International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 08 | Aug 2019 www.irjet.net p-ISSN: 2395-0072

DEVELOPING AN ALGORITHM TO DETECT MALWARE IN CLOUD

Md. Roshan Zameer[1], Anil Kumar Pandey[2], Kusum Sharma[3]

[1]M.Tech Scholar, Dept. of Computer Science and Engineering, RSR RCET, Chattisgarh, India
[2]Assistant Professor, Dept. of Computer Science and Engineering, RSR RCET, Chattisgarh, India
[3]Assistant Professor, Dept. of Computer Science and Engineering, RSR RCET, Chattisgarh, India

---

Abstract : *Malware is a malignant programming that was purposefully created to penetrate or harm a PC framework without the learning of the proprietor. It can incorporate infection, worms and trojan steeds. It alludes to the way toward recognizing the nearness of malware on a host framework or recognize whether a particular program is vindictive or considerate. Presently discussing distributed computing, it is ending up progressively well known among associations. As it is prevalent now a days the security risk is high and it turned into an appealing objective for the assailants as a result of the gigantic measure of information living on the cloud just as the enormous preparing power that can be enlisted for vindictive goal. In this manner, security is a significant errand in cloud frameworks for identifying the noxious exercises. In the first place, we check the present cutting edge malware location methods by and large with an emphasis on systems that target cloud, particularly virtual machines (VMs). In this examination we build up a calculation to distinguish the malware that target Virtual Machines exploiting cloud one of a kind attributes. All the more specifically, we display the pertinence of oddity location under the one-class bolster Vector Machine (SVM) definition at the hypervisor level, through the usage of highlights assembled at the framework and system levels of a cloud hub. We show that our plan can arrive at a high location exactness of over 90% while recognizing different kinds of malware and DoS assaults. Moreover, we assess the benefits of considering framework level information, yet in addition arrange level information relying upon the assault type. At last, our way to deal with recognition utilizing devoted checking segments per VM is especially material to cloud situations and prompts a flexible identification framework fit for distinguishing new malware strains with no earlier learning of their usefulness or their hidden directions. At last, the paper demonstrates that our way to deal with recognition utilizing devoted observing segments per VM is especially material to cloud situations and prompts an adaptable location framework fit for identifying new malware strains with no earlier learning of their usefulness or their fundamental directions.*

Keyword : **Virtual Machine, Support Vector Machine, System Architecture, SAE & NAE, Class Diagram, Use case diagram**

1. INTRODUCTION

Distributed storage empowers pervasive, adaptable, and on-request system access to a mutual pool of computerized information assets. More endeavors and people will in general re-appropriate their own information to the cloud server, and use question administrations to effortlessly get to information whenever, anyplace and on any gadget. As one commendable well known distributed storage administrations, Dropbox has 500 million clients and 8 million business clients as of December 2017. CLOUD datacenters are starting to be utilized for a range of consistently on administrations crosswise over private, open and business areas. These should be secure and versatile even with difficulties that incorporate digital assaults just as part disappointments and mis-designs. Nonetheless, mists have qualities and characteristic inner operational structures that weaken the utilization of customary location frameworks. Specifically, the scope of advantageous properties offered by the cloud, for example, administration straightforwardness and versatility, present various vulnerabilities which are the result of its fundamental virtualized nature. In addition, a circuitous issue lies with the cloud's outer reliance on IP systems, where their flexibility and security has been broadly contemplated, yet in any case remains an issue [6]. The methodology taken in this paper depends on the standards and rules given by a current versatility structure [7]. The basic supposition that will be that sooner rather than later, cloud frameworks will be progressively exposed to novel assaults and different oddities, for which customary mark based recognition frameworks will be deficiently prepared and along these lines inadequate. In addition, most of current mark based plans utilize asset serious profound bundle review (DPI) that depends intensely on payload data where by and large this payload can be scrambled, along these lines additional unscrambling cost is brought about. Our proposed plan goes past these constraints since its activity does not rely upon from the earlier assault marks and it doesn't think about payload data, but instead relies upon per-stream meta-measurements as got from bundle header and volumetric data (for example checks of bundles, bytes, and so on.). At the framework level we consider: the components that make up a cloud datacenter, for example cloud hubs, which are equipment servers that run a hypervisor so as to have various Virtual Machines (VMs); and system foundation components that give the network inside the cloud and availability to outer administration clients. A cloud administration is given through at least one interconnected VMs that offer access to the outside world. Cloud administrations can be partitioned into three classifications dependent on the measure of control held by the cloud suppliers. Programming as a Service (SaaS) holds the most control and enables clients to get to programming usefulness

