (B,A). Hence the entry topology(2, (B,A))=2. A blank entry in the index indicates that there is no node of type t at a distance d from node n along the corresponding metapath. As we shall see in Section IV-A, the topology index plays a crucial role in reducing the search space by pruning away candidate graph nodes that cannot be instantiated for a given query node.

III. *TOP-K* INTERESTING SUBGRAPH QUERY PROCESSING

Given a query Q with node set VQ and edge set EQ, top-K matching subgraphs are discovered by traversing the sorted edge lists in the top to bottom order with the following speedup heuristics. First for each node in V_Q, a set of nodes from the graph that could be potential candidates for the query node, is identified using the topology index (Algorithm 1). The edges in the sorted edge lists that contain nodes other than the potential candidate nodes are marked as invalid. This prunes away many edges and speeds up the edge list traversal. The query Q is then processed using these edge lists in a way similar to the *top-K* join query processing (Section IV-B) adapted significantly to handle network queries. The approach discussed in Section IV-B is further made faster by the tighter upper bound scores computed using the MMW index (Algorithm 2). We will discuss these in detail in this section.

Algorithm 1 Candidate Node Filtering Algorithm

Input: (1) Query Node *q*, (2) Graph Node *p*, (3) *topology* [p], (4) *queryTopology*[q], (5) Index Parameter *D*
Output: Is *p* a potential candidate node for query node *q*?
1: **for** $d = 1 \ldots D$ **do**
2: **for** $mp = 1 \ldots T^d$ **do**
3: if *queryTopology*[*q*][*d*][*mp*] > *topology* [*p*][*d*][*mp*] **then**
4: Return False
5: Return True

Candidate Node Filtering Algorithm

The proposed candidate node filtering approach is summarized in Algorithm 1. For each distance value d, all possible metapaths of length d are checked. By comparing the topology for all metapaths with the corresponding query Topology values (Step 3), it can be inferred whether the candidate p is valid to be an instantiation of

query node q for some match. The time complexity is O(DTD+1). Candidate pruning leads to shortening of the edge lists associated with any of the query edges. For example, nodes 2, 8 and 10 get pruned for the query node Q2. Thus, the edge list corresponding to the query edge (Q2,Q3) will have the following AA edges marked as invalid: (2,3), (8,9) and (10,9).

B. Top-K Match Computation

In this sub-section, we describe the top-K algorithm to perform interestingness scoring and matching simultaneously. The algorithm is based on the following key idea. A top-K heap is maintained which stores the best K answers seen so far. The sorted edge lists are traversed from top to bottom. Each time an edge with maximum edge weight from any of the lists is picked and all possible size-1 matches in which that edge can occur are computed. Candidate size-1 matches are grown one edge at a time till they grow to the size of the query. Partially grown candidate matches can be discarded if the upper bound score of these matches falls below the minimum element in the top-K heap. The algorithm terminates when no subgraph using the remaining edges can result into a candidate match with upper bound score greater than the minimum element in the top-K heap. We discuss the details below.

Definition 7 (Valid Edge). A valid edge e with respect to a query edge qE is a graph edge such that both of its endpoints are contained in the potential candidate set for the correspond *ing query nodes in* qE. *Recall that the potential candidate set for each query node is computed using Algorithm 1.*

The sorted edge lists are quite similar to the lists in Fagin's TA [4]. To traverse the edge lists in the top to bottom order, a pointer is maintained with every edge list. The pointers are initialized to point to the topmost graph edge in the sorted edge list, which is valid for at least one query edge. As the pointers move down the lists, they move to the next valid edge rather than moving to the next edge in the list (as in Fagin's TA).

Definition 8 (Size-c candidate match). A size-c candidate match is a partially grown match such that c of its edges have been instantiated using the matching graph edges.

VI. EXPERIMENTS

We perform experiments on multiple synthetic datasets each of which simulates power law graphs. We evaluate the results on the real datasets using case studies. We perform a comprehensive analysis of the objects in the top subgraphs returned by the proposed algorithm to justify their interestingness. Data and code is available at http://dais.cs.uiuc.edu/manish/RWM/.

A. Synthetic Datasets

We construct 4 synthetic graphs using the R-MAT graph generator in GT-Graph software [2]: G1, G2, G3 and G4 with 103, 104, 105, and 106 nodes respectively. Each graph has a number of edges equal to 10 times the number of nodes. Thus, we consider graphs with exponential increase in graph size. Each node is assigned a random type from 1 to 5. Also, each edge is assigned a weight chosen uniformly randomly between 0 and 1. All the experiments were performed on an Intel Xeon CPU X5650 4-Core 2.67GHz machine with 24GB memory running Linux 3.2.0. The code is written in Java. The distance parameter D for the indexes is set to 2 for both the proposed approach RWM (Ranking While Matching) and the baseline RAM (Ranking After Matching), unless specified explicitly. Also unless specified explicitly, we are interested in computing top 10 interesting subgraphs (K=10) and the execution times mentioned in the tables and the plots are obtained by repeating the experiments 10 times.

Baseline: Ranking After Matching (RAM)

The problem of finding the matches of a query Q in a heterogeneous network G has been studied earlier [20], [27]. In [27], the authors present an index structure called SPath. SPath stores for every node,a list of its typed neighbors at a distance d for $1 \leq d \leq D$. SPath index is then used to efficiently find matches for a query in a path-at-a-time way: the query is first decomposed into a set of shortest paths and then the matches are generated one path at a time. This method is used as a baseline.

Index Construction Time

Figure 4 shows the index construction times for the various indexes. Generating the sorted edge lists is very fast. Even for the largest graph with a million nodes, the sorted edge lists creation takes around 40 seconds. The Topology+MMW (D=2) and SPath (D=2) curves show the time required for construction of these indexes, for various graph sizes. The X axis denotes the number of nodes in the synthetic graphs and the Y axis shows the index construction time in seconds. Note the Y axis is plotted using a log scale.

The index construction time rises linearly as the graph size grows. Also, as expected the index construction time rises as D increases.

Index Size

Figure 4 shows the size of each index for different values of D. The X axis plots the number of nodes in the synthetic graphs and the Y axis plots the size of the index (in KBs) using a logarithmic scale. Different curves plot the sizes of various indexes, and the graph. Note that the size of the topology index and the MMW index for D=2 is actually smaller than the size of the graph. Even when the index parameter is increased to D=3, the topology and the MMW indexes remain much smaller than the SPath index for D=2. For D=3, the SPath index grows very fast as the size of the graph increases. As expected as the graph size increases, the size of each index increases. While the increase is manageable for the Edge lists, the MMW index and the topology index, the increase in SPath index size is humongous.

Query Execution Time

We experiment with three types of queries: path, clique and general subgraphs, of sizes from 2 to 5. We present a comparison of different techniques for the graph G2 using the indexes with *D*=2.

