ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

IJIRSET

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

A Decentralized Service Discovery Approach on Peer-To-Peer Networks

P.Thangalakshmi, S. Kamalesh

PG Scholar, M.E Network Engineering, Velammal College of Engineering and Technology, Madurai , India.

Assistant Professor, Department of Information Technology, Velammal College of Engineering and Technology, Madurai, India.

***Abstract*—** This project is Service-Oriented Computing (SOC) is rising as a paradigm for developing distributed applications. An essential issue of utilizing SOC is to own an ascendible, reliable, and strong service discovery mechanism. However, ancient service discovery strategies mistreatment centralized registries will simply suffer from issues like performance bottleneck and vulnerability to failures in massive ascendible service networks, therefore functioning abnormally. During this paper we have a tendency to propose a P2P call tree induction algorithmic program during which each tree peer learns and maintains the proper call compared to a centralized state of affairs. Our algorithmic program is totally decentralized, asynchronous, and adapts swimmingly to changes within the information and also the network. In meta-algorithm that uses completely different classification models in line with a learning technique known as boosting. The algorithmic program could be a learning methodology which will handle weighted instances. Finally, we propose a rule induction algorithm to produces better accuracy than the original decision tree classifier.

Keywords— J48, Peer- to- Peer (P2P), WEKA

I. INTRODUCTION

Decentralized P2P networks have many benefits over traditional client-server networks. These networks scale indefinitely while not increasing search time or the requirement for costly centralized resources. They utilize the process and networking power of the end-users' machines since these resources forever grow in direct proportion to the network itself. As computing and communication over wired and wireless networks advanced, several pervasive distributed computing environments like net internet, intranets, LANs, adhoc wireless networks, and P2P networks have emerged call tree may be powerful applied math and Machine learning technique is wide used for information classification, Prognostic modelling and additional. Given a collection of

learning example (attribute values and corresponding category labels) at a one location, there exist many well-known methods to build a call tree like ID3 and C4.5. However, there can be several situations in which the data is distributed over a large, dynamic network containing no special server or client nodes. A typical example is a Peer-to-Peer (P2P) network. Performing data mining tasks such as building decision trees is very challenging in a P2P network because of the large number of data sources, the asynchronous nature of the P2P networks, and dynamic nature of the data. A scheme which centralizes the network data is not scalable because any change must be reported to the central peer, since it might very well alter the result. In this paper we propose a P2P decision tree induction algorithm in which every peer learns and maintains the correct decision tree compared to a centralized scenario. Our algorithm is completely decentralized, asynchronous, and adapts smoothly to changes in the data and the network. The algorithm is efficient in the sense that as long as the decision tree represents the data, the communication overhead is low compared to a broadcast-based algorithm. As a result, the algorithm is highly scalable. When the data distribution changes the decision tree is updated automatically. Our work is the first of its kind in the sense that it induces decision trees in large P2P systems in a communication-efficient manner without the need for global synchronization and the tree is the same that would have been induced given all the data to all the peers.

II. MOTIVATION

P2P networks are quickly emerging as large-scale systems for information sharing. Through networks such as Kazaa, e-Mule, Bit Torrents, consumers can readily share vast of amounts of information. While initial consumer interest in P2P networks was focused on the value of the data, more recent research such as P2P web community formation argues that the consumers will greatly benefit from the knowledge locked in the data. For instance, music recommendations and sharing systems are a thriving industry today a sure sign of the value consumers have put on this application. However, all existing systems require that users submit their listening habits, either explicitly or implicitly, to centralized processing. Such centralized processing can be problematic because it can result in severe performance bottleneck. Wolff *et al.* have shown that centralized processing may not be a necessity by describing distributed algorithms which compute association rules (and hence, recommendations) in-network in a robust and scalable manner. Later, Gilbert *et al* Showed that it is relatively easy, given an in-network knowledge discovery algorithm, to produce a similar algorithm which preserves the privacy of users in a well defined sense.[10] Have demonstrated the collaborative use of features for organizing music collections in a P2P setting. Users would prefer not to have the organization of their data change with every slight change in the data of the rest of the users. At the same time, users would also not like the quality of that organization to degrade a lot. As suggested in this paper, monitoring the change and updating the model seems to hit a good balance between being up-to-date and robust. Another application which offers high value to the consumers is failure determination. In failure determination, computer-log data which may have relation to the failure of software and this data are later analyzed in an effort to determine the reason for the failure. Data collection systems are today integral to both the Windows and Linux operating systems. Analysis is performed off-line on a central site and often uses knowledge discovery methods. Still, home users often choose not to cooperate with current data collection systems because they fear for privacy and currently there is no immediate benefit to the user for participating in the system. Collaborative data mining for failure determination can be very useful in such scenarios, and resolve concerns of both privacy and efficiency.1 However, failure determination models can be quite complex and developing such models require a lot of expert knowledge. The method we describe allows computing a model centrally, by whatever means deemed suitable, and then testing it against user's data without overloading the users or requiring them to submit their data to an untrusted party.

