Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

# Cloud based Malware Detection System

Vinay J, Zuhair Bilal, Rohit K, Syed Ali Zuhaib
Department of Computer Science and Engineering
Dayananda Sagar University
Bengaluru, India

Dr. Mouleeswaran S K
Associate Professor
Department of Computer Science and Engineering
Dayananda Sagar University
Bengaluru, India

*Abstract*— **Now a days the cloud environment has seen a sudden growth in its users and about 52% of the organisations have stepped up to use the cloud infrastructures just because of its flexible resources and economic scale it provides. When it comes to malware, malware is any type of software or a bug which tries to self-replicate or harm the hardware or a software of a system. These types of attacks are not known to the human eye as they are built with malicious intent to harm any system in use. So overall to overcome the problems faced and make a flexible solution, we propose a framework where Machine learning algorithms are used to find the best features from the data set provided by us and give an accuracy report, and the classifiers used among them best algorithm prediction will be used to extract the best features and use them. The .exe file features will be extracted and compared with the best algorithms extracted features to detect weather a give input file from a user is malware or a legitimate file. The use of the cloud infrastructure will be to deploy the said project so that it becomes convenient for a user to deploy the model.**

***Keywords—Machine Learning, Malware, detection, Legitimate file, Malware file, Cloud computing.***

## I. INTRODUCTION

Malware is defined as software designed to infiltrate or damage a computer system without the owner's informed consent. Malware is actually a generic definition for all kind of computer threats. A simple classification of malware consists of file infectors and stand-alone malware. Another way of classifying malware is based on their particular action: worms, backdoors, trojans, rootkits, spyware, adware etc. Malware detection through standard, signature based methods is getting more and more difficult since all current malware applications tend to have multiple polymorphic layers to avoid detection or to use side mechanisms to automatically update themselves to a newer version at short periods of time in order to avoid detection by any antivirus software. For an example of dynamical file analysis for malware detection, via emulation in a virtual environment. Here we give a few references to exemplify such methods. Boosted decision trees working on n-grams are found to produce better results than both the Naive Bayes classifier and Support Vector Machines Uses automatic extraction of association rules on Windows API execution sequences to distinguish between malware and clean program files. The *main steps* performed through this framework are sketched as follows:

1. A set of *features* is computed for every binary file in the training or test *datasets*, based on many possible ways of analyzing a malware.
2. A machine learning system based firstly on one-sided perceptrons, and then on feature mapped one-sided perceptrons and a kernelized one-sided perceptrons, combined with feature selection based on the F1 and F2 scores, is trained on a medium-size dataset consisting of clean and malware files.

## II. BACKGROUND

### A. Cloud Computing

With the Internet's ubiquity in modern living, many argue that some level of cloud computing is now a common occurrence. This research heavily focuses on cloud computing technology, and thus requires a formal definition of cloud computing. Cloud computing cannot be easily defined. There are many definitions, which share the same common denominator: the Internet. Cloud computing is a way to use the Internet in the daily life of a single machine or single room, using all the tools installed on computers [Figure 1]. It is also the ability to use shared computing resources with local servers handling applications. With cloud computing users do not worry about the location and the storage of their data. They just start using the services anywhere and at any time. The main driver of this technology is Virtualization (Hypervisor) and virtual appliance.

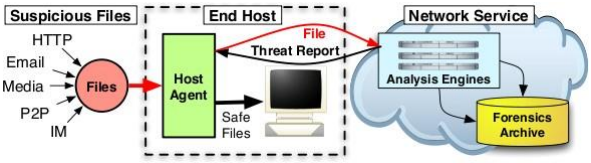


Fig. 1. The Flow of the Process of the cloud computing systems

Cloud computing offers different service models that allow customers to choose the appropriate service model that fits their environment needs, Cloud service models are software as

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

a service (SaaS), Platform as a service (PaaS), and Infrastructure as a service (IaaS):

- Software-as-a-service (SaaS): The consumer uses the provider's applications, which are hosted in the cloud. For example, Salesforce.com CRM Application.
- Platform-as-a-service (PaaS): Consumers deploy their own applications into the cloud infrastructure. Programming languages and application development tools used must be supported by the provider. For example, Google Apps.
- Infrastructure-as-a-service (IaaS): Consumers are able to provide storage, network, processing, and other resources, and deploy and operate arbitrary software, ranging from applications to operating systems.