	$\|V_Q\|= 2$	$\|V_Q\|= 3$	$\|V_Q\|= 4$	$\|V_Q\|= 5$
RAM	245	2004	14628	169328
RWM0	15	32	43	122
RWM1	19	36	98	178
RWM2	20	40	442	6887
RWM3	218	1733	2337	3933
RWM4	18	34	42	118

Table I
Query Execution time (MESC) for Path queries
(Graph G2 and Indexes with D=2)

	$\|V_Q\|= 2$	$\|V_Q\|= 3$	$\|V_Q\|= 4$	$\|V_Q\|= 5$
RAM	144	8698	34639	174992
RWM0	11	376	14789	229236
RWM1	14	448	16789	200075
RWM2	13	567	19089	201709
RWM3	157	2279	17184	161545
RWM4	12	347	13567	198617

The tables I- II show the average execution times for an average of 10 queries per experimental setting each repeated 10 times. The six different techniques are as follows: RAM (the ranking after matching baseline), RWM0 (without using the candidate node filtering), RWM1 (without using the MMW index), RWM2 (same as RWM1 without the pruning any partially grown candidates), RWM3 (same as RWM1 without the global Top-***K*** quit check), RWM4 (same as RWM1 with the MMW index). Clearly, RAM takes muchlonger execution times for all types of queries. We observed that the larger the number of candidate matches, the more the execution time gap between the RAM method and the RWM methods. An interesting case is $|V_Q|=5$ for the clique queries. Actually there are very few (less than 10) cliques of size 5 of a particular type in the graph. Hence, we can see that almost all the approaches take almost the same time. In this case, the Top-***K*** computation overheads associated with the RWM approaches and lack of pruning result in relatively lower execution time for RAM.

Next, note that RWM4 usually performs faster than RWM1. The time savings are higher for the path queries compared to the subgraph or clique queries. This is expected because the upper bound scores computed in RWM4 are tighter only if most of the query structure can be covered by the non overlapping paths. Also, RWM0 performs slightly better than RWM4 for smaller query sizes, but candidate node filtering helps significantly as query size increases.

Table-III shows the time split between the candidate filtering step and the actual Top–***K*** execution. Note that the candidate filtering takes a very small fraction of the total query execution time.

	$\|V_Q\|=2$	$\|V_Q\|=3$	$\|V_Q\|=4$	$\|V_Q\|=5$
RAM	158	3186	39294	469962
RWM0	10	165	824	4660
RWM1	12	195	1022	5891
RWM2	12	212	3135	27363
RWM3	111	1486	3978	9972
RWM4	12	165	791	4518

Table –III

Query Execution Time (Msec) for Subgraph Queries (Graph G2 and Indexes With D=2)

Query size / Query Type	$\|V_Q\| = 2$		$\|V_Q\| = 3$		$\|V_Q\| = 4$		$\|V_Q\| = 5$	
	CFT	TET	CFT	TET	CFT	TET	CFT	TET
Path	8	10	10	24	10	32	12	106
Clique	5	6	8	##	9	13538	9	2E+05
Subgraph	6	6	9	##	10	781	12	4506

Table –IV

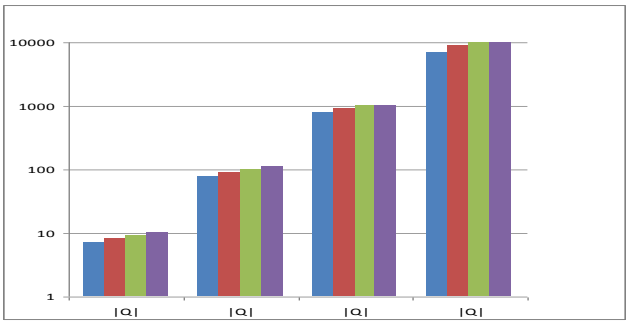


Fig. 5 Query Execution Time for Different Values of ***K***

Scalability Results

We run the 20 path and general subgraph queries (each 10 times) over all the 4 synthetic graphs using RWM4 and present the results in Table IV. The table shows that the execution time increases linearly with the graph size, and exponentially with the query size. Even though the execution time is exponential in query size, (1) that is the case with most subgraph matching algorithms, and (2) intuitive user queries are limited in size by limits of human interpretability for most applications. Effect of Varying the K Figure-5 shows the effect of varying K on 20 path and general subgraph queries on graph G2 using RWM4. As expected, the query execution time increases as K increases. However, the increase in execution time is reasonably small enough making the system usable even for larger values of K.

	$\|V_Q\| = 2$	$\|VQ\| = 3$	$\|VQ\| = 4$	$\|VQ\| = 5$
# SIZE-1 Candidates	9.55	7.87	4.38	1.63
# SIZE-2 Candidates		29.28	19.31	8.94
# SIZE-3 Candidates			24.42	24.5
# SIZE-4 Candidates				14.61

TABLE V

Number of candidates as percentage of total matches for different query sizes and candidates sizes

Table-V shows the percentage of candidates of different sizes with respect to the total number of matches. The results shown in this table are obtained by running the algorithm for the 20 path and subgraph queries on graph G2. We removed the clique queries because the number of cliques of size 5 matching such queries is less than 10 and hence no pruning occurs. Note that on an average, the number of candidates is around 14% of the total number of matches. Clearly, for subgraph queries there are candidates of higher sizes also, but the number of such candidates is much smaller (< 1%) compared to the number of matches, and so we do not show them here.

V. RELATED WORK

The network (graph) query problem can be formulated as a selection operator on graph databases and has been studied first in the theory literature as the subgraph isomorphism problem [3], [14], [20]. One way of answering network queries is to store the underlying graph structure in relational tables and then use join operations. However, joins are expensive, and so fast algorithms have been proposed for approximate

	DBLP	Wikipedia
Nmber of Nodes	138 K	670K
number of edges	1.6M	4.1M
number of types	3	10
Sorted Edge List Index Size	50 MB	261 MB
Topology Index Size	5.8 MB	148 MB
MMW Index Size	11.4 MB	249 MB
Spath Index Size	4.3 GB	13.7 GB
Sorted Edge List Construction Time	12 sec	23 sec
Topology + MMW Construction Time	461 min	1094 min
Average Query Time	100 sec	42 sec

graph matching as well as for exact graph matching. A problem related to the proposed problem is: given a subgraph query, find graphs from a graph database which contain the subgraph [16], [22], [29]. All top-K processing algorithms are based on the Fagin et al.'s classic TA algorithm [4]. Growing a candidate solution edge-by-edge in a network can be considered to be similar to performing a join in relational databases. The candidates are thus grown one edge at a time much like the processing of a top-K join query [11] and as detailed in Section IV-B. However, we make the top-K join processing faster by tighter upper bounds computed using the MMW index and list pruning using the topology index. The top-K joins on networks with the support of such graph indexes is our novel contribution. The proposed problem is also related to the team selection literature. However, most of such literature following the work of Lappas et al. [13] focuses on clique (or set) queries [10], unlike the general subgraph queries handled by the proposed approach. Top- K matching subgraphs can also be considered as statistical outliers. Compared to our previous work on outlier detection from network data [6], [7], [8], [9], we focus on query based outlier detection in this work. For more comparisons with previous work, please refer to Section I.

VI. CONCLUSION

In this paper, we studied the problem of finding top-K interesting subgraphs corresponding to a typed unweighted query aplied on a heterogeneous edge-weighted information network. The problem has many practical applications. The baseline ranking after matching solution is very inefficient for large graphs where the number of matches is humongous. We proposed a solution consisting of an offline index construction phase and an online query processing phase. The low cost indexes built in the offline phase capture the topology and the upper bound on the interestingness of the metapaths in the network. Further, we proposed efficient top-K heuristics that exploit these indexes for answering subgraph queries very efficiently in an online manner. Besides showing the efficiency and scalability of the proposed approach on synthetic datasets, we also showed interesting subgraphs discovered from real datasets like Wikipedia and DBLP. In the future, we plan to study this problem in a temporal setting.

