Cloud Antivirus Cost Model using Machine Learning

Ali Abdullah Hamzah, Sherif M. Khattab, Salwa S. El-Gamal

Faculty of Computers and Information
Cairo University, Egypt
allawi.cs@gmail.com, s.khattab@fci-cu.edu.eg, s.el-gamal @fci-cu.edu.eg

***Abstract*— An important cloud computing is a new generation of computing and is based on virtualization technology. More and more applications are being deployed in cloud environments. Malware detection or antivirus software has been recently provided as a service in the cloud. A cloud antivirus provider hosts a number of virtual machines each running the same or different antivirus engines on potentially different sets of workloads (files). From the provider's perspective, the problem of optimally allocating physical resources to these virtual machines is crucial to the efficiency of the infrastructure.**

We propose a search-based optimization approach for solving the resource allocation problem in cloud-based antivirus deployments. An elaborate cost model of the file scanning process in antivirus programs is instrumental to the proposed approach. The general architecture is presented and discussed, and a preliminary experimental investigation into the antivirus cost model is described. The cost model depends on many factors, such as total file size, size of code segment, and count and type of embedded files within the executable. However, not a single parameter of these can be reliably used alone to predict file scanning time.

Thus, a machine-learning approach that combines all these parameters as features is used to build a classifier for antivirus file scanning time. The best results we obtained were using the Decision Tree classifier. The highest F-measure value was 0.91, the highest F-measure value using logitboost was 0.87, the highest F-measure value using support vector machine was 0.85 and the highest F-measure value using naïve Bayes was 0.82. We evaluated the accuracy of the classification model versus linear regression model using the Root Mean Square (RMS) measure. We found that the classification model is more accurate than linear regression model, whereas the values average of RMS were 0.988 second and 2.44 second for classification model and linear regression model, respectively.

Keywords - cloud Antivirus, Virtualization, Resource Allocation, Antivirus, Machine Learning, Cost Model.

I. INTRODUCTION

Cloud computing allows for the management and provisioning of resources (e.g., software, CPU, memory, I/O, network bandwidth, applications, and information) as services provided over the Internet on demand. Services are presented in three layers: SaaS (software as a service), such as Facebook and YouTube), PaaS (platform as a services), such as Microsoft Azure and Google AppEngine, and IaaS (infrastructure as a service), such as Amazon EC2 and GoGid [1]. Cloud computing is based on virtualization, which abstracts physical computer resources and allows users to create and run multiple Virtual Machines (VMs) on a single physical machine, whereby each Virtual Machine (VM) is allocated a share of the physical machine resources. The virtual machines can potentially run different operating systems with different applications [2].

Antivirus (AV) software is one of the most important applications in information technology to prevent and remove malware. The effectiveness of traditional host-based AV is limited on the long term in detecting many modern threats for many reasons, such as the time window from the time a virus's signature is released to the time the signature is delivered to the AV running on a device, and the high resource requirement, which may make mobile device users opt for turning off their antivirus [3].

Providing antivirus as a service in the cloud, such as the CloudAV model [4], addresses the above-mentioned limitations. Cloud antivirus provider hosts one or more malware analysis engines and provides its service to remote clients over the internet. Files are sent from the clients to the cloud, get scanned, and the result is sent back to the client. The concept of a cloud antivirus is simple: instead of running complex analysis software on every end-host, each end-host runs a lightweight process. This lightweight process discovers potential threats to the system and sends them to a network-accessible service for analysis.

This paper investigate problem of resource allocation in a CloudAV-like setting. A cloud antivirus provider hosts a number of virtual machines, each running the same or different antivirus engine on different workloads (files). Lightweight processes on end-hosts collect and upload suspect files to the cloud for scanning. The performance of the cloud antivirus infrastructure is affected by the resource allocation to the virtual machines. Optimizing resource allocation would improve resource utilization

The rest of this paper is organized as follows. Related work is described in Section II. The automatic cloud antivirus configurator (ACAC) is described in Section III. Section IV describes the linear AV cost model in ACAC. Whereas, the proposed classification-based cost model in ACAC is shown in Section V. An experimental evaluation of the AV classification-based cost model is shown in Section VI. A comparative evaluation of AV classification-based cost model and AV linear cost model is described in Section I. The discussion and limitation are discussed in VIII and IX and respectively. Finally, conclusions and future work are in Section X.

