International Journal of Computer Applications (0975 – 8887)
Volume 19– No.3, April 2011

A Statistical Approach for the Assessment of QOS and Performance in Grid Computing Environment

Jahangir khan
Department of computer science & IT
Sarhad University Of Science & Information Technoloogy
Peshawar 25000, Pakistan

Farhan Shafiq
Graduate School Of Science & Engineering,PAF-KIET, PAF Base Korangi Creek
Karachi75190, Pakistan

Muhammad Khalid khan
Graduate School Of Science & Engineering,PAF-KIET, PAF Base Korangi Creek
Karachi75190, Pakistan

ABSTRACT
Grid computing enables resource sharing and dynamic allocation of distributed heterogeneous computational resources while minimizing the associated ownership and operating cost. In grid computing there is no matter where data located is or which computer processes a task. Quality of Service parameters plays a key role in selecting Grid resources and optimizing resources usage efficiently. This research study focuses on the best resource selection and allocation for ensuring the QoS in a grid environment. We have presented a new approach inspired by statistical ranking and selection schema for best resource selection to get optimal results and better decision making. For QoS over the grid, we have considered few parameters of each node on the available grid. Based on the values of those parameters, we have developed a ranking and threshold selection criteria for each parameter at each node. The ASS scheme will consider only those nodes for the Next Grid (NGrid) that have at least X% free resources at each node. Using this proposed scheme we prepared a training dataset. Using training dataset, we applied multiple binary logistic regression over the training data and prepared a binary logit model. This data model will work as the predictor for the selection of only those nodes for NGrid that have at least X% free resources. Experiment considered a Grid environment composed of heterogeneous resources. In the data training phase 301 nodes are examined and we get 94.0 % prediction accuracy while prediction accuracy in validation phase is 96.79%. in the validation phase no of nodes are 156. While the Model Error is 3.21% . The main concern for future work is to develop an automated tool which gives the fully support to our mechanism in a grid environment, also we pretend to extend this approach measuring and modeling network quality of service (QoS) parameters.

General Terms
QOS, Analysis, Treshold value , Testing and analysis.

Keywords
CPU clock, Grid protocol Architecture, Efficiency evaluation, Quality of services, next Grid.

1. INTRODUCTION
Grid computing has the potential to drastically change enterprise computing as we know today. This advancement of technology play great role in the field biological sciences, Earth science, physics, Astronomy and mathematical research. The academia and the industry take great advantage of these significant changes in computing world. The main concept of grid computing is viewing computing as a utility. Grid computing enables resource sharing and dynamic allocation of distributed heterogeneous computational resources while minimizing the associated ownership and operating cost. These resources may be geographically distributed over the network or internet and follows heterogeneous protocols made by different vendors. In grid computing there is no matter where data located is or which computer processes a task.Computing grids and electrical power grids works in same manner. They have some common attributes in their infrastructure. An electrical power grid supplies the electricity to their customers at a cost e.g. X Rs./ watt hour . Grids may provide different services for computation to users. These are usually own by different organization and sites. If some one on the grid wants to use these services they can use it and pay charges for the services used to service providers at a specified costing modules. There is a service level agreement (SLA) established between the user and the Grid Service provider (GSP). This agreement furnishes all the conditions for using service. In order to use these recourses in an efficient manner the recourse allocation manager must be able to select the best service for their tasks, based on criterion of the service required. Quality of Service (QoS) metrics plays an important role in selecting and optimizing grid resources usage efficiently.Grids are the heterogeneous and complex systems composed of larger number of distributed components belonging to diverse domains. Assuring the QoS in this kind of system is a critical job because a number of vendors may offer similar services to the users on the grid. These services providers would have their own costing mechanism and may allocated resources according to a pre-specific criterion. A user often uses go for the lowest budget service for the specific task. Lowest budget service however may not give the best QoS or optimum results. Since solution of the problem often require multiple tasks choosing service on the basis of lowest budget criterion would lack optimal, resulting in poor decision making or poor results.

In Grid computing , QOS is not limited to network bandwidth but extends to the processing and storage capabilities of the nodes. Thus the focus is on the degree a Grid can provide endto-end QOS rather than providing only QOS on the network. When a Grid job has QOS requirements, it may be necessary to negotiate a service level agreement (SLA) to enforce the desired level of service. Resource reservation is one of the ways of providing guaranteed QOS in a Grid with dedicated resources [21]. User's demands could be described as QoS requirements, a QoS set consists of all these requirements to depict qualitative characteristics of a grid service. So grid service matchmaking is to compare corresponding elements in the QoS set between two services[5]. In order to use distributed resources efficiently, users supervised by the resource management system and the scheduler to select the best resources for their tasks, based on quality aspects of resources. Different type of QoS parameter acquire the attention of the user because QoS parameter play a vital role in selecting the best resources an negotiating service level agreement (SLA) between user and service provider. Service level Agreement (SLA) is a contract between service provider and the user. This agreement defines the terms and conditions to use the service and usually associate a cost and penalties with desire level of service.

The paper is organised as : Section 2 describes Grid computing, Section 3 briefs about implementation and testing, Section 4 proceeds for the results, second last section describes proposed approach validation and lat one concludes the paper.

2. GRID ARCHITECTURE

There are two important grid architectures: One of them is the five-layer sand- leakage model raised by Ian Foster; another is he Open Grid Service Architecture raised by IBM and Globus Alliance and Web Service.

2.1 Sand-Leakage Model

The functions of operation, management and usage sand leakage model to share resources are separated to five different layers. The more to low, the more it approaches to the physical sharing and related to the particular source; the more to up layer, the more the detailed characters of sharing sources couldn't be felt and are related to the abstract resources and don't take care of the low layer resources. Ian Foster and others define the sand leakage model to be five layers: They are fabric, connectivity, resource, collective and application. Figure 1 shows the send leakage model [2] than main attention of the architecture is on the interoperability among resource providers ad users to establish the sharing relationship. This interoperability means common protocols at each layer of the architecture model which leads to

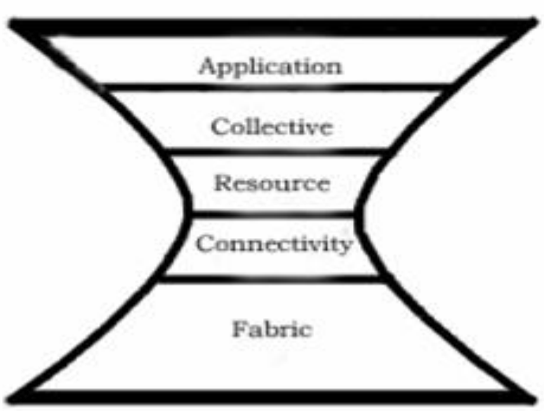


Figure 1. Send Leakage Model

the definition of grid protocol architecture as shown in Figure 2.