### *B. Related Work*

As a matter of fact "cloud computing" concepts date backward to the 1950s, when large-scale mainframes were made available to schools and corporations, In addition, the on-demand computing concept of the cloud model went back to the time-sharing era in the 1960s. Therefore, many of the cloud computing security issues are arguably quite similar to the ones that were introduced during the Internet expansion era. However, Malware detection in a Cloud what we now commonly refer to as cloud computing is the result of an evolution of the widespread adoption of Virtualization, service-oriented architecture, autonomic, and utility computing. Details such as the location of infrastructure or component.

Devices are unknowns to most end-users, who no longer need to be thorough, understand or control the technology infrastructure that supports their computing activities. There are several previous studies related to this research dealing with all of cloud computing and its structure as well as detection systems used for each of the Static analysis, detection: Signature Optimizing Pattern Matching and Dynamic analysis detection: Heuristic.

### *C. In Cloud Computing*

This novel paradigm provides significant advantages over traditional host-based an antivirus, including better detection of malicious software, enhanced forensic's capabilities, improved deployable and manageable retrospective detection. Use a production implementation and real-world deployment of the Cloud AV platform. An approach for combined malware detection and kernel rootkit prevention in virtualized cloud computing environments, and all running binaries in virtual instance are intercepted and submitted to one or more analysis engines. Besides a complete check against a signature database, lives introspection of all system calls is performed to detect yet unknown exploits or malware.

Malware detection has been an important issue in computing since the late '80s. Since then the predominant method of malware detection has been to scan a computer system for infection by matching malware signatures to files on the computer. Although detection of known samples is extremely reliable, signature based detection only works for malware that has been obtained, analyzed and a suitable signature identified.

Decreased the signature mapping cost by optimizing signature library, taking advantage of common conduct characteristics of viruses such as self-replicate and seasoning, and proposed optimization policy against this scalability issue with the help of data mining. Moreover, he decreased the number of unnecessary signature matching and raises efficiency of that comparison procedure by rearrangement within a signature library. In Heuristic detection, Treadwell suggested analyzing the obfuscation pattern before unpacking, providing a chance to prevent malware from further execution.

## III. SYSTEM DESIGN

The framework configuration prepare develops general structure building outline. Programming diagram incorporates addressing the item system works in a shape that might be changed into at least one anticipates. The essential demonstrated by the end customer must be placed in a systematical way. Diagram is a creative system; an extraordinary design is the best approach to reasonable structure. The structure "Layout" is portrayed as "The methodology of applying distinctive frameworks and guidelines with the ultimate objective of describing a strategy or a system in sufficient purpose important to permit its physical affirmation". Diverse design segments are taken after to add to the system. The design detail depicts the segments of the system, the sections or segments of the structure and their appearance to end-customers.

### *A. Proposed System*

The clean files in the training database are mainly system files (from different versions of operating systems) and executable and library files from different popular applications. We also use clean files that are packed or have the same form or the same geometrical similarities with malware files (e.g use the same packer) in order to better train and test the system. The malware files in the training dataset have been taken from the Virus Share collection. The test dataset contains malware files from the dataset collection and clean files from different operating systems (other files that the ones used in the first database). Large dataset was used for testing the scaling up capabilities of the used machine learning algorithms.