REFERENCES

[1] P. Bogdanov, M. Mongiov`ı, and A. K. Singh. Mining Heavy Subgraphs in Time-Evolving Networks. In *ICDM*, pages 81–90, 2011.

[2] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-mat: A recursive model for graph mining. In *SDM*, pages 442–446, 2004.

[3] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (Sub)Graph Is omorphism Algorithm for Matching Large Graphs. *TPAMI*, 26(10):1367– 1372, 2004.

[4] Y. Tian, R. C. Mceachin, C. Santos, D. J. States, and J. M. Patel. SAGA: A Subgraph Matching Tool for Biological Graphs. *Bioinformatics*, 23(2):232–239, Jan 2007.

[4] R. Fagin, R. Kumar, and D. Sivakumar. Comparing Top-K Lists. In *SODA*, pages 28–36, 2003.

[5] G. Gou and R. Chirkova. Efficient Algorithms for Exact Ranked Twigpattern Matching over Graphs. In *SIGMOD*, pages 581–594, 2008.

[6] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier Detection for Temporal Data. In *SDM*, 2013.

[7] M. Gupta, J. Gao, and J. Han. Community Distribution Outlier Detection in Heterogeneous Information Networks. In *ECML PKDD*, pages 557– 573, 2013.

[8] M. Gupta, J. Gao, Y. Sun, and J. Han. Community Trend Outlier Detection using Soft Temporal Pattern Mining. In *ECML PKDD*, pages692–708, 2012.

[9] M. Gupta, J. Gao, Y. Sun, and J. Han. Integrating Community Matching and Outlier Detection for Mining Evolutionary Community Outliers. In *KDD*, pages 859–867, 2012.
[10] M. Gupta, J. Gao, X. Yan, H. Cam, and J. Han. On Detecting Association-Based Clique Outliers in Heterogeneous Information Networks. In *ASONAM*, 2013.
[11] I. F. Ilyas, W. G. Aref, and A. K. Elmagarmid. Supporting Top-K Join Queries in Relational Databases. *VLDB Journal*, 13(3):207–221, Sep 2004.
[12] G. Karypis and V. Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *J. Sci. Comp*, 20(1):359–392, Dec 1998.
[13] T. Lappas, K. L iu, and E. Terzi. Finding a Team of Experts in Social Networks. In *KDD*, pages 467–476, 2009.
[14] B. D. McKay. Practical Graph Isomorphism. *Congressus Numerantiu m*, 30:45–87, 1981.
[15] Y. Qi, K. S. Candan, and M. L. Sapino. Sum-Max Monotonic Ranked Joins for Evaluating Top-K Twig Queries on Weighted Data Graphs. In *VLDB*, pages 507–518, 2007.
[16] S. Ranu and A. K. Singh. GraphSig: A Scalable Approach to Mining Significant Subgraphs in Large Graph Databases. In *ICDE*, pages 844– 855, 2009.
[17] Y. Sun, Y. Yu, and J. Han. Ranking-based Clustering of Heterogeneous Information Networks with Star Network Schema. In *KDD*, pages 797–806, 2009.
[18] Z. Sun, H. Wang, H. Wang, B. Shao, and J. Li. Efficient Subgraph Matching on Billion Node Graphs. *PVLDB*, 5(9):788–799, May 2012.
[20] J. R. Ullmann. An Algorithm for Subgraph Isomorphism. *J. ACM*, 23(1):31–42, Jan 1976.
[21] X. Yan, B. He, F. Zhu, and J. Han. Top-K Aggregation Queries ov er Large Networks. In *ICDE*, pages 377–380, 2010.
[22] X. Yan, P. S. Yu, and J. Han. Substructure Similarity Search in Graph Databases. In *SIGMOD*, pages 766–777, 2005.
[23] J. Yang, W. Su, S. Li, and M. M. Dalkilic¸. WIGM: Discovery of Subgraph Patterns in a Large Weighted Graph. In *SDM*, pages 1083– 1094, 2012.
[24] Y. Yuan, G. Wang, L. Chen, and H. Wang. Efficient Subgraph Similarity Search on Large Probabilistic Graph Databases. *PVLDB*, 5(9):800–811, May 2012.
[25] X. Zeng, J. Cheng, J. X. Yu, and S. Feng. Top-K Graph Pattern Matching: A Twig Query Approach. In *WAIM*, pages 284–295, 2012.
[26] S. Zhang, J. Yang, and W. Jin. Sapper: Subgraph indexing and approximate matching in large graphs. *PVLDB*, 3(1):1185–1194, 2010.
[27] P. Zhao and J. Han. On Graph Query Optimization in Large Networks. *PVLDB*, 3(1):340–351, 2010.
[28] F. Zhu, Q. Qu, D. Lo, X. Yan, J. Han, and P. S. Yu. Mining Top-K Large Structural Patterns in a Massive Network. *PVLDB*, 4(11):807–818, 2011.
[29] Y. Zhu, L. Qin, J. X. Yu, and H. Cheng. Finding Top-K Similar Graphs in Graph Databases. In *EDBT*, pages 456–467, 2012.
[30] L. Zou, L. Chen, and Y. Lu. Top-K Subgraph Matching Query in a Large Graph. In *PIKM*, pages 139–146, 2007.
[31] L. Zou, L. Chen, and M. T. ¨Ozsu. Distance-join: Pattern Match Query in a Large Graph Database. *PVLDB*, 2(1):886–897, Aug 2009.

ISSN (e): 2250 – 3005 || Volume, 05 || Issue, 10 ||October – 2015 ||
International Journal of Computational Engineering Research (IJCER)

A Study on Video Steganographic Techniques

Syeda Musfia Nasreen , Gaurav Jalewal, Saurabh Sutradhar

Abstract

Data hiding techniques have taken important role with the rapid growth of intensive transfer of multimedia content and secret communications. The method of Steganography is used to share the data secretly and securely. It is the science of embedding secret information into the cover media with the modification to the cover image, which cannot be easily identified by human eyes. Steganography algorithms can be applied in audio, video and image file. Hiding secret information in video file is known as video steganography. Video Steganography means hiding a secret message that can be either a secret text message or an image within a larger one in such a way that just by looking at it, an unwanted person cannot detect the presence of any hidden message. For hiding secret information in the video, there are many Steganography techniques which are further explained in this paper along with some of the research works done in some fields under video steganography by some authors. The paper describes the progress in the field of video Steganography and intends to give the comparison between its different uses and techniques.

Keywords: *video steganography, LSB method, data hiding, embed, stego video, AVI, PSNR.*

I. Introduction

The premise from which to measure a secure video steganography system is to assume that the opponent knows the system being employed, yet still cannot find any evidence of the hidden message. Video steganography algorithm tries to replace the redundant bits of the cover medium by the bits of the secret medium. Now the availability of those redundant bits to be inserted in the cover media depends on the quality of video or sound. Military, industrial applications, copyright, intellectual property rights etc. are some of the most commonly used applications of video steganography.