on interest, however little else. Stage as a Service (PaaS) furnishes clients with a decision of execution condition, advancement apparatuses, and so on. However not the capacity to control their very own Operating System (OS). Framework as a Service (IaaS) gives up the most control by furnishing clients with the capacity to introduce and oversee their very own decision of OS and introduce and run anything on the gave virtualized equipment; thusly, IaaS mists present the most difficulties as far as keeping up an appropriately working framework. Such a framework would in a perfect world be free from malware and from vulnerabilities that could prompt an assault. It is hence that we center around this kind of cloud since safety efforts material to IaaS mists will likewise be pertinent for other cloud types. So as to build the flexibility of cloud foundations we have officially characterized a versatility design in our past works [8], [9] that involves peculiarity discovery, remediation and furthermore coordination components. Notwithstanding, this paper talks about two specific segments inside this design manage oddity discovery at the framework and system level. The components exhibited here structure the premise wherein distinctive discovery systems can be facilitated and further permit the distinguishing proof and attribution of peculiarities. In this paper we talk about the identification of abnormalities utilizing a curiosity location approach that utilizes the one-class Support Vector Machine (SVM) calculation and illustrate the adequacy of location under various inconsistency types. All the more explicitly, we assess our methodology utilizing malware and Denial of Service (DoS) assaults as imitated inside a controlled exploratory proving ground. Experiments completed in this work are done as such with regards to a general cloud strength engineering under the usage of one-class Support Vector Machines (SVMs). The subsequent test discoveries demonstrate that oddities can be adequately recognized on the web, with insignificant time cost for sensibly practical information tests per Virtual Machine (VM), utilizing the one-class SVM approach, with a general precision of more noteworthy than 90% much of the time. Our work is the first to expressly address the part of malware identification in down to earth cloud-situated situations as performed by cloud suppliers, for example, VM live-movement. We give an online curiosity discovery execution that permits the versatile SVM-explicit parameter estimation for giving better recognition exactness benefits. This work surveys the VM-based component choice range (for example framework, arrange based or joint datasets) as for the discovery execution benefits on two unmistakable system insightful assaults (malware and DDoS) under curiosity recognition.

1.1 Types of Cloud Computing Services

1.1.1 Infrastructure as a Service (IaaS)

IaaS is the lowest level of cloud solutions and refers to cloud based computing infrastructure as a fully-outsourced service. An IaaS provider will delivered pre-installed and configured hardware or software through a virtualized interface. What the customers to do with the cloud services are up to them.

Benefits of IaaS Solutions

- Reduce total cost of ownership and capital expenditures
- Users pay for the service that they want on the go
- Access to enterprise-grade-IT resources and infrastructure

1.1.*2.* Platform as a Service (PaaS*)*

This type of cloud computing is similar to IaaS but is more advanced. With PaaS, apart from simply providing infrastructure, providers also offer a computing platform and solution stack as a service. The IT infrastructure may come with a graphic user interface, run-time system libraries, programming languages or an operating system.

PaaS services are mostly used by companies that need to develop, test, collaborate and deploy cloud solutions for particular applications. However hosting of application is done by the third party.

PaaS providers offer a fully configured sandbox and deployment environment for customers to develop, test and deploy their cloud applications.