II. RELATED WORK

Traditional antivirus on the long run has to deal with new threats and increasing vulnerabilities. Hence, antivirus must be more scalable and appropriate for new developments. Particularly in cloud computing, malware detection is provided as antivirus service in the cloud. Currently, mobile security solutions mirror the traditional desktop model to enhance security in mobile services by moving functionality of antivirus off-device to the network and employing multiple virtualized malware detection engines [5]. A lightweight process at the client sends files to a server in the cloud for scanning, allowing for integrated antivirus software, behavioral simulation, and other deep-analysis engines from multiple vendors providing better detection of malware [4].

The executable analysis currently provided in host-based antivirus software can be more efficiently and effectively provided as an in cloud network service. Suggestions to be at each end host lightweight to send files or malware to network for analysis inside an enterprise network or a service provider cloud. This means integrated antivirus software, behavioral simulation and other analysis engines from multiple vendors provide a better detection for malware [6].

There are some products of cloud-based antivirus. **Panda Cloud Antivirus** is considered one of the cloud antivirus products. It is a lightweight layer of defense for any windows-based system. It is an anti-malware program, which uploads the files to analysis and offloads most of the detection and processing to the cloud [3]. **Immunet's antivirus** software adapts rapidly to protect against threats and uses community awareness for intelligent protection, while never slowing down your PC [7]. Symantec cloud antivirus services provide essential protection while virtually eliminating the need to manage hardware and software on site [8].

On the other hand, a case study in this work is **ClamAV** is considered as open source anti-virus toolkit for UNIX. The main purpose of it is e-mail scanning on email gateways. It is the most widely used open source anti-virus scanner available. ClamAV relies on two pattern matching algorithms: Aho-Corasick and Multi-pattern version of Boyer-Moore [9].

A novel anti-malware system as an extension to Clam AV called Split Screen, whereby, performs an additional screen step prior to signature matching phase and filters non-infected files, identifies malware signature. The difference between cloud AV and Split Screen, the client in Cloud AV sends files to a central server for analysis, while in SplitScreen, clients send only their matched signature [10].

There are currently several researches to optimize performance in Anti-Virus, MRSI algorithm Achieved more speed without excessive memory usages. Various performance evaluation tests show that MRSI outperformed the current algorithm implemented in Clam-AV in many aspects [11]. A novel implementation which requires small memory space and achieves high throughput performance with reduction in memory requirement for 5,000 patterns randomly selected from ClamAV [12].

There have been several research attempts to understand the impact of the antivirus on performance. One study characterizes antivirus workload execution and finds that the number of CPU cycles increased when the antivirus was installed and running on the user machine as compared to when the antivirus was not running. This study was on the hardware level using the Virtutech Simics Toolset Simulator (VSTS). Their results indicate that increasing the CPU allocated to VMs that host heavy workloads could optimize the performance in Cloud antivirus. On the other hand, migration of the antivirus from traditional hosts into the cloud indeed optimizes the performance in user machines [13].

Another work studies antivirus performance characterization and finds that the increase of CPU usage when an antivirus is running is because the antivirus caused the programs to spend more time in creating more file IO operation, more page faults, more system calls, and more threads. This study was done on the operating system level using Event Tracing for Windows (ETW). Similar to [13], their results indicate that increasing the CPU allocated to VMs that host heavy workloads could optimize the performance in Cloud antivirus. On the other hand, migration of the antivirus from traditional hosts into the cloud indeed optimizes the performance in user machines [14].

III. Automatic Cloud Antivirus Configurator(ACAC)

This Section describes Automatic Cloud Antivirus Configurator (ACAC) which proposed in our previous work as a solution of the problem of resource allocation for cloud-based antivirus [15].

A. Analysis for cloud-based antivirus

This section describes the problem of allocating physical resources to virtual machines running antivirus software in a cloud computing setting. Virtualization is a common enabling technology in cloud computing. In Cloud Computing, the virtualization technology is used to share resources among virtual machines that run on physical servers. Resource allocation is one of the problems faced in order to optimize performance of the applications running inside the virtual machines. These applications are called (cloud appliances). Deciding on how much physical resource to allocate to each virtual machine not only affects the performance of cloud appliances but also the efficiency of the underlying physical infrastructure.

Application performance is affected by the allocation of physical resources to the VMs. Different applications demand different resource allocations, and moreover, different workloads within the same application pose different resource requirements. For example, the performance of some applications and workloads is optimized by allocating more CPU to their appliances, whereas other applications and workloads benefit more from higher memory or I/O allocation. This heterogeneity motivates the need to control and configure resource allocation to the virtual machines.