The advantages of using video stream as the cover file are to get extra security against the attacker because the video file is much more complex that the image file. One more advantage of embedding the secret data to the video is that the secret data is not recognized by the human eye as the change of a pixel color is negligible. In video steganography, we can also very secretly hide data in audio files as it contains unused bits. We can store secret data up to about four least significant bits in the audio file. So it is more beneficial to use video steganography rather than other steganography methods when we need to store more amounts of secret data. [1]

1.1. Techniques of video steganography

There are various techniques of video steganography. The best technique is to hide the secret data without reducing the quality of the cover video, so that it cannot be detected by naked eyes. The embedded video is known as the "stego" video which is sent to the receiver side by the sender.[2]

Variety of video steganography techniques are used now days, to secure important information. Some much known techniques are explained briefly in the following:

1.2. LSB (Least Significant Bit) method

LSB is said to be the best method for data protection because of its simplicity and commonly used approach. It is the most easiest and effective way of embedding data. In LSB, the cover video's pixel values are extracted which are in bytes, then its LSB are substituted by the bits of the secret message that we will embed. Now since we change only the lsb bits of the host video, it doesn't gets distorted and almost looks alike as the original video.[3]

1.3. Non-uniform rectangular partition

This method is for uncompressed videos. In non-uniform rectangular partition, data hiding is done by hiding an uncompressed secret video file in the host video stream. But we have to make sure that both the secret as well as the cover file should be of almost the same size. Each of the frames of both the secret as well as cover videos is applied with image steganography with some technique. The secret video file will be hidden in the leftmost four least significant bits of the frames of the host video. [3]

1.4. Compressed video steganography

This method is done entirely on the compressed domain. Data can be embedded in the block of I frame with maximum scene change and in P and B block with maximum magnitude of motion vectors. The AVC encoding technique yields the maximum compressing efficiency. [3]

1.5. Anti-forensics technique

Anti-forensic techniques are actions taken to destroy, hide and/or manipulate the data to attack the computer forensics. Anti-forensic provides security by preventing unauthorized access, but can also be used for criminal use also. Steganography is a kind of anti- forensic where we try to hide data under some host file. Steganography along with anti- forensics makes the system more secure. [3]

1.6. Masking and filtering

Masking and filtering are used on 24 bits/pixel images and are applicable for both colored and gray scale images. It is like watermarking over an image and doesn't affect the quality of that image. Unlike other steganography techniques, in data masking the secret message is so processed such that it appears similar to a multimedia file. Data masking cannot easily be detected by traditional steganalysis.[3]

II. Related works

In 2009, Eltahir, L. M. Kiah, and B. B. Zaidan presented a high rate video streaming steganography system based on least significant bit method.[22] The results of using this method on instant images saves up to 33.3% of the image for data hiding which is an enhancement for LSB. The idea of the suggested method is by using 3-3-2 approach, which uses the LSB of RGB (red, blue, green) colors in 24 bits image. The method here takes the least 3 bits of red color, 3 bits of green color and only 2 bits from blue color because human vision system is more sensitive to blue than red and green, to come up with 1 byte which is used for data hiding. So to make the outcome image look almost the same as the original, the 3-3-2 approach is very efficient.
The result was found to be good and the size of data was substantial i.e. about 33.3% from the size of image. They didn't found any difference between two frames and their histograms, especially for human vision system. [4]

In the year 2011 ShengDun Hu, KinTak U presented a video steganography system based on non-uniform rectangular partition. This technique is used in uncompressed videos. In this method a secret video is hidden in a cover video, both should be of almost the same size. In each frame of both the videos, a mechanism is applied for hiding the video stream. The frame length of the cover video should be greater than or equal to the frame length of the secret video, in order to hide the secret video in to the cover or host video. Each frame of secret video is portioned in to non-uniform rectangular part which is encoded. The secret video stream is hidden in the leftmost four least significant bits of each frame of the host video stream.
Results of using this technique showed no distortion, so no one will think that any kind of data is being hidden in the frames. All the PSNR values of the frames were larger than 28db. [5]

In 2014, R. Shanthakumari and Dr.S. Malliga presented a paper on Video Steganography using LSB matching revisited algorithm, where they have taken a video stream of AVI format. In the paper they have initially splitted the cover video into frames. Now, the message can be embedded in multiple frames, therefore size of a message does not matter in video steganography. After embedding the secret data in multiple frames of the cover video stream, all the frames are then grouped together to form the stego video, which is then again will be splitted into frames and data will be extracted in the receiver side.
The proposed method in this paper was found to have two problems which were low embedding rate and lack of security. LSBMR algorithm has a low replacement rate and hence the Mean Square Error (MSE) is low, as a result of which LSBMR is more secured than the LSB algorithm for data hiding. The PSNR value decreases on increase of the embedding unit.[6]

In 2015, Vivek Kapoor and Akbar Mirza presented a paper on Enhanced LSB based Video Steganographic System for Secure and Efficient Data Transmission. At first, they have divided the host video into frames, then the text file to be embedded is compressed using the ZIP compressor and the bytes are generated. The advantage of sending compressed data over the network is to minimize the payload and reduce extra burden of the network. Then the color of each color pixels is calculated in RGB 24 bit format. Then chunks of bits are created from the extracted bytes of the secret message. Then these chunks are embedded in the video frames. Text files are then combined with video frames and the file is sent. The proposed method is designed for MPEG format; however it can work with other video file formats also like AVI, 3GP by doing some modification in it. They have calculated the Data Quality, Mean Squared Error(MSE) and Peak Signal to noise ratio(PSNR) of both the original and the stego video and it was found that Mean Square Error and Peak Signal to Noise Ratio is low enough that it cannot be noticed easily in the Steganalysis process. Then they have also compared their method with LSB method and they found that the proposed method gives better MSE and PSNR values than the LSB method. [7]

In 2013, Hemant Gupta, Dr. Setu Chaturvedi presented a video steganography through LSB based hybrid approach. This method is used in AVI videos. The video is converted into 20 equal gray scale images. Data hiding is done is done in the host video by using Single bit, two bit, three bit LSB substitution and after that Advanced Encryption Standard Algorithm is applied. After processing the source video by using the data hiding procedures, the encrypted AVI Video is sent by the sender and decryption is performed by the receiver. They have found the PSNR and correlation factor between Original and embedded image for 1 bit LSB & 2 bit LSB & 3 bit LSB Substitution and AES method. It is observed that PSNR value decreases and security increases with the increase of LSB substitution bit. In this paper they have found no correlation relation between original image and encrypted image for different frames.[8]