III. LITERATURE SURVEY

Chord to distribute and discover services are a decentralized manner. Based on the service publication approach, Chord4S supports QoS-aware service discovery. [1] AdaBoost has been used for distribute analysis and in parallel processing so far. The learning algorithm is able to handle instances weighted instances change the way it calculates the classifier error. [2] This paper was developed algorithm by Ross Quinlan. The decision tree techniques (J48 (C4.5), NBTree and in data mining were evaluated and compared on basis of accuracy and Error Rate. [5]. this paper also analyzes data set properties to find relations between them and the classification algorithms and pruning methods. [9] It is utilize message-based connectivity structure can considerably reduce the messaging cost, and provide better utilization of resources, which in turn improves the quality of service of the applications executing over decentralized peer-to-peer networks[16]

IV. PROBLEM SCOPE OF THE PROJECT

Traditional service discovery approaches of the web services technology are based on Universal Description, Discovery, and Integration (UDDI). However, centralized service registries used by UDDI may easily suffer from problems in an open SOC environment. To overcome the problems The Peer-to-Peer (P2P) technology provides a universal approach to improving reliability, scalability, and robustness of distributed systems by removing centralized infrastructures. Based on Distributed Hashing Table (DHT), structured P2P systems can achieve even data distribution and efficient query routing by controlling the topology and imposing constraints on the data distribution. In this technology also problems occur. It is suffered from such as Problems; they are Bottleneck, Vulnerability to failure in large scale network. Large scale service network, largely distributed, unexpected failure of nodes cannot be avoided high cost and loss of control.

V. ARCHITECTURE

P2P decision tree induction algorithm in which every peer learns and maintains the correct decision tree compared to a centralized scenario. Our algorithm is completely decentralized, asynchronous, and adapts smoothly to changes in the data and the network, shown in Fig.1The client request the service to P2P then collect the dataset and using the ID3 algorithm is efficient in the sense that as long as the decision tree represents the data, the communication overhead is low compared to a broadcast-based algorithm. As a result, the algorithm is highly scalable. When the data distribution changes, the decision tree is updated automatically Our work is the first of its kind in the sense that it induces decision trees in large P2P systems in a communication-efficient manner without the need for global synchronization and the tree is the same that would have been induced given all the data to all the peers.

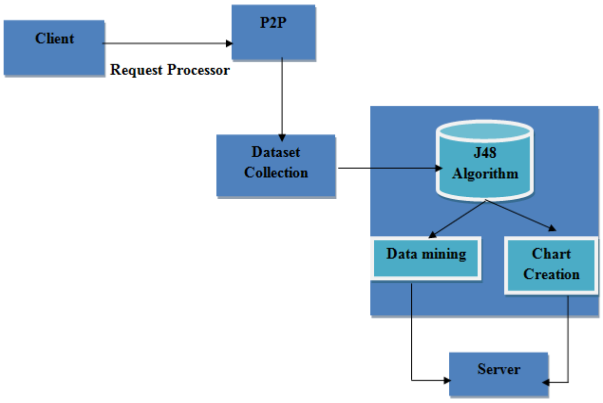


Fig. 1 Decentralized P2P Network in Data mining

VI.METHODOLOGIES

A. Data set Creation

A collection of related sets of information that is composed a separate elements but can be manipulated as a unit by a computer. Datasets consist of all of the information gathered during a survey which needs to be analyzed. Learning how to interpret the results is a key component to the survey process.