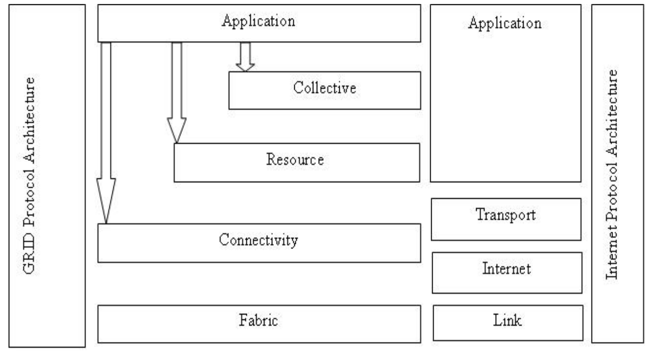


Figure 2. Grid Protocol Architecture

2.2 Open Grid Service Architecture (OGSA)

The grid infrastructure is mainly concerned with the creation , management, and the application of dynamic coordianated resources and services. These dynamic and coordinated resources and services are complex. They may be individual or collection of entities with a short or long lifespan. These resources may be constituted from single or from multiple institution so as to provide a homogenous set of functionalities. Even though the complexity and difference in resources and services may vary within every virtual organization, they are all agreed to deliver a QoS features including common security semantics., work flow resource management, problem determination, failover, and service level management. The QoS features require a well-defined architecture to achieve the desire level of service quality. This prompted for the introduction of Open Grid Service Architecture (OGSA) to support the creation , maintenance and application of ensembles of service maintained by virtual organization (VO). (Foster, Kesselman, & tuecke). OGSA architecture is layerd architecture, as shown in Figure 3 , with clear separation of the functionality at each layer. The core architecture layer are Open Grid Service Infrastructure (OGSI), which provides the base infrastructure , and OGSA core platform services , which are a set of standard service including policy, logging, service level management ,and so on. The high level applications and services use these lower layer core platform components and OGSI that become part of a resource sharing grid [1].

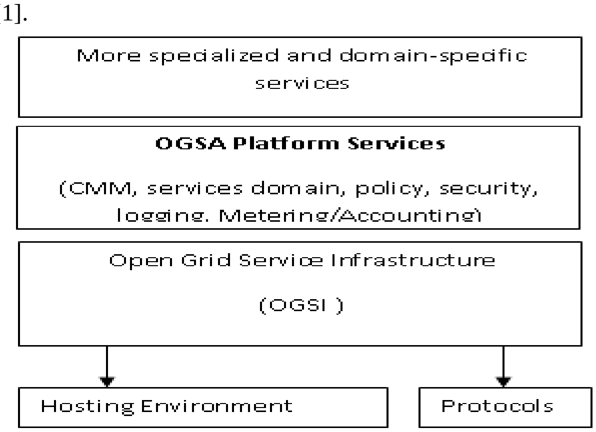


Figure 3. OGSA Platform Architecture

2.3 Service Computing

A distributed system consists of a set of software agents that all works together to implement some intended functionality. The agents in a distributed system do not operate in the same processing environment, so they must communicate by hardware/software protocols stack that are intrinsically less reliable then direct code invocation and shared memory. Service oriented architecture (SOA) is a specific type of distributed system in which the agents are software services that perform some well-defined operation and this type of architecture can be invoked outside of the context of a large application. This can be inferred a service as acting as a user-facing software component of a large application.In a service oriented Architecture (SOA) business purchases functionality in chunks. Rather than buy huge software permanently, organizations will buy services as they needed much like buying an air ticket for traveling as opposed to buy an aero plane. The most common understood SOA architecture is Web and Web services. Consequently grid community expressed interest in service oriented computing. A service is natural progression from component based software development and as a mean to integrate different component development framework. A service may be defined as a behavior that is provided by a component for use by another component base on network addressable interface contract. Such a contract specifies the set of operations that one service can invoke on another. Such an interface may also contain additional attributes like performance of the service the cost of accessing and using the service and detail of ownership and access rights associated with services.

2.4 Charging/Pricing of the service.

In grid computing environment because resources are distributed, heterogeneous in nature and obey diverse protocols and having different costing models in this kind of system pricing of the service is most important and crucial job. To illustrate To illustrate the problem, let's consider the follow scenario from paper [3] where there is great difference in the processing power between the devices, a doctor has in your hands a PDA, he works in a heart care center. Into your device, the doctor keeps the last tomography images of a patient with a serious sickness. For an exact analysis of the patient's condition, they request a 3D Reconstruction for this case, this task require a large computational power and the PDA can't realize this task, so the PDA, that are connected to a Grid, begin a auction with another node, request a budget for this service. So, some devices response the request, and the PDA elect the node that have the lower price to do the task. After the task is complete, the doctor analyze the 3D image and if the quality is approved the cost of the service is computed in the patient's care plan. In the above scenario many grid nodes respond against the request of the user according to their computational power and ability to generate the 3D image of the given task. Among the different service providers only the low price service being selected to solve the problem. As far as low price concern user satisfaction is also mandatory means the out put of the submitted job must satisfy the user requirement means the acquired service is a low price as well as a quality service. May be the job submitted to a low price service is not a quality service and have not the computational power to solve the problem at desired level of satisfaction. In grid environment where we can chose the service from different published services it must be guaranteed the quality of service(QoS).

3. IMPLEMENTATION AND TESTING

For validating our model, we examine 300 nodes. All nodes having Windows Xp Professional Operating System, 512 M.B of RAM , variable Storage Capacity(HDD) in Giga bytes(GB) and 2.0 GHz, 2.4 GHz, 2.53 GHz and 2.8 GHz Intel processors.

Parameters Values

In this experiment we use four type of parameters given as below.

- C.P.U. Speed
- C.P.U. Load(C.P.U Usage in %)
- R.A.M_available
- Storage_Capacity_avialable

For extracting the value of available strength of above parameters, we build a simple component which extract parameter information from each node. Figure 4 shows the value of each parameter at each node fetch by the component.