On the other hand, antivirus is an important cloud application that needs a mechanism for improving the performance of the scanning process during huge user load. In the same time, there is a need from the provider side to minimize the cost by good utilization of resources. Optimizing the resource allocation, in particular the CPU resource, is considered in this work.

B. Design of the Automatic Cloud Antivirus Configurator (ACAC)

This section describes in a glance the solution proposed in our previous to the resource allocation problem [15].

The proposed solution is based on a search algorithm that enumerates possible resource allocations and evaluates them based on a cost model. The cost model provides the search algorithm with an estimate of the CPU time needed to scan the workload files on a particular VM with a given CPU allocation. The parameters of the cost model may need to be adjusted in order to reflect the virtual environment under the provided CPU allocation. This adjusting process is called (parameter calibration).

Figure 1 depicts the architecture of the proposed solution, namely the Automatic Cloud Antivirus Configurator (ACAC). ACAC is divided into three modules: the configuration enumerator, which includes the search algorithm, the parameter calibration process, and the AV cost model, which estimates the cost of scanning a set of files under a given CPU allocation. This architecture is inspired by the Virtual Design Advisor [16], which has been proposed to solve the resource allocation problem in a different context. ACAC gives recommended configurations for multiple VMs that run different workloads.

The Figure 1 gives the recommend resources allocation for VMs based on the nature of workloads. It is constructed of three modules: enumeration module (optimization search algorithm), Calibration process, and AV performance.

1) Configuration Enumeration Module

The configuration enumeration module is used to enumerate resource allocations for the VMs. It implements a search algorithm, such as greedy search and dynamic programming, for enumerating and searching candidate resource allocations [16]. The greedy algorithm makes the decisions of increasing and decreasing the resources allocated to VMs based on the estimated cost of the given workloads.

2) Calibration Process

Calibration is a process used for mapping each candidate resource allocation to a set of parameter values in the antivirus cost model. In the cost model, a set of parameterized equations are used to compute a cost estimate. Some of the parameters may need to be adjusted according to the resource allocation to the virtual machine. The calibration process adjusts these parameters before applying the equations. The calibration process using Linear Cost Model details are listed in our previous [15].

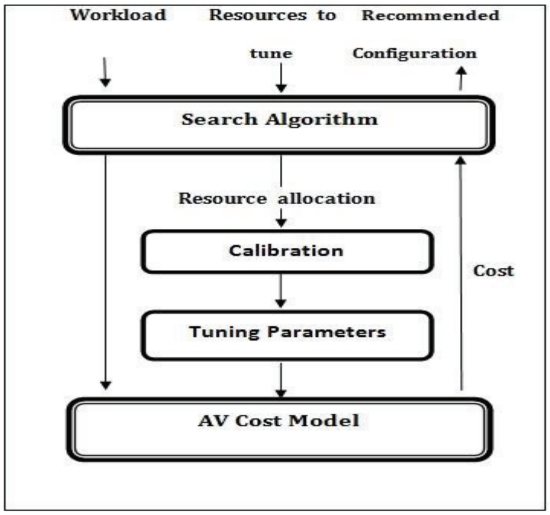

Fig.1 Automatic Cloud Antivirus Configuration (ACAC)

3) AV Cost and Performance Model

ClamAV is an open-source antivirus toolkit for Unix [9]. The main purpose of it is e-mail scanning on email gateways. It is the most widely used open-source antivirus and is based on pattern matching algorithms (i.e., signature-based). ClamAV uses signature-based strategy in the scanning process, whereby it uses two algorithms, namely multi-pattern Boyer-Moore and Aho-Corasick. Theoretically, the running time for the single-pattern Boyer-Moore algorithm is O(N/M) and for Aho-Corasick is O(N), where N is the text size and M the pattern size.

A cost model in computer science is set of mathematical equations or another model that converts resource data into cost data. In this work, the cost model is implemented into two models: linear model and classification model. These two models are clarified in details in next two sections.

IV. LINEAR AV COST MODEL IN ACAC

The cost model, the first cost model in ACAC, as linear (e.g. mathematical measure to get the relationship between two variables such as size and time as depicted in Figure 2) is calculated using this equation:

$$y = A * x + B$$

Where A, B are constant values resulted of calibration process (More details are listed in our previous work [15]).