### *B. System Architecture*

The architectural configuration procedure is concerned with building up a fundamental basic system for a framework. It includes recognizing the real parts of the framework and interchanges between these segments. The beginning configuration procedure of recognizing these subsystems and building up a structure for subsystem control and correspondence is called construction modeling outline and the yield of this outline procedure is a portrayal of the product structural planning. The proposed architecture for this system

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

is given below. It shows the way this system is designed and brief working of the system.

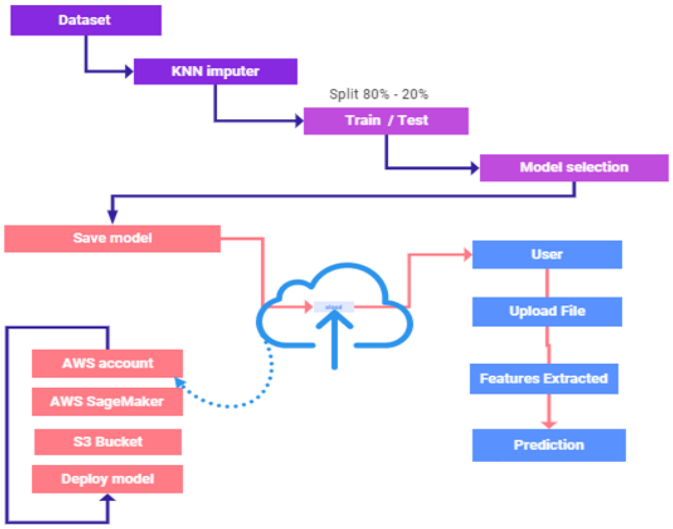


Fig. 2. System Architecture Model

*C. Data Flow Diagrams*

DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

- Logical information flow of the system
- Determination of physical system construction requirements
- Simplicity of notation
- Establishment of manual and automated systems requirements

**DFD Components** - DFD can represent Source, destination, storage and flow of data using the following set of components.

**Entities** - An external entity is a person, department, outside organization, or other information system that provides data to the system or receives outputs from the system.

**Process** - any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules.

**Data Storage** - files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as "Orders."

**Data Flow** - the route that data takes between the external entities, processes and data stores.

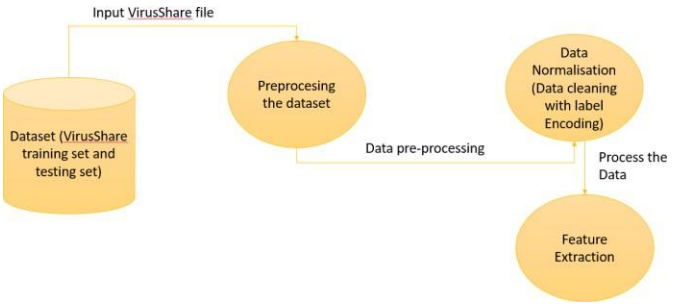


Fig. 3. Data Flow Diagram – L0

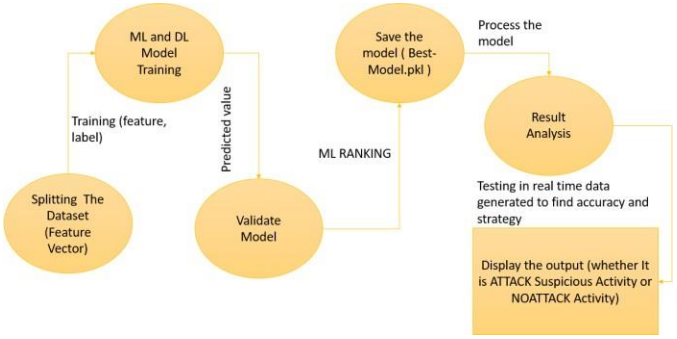


Fig. 4. Data Flow Diagram- L1

*D. Use Case Diagrams*

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows −

- Used to gather the requirements of a system.
- Used to get an outside view of a system.
- Identify the external and internal factors influencing the system.
- Show the interaction among the requirements are actors.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