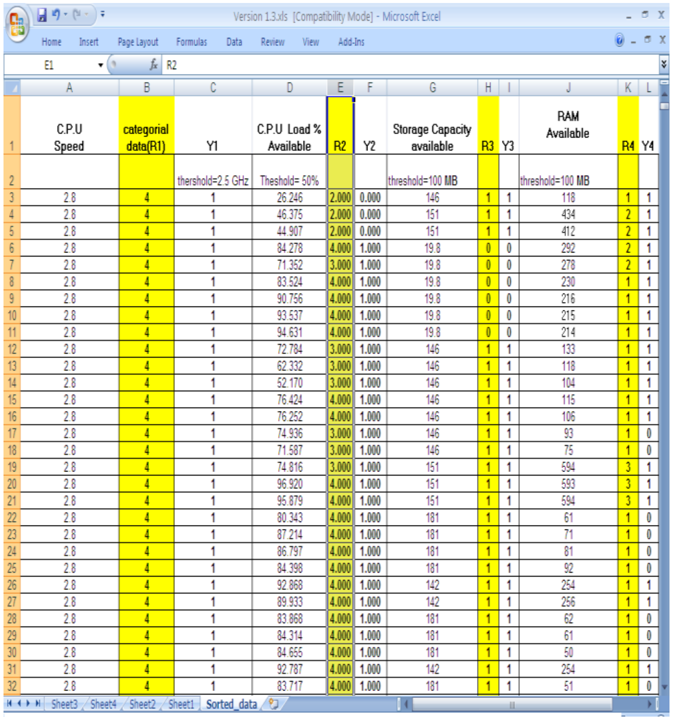
	A	B	C	E	F	G	H	I
1	C.P.U Speed	C.P.U Usage	C.P.U Load % available	Storage Capacity available	Total RAM	RAM Available	OSName	
2	2.8 GHz	54.24236%	45.76%	146.GB	512 MB	134 MB	Microsoft Windows XP Professional	
3	2.8 GHz	73.75436%	26.25%	146.GB	512 MB	118 MB	Microsoft Windows XP Professional	
4	2.8 GHz	27.21615%	72.78%	146.GB	512 MB	133 MB	Microsoft Windows XP Professional	
5	2.8 GHz	37.6681%	62.33%	146.GB	512 MB	118 MB	Microsoft Windows XP Professional	
6	2.4 GHz	36.57122%	63.43%	146.GB	512 MB	109 MB	Microsoft Windows XP Professional	
7	2.8 GHz	47.83043%	52.17%	146.GB	512 MB	104 MB	Microsoft Windows XP Professional	
8	2.8 GHz	23.57594%	76.42%	146.GB	512 MB	115 MB	Microsoft Windows XP Professional	
9	2.8 GHz	23.74848%	76.25%	146.GB	512 MB	106 MB	Microsoft Windows XP Professional	
10	2.4 GHz	24.65337%	75.35%	146.GB	512 MB	103 MB	Microsoft Windows XP Professional	
11	2.53 GHz	25.11831%	74.88%	146.GB	512 MB	97 MB	Microsoft Windows XP Professional	
							Microsoft Windows XP	

ComputersDescription / ComputersDescription (2) / Sorted_data1

Figure 4. Parameters Value

Resource Categorization by Using Scoring Scheme

After fetching the values of each parameter we applying scoring scheme describe in above section. Figure 5 shows the score of each parameter at each node. Yellow color columns show the categorical data of the resources.



Version 1.3.xls [Compatibility Mode] - Microsoft Excel

	A	B	C	D	E	F	G	H	I	J	K	L
1	C.P.U Speed	categorial data(R1)	Y1	C.P.U Load % Available	R2	Y2	Storage Capacity available	R3	Y3	RAM Available	R4	Y4
2			thershold=2.5 GHz	Theshold= 50%			threshold=100 MB			threshold=100 MB		
3	2.8	4	1	26.246	2.000	0.000	146	1	1	118	1	1
4	2.8	4	1	46.375	2.000	0.000	151	1	1	434	2	1
5	2.8	4	1	44.907	2.000	0.000	151	1	1	412	2	1
6	2.8	4	1	84.278	4.000	1.000	19.8	0	0	292	2	1
7	2.8	4	1	71.352	3.000	1.000	19.8	0	0	278	2	1
8	2.8	4	1	83.524	4.000	1.000	19.8	0	0	230	1	1
9	2.8	4	1	90.756	4.000	1.000	19.8	0	0	216	1	1
10	2.8	4	1	93.537	4.000	1.000	19.8	0	0	215	1	1
11	2.8	4	1	94.631	4.000	1.000	19.8	0	0	214	1	1
12	2.8	4	1	72.784	3.000	1.000	146	1	1	133	1	1
13	2.8	4	1	62.332	3.000	1.000	146	1	1	118	1	1
14	2.8	4	1	52.170	3.000	1.000	146	1	1	104	1	1
15	2.8	4	1	76.424	4.000	1.000	146	1	1	115	1	1
16	2.8	4	1	76.252	4.000	1.000	146	1	1	106	1	1
17	2.8	4	1	74.936	3.000	1.000	146	1	1	93	1	0
18	2.8	4	1	71.587	3.000	1.000	146	1	1	75	1	0
19	2.8	4	1	74.816	3.000	1.000	151	1	1	594	3	1
20	2.8	4	1	96.920	4.000	1.000	151	1	1	593	3	1
21	2.8	4	1	95.879	4.000	1.000	151	1	1	594	3	1
22	2.8	4	1	80.343	4.000	1.000	181	1	1	61	1	0
23	2.8	4	1	87.214	4.000	1.000	181	1	1	71	1	0
24	2.8	4	1	86.797	4.000	1.000	181	1	1	81	1	0
25	2.8	4	1	84.398	4.000	1.000	181	1	1	92	1	0
26	2.8	4	1	92.868	4.000	1.000	142	1	1	254	1	1
27	2.8	4	1	89.933	4.000	1.000	142	1	1	256	1	1
28	2.8	4	1	83.868	4.000	1.000	181	1	1	62	1	0
29	2.8	4	1	84.314	4.000	1.000	181	1	1	61	1	0
30	2.8	4	1	84.655	4.000	1.000	181	1	1	50	1	0
31	2.8	4	1	92.787	4.000	1.000	142	1	1	254	1	1
32	2.8	4	1	83.717	4.000	1.000	181	1	1	51	1	0

Sheet3 Sheet4 Sheet2 Sheet1 Sorted_data

Figure 5. Scoring of Parameters

Threshold Test

In this research study we suggest to use threshold value for each QoS parameter . In this experiment we use the following threshold values for our QoS parameters.

- 2.0 GHz for CPU Speed
- 50% for CPU_Load
- 100 M.B. for RAM
- 100 MB for Storage Capacity.

Now by providing threshold values for used parameters we predict that the node is willing to serve the user or not. Here if the value of a parameter is passes the threshold test it means that resource is willing for serve the user, as combining all the parameters to know the willingness we set the criteria that if 75% resources at a particular node are pass the threshold test then it means that node is willing to serve the user.After scoring the resources we conduct threshold test and the result of threshold test is in the form of binary numbers (0,1). 0 means that the node is not willing to serve the user and 1 means that the node is willing to serve the user. Following is the node assessment function that shows the willingness and reluctance of the node that shows the willingness and reluctance of the node.

$$R_{ij} = \begin{cases} 1 & , \ if\ Rij \geq dj \\ 0 & , \ if\ Rij < dj \end{cases}$$

Here Rij is the resource value and dj is the threshold value of the resource. If resource passes the threshold test then it returns binary value 1 and in the case of failure it returns binary value 0. These binary values show the willingness and reluctance of the nodes. Binary 1 means resource is willing to serve the user and binary 0 means that the resource is not wiling to serve the user. Following is the code segment of the threshold test.

```
If ( parameter_value ) >= threshold value

        Resource is willing to serve the user

Else

        Resource is not willing to serve the user
```

Figure 6 shows the threshold test result applying on the values of each parameter.

Yellow color columns show the threshold test results. Our QoS parameters and their corresponding threshold values are as under.

2.0 GHz for CPU Speed, 50% for CPU_Load , 100 M.B. for RAM, 100 MB for Storage Capacity.

Training Data Set

After applying threshold values we get the Y values for each parameter in the form of binary values , on the basis of each parameter binary value , we set the criteria that if 75% (3 out of 4 parameters) of the resources are getting binary value 1 it means that the node is willing for serve the user. In the Figure 7 Green column shows the node willingness and reluctances values.