Dataset

```
0,tcp,auth,SF,9,36,0,0,1,0,0,1,normal
0,tcp,daytime,S0,0,0,0,0,0,0,0,249,DOS
0,tcp,auth,SF,10,37,0,0,1,0,0,1,normal
0,tcp,auth,SF,10,38,0,0,1,0,0,1,normal
0,tcp,daytime,S0,0,0,0,0,0,0,0,252,DOS
0,tcp,daytime,S0,0,0,0,0,0,0,0,217,probe
0,tcp,daytime,SH,0,0,0,0,0,0,0,1,probe
0,tcp,daytime,S0,0,0,0,0,0,0,0,245,DOS
0,tcp,bgp,S0,0,0,0,0,0,0,0,249,DOS
0,tcp,bgp,S0,0,0,0,0,0,0,0,250,DOS
0,tcp,bgp,S0,0,0,0,0,0,0,0,253,DOS
```

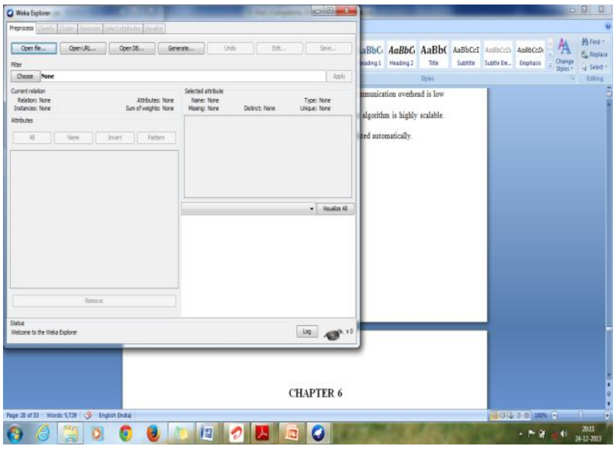


Fig. 2 Open Weka

B. Upload Data set

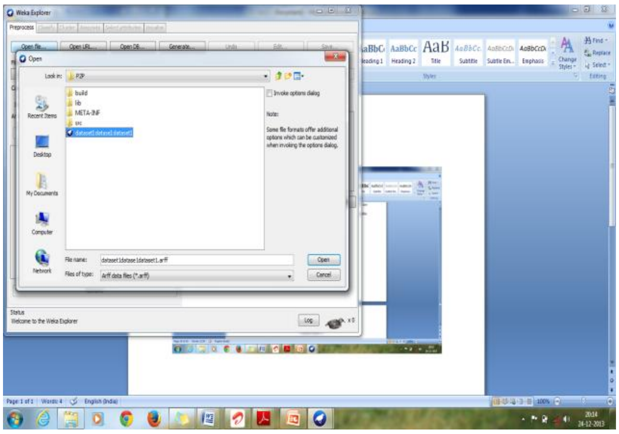


Fig. 3 Dataset Upload

C. J48 Implementation

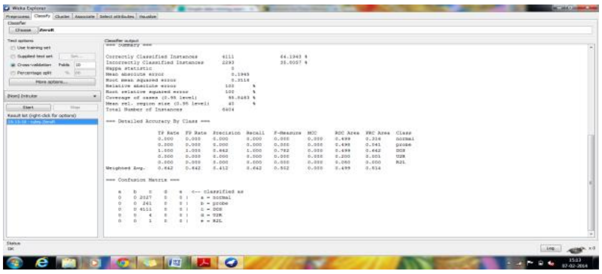
J48 [QUI93] implements Quinlan‟s C4.5 algorithm [QUI92] for generating a pruned or un pruned C4.5 decision tree. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by J48 can be used for classification. J48 builds decision trees from a set of labeled training data using the concept of information entropy. It uses the fact that each attribute of the data can be used to make a decision by splitting the data into smaller subsets. J48 examines the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. To make the decision, the attribute with the highest normalized information gain is used. Then the algorithm recurs on the smaller subsets. The splitting procedure stops if all instances in a subset belong to the same class. Then a leaf node is created in the decision tree telling to choose that class. But it can also happen that none of the features give any information gain. In this case J48 creates a decision node higher up in the tree using the expected value of the class. J48 can handle both continuous and discrete attributes, training data with missing attribute values and attributes with differing costs. Further it provides an option for pruning trees after creation.