In 2013, Pooja Yadav, Nischol Mishra and Sanjeev Sarma presented a video steganography technique with encryption and LSB substitution. In their technique, they had 2 video streams called Host and the Secret video with same number of frames and equal frames per second (12 frames and 15 fps). A header of 8 bits is used for representing the frame size of the hidden video and was appended in the beginning of each frame. After this the appended header was encrypted along with the secret frames by using symmetric encryption. Using sequential encoding, the secret video frames are encoded to the host video frames and then from the encoded frames the secret video is generated. They used XOR transformation for encrypting the data with secret keys and decrypting the secret message to retrieve the original information. For sequential encoding they used a pattern of BGRRGBGR (Blue (B), green (G) and red (R)) to encode the message in the LSB. They used 2 AVI video files and found the PSNR value of the stego video as 35 dB which was the same as the host video.
To find the result they have calculated the PSNR value frame by frame and maximum, minimum and average PSNR value of total frames. Frame by frame comparison of the host and the embedded video stream shows that there was no distortion in the stego video. According to the calculated PSNR values, it was observed that there was much similarity between the host and the embedded video. The host video was found to be distortion less and also the recovered video stream had also an acceptable quality. [9]

Ramadhan J. Mstafa and Khaled M. Elleithy, Senior *Member, IEEE, Department* of Computer Science and Engineering, University of Bridgepor, proposed a highly secured method of video steganography by using Hamming Code (7, 4). In their project, they used 9 video files as cover and 1 secret image which were to be hidden. At first, they generated the frames from the video stream and then separated each frames into Y, U & V components. Then by the use of a special key, all pixel positions of video are randomly ordered. A binary image is used as the message which was converted to a 1-dimensional array and the position of the message is changed by a key. Now, 4 bits of the message is encoded using Hamming Code (7, 4) encoding technique. Now, the encoded data is XORed with the random values and the result is embedded in 1 pixel of Y, U and V components. The pixels are then reordered in their original position and the final stego video was rebuilt from the embedded frames. Similar steps are involved in the data extraction process.
The stego video has mostly the same quality as the original video because of the low modification on the host video stream. The visual quality is measured by the PSNR and all the obtained experimental results have a PSNR above 51 dBs. Using this method, attackers are not likely to be suspicious since they have a good visual quality for stego videos. The algorithm is much secured because security has been satisfied by having more than one key to embed and extract the secret message. [10]

In 2008, Bin Liu et al proposed a new steganography algorithm for compressed video Bit streams. In this technique, the embedding and detecting data is done only in compression domain and no decompression is needed here. The cover video was first compressed by eliminating temporal, spatial and statistical redundancies.

The video was then divided into several slow speed and single scene video sub-sequences. After the scene detection process, they embedded the secret message in the video file without any distortion. Finally the embedded video was tested for Steganalysis to check the presence of hidden data in the video. They constantly adjusted the scale factor for manipulating the hidden data strength until the analyzer was unable to detect the hidden message.

The PSNR value and the correlation value changes the magnitude of the stego-video, which says that the perception quality and intraframe correlation of test video is little changed. There is no noticeable change in the visual quality of the compressed video, also their system was found to highly secured as they were continuously testing the video for Steganalysis.[11]

III. Comparative Analysis

This paper presented a background of Steganography and a comparative study of some Steganographic techniques. There are two important parameters of evaluating all Steganography technique, first is imperceptibility and the second is capacity. Imperceptibility means the embedded data must be imperceptible to the observer and computer analysis. Capacity means maximum payload is required, *i.e.* maximum amount of data that can be embedded into the cover image without losing the fidelity of the original image. The results of surveying the papers in different techniques of video steganography showed that all the methods possess the ability to hide data without noticing changes in their properties.

It was found that in [4], they have used the 3-3-2 approach along with LSB and the result was found to be good and about 33.3% from the size of image can be used for data hiding. In other words, in the space of 5 images, 500 pages of data could be stored without resizing. Similarly, 1 second in certain video types contains approximately 27 frames, which in turn creates a lot of room for hiding data. In [5], Results of using this technique showed no visual distortion in the host file and even the quality of the new video generated can be accepted for practical use. In [6], it has been known that in the LSB algorithm due to high replacement rate, MSE value is high. So it lacks from security**.** In case of LSBMR algorithm due to low replacement rate, MSE value is low which makes it secure when compared to LSB algorithm. In their method, intruder may not be able to identify the presence of the secret message inside the frame. Also, the comparison with the original video never gives the original secret message, which ensures additional security.

In [7], videos of different sizes and resolutions are tested for their method and they have got successful in keeping the MSE and PSNR value low enough that it cannot be noticed easily in the Steganalysis process. They have provided a comparison between the basic LSB method and their method gave better values of MSE and PSNR than the LSB method. The average PSNR of the proposed LSB embedding technique (per pixel, RGB) to the traditional layering technique in which embedding is done by layers of RGB. They found an improvement of about 1.5 dB in their PSNR value when compared to the traditional LSB technique and also a lesser MSE which means in detectability. In [8], the authors calculated PSNR value for different amount of LSB substitution. For 1 LSB substitution, the PSNR value was found between 45-50 for different no of frames. For 2 bit LSB substitution, PSNR was found to be in the range of 40-45. And for 3 bit LSB substitution, PSNR value was about 35. By the use of AES encryption, their method was more secured as compared to traditional LSB techniques. In [9], their results showed that no visual distortion is there in the host video stream and even the quality of the recovered secret video is also acceptable in practical. In [10], use of Hamming code makes the technique highly efficient and more secured. The authors used more than 1 key and thus have obtained a high level of security as compared to traditional steganographic methods like LSB substitution where only one XOR encryption is used. In [11], unlike other steganographic technique, the authors have implemented a closed loop feedback steganalysis to test their project's immunity towards steganalysis. The complete project was done in compressed domain hence avoiding decompression process.

With continuous advancements in technology it is expected that in the near future more efficient and advanced techniques in steganalysis will emerge that will help law enforcement to better detect illicit materials transmitted through the Internet.

IV. Conclusion

In the era of fast information interchange using internet and World Wide Web, video Steganography has become essential tool for information security. This paper gave an overview of different video steganographic techniques its major types and classification of steganography which have been proposed in the literature during last few years.

References

[1] https://courses.cs.washington.edu/courses/csep590/06wi/finalprojects/chakraborty.doc

[2] Arup Kumar Bhaumik, Minkya choi, "Data hiding in video" IEEE International journal of database application,vol.2no.2 June 2009.pp.9-15

[3] https://edupediapublications.org/journals/index.php/ijr/article/view/678/309

[4] M. E. Eltahir, L. M. Kiah, and B. B. Zaidan, "High Rate Video Streaming Steganography," in Information Management and Engineering, 2009. ICIME '09. International Conference on, 2009, pp. 550-553.

[5] ShengDun Hu, KinTak U," A Novel Video Steganography based on Non-uniform Rectangular Partition ",IEEE International Conference on Computational Science and Engineering,pp 57-61, Aug.2011.

[6] R. Shanthakumari and Dr.S. Malliga," Video Steganography using LSB matching revisited algorithm", IOSR Journal of Computer Engineering ,Volume 16, Issue 6, Ver. IV(Nov – Dec. 2014), PP 01-06

[7] Vivek Kapoor and Akbar Mirza, "An Enhanced LSB based Video Steganographic System for Secure and Efficient Data Transmission", International Journal of Computer Applications (0975 – 8887) Volume 121 – No.10, July 2015

[8] Hemant Gupta and Dr. Setu Chaturvedi,"video steganography through LSB based hybrid approach", International Journal of Engineering Research and Development, Volume 6, Issue 12 (May 2013), PP. 32-42

[9] Pooja Yadav, Nischol Mishra and Sanjeev Sarma, "video steganography technique with encryption and LSB substitution", 2013, School Of Information Technology, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India

[10] Ramadhan J. Mstafa and Khaled M. Elleithy, Senior Member, IEEE, Department of Computer Science and Engineering University of Bridgeport Bridgeport, CT 06604, USA, "A Highly Secure Video Steganography using Hamming Code (7, 4) "

[11] Bin Liu, Fenlin Liu, Chunfang Yang and Yifeng Sun , "Secure Steganography in Compressed Video Bitstreams",The Third International Conference on Availability, Reliability and Security

ISSN (e): 2250 – 3005 || Volume, 05 || Issue, 10 ||October – 2015 ||
International Journal of Computational Engineering Research (IJCER)

Study on groundwater quality in and around sipcot industrial complex, area cuddalore district,tamilnadu.