After getting Response values from the grid nodes, we give the parameters data to SPSS software to estimate the coefficients values for the Logistic Regression Model. SPSS perform necessary calculation on given data set and estimate the coefficients i.e. $\beta_1, \beta_2, \beta_3\, and\ \beta_4$ and constant values i.e. β_0. Following figures [9,10] are the some screen shots of SPSS software.

SPSS Screen Shots

Figure 8 shows the parameters value provided to the SPSS software for calculating the coefficient and constant values.

Figure 9 and 10 show the output of data analysis and model fitting phase.

Following is the table 1 that shows the estimated values of the coefficients of Our logistic regression model.

Table 1. Coefficient and Constant of Logistic regression Model

	β	S.E	P-Value	Significant (P-value < 5%)
C.P.U Speed	14.010	3.069	0.000	Significant
C.P.U Load	0.086	0.022	0.000	Significant
Storage	0.050	0.012	0.000	Significant
RAM	0.048	0.011	0.000	Significant
Constant	-53.133	10.932	0.000	Significant

The Logistic Regression model is given as under.

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4$$

In our Logistic model parameters and coefficients are given in the following table 2.

Table 2. Parameter, Coefficient and Constant table

Parameter Name	Variable	Coefficient	Values
CPU_Speed	X1		14.010
CPU_Load	X2		.086
Storage Capacity	X3		.050
RAM	X4		.048
constant			-53.133

So our Logistic Regression Model Look like this.

$$Y = -53.133 + 14.010X_1 + 0.86X_2 + 0.050\ X_3 + 0.048\ X_4$$

Type I ERROR

α=Type-I error= Prob (Rejecting a node that should be included in the Next grid).

Type II Error

β=Type-II error= Prob (Accpeting a node that should not be in the Next grid).

4. RESULTS

The experiment considered a Grid environment composed of 300 nodes with heterogeneous resources. All 300 nodes are examined and parameters values are collected at a certain time stamp. All nodes having Windows Xp Professional Operating System, 512 M.B. of RAM , variable Storage Capacity(HDD) in Giga bytes(GB) and 2.0 GHz, 2.4 GHz, 2.53 GHz and 2.8 GHz Intel processors.

Coefficient values

Coefficient values for logistic Regression Model is given is following table 3.

Table 3. Coefficent Value

	β
C.P.U Speed	14.010
C.P.U Load	0.086
4.1 Logistic Regression Model Storage	0.050
RAM	0.048
Constant	-53.133

$Y = -53.133 + 14.010X_1 + 0.86X_2 + 0.050\ X_3 + 0.048\ X_4$

Classification Table

Table 4. Classification Table

	Observed Y	Predicted Y .00	Predicted Y 1.00	Predicted Percentage Correct
Y	.00	60	12	83.3
	1.00	6	223	97.4
Overall Percentage				94.0

According to classification table 4 we get the 94.0% correct prediction result for allocation or selection the best resources for a grid job. The statistics presented in classification table is generated by the SPSS on the basis of training data set . the statistics shows that there are 301 nodes involved in this experiment.

a. Total 60 nodes predicted “Not-Available” for a grid job by using scoring and threshold test approach(Observed Y), and this result ensure by the Logistic Regression Model (Predicted Y).
b. Total 223 nodes predicted “Available” for a grid by using scoring and threshold test approach (Observed Y), and this result ensure by the Logistic Regression Model (Predicted Y).
c. Total 6 nodes predicted “Available” for a grid job by using scoring and threshold test approach(Observed Y), and this result expel by the Logistic Regression Model(Predicted Y).
d. Total 12 nodes predicted “Not-Available” for a grid job by using scoring and threshold test approach(Observed Y), and this result expel by the Logistic Regression Model (predicted Y).

5. PROPOSED APPROACH VALIDATION

After training the data set we get the coefficient and constant values and by applying these constant and coefficient values we get the Logistic Regression Model for resource prediction for a grid job. The model is given below.

$$Y = -53.133 + 14.010X_1 + 0.86X_2 + 0.050\ X_3 + 0.048\ X_4$$

For validating our result obtained by the training data set we provide QoS parameters value to our model and get the predicted value for willingness or reluctance to serve the grid user.Result of the data set of 156 node for validation of our model.

In the validation test we get the 5 out of 156 wrong prediction, i-e. Type-I and Type-II error. Observation no 25, 32 and 43 are the example of Type-I error and two observation numbers are the example of Type-II error. ***Model Error % is 3.21%*** and ***Model Prediction Accuracy is 96.79%*** testing for the above mentioned nodes.

6. CONCLUSION

This research study focuses on the best resource selection and allocation for ensuring the QoS in a grid environment. We have presented a new approach inspired by statistical ranking and selection schema for best resource selection to get optimal results and better decision making. Multiple binary logistic model for prediction of nodes for Next Grid for a new job assignment that atleast X% free in their resources is proposed. Experiment considered a Grid environment composed of heterogeneous resources. In the data training phase 301 nodes are examined and we get 94.0 % prediction accuracy while prediction accuracy in validation phase is 96.79%. in the validation phase no of nodes are 156. While the Model Error is 3.21% . The main concern for future work is to develop an automated tool which gives the fully support to our mechanism in a grid environment, also we pretend to extend this approach measuring and modeling network quality of service (QoS) parameters.

7. ACKNOWLEDGMENTS

With core of heart we are very pleased to **PAF-KIET** & **Sarhad University pakistan** for providing us the research facility, we are also greatful to **IJCA** for their full support and lastly we are thankful to our family for their spiritual prays.

8. REFERENCES

[1] Grid Computing by Joshy Joseph, Craig Fellenstein ch 02 pages#48

[2] Xu Bing, "A Study of the Grid Technology,", in proceeding International Conference on Advanced Computer Control, icacc 2009, pp.729-733

[3] Marcelo Massocco Cendron and Carlos Becker Westphall,"A price-based Task Scheduling for Grid Computing" in Proceedings of the Seventh International Conference on Networking Pages 762-766

[4] Kai-jian Liang, Quan Liang and Yang Yang, "The Strategy and Method of QoS Parameter Processing for Grid Service Matchmaking" in Proceedings of the First International Conference on Innovative Computing, Information and Control vol. 1, pp.381-384

[5] Safiullah Faizullah and Ivan Marsic " Pricing QoS: Simulation and Analysis " in proceeding of the 34th Annual Simulation Symposium (SS01) page:193

[6] Allan Snavely , Greg Chun , Henri Casanova, Rob F. Van der Wijngaart and Michael A. Frumkin "Benchmarks for grid computing: a review of ongoing efforts and future directions" ACM SIGMETRICS Performance Evaluation Review Volume 30 Issue 4 (March 2003) Pages: 27 - 32

[7] George Tsouloupas Marios D. Dikaiakos "Ranking and Performance Exploration of Grid Infrastructures: An Interactive Approach" in proceeding Grid Computing, 7th IEEE/ACM International Conference page(s): 313-314