Program:

```
package weka.classifiers.trees;
import weka.classifiers.AbstractClassifierTest;
import weka.classifiers.Classifier;
import junit.framework.Test;
import junit.framework.TestSuite;
public class J48Test extends AbstractClassifierTest {
public J48Test(String name) { super(name);  } /**
Creates a default J48 */
public Classifier getClassifier() {
return new J48();
}
public static Test suite() {
return new TestSuite(J48Test.class);
}
public static void main(String[] args){
junit.textui.TestRunner.run(suite());
}
}
```

D. Resource Allocation

Resource allocation is the scheduling of activities and the resources required by those activities while taking into consideration both the resource availability and the project time. Cost of resources varies significantly depending on configuration for using them. The classifier is evaluated by cross-validation, using the number of folds that are entered in the Folds text field.

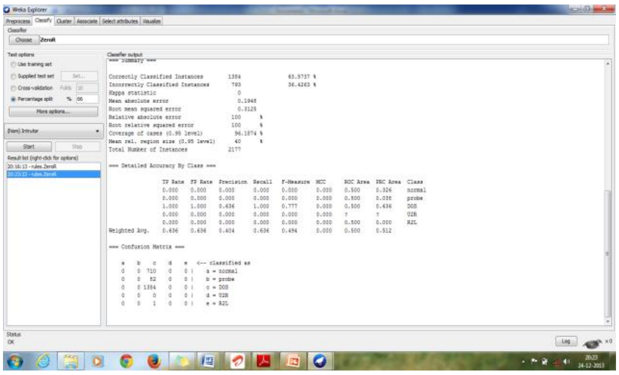


```
=== Summary ===

Correctly Classified Instances        4111               64.1943 %
Incorrectly Classified Instances      2293               35.8057 %
Kappa statistic                          0
Mean absolute error                      0.1945
Root mean squared error                  0.3118
Relative absolute error                100      %
Root relative squared error            100      %
Coverage of cases (0.95 level)          95.8463 %
Mean rel. region size (0.95 level)      40      %
Total Number of Instances             6404

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    0.000      0.000    0.000      0.000    0.499     0.316     normal
                 0.000    0.000    0.000      0.000    0.000      0.000    0.498     0.041     probe
                 1.000    1.000    0.642      1.000    0.782      0.000    0.499     0.642     DOS
                 0.000    0.000    0.000      0.000    0.000      0.000    0.200     0.001     U2R
                 0.000    0.000    0.000      0.000    0.000      0.000    0.060     0.000     R2L
Weighted Avg.    0.642    0.642    0.412      0.642    0.502      0.000    0.499     0.514

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
    0    0 2027    0    0 |    a = normal
    0    0  261    0    0 |    b = probe
    0    0 4111    0    0 |    c = DOS
    0    0    4    0    0 |    d = U2R
    0    0    1    0    0 |    e = R2L
```

Fig. 4Cross-validation test

E. Percentage Validation Test

The classifier is evaluated on how well it predicts a certain percentage of the data which is held out for testing. The amount of data held out depends on the value entered in the % field.



```
=== Summary ===

Correctly Classified Instances        1384               63.5737 %
Incorrectly Classified Instances       793               36.4263 %
Kappa statistic                          0
Mean absolute error                      0.1948
Root mean squared error                  0.3125
Relative absolute error                100      %
Root relative squared error            100      %
Coverage of cases (0.95 level)          96.1874 %
Mean rel. region size (0.95 level)      40      %
Total Number of Instances             2177

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.326     normal
                 0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.038     probe
                 1.000    1.000    0.636      1.000    0.777      0.000    0.500     0.636     DOS
                 0.000    0.000    0.000      0.000    0.000      0.000    ?         ?         U2R
                 0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.000     R2L
Weighted Avg.    0.636    0.636    0.404      0.636    0.494      0.000    0.500     0.512

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
    0    0  710    0    0 |    a = normal
    0    0   82    0    0 |    b = probe
    0    0 1384    0    0 |    c = DOS
    0    0    0    0    0 |    d = U2R
    0    0    1    0    0 |    e = R2L
```