Inbanila.T, Arutchelvan.V
Department of civil Engneering, Annamalai university,Chidambaram, India.

ABSRACT

STATE INDUSTRIES PROMOTION CORPORATION OF TAMIL NADU(SIPCOT) cuddalore phase 1 has estabilished in 1984 at an extent of 518.79 acres. currently between 26 and 29 functional units are lie within phase1 of the industrial estates.At least 10 villages lie within or in the vicinity of the industrial complex. Till date no sites has been developed for secure storage of hazardous wastes generated by the industries in the estate. In absence of such facilities factories have dumped these wastes on neighbouring lands and in open pits. By the industries own admission,out of the 20 million litres of fresh water required by the companies, 18 million litres (90%) of the water is released back to their environment as toxic effluents.These poisons have leached into the ground water and contaminated the water resources of communities living around the factory. This study was carried out to asses the Quality of ground water in and around SIPCOT industrial complex in cuddalore district. The Quality was assessed in terms of physico chemical parameters.Ground water samples were collected from 30 locations in and around the study area and analyzed (APHA,1998) to know the present status of the Ground water Quality. The results were compared with standards prescribed by ISI 10500-91.It was found that the ground water was contaminated at few sampling locations.The remaining locations shows that the parameters are within the desirable limits and fit for drinking purpose.

Keywords: *Ground water, water Quality,SIPCOT.*

I. INTRODUCTION

Ground water is water that found underground in voids and fractures,cracks and space in soil. Ground water forms a major source of drinking water foe the urban and rural population of India.Besides being the primary source of water supply for domestic use,it is almost the most source of irrigation. It has become evident that the ground water has been a major contribution to meet the ever increasing depend of water .Ground water is a gift of nature, is about 210 billion m3 includes recharge through infiltration, seepage and evaporation. The salinity intrusion and industrial pollution of ground water are to key reasons for deterioration of water quality. The objective of this study is to analyze the quality of ground water due to the discharge of waste from SIPOT industries.

II. STUDY AREA

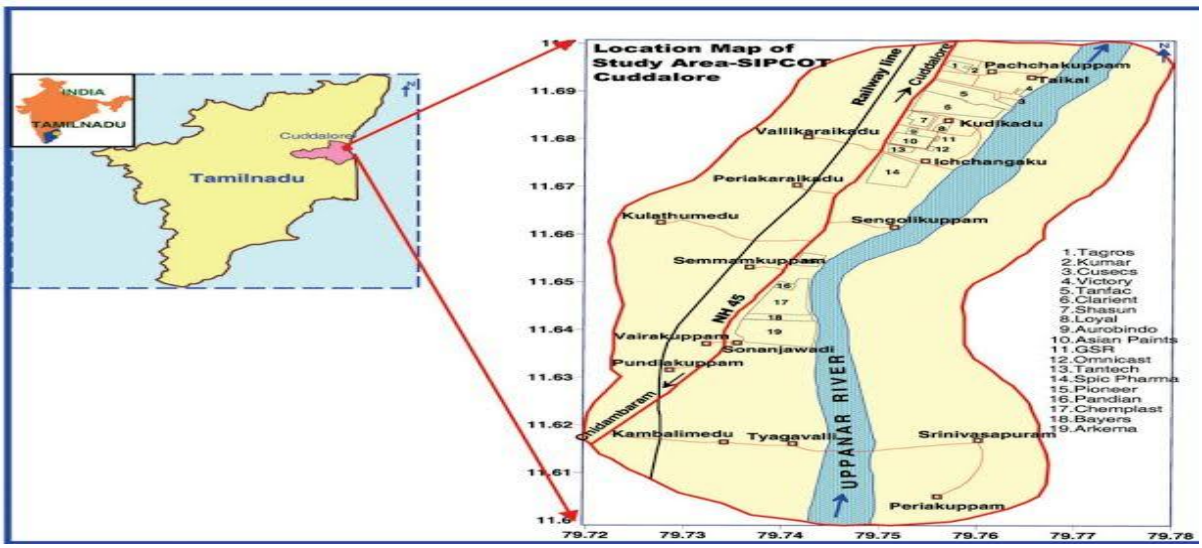
Cuddalore is the heartland of Tamilnadu,located 200km south of chennai and lessthan 25km south of Pondichery is a developing industrial city,lying between latitude 11 43 north and longitude 79 49 east. It is the port town from ancient times with historical trades lies to the occident and the orient.The 27 sg.km district comprises 6 taluks and136 panchayat villages. Bore well water is using for drinking and irrigation purpose in this district.SIPCOT has estabilished in 1984 at an extent of 518.79 acres.

It is located 8 km from Cuddalore to Chidambaram road,stretching from Pachaiyankuppam in the north to semmankuppam in the south.Phase II will cover 88 hectares(200 acres).curently between 26 and 29 functional units are lieing within phase I of the industrial estate on the western bank of the river uppanar.these companies manufacture pesticides and pharmaceuticals and intermediates,chemicals,plastics and plastics additives,dyes and intermediates and textiles.

At least 10 villages lie within the vicinity of the industrial complex. At least 2000 peoples are estimated to be lying in the potential impact of SIPCOT taken taken from SIPCOT area community environmental monitors.

The village pachaiyankuppam is located at the north of the SIPCOT complex behind tagro's chemicals. Kudikadu lies on the eastern side of vanavil dyes and shasun chemicals uppanar towards band of uppanar behind asian paints. Echaukadu is located between pharma to the south and tanfac agro chemicals in the north. Sangolikuppam lies near the north of pioneer chemical. Semmakuppam lies immediate south of pioneer miyagi chemicals and sonnanchavadi is locatd on he southern end of sipcot, south of Aerokema peroxides and bayer. Groundwater was earlier available at 30 feet or less is now difficult to find even at 800 feet according to sipcot resideuts .

The study area receives about an annual rainfall of 1,162mm. Ground water in the area is overexploited for agriculture and industrial purposes are predominant land use , which includes sailing in the coastel aquifers.



III. MATERIALS AND METHODS

To asses ground water Quality in and around SIPCOT area, 30 sampling locations are selected in around the study area. Water samples were collected during pre-monsoon season during (20.8.2012). Samples were analyzed for different phisico - chemical parameters such as pH, Electrical conductivity(EC) ,Total Dissoived solids(TDS), Turbidity ,Total Alkalinity(TA), Total Hardness(TH) , Calcium (ca), Magnesium (Mg^{2+}) , Sodium(Na), Potassium (K), Chlorides(Cl), Nitrate(NO3), Sulphate(S04), etc as per the standard procedure APHA (1995).