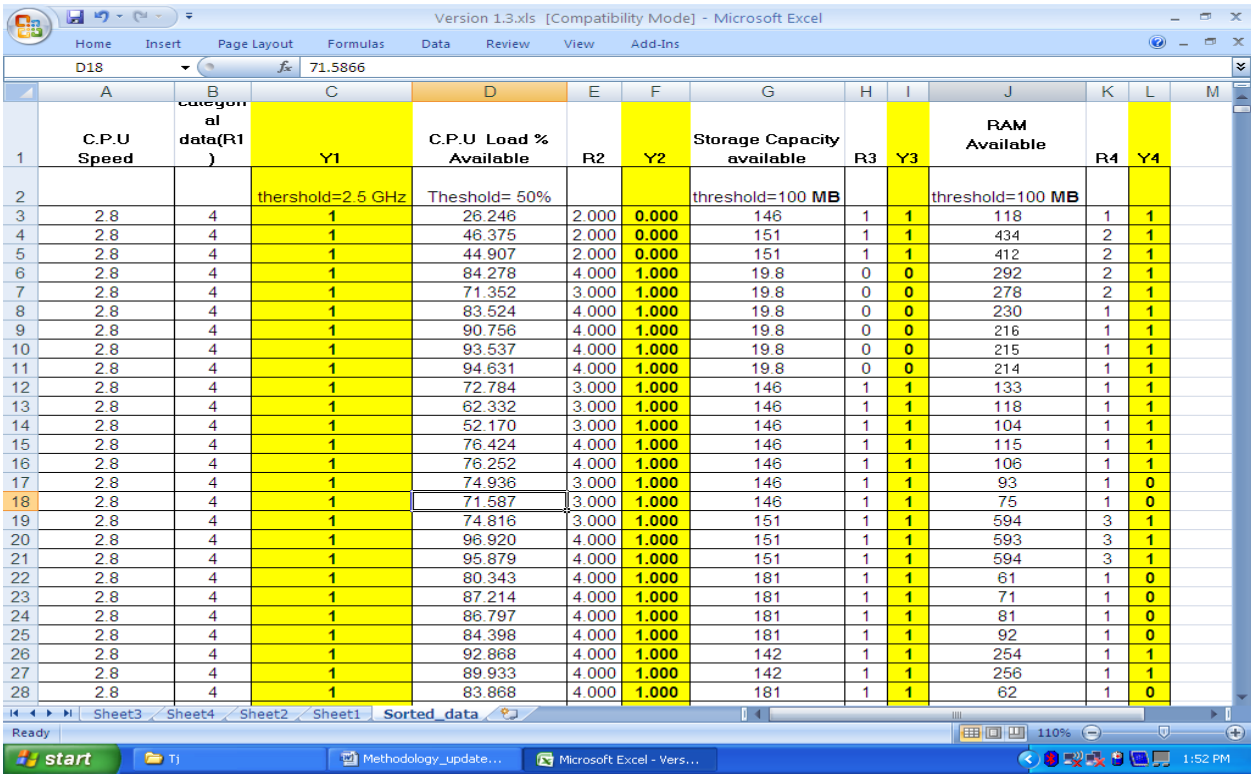
[8] Yongwei Wu, Yulai Yuan, Guangwen Yang, Weimin Zheng, "Load prediction using hybrid model for computational grid," in proceeding of the 8th IEEE/ACM International Conference on Grid Computing, 2007 pages:235-242.

[9] Rimantas Plestys, Gytis Vilutis, Donatas Sandonavicius, Kaunas, LITHUANIA Rasa Vaskeviciute ,Rimantas Kavaliunas" The Measurement of Grid QoS Parameters" in Proceedings of the ITI 2007 29th Int. Conf. on Information Technology Interfaces, June 25-28, 2007, Cavtat, Croatia

[10] Seyed Masoud Sadjadi, Shu Shimizu, Javier Figueroa, Raju Rangaswami, Javier Delgado,Hector Duran, Xabriel J. Collazo-Mojica " A Modeling Approach for Estimating Execution Time of Long-Running Scientific Applications " in Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on Publication Date: 14-18 April 2008 pages: 1-8

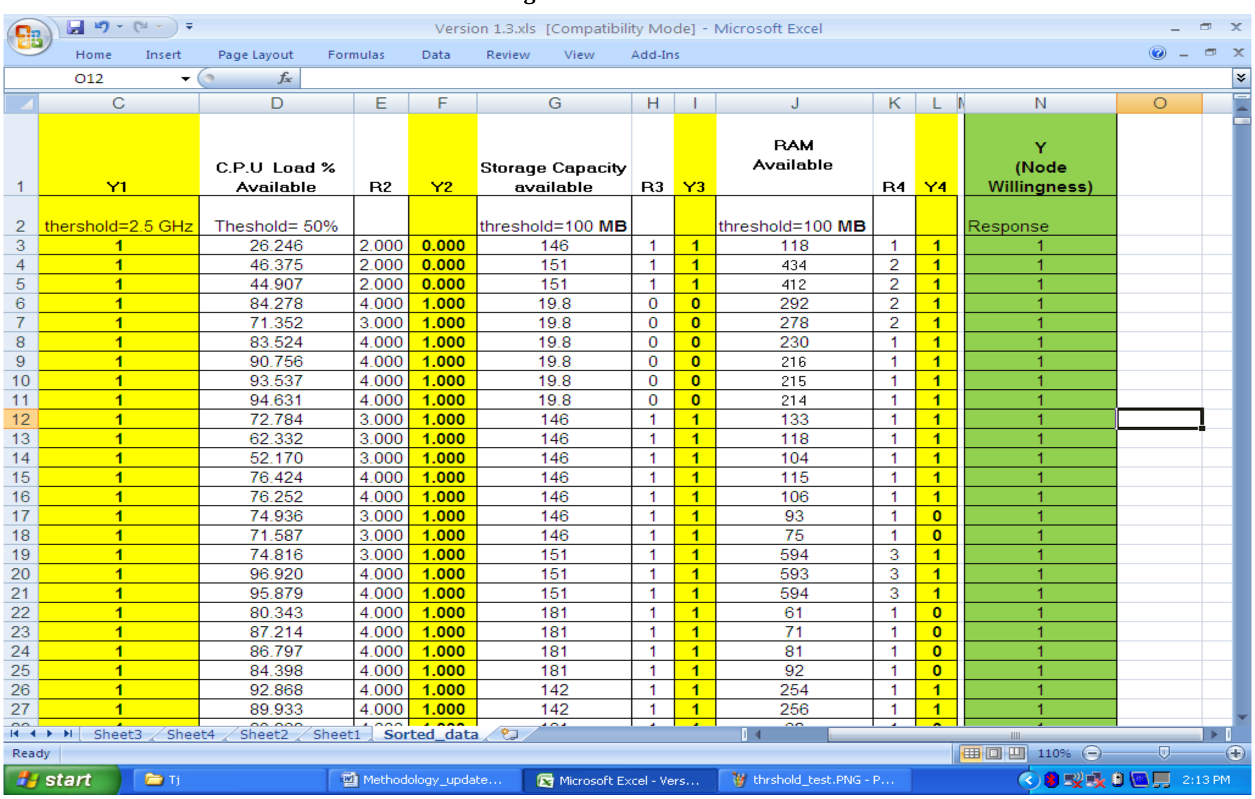
[11] David Allenotor, Ruppa K. Thulasiram, "G-FRoM: Grid Resources Pricing A Fuzzy Real Option Model," in proceeding Third IEEE International Conference on e-Science and Grid Computing (e-Science 2007), pages.388-395