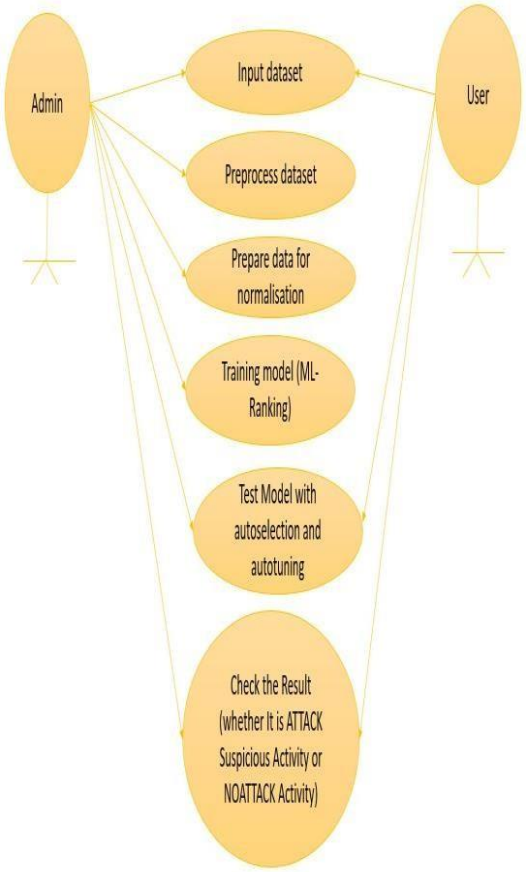


Fig. 5. Use case Diagram of the Model

### *E. Sequence Diagram*

A sequence diagram is a system is an interaction diagram that shows how process operates with one and other and in what order. It's a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and sequence of messages exchange between the objects needed to carry out the functionality of the scenario. Sequence diagram are sometimes called event diagrams or event scenarios.

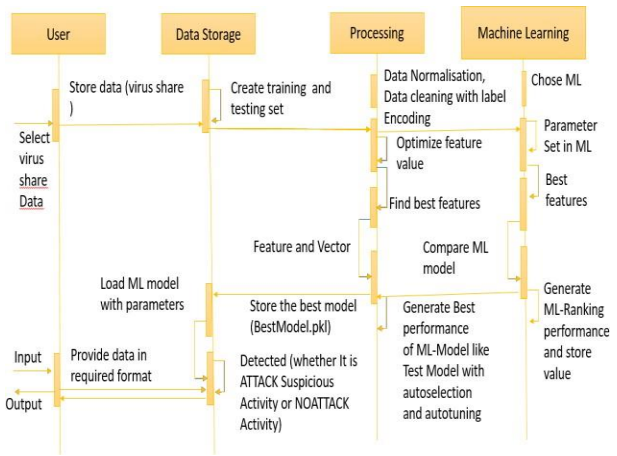


Fig. 6. Sequence Diagram of the Model

## IV. MODULES

This stage is the underlying stage in moving from issue to the course of action space. Accordingly, starting with what is obliged; diagram takes us to work towards how to full fill those requirements. System plot portrays all the critical data structure, record course of action, yield and genuine modules in the structure and their Specification is picked. This assumes an essential part on the grounds that as it will give the last yield on which it was being working.
In our work we are using some modules, these modules are listed below.

### *A. Gathering Data*

Once you know exactly what you want and the equipment's are in hand, it takes you to the first real step of machine learning- Gathering Data. This step is very crucial as the quality and quantity of data gathered will directly determine how good the predictive model will turn out to be. The data collected is then tabulated and called as Training Data.

#### 1. THE VIRUS-SHARE DATASET

The VirusShare dataset is a repository of malware samples to provide security researchers, incident responders, forensic analysts, and the morbidly curious access to samples of live malicious code.

### *B. Data Preparation*

After the training data is gathered, you move on to the next step of machine learning: Data preparation, where the data is loaded into a suitable place and then prepared for use in machine learning training. Here, the data is first put all together and then the order is randomized as the order of data should not affect what is learned.