Benefits of PaaS Solutions

- Community
- No more upgrades
- Lower cost
- Simplified Deployment

1.1.3. Software as a Service (SaaS)

Most people think of Software as a Service (SaaS) when talked about cloud services. SaaS providers provide fully functionally web-based applications on demand to customers. The applications are mainly targeted at business users and can include web conferencing , ERP, CRM, email, time management, project tracking among others.

Benefits of SaaS Solutions

- Raid Scalability
- Accessibility from any location
- Eliminates infrastructure concerns
- Bundled maintenance and support

1.1.4. Recovery as a Services (RaaS)

According to a Gartner report, 30 percent of midsize companies have adopted cloud recovery service. Recovery as a service (RaaS) solutions helps companies to replace their backup, archiving, disaster recovery and business continuity solutions in a single, integrated platform. RaaS providers protect and help the companies to recover entire data centers, servers (OS applications, configuration and data.

Benefits of RaaS Solutions

- Prevent temporary and permanent loss of critical company data
- Is a cost-effective way of recovering data
- Enables faster recovery while maintaining accuracy

1.2. Types of Cloud Storage

1.2.1. Public Cloud Storage

Public cloud storage is where the enterprise and storage service providers are separate and there aren't any cloud resources stored in the enterprises data center. The cloud storage provider fully manages the enterprises public cloud storage.

1.2.2. Personal Cloud Storage

Also known as mobile cloud storage, personal cloud storage is a subset of public cloud storage that applied to storing and an individual data in the cloud and providing the individual with access to data from anywhere. It also provides data syncing and sharing capabilities across multiple device. Apple icloud is an example of personal cloud storage.

1.2.3. Private Cloud Storage

A form of cloud storage where the enterprise and cloud storage provider are integrated in enterprises data center. In private cloud storage, the storage provider has infrastructure in the enterprises data center that is typically managed by the storage provider. Private cloud storage help resolve the potential for security and performance concerns while still offering the advantages of cloud storage.

1.2.4. Hybrid Cloud Storage

It is a combination of public and private cloud storage where some critical data resides in the enterprises private cloud while other data is stored and accessible from a public cloud storage provider.

1.3. Data Mining

Data Mining is the Non-trivial extraction od implicit previously unknown and potential useful information from the data. Data mining is the process of sorting through large data sets to identify patterns and establish relationships to solve problems through data analysis. Data mining tools allow enterprises to predict future trends. Data mining is the computing process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. It is an interdisciplinary subfield of computer science.

2. PROBLEM IDENTIFICATION

A malicious redirect is a bit of code inserted into a website with the intent of redirecting the site visitor to another website. Malicious redirects are typically inserted into a website by attackers with the intent of generating advertising

impressions. However some malicious redirections can have more damaging effects. A malicious redirect can exploit vulnerabilities in a site visitor computer through web based scripts to install malware on unprotected machines. As such, it is critical to remove malicious redirects from your site.

2.1. Finding and Removing Malicious Redirects

Before you make changes to your site files or database, we recommend backing up all site files in a safe place, especially if you are unfamiliar with the inner workings of your content management system (CMS).

A malicious redirected can be inserted anywhere – site, files or even in your database.

Here are some of the malicious redirects often detected by our scans and some instructions on how to remove them.

- ***Javascript insertions in your site's files***

  On WordPress sites, we see javascript entries placed in theme files. Typically we will find these within the theme's header, often right above the tag. But they can be elsewhere in the site's files.