In practice, however, there is more than one parameter that affects the cost, such as the file size, the number of file types embedded within a file, the number of signatures targeted for each file type and code segment. Hence, combining these parameters as features in a cost model would give a more accurate estimate for the cost that would guide the search algorithm to the right allocation for CPU. This leads to improve the performance to antivirus by good utilization for CPU that satisfying user needed. The features which affected the scanning time are explained in the next subsection.

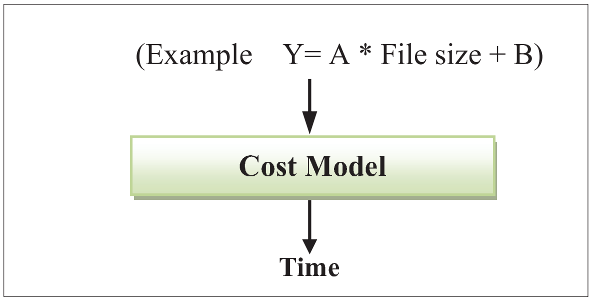

Fig.2 Linear Cost Model Example

A. *Features Affecting Antivirus Scanning Time*

In this subsection, the parameters that affect the cost of the scanning process are described. Modeling the cost of the workload for a given resource allocation is important for all kinds of workload for most application. For example, in related work in database application have the unique advantage that their query optimizer already provides a model for the cost of workload execution [16, 17]. Thus, our approach produces a cost model for antivirus application.

As appeared in the experiments in our previous [15], the cost model of antivirus application is not linear, and more than one parameter affect the cost (time). The next section experimentally digs deeper into understanding the cost model of an antivirus program. Figure 3 shows some of the inputs to the cost model (i.e., parameters that affected the cost model) which is considered an instrumental part of the proposed architecture.

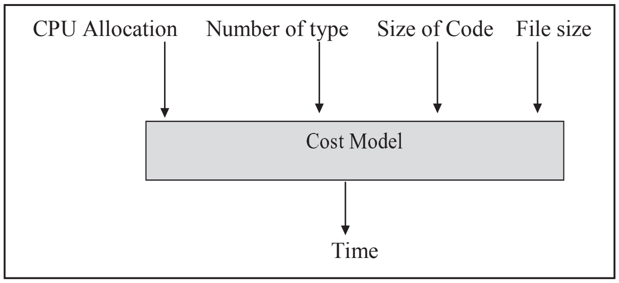

Fig.3 The AV Performance Model

V. CLASSIFICATION-BASED COST MODEL IN ACAC

According to the features which affect on the scanning time, some of the classification techniques are used to improve ACAC cost model. We implemented these techniques using classification algorithms, whereby, the scanning time for the dataset (i.e., the features of set of files that observed during scanning process) splits into four categories (classes).

The step-by-step scenario for classification process that we implemented is depicted in Figure 4. The scenario stages proceed as follows:

A. *First stage*

In this stage, the dataset is combined and divided it on four classes based on the scanning process time under each CPU allocations (100%, 70%, 50%, and 30%) whereby the four classes are:

1. First Class is : C1=(<=30 sec)
2. Second Class is : C2=(>=31&&<=60 sec)
3. Third Class is: C3=(>=61&&<=90 sec)
4. Fourth Class is : C4=(>90 sec)

B. *Second stage*

In this stage, the dataset is trained using the classification algorithm (e.g., decision tree, naïve Bayes, support vector machine) to build a model.

C. *Third stage*

In this stage, the model is tested to assign the file class based on classification technique. The classifier assigned each entering file into specific class.