Fig 5 Percentage validation test in weka

F. Visualizing

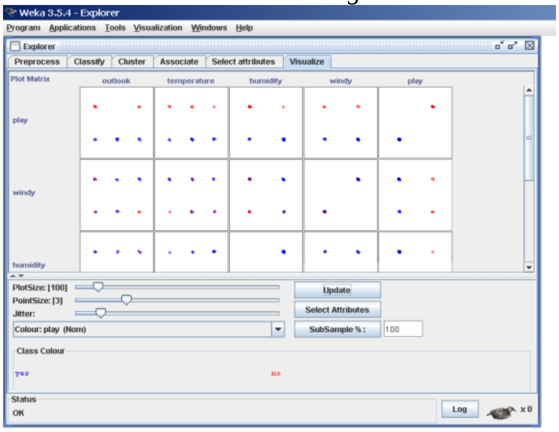
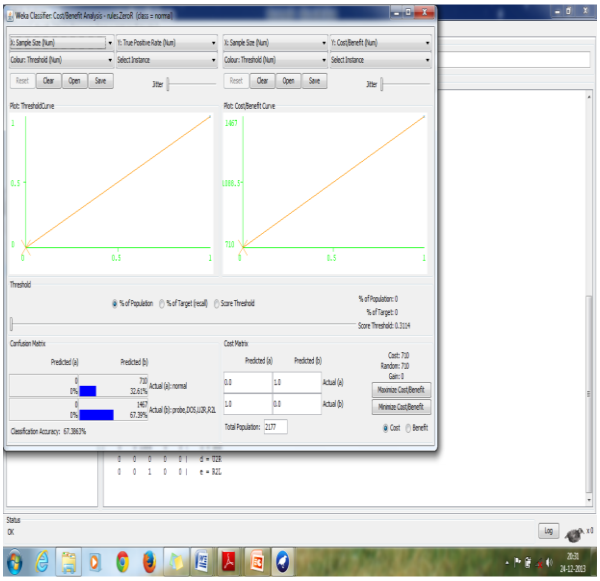


Fig. 6 Visualizing

G.Cost/Benefit Analyze [Normal]



Weka Classifier: Cost/Benefit Analysis - rules.ZeroR (class = normal)

Threshold: ● % of Population ○ % of Target (recall) ○ Score Threshold

% of Population: 0
% of Target: 0
Score Threshold: 0.3114

Confusion Matrix

	Predicted (a)	Predicted (b)	
	0 0%	710 32.61%	Actual (a): normal
	0 0%	1467 67.39%	Actual (b): probe,DOS,U2R,R2L

Classification Accuracy: 67.3863%

Cost Matrix

Predicted (a)	Predicted (b)	
0.0	1.0	Actual (a)
1.0	0.0	Actual (b)

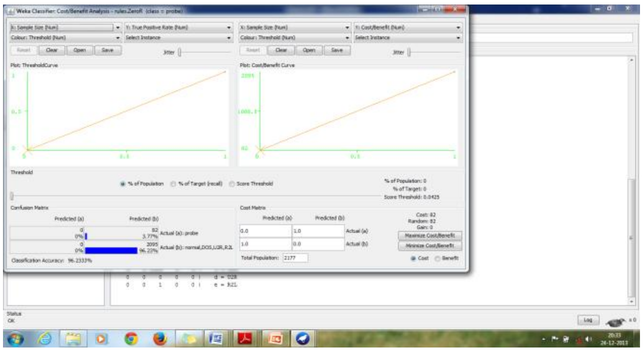
Total Population: 2177

Cost: 710
Random: 710
Gain: 0

● Cost ○ Benefit

Fig. 7 Normal Cost Analyze

H. Cost/Benefit Analyzes [PROB]



Weka Classifier: Cost/Benefit Analysis - rules.ZeroR (class = probe)