IV. RESULT AND DISCUSSION

Table:1 The analytical results are given in Waterquality Standards for drinking water

S.NO	PARAMETER	ISI 10500-91
1.	PH	6.5-8.5
2.	EC	-
16.	POTASSIUM	-
3.	TDS	500
4.	TURBIDITY	10
5.	TOTAL ALKALINITY	200
6.	TOTAL HARDNESS	300
7.	CALCIUM	75
8.	MAGENESIUM	30
9.	SODIUM	200
10.	IRON	0.3
11.	MANGANESE	0.1
12.	CHLORIDE	250
13.	FLUORIDE	1.5
14.	SULPHATE	200
15.	NITRATE	45

All the parameters are in mg/l, except pH and turbidity in NTU.

The values are compared with BIS standard for drinking water IS: 10050:1991
The findings are discussed below.

Table-2
Physico-chemical characteristics of groundwater of in and around sipcot area, cuddalore

S.NO	HABITATION	SEASON	TURB	EC	TDS	PH	T.ALK	T.H	CA	MG	NA	K	FE	MN	NO3	CL	F	SO4
1	THIRUCHIOPURAM	PRE-MONSOON	1.0	515	361	7.1	95	160	43.2	12	28	9	0.0	0	7	77	0.4	29
2	THIYAGAVALLI	PRE-MONSOON	1	1090	763	6.8	172	288	65.6	30	76	27	0.0	0	3	220	0.3	25
3	LENNINNAGAR	PRE-MONSOON	0.6	290	203	6.7	65	100	25.6	9	12	5	0.00	0	6	40	0.1	4
4	AMBETHKAR NAGAR	PRE-MONSOON	0.8	205	144	6.6	69	70	16	7	31	9	0.16	0	2	19	0	3
5	PERIYARNAGAR	PRE-MONSOON	0.9	440	308	6.5	82	152	36.2	15	18	6	0.0	0	2	90	0.1	25
6	NADUTHITTU	PRE-MONSOON	1	320	224	7.1	65	120	41.6	4	9	3	0.0	0	3	54	0.0	6
7	NOCHIKADU	PRE-MONSOON	2.6	445	312	7.0	129	176	36.8	20	11	4	0.40	0	2	36	0.0	26
8	CHITHIRAIPETTAI	PRE-MONSOON	1.5	535	375	7.3	168	200	48	19	52	13	0.13	0	3	65	0.1	31
9	POONDIYANKUPPAM	PRE-MONSOON	2.6	180	126	6.6	17	56	96	8	10	3	0.40	0	9	34	0.1	12
10	MANDAPAM	PRE-MONSOON	1	2600	182	6.6	22	80	19.2	8	14	6	0.0	0	3	65	0.1	13
11	SEMMANKUPPAM	PRE-MONSOON	1.3	530	371	6.7	108	210	48	22	70	25	0.13	0	7	537	0.4	31
12	SEMMANKUPPAM COLONY	PRE-MONSOON	1.5	550	385	6.6	206	160	40	14	51	14	0.13	0	3	28	0.1	11
13	SONANCHAVADI	PRE-MONSOON	1.6	815	571	7.1	215	248	44.8	33	47	12	0.27	0	3	108	0.1	7
14	SONANCHAVADI METTUTHERU	PRE-MONSOON	1	605	424	6.6	65	152	40	12	48	12	0.0	0	8	107	0.1	56
15	VAIRANKUPPAM	PRE-MONSOON	3.4	1815	1271	6.9	172	580	118	68	88	35	0.8	0	4	444	0.5	103
16	VAIRANKUPPAM COLONY	PRE-MONSOON	26	560	392	6.7	151	152	38.4	13	38	13	2.67	0	3	64	0.4	20
17	THACHAN COLONY	PRE-MONSOON	1	1510	1057	6.6	125	460	136	29	82	30	0.13	0	1	262	0.6	194
18	SANGOLIKUPPAM	PRE-MONSOON	1.0	935	655	6.9	267	236	64	18	71	22	0.13	0	3	92	0.1	26
19	SANGOLIKUPPAM COLONY	PRE-MONSOON	1.0	935	655	6.9	267	236	64	18	71	22	0.13	0	3	92	0.1.	26
20	ECHANKADU	PRE-MONSOON	1.5	580	460	6.9	159	200	56	14	140	52	0.13	0	5	28.5	0	20
21	SEDAPALAYAM	PRE-MONSOON	1 0.8	330 390	231 273	6.6 6.6	43 69	104 110	28 28	8 10	16 24	7 10	0.0 0.0	0 0	4 3	50 56	0.2 0.2	30 24
22	CHINNAKARAIKADU	PRE-MONSOON	0.8	330	231	6.9	108	110	28	10	65	35	0.13	0	3	23	0.1	11
23	KARAIKADU	PRE-MONSOON	0.8	940	658	6.8	172	288	80	21	50	19	0.1	0	3	133	0.2	68
24	KARAIKADU COLONY	PRE-MONSOON	1.2	1225	858	6.7	237	360	95	27	80	23	0.0	0	6	70	0	20
25	KUDIKADU	PRE-MONSOON	1.0	1375	965	7.2	172	400	131	17	87	24	0.13	0	7	272	0.5	32
26	KUDIKADU (ROYAL FABRICS)	PRE-MONSOON	1	920	644	6.8	159	268	60.8	28	58	14	0.13	0	3	142	0.2	32
27	PERIYAKARAIKADU	PRE-MONSOON	1	865	606	7.0	43	260	57.6	24	100	32	0.13	0	2	266	0.2	32
28	PERIYAKARAIKADU COLONY	PRE-MONSOON	1.2	550	371	7.7	86	164	40	15	28	10	0.13	0	1	65	0.6	44
29	THAIKAL	PRE-MONSOON	1	730	511	6.7	155	260	48	19	25	11	0.0	0	5	92	0.1	41
30	PACHAYANKUPPAM	PRE-MONSOON	0.7	650	455	6.8	172	140	40	10	50	19	0.0	0	3	65	0.1	21

All the parameters are in mg/l, except pH and turbidity.is expressed in NTU,EC in micromhos/cm

PH:

The low PH value may cause corrosion in containers and pipe lines, while the high may produce sediments,deposits and difficult in chlorination for disinfection of water(sudhakar gummadi et al 2013).In the collected samples the values are with in the permissible limit.There is no abnormal change in the ground water samples.

Turbidity:

Turbidity was in the range of 0.6-26 mg/l.
Out of 30 sampling locations, turbidity exceeded the desirable limit of 5mg/l in one location.

Total Dissolved Solids:

TheTotal solids in water are due to the presence of sodium,potassium,calcium, magnesium,manganese,carbonates,chlorides,organic matrer,other particles.(Bhattacharya T., et al(2012)).

TDS was found in the range of 126-127mg/l. From the 31 sampling locations,15 locations exceeded the desirable limits of 550 mg/l. The highest value was recorded in Vairankuppam(BW).

Electrical Conductivity(EC):

Signifies the amount of total dissolved solids.EC values were in the range of 180-1530 micromnos/cm. High EC value was observed in Echankadu (HP) indicating the high amount of dissolved inorganic substance in ionized form. pH varies from 6.5-7.7 and were found within the limit prescribed by ISI.