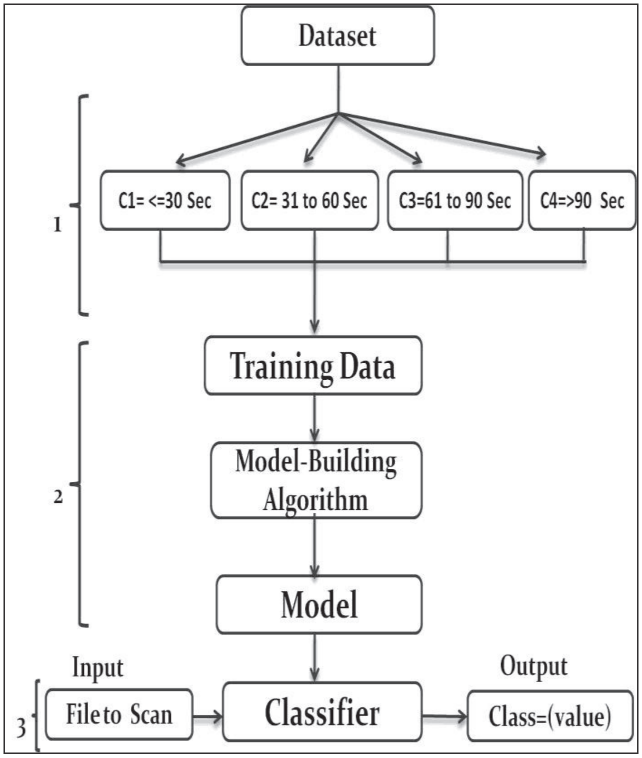

Fig.4 Classification Process to four classes Flowchart

VI. EVALUATION OF COST MODEL USING CLASSIFICATION ALGORITHMS

In this section, we describe the steps of experiments conducted with classification algorithm and the results of each classifier.

A. Dataset

Initially, we created a data set to group of EXE file that contain different features, whereby each feature (e.g., file size, number of embedded types, code size) has different values within files of the dataset. In addition, we conducted the scanning process on more than one CPU allocation, so we divided the dataset into a number of classes based on the time taken to scan the file, for example, <=30 seconds for Class 1, 31-60 seconds for Class 2, 61-90 seconds for Class 3, and more than 90 for Class 4.

B. The Used Program

We used **WEKA** software to achieve classification process. **WEKA** software, formally called Waikato Environment for Knowledge Learning, is a computer program that was developed at the University of Waikato in New Zealand for the purpose of identifying information from raw data gathered from agricultural domains. WEKA supports many different standard data mining tasks such as data preprocessing, classification, clustering, regression, visualization and feature selection [18].

C. Experimental Steps

We conducted many experiments to measure the accuracy of the dataset. We used different train-test split percentages and four classifiers techniques, Decision Tree, Naïve Bayes, logitboost and Support Vector Machine. We measured the F-measure, which is a measure of accuracy [19]. It considers both the precision ***p*** (and specificity) and the recall ***r*** (or sensitivity) of the test to compute the score. p is the number of true positives (correct results) divided by the number of true positives plus the number of false positives (all returned results) [20]. ***r*** is the number of true positives (correct results) divided by the number of true positives plus the number of false negatives (results that should have been returned) [19, 21].

D. Experimental Results

In experiments, we repeated the experiment ten times, and we report the true positive rate, false positive rate, recall, precision and the F-measure for algorithms used at each different training set size as shown down, whereby the 66% as training and 34 % as testing to create and test the model.

Figure 5 depicts the results of true positives rate with different training set sizes for set of (EXE) features for the four classifiers techniques, Decision Tree, Naïve Bayes, logitboost and support vector Machine. On the other hand, Figure 6 shows results of false positives rate with different training set sizes for set of (EXE) features for these techniques. Also, Figure 7 shows the results of Precision with Different training set sizes for set of (EXE) features. Moreover, Figure 8 shows results of recall with different training set sizes for set of (EXE) features. Finally, Figure 9 shows the results of F-measures with different training set sizes for set of (EXE) features. The conclusion of this analysis is that the decision tree has the highest F-measure as compared to the other three techniques.

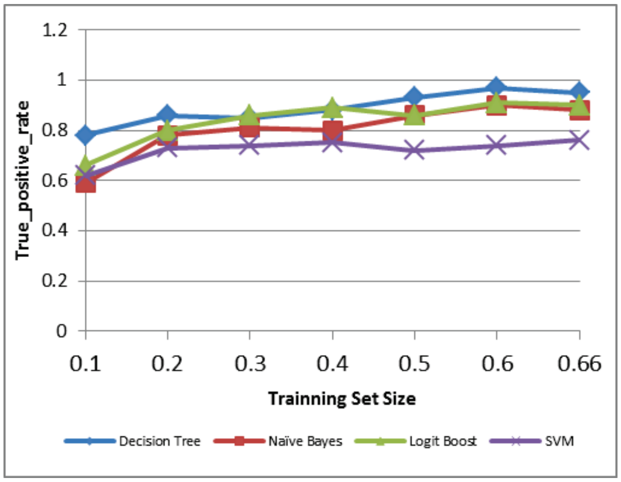

Fig.5 Results of true positives rate with different Training Set Sizes for set of (EXE) features