List of figures



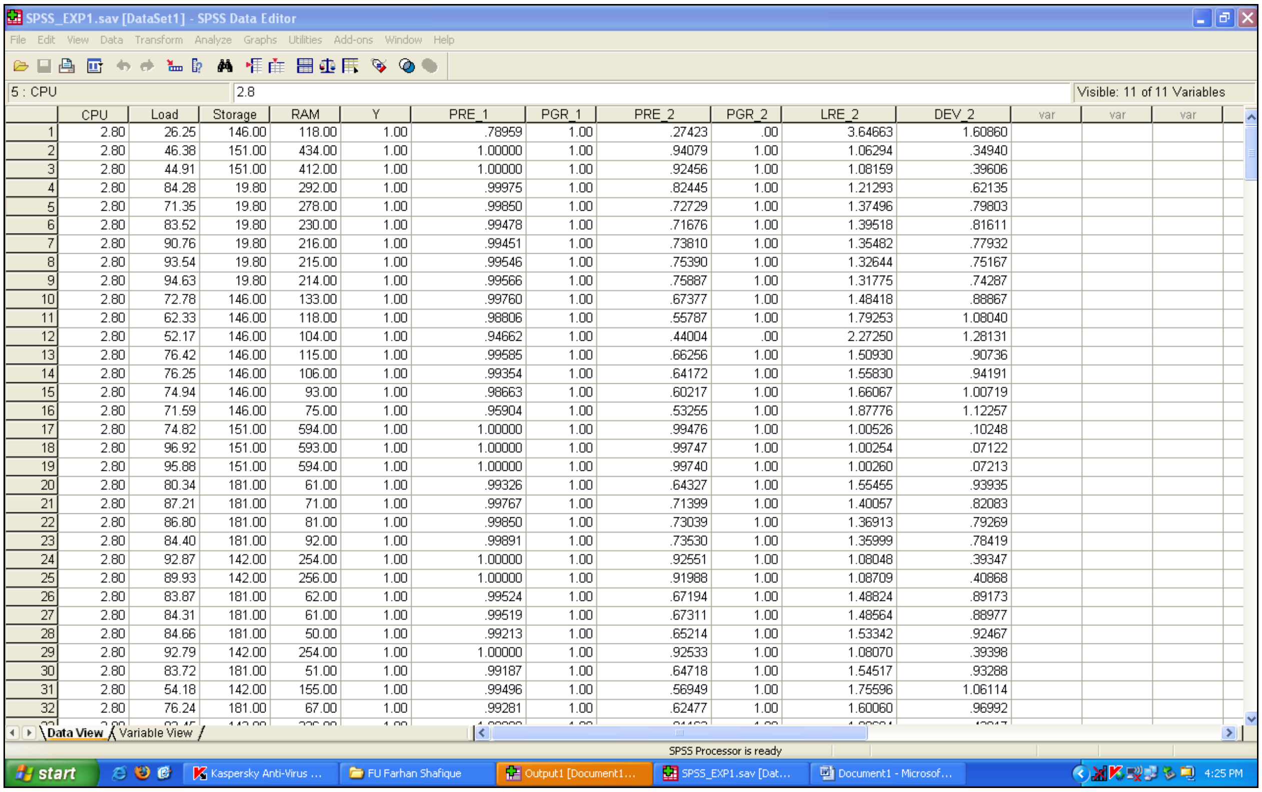
	C.P.U Speed	categorical data(R1)	Y1	C.P.U Load % Available	R2	Y2	Storage Capacity available	R3	Y3	RAM Available	R4	Y4
2			thershold=2.5 GHz	Theshold= 50%			threshold=100 MB			threshold=100 MB		
3	2.8	4	1	26.246	2.000	0.000	146	1	1	118	1	1
4	2.8	4	1	46.375	2.000	0.000	151	1	1	434	2	1
5	2.8	4	1	44.907	2.000	0.000	151	1	1	412	2	1
6	2.8	4	1	84.278	4.000	1.000	19.8	0	0	292	2	1
7	2.8	4	1	71.352	3.000	1.000	19.8	0	0	278	2	1
8	2.8	4	1	83.524	4.000	1.000	19.8	0	0	230	1	1
9	2.8	4	1	90.756	4.000	1.000	19.8	0	0	216	1	1
10	2.8	4	1	93.537	4.000	1.000	19.8	0	0	215	1	1
11	2.8	4	1	94.631	4.000	1.000	19.8	0	0	214	1	1
12	2.8	4	1	72.784	3.000	1.000	146	1	1	133	1	1
13	2.8	4	1	62.332	3.000	1.000	146	1	1	118	1	1
14	2.8	4	1	52.170	3.000	1.000	146	1	1	104	1	1
15	2.8	4	1	76.424	4.000	1.000	146	1	1	115	1	1
16	2.8	4	1	76.252	4.000	1.000	146	1	1	106	1	1
17	2.8	4	1	74.936	3.000	1.000	146	1	1	93	1	0
18	2.8	4	1	71.587	3.000	1.000	146	1	1	75	1	0
19	2.8	4	1	74.816	3.000	1.000	151	1	1	594	3	1
20	2.8	4	1	96.920	4.000	1.000	151	1	1	593	3	1
21	2.8	4	1	95.879	4.000	1.000	151	1	1	594	3	1
22	2.8	4	1	80.343	4.000	1.000	181	1	1	61	1	0
23	2.8	4	1	87.214	4.000	1.000	181	1	1	71	1	0
24	2.8	4	1	86.797	4.000	1.000	181	1	1	81	1	0
25	2.8	4	1	84.398	4.000	1.000	181	1	1	92	1	0
26	2.8	4	1	92.868	4.000	1.000	142	1	1	254	1	1
27	2.8	4	1	89.933	4.000	1.000	142	1	1	256	1	1
28	2.8	4	1	83.868	4.000	1.000	181	1	1	62	1	0