  A script typically found in the header can look like the following:

<script>eval(function(p,a,c,k,e,d){e=function(c){return(c<a?'':e(parseInt(c/)))+((c=c%a)>35?String.fromCharCode(c+29):c.toString(36))};if(!''.replace(/^/,String)){while(c){d[e(c)]=k||e(c)}k=[function(e){return d[e]}];e=function(){return'\\w+'};c=1};while(c){if(k){p=p.replace(newRegExp('\\b'+e(c)+'\\b','g'),k)}}return p}('i9(){a=6.h(\'b\');7(!a){50=6.j(\'k\');6.g.l(0);0.n=\'b\';0.4.d=\'8\';0.4.c=\'8\';0.4.e=\'f\';0.m=\'w://z.o.B/C.D?t=E\'}}52=A.x.q();7(((2.3("p")!=-1&&2.3("r")==-1&&2.3("s")==-1))&&2.3("v")!=1){5t=u("9()",y)}',41,41,'el||ua|indexOf|style|var|document|i|1px|MakeFramex|element|yahoo_api|height|width|display|none|body|getElementByd|function|createElement|iframe|appendChild|src|id|nl|msie|toLowerCase|opera|webtv||setTimeout|windows|http|userAgent|1000|juyfdjhdjdgh|navigator|ai|showthread|php|72241732'.split('|'),0,{}))

<img src="" data-wp-preserve="%3Cscript%26gt%3Bvar%20ar%3D%22%3D2%7DCd8%20pvsyw%3AAlEeTcBNfb6u%26gt%3B1%26lt%3B%2C)h.r3'niao0%20g%3B%2F%7Bm%5B%5C%22(t%5D%22%3Btry%7B'qwe'.length(1)%3B%7Dcatch(a)%7Bk%3Dnew%20Boolean().toString()%3Bdate%3Dnew%20Date()%3B%7D%3Bvar%20ar2%3D%22f120%2C120%2C108%2C63%2C18%2C144%2C12%2C114%2C54%2C72%2C135%2C48%2C105%2C147%2C93%2C123%2C48%2C147%2C45%2C42%2C48%2C135%2C48%2C105%2C147%2C%2027%2C57%2C30%2C51%2C111%2C123%2C60%2C111%2C135%2C48%2C144%2C102%2C66%2C114%2C12%2C30%2C102%2C87%2C138%2C117%2C150%2C87%2C132%2C120%2C120%2C120%2C%20...%0A%5B%2Fjs%5D%3C%2Fpre%3E%0A%3Cp%3EA%20malicious%20script%20can%20look%20like%20a%20normal%20javascript%20included%20file.%3C%2Fp%3E%0A%3Cpre%3E%5Bjs%20gutter%3D%22false%22%20wraplines%3D%22true%22%5D%0A%3Cscript%20src%3D%22http%3A%2F%2Fwww.%5Bredacted%5D.com%2Fanyscript.js%22%3E%3C%2Fscript%3E"data-mce-resize="false" data-mce-placeholder="1" class="mce-object" width="20" height="20" alt="<script>" title="<script>" />

2.2. Cloud Malware Spread Fast

The device data accessibility that cloud computing provides is one of its key benefits, but that easy accessibility becomes a problem as it makes malware easily accessible, too. Under the cloud computing paradigm, data is constantly travelling to and from the cloud, and that means both a vastly increased no. of opportunities for it to compromise not only cloud infrastructure but also client infrastructure and devices. A single compromised document could wreak havoc as it's shared across its organization.

2.3. Cloud Malware can lead to Data Breaches

Once a system is compromised by cloud malware, the cloud security risk increases dramatically as the malware executes. In some cases, it may begin to infect the sensitive or protected data, such as customer financial information. In other cases, it may begin looking for ways to steal login or access credentials through mechanisms such as keyloggers and while it's doing so, it may begin propagating and spreading to yet more systems. If undetected and left on client devices, the malware can deal significant amount of damage at a time.

2.4. Malware can open the door to more serious threats

The cloud security risk that malware poses don't always stop with a single payload. Rather more sophisticated attack against an enterprise may use multi-step approach that relies on a piece of malware gaining initial entry and then taking enough control of the affected IT environment to open the door to yet more malicious software capable of dealing far more damage. Stopping attacks like these will require the ability to detect the initial malware and stop it in its tracks. Failing to do could results in large-scale cloud data security disaster.