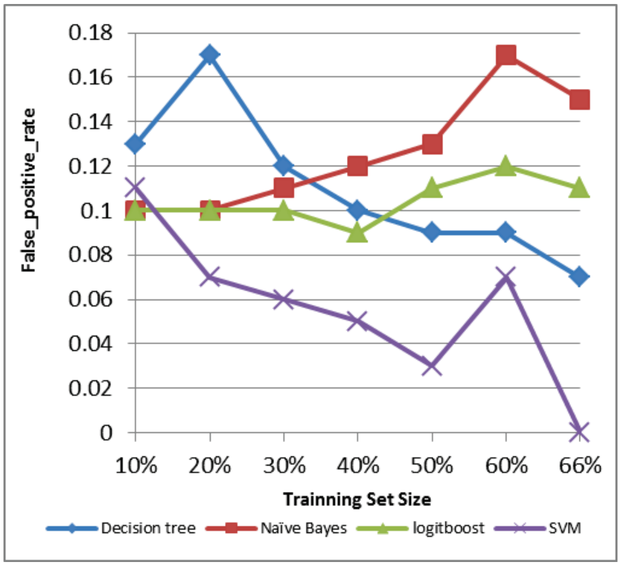

Fig.6 Results of false positives rate with different Training Set Sizes for set of (EXE) features

VII. COMPARATIVE EVALUATION OF CLASSIFICATION MODEL VERSUS LINEAR REGRESSION MODEL

Root mean square is measure used to predicate error in cost model or in predicate model. We implemented this measure between linear regression model versus classification model by root mean square equation

$$\boldsymbol{RMS} = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 \;\dots\dots\dots.. (y_n - x_n)^2}$$

In the linear regression model, RMS was calculated for each allocation at each scanning process (e.g., scanning process at CPU allocation 100%, 70%, 50%, and 30%). By substituting into the RMS equation with values from the tables, the average RMS of the error for the linear regression model was 2.44 seconds.

In the classification model we set the CPU allocation as one of the attributes. The dataset was split into 66% training and 34% testing. The RMS error was computed on the testing part of the dataset for 56 classes. It was found average 0.988 seconds. Figure 10 shows the RMS error percentage for linear regression model verses classification model.

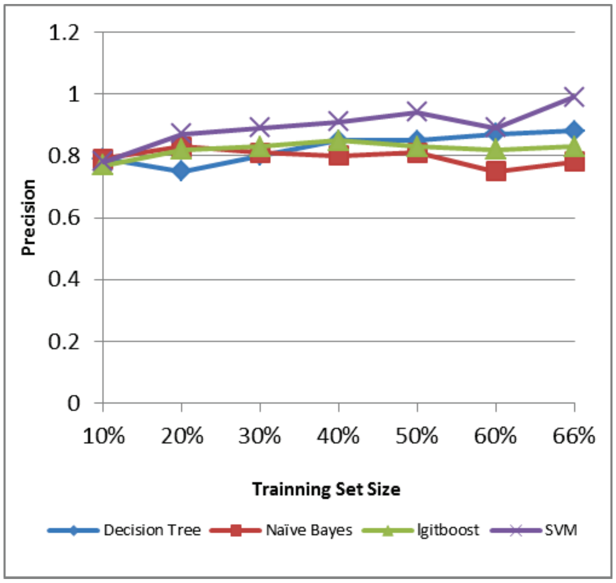

Fig.7 Results of Precision with Different Training Set Sizes for set of (EXE) features

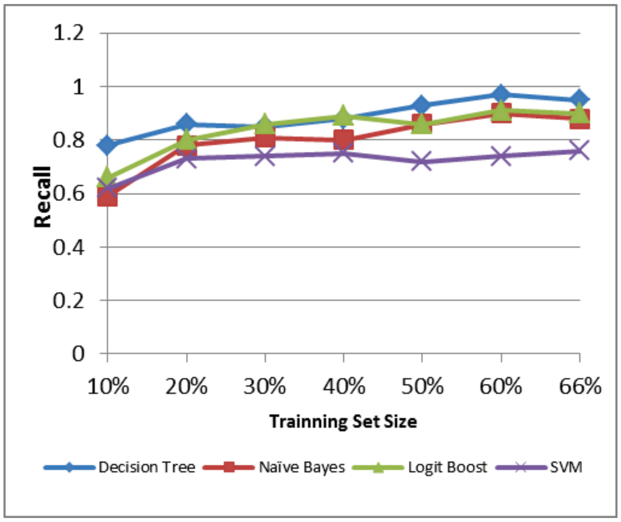

Fig.8 Results of Recall with different Training Set Sizes for set of (EXE) features