Figure 6. Threshold test



	Y1	C.P.U Load % Available	R2	Y2	Storage Capacity available	R3	Y3	RAM Available	R4	Y4	Y (Node Willingness)
2	thershold=2.5 GHz	Theshold= 50%			threshold=100 MB			threshold=100 MB			Response
3	1	26.246	2.000	0.000	146	1	1	118	1	1	1
4	1	46.375	2.000	0.000	151	1	1	434	2	1	1
5	1	44.907	2.000	0.000	151	1	1	412	2	1	1
6	1	84.278	4.000	1.000	19.8	0	0	292	2	1	1
7	1	71.352	3.000	1.000	19.8	0	0	278	2	1	1
8	1	83.524	4.000	1.000	19.8	0	0	230	1	1	1
9	1	90.756	4.000	1.000	19.8	0	0	216	1	1	1
10	1	93.537	4.000	1.000	19.8	0	0	215	1	1	1
11	1	94.631	4.000	1.000	19.8	0	0	214	1	1	1
12	1	72.784	3.000	1.000	146	1	1	133	1	1	1
13	1	62.332	3.000	1.000	146	1	1	118	1	1	1
14	1	52.170	3.000	1.000	146	1	1	104	1	1	1
15	1	76.424	4.000	1.000	146	1	1	115	1	1	1
16	1	76.252	4.000	1.000	146	1	1	106	1	1	1
17	1	74.936	3.000	1.000	146	1	1	93	1	0	1
18	1	71.587	3.000	1.000	146	1	1	75	1	0	1
19	1	74.816	3.000	1.000	151	1	1	594	3	1	1
20	1	96.920	4.000	1.000	151	1	1	593	3	1	1
21	1	95.879	4.000	1.000	151	1	1	594	3	1	1
22	1	80.343	4.000	1.000	181	1	1	61	1	0	1
23	1	87.214	4.000	1.000	181	1	1	71	1	0	1
24	1	86.797	4.000	1.000	181	1	1	81	1	0	1
25	1	84.398	4.000	1.000	181	1	1	92	1	0	1
26	1	92.868	4.000	1.000	142	1	1	254	1	1	1
27	1	89.933	4.000	1.000	142	1	1	256	1	1	1

Figure 7. Node Response



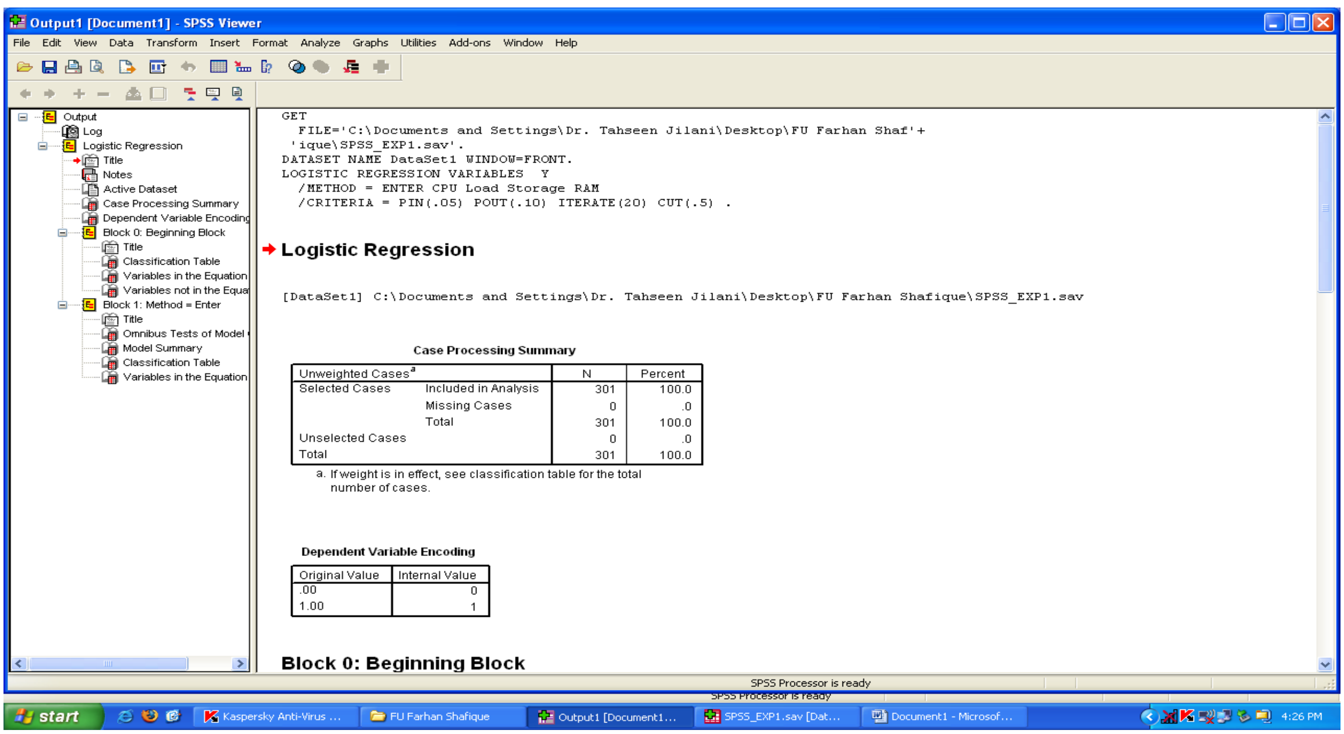
SPSS_EXP1.sav [DataSet1] - SPSS Data Editor

5 : CPU 2.8 — Visible: 11 of 11 Variables

	CPU	Load	Storage	RAM	Y	PRE_1	PGR_1	PRE_2	PGR_2	LRE_2	DEV_2
1	2.80	26.25	146.00	118.00	1.00	.78959	1.00	.27423	.00	3.64663	1.60860
2	2.80	46.38	151.00	434.00	1.00	1.00000	1.00	.94079	1.00	1.06294	.34940
3	2.80	44.91	151.00	412.00	1.00	1.00000	1.00	.92456	1.00	1.08159	.39606
4	2.80	84.28	19.80	292.00	1.00	.99975	1.00	.82445	1.00	1.21293	.62135
5	2.80	71.35	19.80	278.00	1.00	.99850	1.00	.72729	1.00	1.37496	.79803
6	2.80	83.52	19.80	230.00	1.00	.99478	1.00	.71676	1.00	1.39518	.81611
7	2.80	90.76	19.80	216.00	1.00	.99451	1.00	.73810	1.00	1.35482	.77932
8	2.80	93.54	19.80	215.00	1.00	.99546	1.00	.75390	1.00	1.32644	.75167
9	2.80	94.63	19.80	214.00	1.00	.99566	1.00	.75887	1.00	1.31775	.74287
10	2.80	72.78	146.00	133.00	1.00	.99760	1.00	.67377	1.00	1.48418	.88867
11	2.80	62.33	146.00	118.00	1.00	.98806	1.00	.55787	1.00	1.79253	1.08040
12	2.80	52.17	146.00	104.00	1.00	.94662	1.00	.44004	.00	2.27250	1.28131
13	2.80	76.42	146.00	115.00	1.00	.99585	1.00	.66256	1.00	1.50930	.90736
14	2.80	76.25	146.00	106.00	1.00	.99354	1.00	.64172	1.00	1.55830	.94191
15	2.80	74.94	146.00	93.00	1.00	.98663	1.00	.60217	1.00	1.66067	1.00719
16	2.80	71.59	146.00	75.00	1.00	.95904	1.00	.53255	1.00	1.87776	1.12257
17	2.80	74.82	151.00	594.00	1.00	1.00000	1.00	.99476	1.00	1.00526	.10248
18	2.80	96.92	151.00	593.00	1.00	1.00000	1.00	.99747	1.00	1.00254	.07122
19	2.80	95.88	151.00	594.00	1.00	1.00000	1.00	.99740	1.00	1.00260	.07213
20	2.80	80.34	181.00	61.00	1.00	.99326	1.00	.64327	1.00	1.55455	.93935
21	2.80	87.21	181.00	71.00	1.00	.99767	1.00	.71399	1.00	1.40057	.82083
22	2.80	86.80	181.00	81.00	1.00	.99850	1.00	.73039	1.00	1.36913	.79269
23	2.80	84.40	181.00	92.00	1.00	.99891	1.00	.73530	1.00	1.35999	.78419
24	2.80	92.87	142.00	254.00	1.00	1.00000	1.00	.92551	1.00	1.08048	.39347
25	2.80	89.93	142.00	256.00	1.00	1.00000	1.00	.91988	1.00	1.08709	.40868
26	2.80	83.87	181.00	62.00	1.00	.99524	1.00	.67194	1.00	1.48824	.89173
27	2.80	84.31	181.00	61.00	1.00	.99519	1.00	.67311	1.00	1.48564	.88977
28	2.80	84.66	181.00	50.00	1.00	.99213	1.00	.65214	1.00	1.53342	.92467
29	2.80	92.79	142.00	254.00	1.00	1.00000	1.00	.92533	1.00	1.08070	.39398
30	2.80	83.72	181.00	51.00	1.00	.99187	1.00	.64718	1.00	1.54517	.93288
31	2.80	54.18	142.00	155.00	1.00	.99496	1.00	.56949	1.00	1.75596	1.06114
32	2.80	76.24	181.00	67.00	1.00	.99281	1.00	.62477	1.00	1.60060	.96992