Total alkalinity:

Total alkanity of water is due primarily to the salts of week acids. Bicarbonate represents the major of alkalinity. Total alkanity was in the range of 17-280mg/l. Thee highest value 270 mg/l was found in Chinnakaraikadu,whereas the desirable limit is 200mg/l. Out of 30 samples,4 samples (Sonachavadi, Sangolikuppam(colony), Echankudi & Chinnakaraikadu) are exceeding the limit.

Total Hardness

Hardness of water mainly depends upon the amount of calcium or magnesium salts or both. Hardness may also be caused by ferrous and manganese (Kavitha Kirubavathi A, 2010) .Total Hardness in the study area was in the range of 56 mg l -580 mg/l.The highest value is found in Vairankuppam ,where as the desirable limit is 300mg/l.

Calcium (ca) : Calcium may dissolve readily from carbonate rocks and lime stones or be leached from soils. But calcium is an essential nutritional element for human being and aids in the maintaining the structure of plant cells and soils(Chadrik Route et al 2011) Calcium was found to be in the range of 9.6-136 mg/l. Vairakuppam, Thachan colony ,sangolikuppam and Chinnakaraikadu were above the desirable limit of 75 mg/l.

Magnesium:

Magnesium generally occurs in lesser concentration than calcium because of dissolution of magnesium rich minerals is slow process and calcium is more abundant in earth crust (Varatharathajan N et al., 2013) Magnesium was detected in the range of 4-68 mg/l, whereas the desirable limit is 30 mg/l. From the study area 4 samples such as Sonachavadi, Vairankuppam, Echankadu,Thaikal wre above the desirable limits.

Chlorides:

Excess chloride (>250 mg/l) imparts a salty taste to water . Exessive chlorides in potable water is particularly not harmful but the criteria set for chloride value is based on its potentially high corrosiveness . Desirable limit of chloride in drinking water is 250 mg/l. In the study the chloride concentration was found to be in the range of 34 mg/l-444 mg/l. From this analysis, samples at Vairankuppam, Thachan colony,Echankadu,Kudaikadu and Periyakaraikadu exceeds the desirable limits.

Sodium:

Sodium and potassium elements are directly added into the ground water from industrial and domestic waste and contribute salinity of water (Mohamed Hanifa M, et al., 2013) .Sodium concentrations were found in the range of 9 mg/l-140 mg/l.The values of sodium concentrations of all the samples are within the desirable limits.

Potassium:

Sodium and Potassium are most important minerals occurring naturally. High amount of potassium in the ground water is due to presence of Silicate minerals from igneous and metamorfic rocks (Zahir Hussain A, et al ., 2011) Potassium content in this study was in the range of 3 mg/l-52 mg/l

Nitrate:

The presence of nitrate in ground water may be due to leaching of nitrate with a perculating water . The contamination of ground water may be due to seawage and other waste rich in nitrate (Venkateshwara Rao B, et al ., 2011). Toxicity of nitrates in infants causes methaemoglobiaemia (Basic information in nitrates in drinking water, US EPA-2012). Nitrate was measured in the range of 1-9 mg/l. All the samples are within the desirable limit of 45 mg/l

Fluoride:

High concentration of fluride in ground water may be due to break down of rocks and soil or infiltration of chemical fertilizers from agricultural land . Skeletal fluorosis is on important disease due to presence of high fluoride content in ground water (Mohamed M Hanifa, et al., 2013) Fluoride concentration in the study was in the range of 0.1-0.8 mg/l. All the values are within the desirable limit.

Sulphate:

High concentration fo sulphate may cause gastro intestinal irritation at particularly when magnesium and sodium irons are also present in drinking water resources (Indirani Gupta et,al.,2011). The sulphate concentration varied between 4 mg/l-194mg/l .Desirable limit of sulphate in drinking water is 200 mg/l. All samples are within the desirable limit

V. CONCLUSION

The above studies shows that the ground water in and around SIPCOT in Cuddalore district is not affected with respect to turbidity, TDS ,Total alkalinity ,Total hardness, calcium, Magnesium, Chlorides are exceeding the limitis in the. Villages such as Kudikadu, Sonnanchavadi ,Echankadu ,Vairankuppam, Thachan colony, Periyakaraikadu are lie in a virtual gas chamber surrounded on three sides by chemical industries .Ground water from these villages got polluted due to the industrial discharges,industrial activities and of from illegal dumpings of toxic wastes

REFERENCES:

[1] R.Rajamanickam and S.Nagar,Studies on physio-chemical characteristics ground water in Amaravathy river basin of Karur district,Tamil nadu.2020,29(1):153
[2] Kavitha Kirubaathy.A,Ground water quality of orathupalayam village,Erode district,Tamil nadu,Environ.Monit 2010 20(4):389-392
[3] APHA Standards Methods for the examination of water and wastewater,American public health association,Washington DC.1998,18th Ed.
[4] Sudhir Dahiya and Amarjeet kaur,physio chemical characteristics of underground water in rural areas of tosham subdivision,Bhiwani district,Haryana,J.Environ poll,1999,6(4)281
[5] "R.Rajamanickam and S.Nagar" Impact of Textile Dying Industries on ground water quality in Karur Amaravathi river basin,Tamil nadu-A field study,J.Environ,Science & Engg.Vol.52,2010,No.4,P.315-320
[6] Mohamed Hanipha M. and Zahir Hussain A. Study of ground water quality at Dindigul Town.Tamilnadu,India, International Research Journal of Environment Sciences Vol,2(1), 68-73, January (2013)
[7] Bhattacharya T., Chakraborty S. and Tuck Neha., Physico chemical Characterization of ground water of Anand district, Gujarat, India, I. Res. J. Environment Sci., 1(1), 28-33 (2012)
[8] Zahir Hussain A. and Abdul Jameel. M., Monitoring the quality of groundwater on the bank of Uyyakondan channel of river Cauvery at Tiruchirappalli, Tamilnadu, India, Environmental Monitoring and Assessment, 10.10007/s 10661, 011, 1910–14 (2011)
[9] Chari K.V.R. and Lavanya M.G., Groundwater contamination in Cuddapah urban area, Andhra Pradesh, In Proceedings on regional Workshop of Environmental aspects of groundwater development. KU, Kurukshetra Oct. 17-19, Kurukshetra, India, 130-134 (1994)
[10] Chadrik Rout and Arabinda Sharma., Assessment of drinking water quality, a case study of Ambala cantonment area, Hariyana, India, International Journal of Environmental Sciences, 2(2), 933–945 (2011)
[11] Varadarajan N., Purandara B.K. and Bhism Kumar, Assessment of groundwater quality in Ghataprabha Command area, Karnataka, India, J. Environ. Science and Engg. 53(3), 341-348 (2011)
[12] Venkateswara Rao B., Physico-chemical analysis of selected groundwater samples of Vijayawada rural and urban in Krishna district, Andhra Pradesh, India, International Journal Environmental Sciences, 2(2), 710-714 (2011)
[13] Basic Information in Nitrates in Drinking Water, Basic information about Regulated Drinking Water Contaminants, US-EPA-Environment Protection Agency(2012)