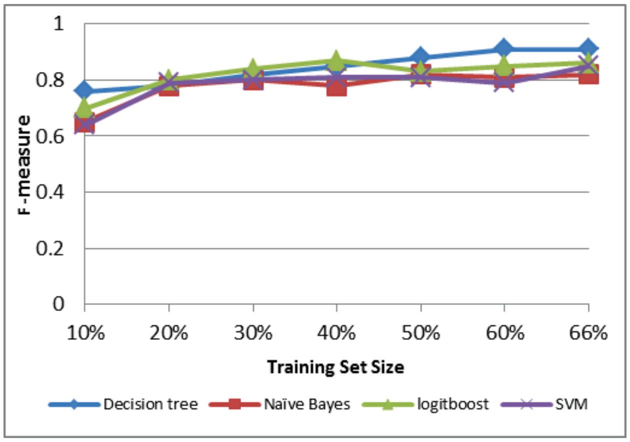

Fig.9 Results of F-measure with different Training Set Sizes for set of (EXE) features

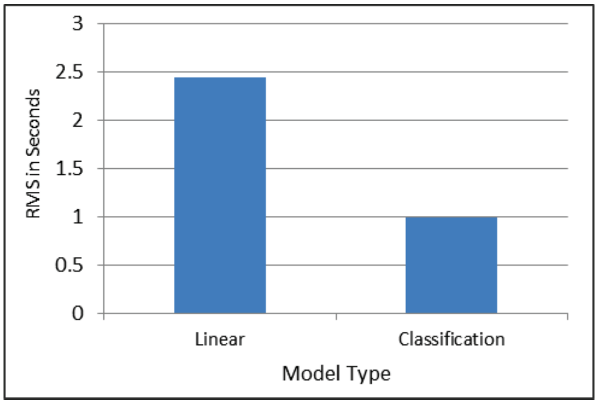

Fig.10 Root Mean Square measure

VIII. DISCUSSION

Resource virtualization is currently being employed at all levels of the IT infrastructure to improve provisioning and manageability, with the goal of reducing total cost of ownership. This means that antivirus systems will increasingly be run in virtualized environments, inside virtual machines. We presented a formulation of resource allocation problem, which focuses on setting resource allocation levels for different antivirus workloads to optimize the performance to antivirus application whereby, the formulation for this problem on the infrastructure (analysis engines) where the workload is on it.

In addition, the work presents the proposed solution for this problem. An important component of this solution is modeling the cost of a workload for a given resource allocation. We initially studied a linear cost model before finding out that the cost (time) is not based just on the file size alone. We found that the cost (time) is affected by more than factor. We took these factors as features and implemented classification model to get a cost model. By measuring the root mean square RMS error between linear regression model and classification model, the classification model achieved less error in estimating the cost whereby RMS= 0.988 for classification model versus RMS=2.44 for linear regression model.

IX. LIMITATIONS

In this research, we found several limitations as follows:

1. There is not specific benchmark of files to implement scanning process.
2. In this study, we implemented scanning process on the EXE files because these files are more infected with viruses according to some search lab such as Symantec Search.
3. The infected files were small in size, limited to just viruses, but it is often the viruses come within a file.
4. We cannot extract some of features for EXE files such as code size.

X. CONCLUSIONS AND FUTURE WORK

The performance of applications on cloud computing is important field to get services as users needs of ownership. Antivirus application is one of these applications that need to improve the performance through huge scanning process. A cost model of antivirus programs is instrumental to the approach adopted in this work.

An accurate cost model would guide the search algorithm to optimal or near-optimal resource allocation. Our preliminary experiments showed that the scan time (e.g., the time taken by an antivirus program to scan a file) depends on the total file size, the size of the code segment, and the count and types of embedded files inside the executable. However, not a single parameter of these can be reliably used to estimate the scan time. Hence, we used a machine-learning approach for all these parameters as features by classifier technique to introduce a cost model.