Data View / Variable View — SPSS Processor is ready

Figure 8. Parameters Value



Output1 [Document1] - SPSS Viewer

```
GET
  FILE='C:\Documents and Settings\Dr. Tahseen Jilani\Desktop\FU Farhan Shaf'+
 'ique\SPSS_EXP1.sav'.
DATASET NAME DataSet1 WINDOW=FRONT.
LOGISTIC REGRESSION VARIABLES  Y
  /METHOD = ENTER CPU Load Storage RAM
  /CRITERIA = PIN(.05) POUT(.10) ITERATE(20) CUT(.5) .
```

Logistic Regression

[DataSet1] C:\Documents and Settings\Dr. Tahseen Jilani\Desktop\FU Farhan Shafique\SPSS_EXP1.sav

Case Processing Summary

Unweighted Cases[a]		N	Percent
Selected Cases	Included in Analysis	301	100.0
	Missing Cases	0	.0
	Total	301	100.0
Unselected Cases		0	.0
Total		301	100.0

a. If weight is in effect, see classification table for the total number of cases.

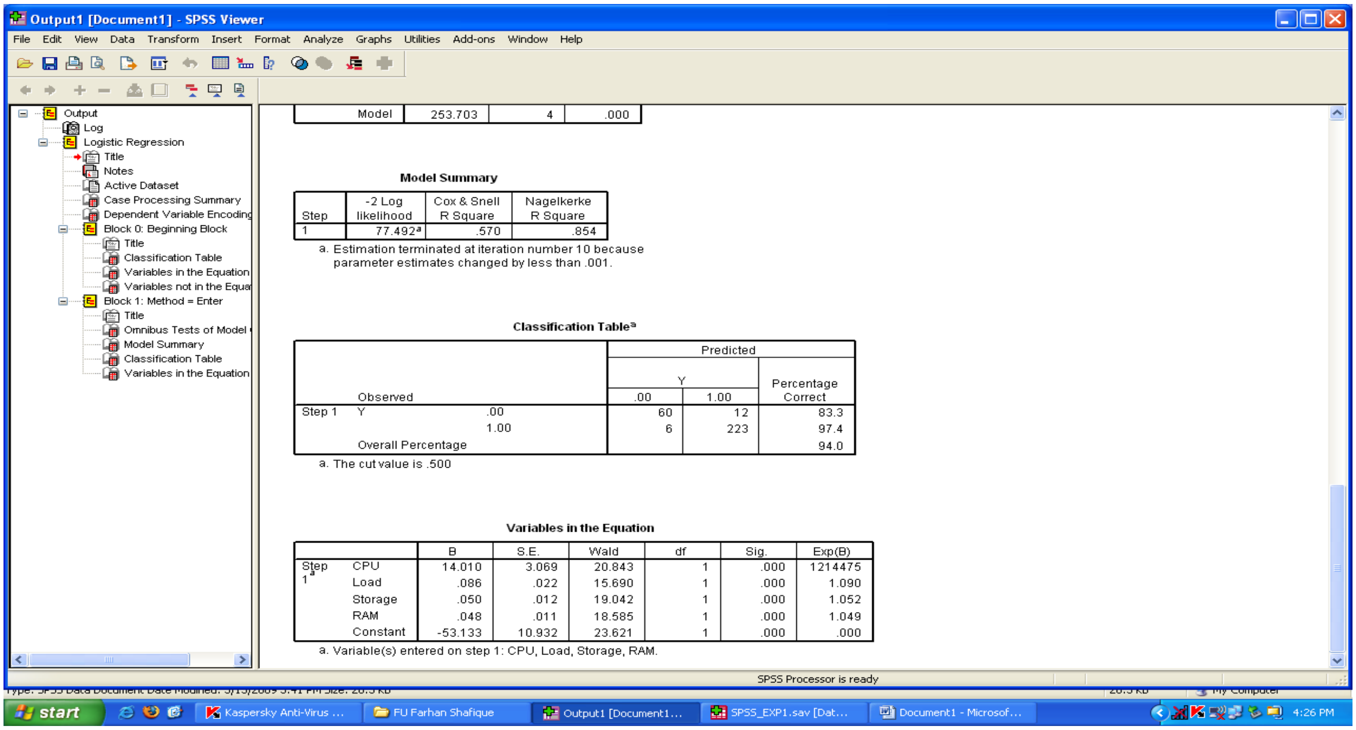
Dependent Variable Encoding

Original Value	Internal Value
.00	0
1.00	1

Block 0: Beginning Block

SPSS Processor is ready

Figure 9. SPSS Output 1



Model	253.703	4	.000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	77.492[a]	.570	.854

a. Estimation terminated at iteration number 10 because parameter estimates changed by less than .001.

Classification Table[a]

Observed			Predicted Y .00	Predicted Y 1.00	Percentage Correct
Step 1	Y	.00	60	12	83.3
		1.00	6	223	97.4
	Overall Percentage				94.0

a. The cut value is .500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	CPU	14.010	3.069	20.843	1	.000	1214475
	Load	.086	.022	15.690	1	.000	1.090
	Storage	.050	.012	19.042	1	.000	1.052
	RAM	.048	.011	18.585	1	.000	1.049
	Constant	-53.133	10.932	23.621	1	.000	.000

a. Variable(s) entered on step 1: CPU, Load, Storage, RAM.

Figure 10. SPSS output 2