Threshold: ● % of Population ○ % of Target (recall) ○ Score Threshold

% of Population: 0
% of Target: 0
Score Threshold: 0.0425

Confusion Matrix

	Predicted (a)	Predicted (b)	
	0 0%	82 3.77%	Actual (a): probe
	0 0%	2095 96.23%	Actual (b): normal,DOS,U2R,R2L

Classification Accuracy: 96.2333%

Cost Matrix

Predicted (a)	Predicted (b)	
0.0	1.0	Actual (a)
1.0	0.0	Actual (b)

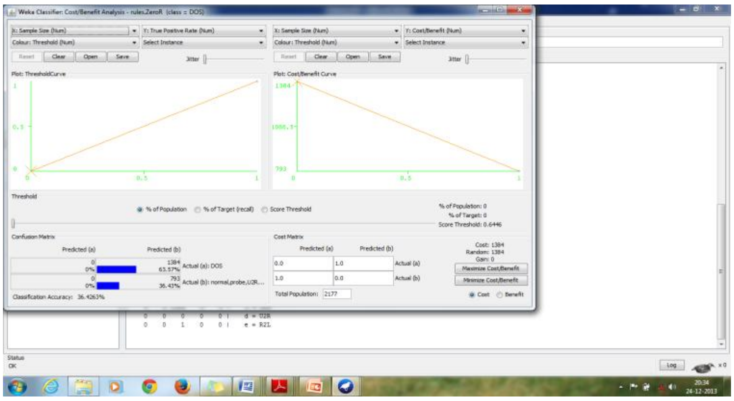
Total Population: 2177

Cost: 82
Random: 82
Gain: 0

● Cost ○ Benefit

Fig. 8 PROB **C**ost Analyze

I. Cost/Benefit Analyzes [DOS]



Weka Classifier: Cost/Benefit Analysis - rules.ZeroR (class = DOS)

Threshold: ● % of Population ○ % of Target (recall) ○ Score Threshold

% of Population: 0
% of Target: 0
Score Threshold: 0.6446

Confusion Matrix

	Predicted (a)	Predicted (b)	
	0 0%	1384 63.57%	Actual (a): DOS
	0 0%	793 36.43%	Actual (b): normal,probe,U2R...

Classification Accuracy: 36.4263%

Cost Matrix

Predicted (a)	Predicted (b)	
0.0	1.0	Actual (a)
1.0	0.0	Actual (b)

Total Population: 2177

Cost: 1384
Random: 1384
Gain: 0

● Cost ○ Benefit

Fig. 9 DOS Cost Analyze

J.Mesurement Metrics

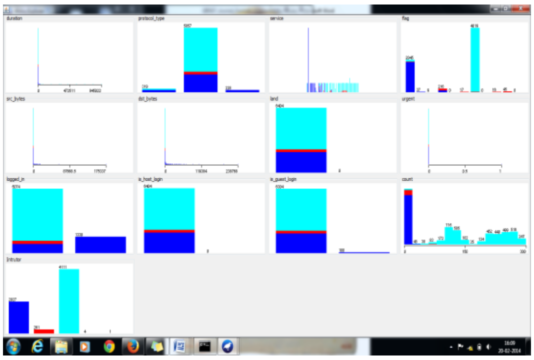


Fig. 10 Measurement Graph

VII. CONCLUSIONS

P2P data processing is recently emerged from distributed data Pre-processing that deals with knowledge analysis in environments with distributed knowledge, computing nodes, and users This survey reveals that albeit P2P data processing algorithms addresses a number of the higher than mentioned options one or different, none of them cowl all the desired options. Analysis towards developing such algorithms has to be compelled to be thought-about. Our algorithmic rule is totally localized, asynchronous, and adapts swimmingly to changes within the knowledge and also the network. The algorithmic rule is economical within the sense that as long because the call tree represents the information, the communication overhead is low compared to a broadcast-based algorithmic rule. As a result, the algorithmic rule is very ascendable. once the information distribution changes, the choice tree is updated mechanically.