We used dataset (EXE) files and extracted features to classify the cost (time) into four classes. The best results we obtained were using the Decision Tree classifier. The highest F-measure value was 0.91, the highest F-measure value using logitboost was 0.87, the highest F-measure value using support vector machine was 0.85and the highest F-measure value using naïve Bayes was 0.8. We evaluated the accuracy of the classification model versus linear regression model using the Root Mean Square (RMS) measure. We found that the classification model is more accurate than linear regression model, whereas the values average of RMS were 0.988 second and 2.44 second for classification model and linear regression model, respectively.

REFERENCES

[1] L. C. Qi Zhang, Raouf Boutaba, "Cloud computing: state-of-the-art and research challenges " *Journal of Internet Services and Applications,* vol. 1, No. 1, pp. 7-18, May 2010.

[2] (2013). *VMware: Benefits of Virtualization* Available: http://www.vmware.com/virtualization/

[3] (2012). *Panda Cloud Antivirus Free Edition*. Available: http://download.cnet.com/Panda-Cloud-Antivirus-Free-Edition/3000-2239_4-10914099.html

[4] J. Oberheide, E. Cooke, and F. Jahanian, "CloudAV: N-version antivirus in the network cloud," presented at the Proceedings of the 17th conference on Security symposium, San Jose, CA, 2008.

[5] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian, "Virtualized in-cloud security services for mobile devices," presented at the Proceedings of the First Workshop on Virtualization in Mobile Computing, Breckenridge, Colorado, 2008.

[6] J. O. a. E. C. a. F. Jahanian, "Rethinking antivirus: Executable analysis in the network cloud," in *In 2nd USENIX Workshop on Hot Topics in Security (HotSec)*, 2007.

[7] (2011). *Immunet antivirus available :* . Available: http://www.immunet.com/main/index.html

[8] (2013). *symantec cloud Available :* . Available: http://www.symantec.com/products-solutions/families/?fid=symantec-cloud

[9] T.kojm. (Nov 2005). *The Clam Antivirus Website*. Available: http://www.clamav.net/

[10] S. K. Cha, I. Moraru, J. Jang, J. Truelove, D. Brumley, and D. G. Andersen, "SplitScreen: enabling efficient, distributed malware detection," presented at the Proceedings of the 7th USENIX conference on Networked systems design and implementation, San Jose, California, 2010.

[11] X. Zhou, B. Xu, Y. Qi, and J. Li, "MRSI: A Fast Pattern Matching Algorithm for Anti-virus Applications," presented at the Proceedings of the Seventh International Conference on Networking, 2008.

[12] T.-H. L. a. N.-L. Huang, "An Efficient and Scalable Pattern Matching Scheme for Network Security Applications," in *Proceedings of the 17th International Conference on Computer Communications and Networks, IEEE ICCCN 2008,*

St. Thomas, U.S. Virgin Islands, August 3-7, 2008, 2008, pp. 951-957.

[13] D. Uluski, M. Moffie, and D. Kaeli, "Characterizing antivirus workload execution," *SIGARCH Comput. Archit. News,* vol. 33, pp. 90-98, 2005.

[14] M. I. Al-Saleh and J. R. Crandall, "Antivirus Performance Characterization: System-Wide View," in *7th Student Conference,* New Mexico, 2011, pp. 56-63.

[15] S. K. Ali Abdullah Hamzah, "Resource Allocation for Antivirus Cloud Appliances," *IOSR Journal of Computer Engineering, (IOSR-JCE)* vol. 10, pp. 2278-0661, 2013.

[16] A. A. Soror, U. F. Minhas, A. Aboulnaga, K. Salem, P. Kokosielis, and S. Kamath, "Automatic virtual machine configuration for database workloads," in *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD'08)*, Vancouver, Canada, 2008, pp. 953-966.

[17] A. A. Soror, U. F. Minhas, A. Aboulnaga, K. Salem, P. Kokosielis, and S. Kamath, "Deploying Database Appliances in the Cloud.," *IEEE Data Eng. Bull.,* vol. 32, No. 1, pp. 13-20, 2009.

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explor. Newsl.,* vol. 11, pp. 10-18, 2009.

[19] J. Rennie, "Derivation of the F-Measure ,in Derivation of the F-Measure," vol. 40, 2004.

[20] R. Wheeler, ""ROC, Precision-Recall : software with condence lmits and Bayes (predicted value) calculations,"in terms of unobservable, continuous underlying variables," 2011.

[21] a. S. K. J. Peltonen, "Generative Modeling for Maximizing Precision and Recall in Information Visualization," in *in Proceedings of the 14th International Conerence on Artificial Intelligence and Statistics*, 2011, pp. 579-587.