This is also a good enough time to do any visualizations of the data, as that will help you see if there are any relevant relationships between the different variables, how you can take their advantage and as well as show you if there are any data imbalances present. Also, the data now has to be split into two parts. The first part that is used in training our model, will be the majority of the dataset and the second will be used for the evaluation of the trained model's performance. The other forms of adjusting and manipulation like normalization, error correction, and more take place at this step.

### *C. Choosing a Model*

The next step that follows in the workflow is choosing a model among the many that researchers and data scientists have created over the years. Make the choice of the right one that should get the job done.

### *D. Training*

After the before steps are completed, you then move onto what is often considered the bulk of machine learning called training where the data is used to incrementally improve the model's ability to predict.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

The training process involves initializing some random values for say A and B of our model, predict the output with those values, then compare it with the model's prediction and then adjust the values so that they match the predictions that were made previously.

This process then repeats and each cycle of updating is called one training step.

### *E. Evaluation*

Once training is complete, you now check if it is good enough using this step. This is where that dataset you set aside earlier comes into play. Evaluation allows the testing of the model against data that has never been seen and used for training and is meant to be representative of how the model might perform when in the real world.

### *F. Parameter Tuning*

Once the evaluation is over, any further improvement in your training can be possible by tuning the parameters. There were a few parameters that were implicitly assumed when the training was done. Another parameter included is the learning rate that defines how far the line is shifted during each step, based on the information from the previous training step. These values all play a role in the accuracy of the training model, and how long the training will take.

For models that are more complex, initial conditions play a significant role in the determination of the outcome of training. Differences can be seen depending on whether a model starts off training with values initialized to zeroes versus some distribution of values, which then leads to the question of which distribution is to be used. Since there are many considerations at this phase of training, it's important that you define what makes a model good. These parameters are referred to as Hyper parameters. The adjustment or tuning of these parameters depends on the dataset, model, and the training process. Once you are done with these parameters and are satisfied you can move on to the last step.

### *G. Prediction*

Machine learning is basically using data to answer questions. So this is the final step where you get to answer few questions. This is the point where the value of machine learning is realized. Here you can finally use your model to predict the outcome of what you want.

The above-mentioned steps take you from where you create a model to where you predict its output and thus acts as a learning path.

## V. METHODOLOGY

### *A. Random Forest Algorithm*

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the **basic steps** involved in performing the random forest algorithm:

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

### *B. Gaussian Naïve Bayes*

Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution.

An approach to create a simple model is to assume that the data is described by a Gaussian distribution with no co-variance (independent dimensions) between dimensions. This model can be fit by simply finding the mean and standard deviation of the points within each label, which is all what is needed to define such a distribution.

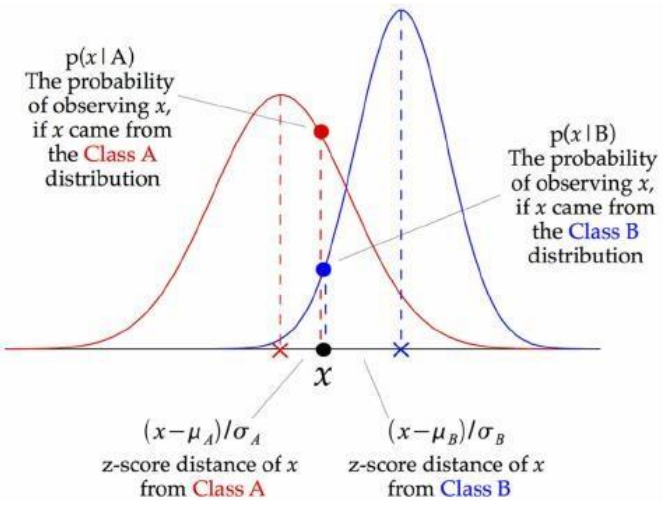


Fig. 7. The above illustration indicates how a Gaussian Naive Bayes (GNB) classifier works.