VIII. Future Work

For the future work, scenario reduction techniques will be applied to reduce the number of scenarios. In addition, the optimal pricing scheme for cloud providers with the consideration of competition in the market will be investigated. Scenario reduction techniques will be applied to reduce the number of scenarios. In addition, the optimal pricing scheme for cloud providers with the consideration of competition in the market will be investigated. We need to predict the future resource needs of VMs. As said earlier, our focus is on Internet applications. One solution is to look inside a VM for application level statistics, e.g., by parsing logs of pending requests. Doing so requires modification of the VM which may not always be possible. Instead, we make our prediction based on the past external behaviors of VMs.

ACKNOWLEDGMENT

We would like to thank our Assistant Professor S.Kamalesh for motivating us in doing such kind of my projects. And thanks to all the web sources.

REFERENCES

[1] Q. He, J. Yan, Y. Yang, R. Kowalczyk and H. Jin, IEEE TRANSACTIONS ON SERVICES COMPUTING, TSC-2009-01-0001C

[2] Department of Computer Science and Engineering, Department of Electrical, Electronic and Information Engineering "G. Marconi", CIRI - ICT, Information and Communication Technologies, University of Bologna, Via Venezia 52, I-47521 Cesena (FC), Italy.

[3] LionShare,2006, http://lionshare.psu.edu/

[4] LimeWire,2008 , http://www.limewire.com/

[5] Jay Golan, Department Of Computer Engineering, College Of Engineering, Pune, Maharashtra, India

[6] B. Raahemi, A. Hayajneh and P. Rabinovitch, Peer-to-Peer IP Traffic Classification Using Decision, International Journal of Business Data Communications and Networking, Volume 3, Issue 4, edited by Jairo Gutierrez © 2007, IGI Global

[7] Ioannidis, S. and Marbach, P. On the Design of Hybrid Peer-to-Peer Systems, SIGMETRICSŠ08, June 2- 6,2008, Annapolis, Maryland, USA.

[8] Frank Dabek, Emma Brunskill, M. Francs Kai-shek, David Karger Robert Morris, Ion Stoica□, Hari Balakrishnan MIT Laboratory for Computer Science

[9] Artificial Intelligence Author: Patrick Ozer Radboud University Nijmegen January 2008.

[10] Kanishka Bhaduri_, Ran Wolff†, Chris Giannella‡ Hillol Kargupta _Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County,1000 Hilltop Circle, Baltimore, Maryland, 21250

[11] Anis ISMAIL1, Aziz BARBAR1 and Ziad ISMAIL2,1 American University of Science & Technology Alfred Naccash Avenue – Ashrafieh,Beirut, Lebanon,2 TELECOM ParisTech Paris, France,ismail.ziad@enst.fr

[12] Ion Stoica□, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan_ MIT Laboratory for Computer Science chord@lcs.mit.edu http://pdos.lcs.mit.edu/chord/

[13] G.Forman and B. Zhang. "Distributed Data Clustering Can Be Efficient and Exact" SIGKDD Explorations, 2(2):34–38, 2000.

[14] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[15] Anis ISMAIL, Aziz BARBAR American University of Science & Technology – AUST Alfred Naccash Avenue – Ashrafieh Beirut, Lebanon

[16] Lakshmish Ramaswamy College of Computing Georgia Institute of Technology Altanta, GA 30324, Bugra Gedik College of Computing Georgia Institute of Technology Altanta, GA 30324

[17] K. Liu, K. Bhaduri, K. Das, P. Nguyen, and H. Kargupta, "Client-side Web Mining for Community Formation in Peer-to-Peer Environments," *SIGKDD Explorations*, vol. 8, no. 2, pp. 11–20, 2006.

[18] Agrawal, R., Mehta, M., Shafer, J., Srikant, R., Arn- ing, A. & Bollinger, T. (1996), The Quest data mining system, in `Proceedings of the 2nd International Conference on Knowledge Discovery in Databases and Data Mining', pp. 244{249}.

[19]Breiman L. (1999), Pasting small votes for classification in large databases and on-line', Machine Learning 36(1), 85{103}.

[20]Breiman, L. & Spector, P. (1994), Parallelizing CART using a workstation network, in `Proc Annual American Statistical Association Meeting'.

[21]"Amazon elastic compute cloud (Amazon EC2)," http://aws.amazon.com/ec2/, 2012.