At every data point, the z-score distance between that point and each class-mean is calculated, namely the distance from the class mean divided by the standard deviation of that class.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

The conditional probability of event A given event B means the probability of event A occurring given that event B has already occurred. Mathematically, the conditional probability of A given B can be denoted as P[A|B] = P[A AND B] / P[B].

### *C. Decision Tree Algorithm*

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

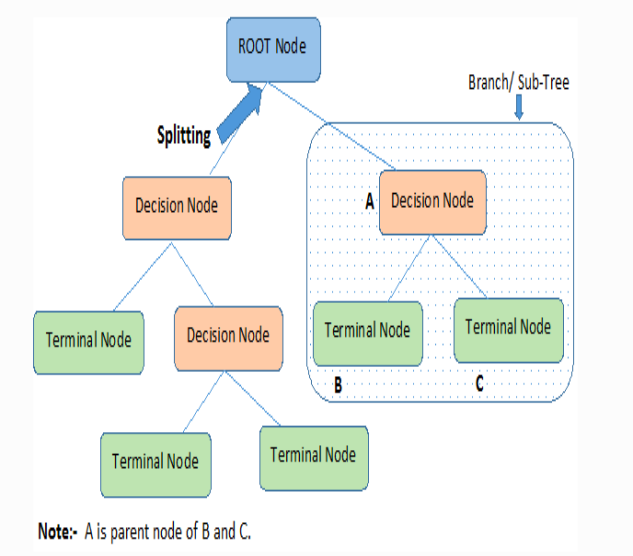


Fig. 8. The above illustration indicates how a Decision Tree Classifier Works

### 1. WORKING PROCEDURE

The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria are different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split, which results in most homogeneous sub-nodes.

## VI. CONCLUSION AND FUTURE WORK

To conclude, we have proposed a system for combined malware detection systems and cloud computing environments, all running binaries and malware are intercepted by submitting to one or more analysis engines, a complete check against a signature database to detect yet unknown exploits or malware. We will suggest increasing in the dependence of cloud computing as consumers increasingly move to cloud computing platforms for their computing needs. In this paper, we reviewed previous work on malware detection, both conventional and in the presence of storage in order to determine the best approach for detection in the cloud. We also argue the benefits of distributing detection throughout the cloud and present a new approach to coordinate detection across the cloud.

In the proposed system, we have used traditional detection techniques (optimizing pattern) as per static signatures and dynamic detection technology (heuristic). Then, we have chosen for safer system methods as well as speed and modern to rival existing anti-virus.

The proposal of this work is to find the best solutions to the problems of anti-viruses and improve performance and find possible alternatives for a better working environment without problems with high efficiency and flexibility.

We used the optimal traditional methods and modern to detect viruses, for unknown and already detected viruses through the signatures and the Heuristic.

Future work in this field will focus on the development of detection systems based on memory introspection and heuristic or statistical detection, as opposed to signature-based detection.

## ACKNOWLEDGMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work. First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean**, **School of Engineering & Technology**, **Dayananda Sagar University** for his constant encouragement and expert advice.It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman**, **Computer Science, and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible. We would like to thank our guide **Dr Mouleeswaran SK, Associate Professor**, **Dept. of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

We would like to thank our Project Coordinator **Dr. Meenakshi Malhotra** and all the staff members of Computer Science and Engineering for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 05, May-2022

## REFERENCES

[1] Microsoft, "Microsoft security intelligence report", [online]:http://www.microsoft.com/technet/security/default.mspx, July December 2006.

[2] Dropbox, Inc., dropbox.com webpage, [Online]: https://www.dropbox.Com/ (accessed 13/04/12).

[3] C. Grace. "Understanding intrusion-detection systems" [J], PC Network Advisor, vol. 122, pp. 11-15, 2000.

[4] S. Subashini, V. Kavitha s.l "A survey of security issues in service delivery models of cloud computing." .Science Direct, Journal of Network and Computer Applications, pp. (1-11) January (2011).

[5] Shirlei Aparecida de Chaves, Rafael Brundo Uriarte and Carlos Becker Westphall "Toward an Architecture for Monitoring Private Clouds." S.l. IEEE December (2011).

[6] Bo Li, Eul Gyu I'm " A signature matching optimization policy for anti-virus programs" Electronics and Computer Engineering, Hanyang University, Seoul, Korea. © IEEE 2011

[7] Chen, Z. & Yoon, J. "IT auditing to assure a secure cloud computing", (2010). [Online]: http://doi.ieeecomputersociety.orgezproxy.umuc.edu/10.1109/SERVICES.2010.118.

[8] J. Oberheide, E. Cooke, and F. Jahanian "CloudAV: N- Version Antivirus in the Network Cloud", In Proceedings of the 17th USENIX Security Symposium (Security'08). San Jose, CA, 2008.

[9] Jon Oberheide, Evan Cooke and Farnam Jahanian "Cloud N-Version Antivirus in the Network Cloud",Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI 48109 (2007).

[10] Matthias Schmidt, Lars Baumg Artner, Pablo Graubner, David Bock and Bernd Freisleben "Malware Detection and Kernel Rootkit Prevention in Cloud Computing Environments." University of Marburg, Germany (2011).

[11] K. Murad, S. Shirazi, Y. Zikria, and I. Nassar, "Evading Virus Detection Using Code Obfuscation" in Future Generation Information Technology, vol. 6485 of Lecture Notes in Computer Science, pp. 394–401, Springer Berlin , Heidelberg, 2010.

[12] Scott Treadwell, Mian Zhou "A Heuristic Approach for Detection of Obfuscated Malware", Bank of America, 1201 Main St, Dallas, TX 75202, © IEEE 2009.

[13] Carlin, S., & Curran, K. "Cloud computing security", International Journal of Ambient Computing and Intelligence.

[14] "Heuristic analysis in Kaspersky Internet Security" [Online]: http://support.kaspersky.com , ID: 8936 , 2013 Mar 01 2013

[15] Algirdas Avizienis, "The n-version approach to fault- tolerant software", IEEE Transactions on Software Engineering, 1985.

[16] Rodrigo Rodrigues, Miguel Castro, and Barbara Liskov. Base, "using abstraction to improve fault tolerance", In Proceedings of the eighteenth ACM symposium on Operating systems principles, New York, NY, USA, 2001.

[17] Lajos Nagy, Richard Ford, and William Allen, "N- version programming for the detection of zero-day exploits", In IEEE Topical Conference on Cybersecurity, Daytona Beach, Florida, USA, 2006.

[18] Carsten Willems and Thorsten Holz. Cwsandbox.[Online]: http://www.cwsandbox.org/, 2007.

[19] Hispasec Sistemas. "Virus total", [Online]: http://virustotal.com, 2004.

[20] Norman Solutions. Norman sandbox whitepaper. http://download.norman.no/whitepapers/whitepaper_Norman_SandBox.pdf, 2003.

[21] Barracuda Networks. "Barracuda spam firewall",[Online]: http://www.barracudanetworks.com, 2007.

[22] Cloudmark, "Cloudmark authority anti-virus",[Online]:http://www.cloudmark.com, 2007.

[23] Alexander Moshchuk, Tanya Bragin, Damien Deville, Steven D. Gribble, and Henry M. Levy, "Spyproxy: Execution-based detection of malicious web content", In Proceedings of the 16th USENIX Security Symposium, August 2007.

[24] Stelios Sidiroglou, Angelos Stavrou, and Angelos D. Keromytis, "Mediated overlay services (moses): Network security as a composable service", In Proceedings of the IEEE Sarnoff Symposium, Princeton, NJ, USA, 2007.