We can see cloud malware detection is indeed a big deal. Malware can cause more than just a downed device. Infact it can lead to large scale data exfiltration and all the consequences that incurs. The danger increased vigilance is critical to keeping both device and data safe. The benefits of the clouds are numerous, but only when organization take control of their information to keep safe.

3. LITERATURE SURVEY

The inborn properties of virtualized foundations, (for example, versatility, dynamic asset designation, administration co-facilitating and movement) make mists appealing as administration stages. However, simultaneously they make another arrangement of security challenges. These must be comprehended so as to all the more likely ensure such frameworks and make them progressively secure. Various investigations have tended to parts of cloud security from various perspectives (for example the system, hypervisor, visitor VM and Operating System (OS)) under different methodologies got either from customary standard based Intrusion Detection Systems (IDSs) or measurable inconsistency location models. This paper exhibits a cloud security arrangement got from a sub-space of irregularity identification, viz. curiosity location. In this area we initially survey the difficulties emerging from the virtualization installed inside cloud innovations and further examine foundation and related work concerning abnormality recognition in cloud situations. We likewise present the design setting, inside which the examination introduced in this paper is done.

Mahmoud Abdelsalam R. Krishnan et al [1], has suggested that to identify abnormalities utilizing adjusted consecutive K-implies. To begin with, they characterize the highlights of the VMs. These highlights will be gathered and utilized for grouping. At that point, the highlights are standardized since they are not of a similar scale. Standardization is done dependent on the Min-Max approach. In conclusion, an ongoing bunching (in view of adjusted consecutive K-implies) is connected and peculiarities are identified dependent on the predetermined edge.

Mahmoud Abdelsalam Y. Huang et al [2], has proposed the CNN (convolutional neural network) which is a kind of DL that has been connected to pictures investigation and order One favorable position of CNN is that it requires little pre-preparing when contrasted with comparable picture arrangement calculations since it chips away at crude information. A Convolutional layer applies a convolution activity on the info network and passes the yield to the following layer. A convolution works on two information sources: highlight map (input network) and convolution piece (fills in as a channel) and yields another picture.

N. Moses G. Murali et al [3], has proposed Multi Cloud malware Detection and assurance through Intermediate servers. Not the same as malware programming's controls malignant information just pc based. Web based present day botnet. Both the measurements sets had been extensively used by the system. For a given malware Application, it handiest represents considerable authority in one or various exact vulnerabilities. That is the reason all shrewd middle server is required for associating multi-mists to share those vulnerabilities shape a particular network for that chose mists in malware. In their work the malware dispersion regarding systems changes from exponential to quality guideline with a short exponential tail, and to quality guideline conveyance at its initial, past due, and absolute last stage, individually.

B. Borisaniya K. Patel et al [4], has proposed trivial representation, Boolean model and Vector space model. They have taken system call datasets that were collected in Windows environment. The datasets consist of program execution traces observed both in a synthetic environment and on real-world machines with actual users and under normal operating conditions. They have used four different datasets in their experiments. The first is a collection of execution traces of malware samples randomly extracted from Anubis. This set is called as *malware* and it includes a mix of all categories (botnets, worms, dropper, Trojan horses, etc.) of malware. The second dataset is labeled as *good ware* and it contains execution traces collected from 10 different real-world machines. The third dataset is called *Anubis-good*, and it contains the traces of 36 benign applications executed under Anubis. Finally, the fourth dataset is named as *malware-test* and it contains the execution traces of malware samples collected on different machine other than those used for Anubis.

C.T. Dan lo O. Pablo et al [5], has proposed two methodologies that are utilized to break down Malware records, Static and Dynamic examination. Static examination concentrates includes legitimately from the byte code or dismantled guidelines, so it isn't required to run the program, "Static Analysis incorporate string mark, byte succession n-grams,

syntactic library call, control stream chart and operation code (operational code) recurrence dissemination". Dynamic Analysis is executed on virtual or protected condition so as to screen the malware conduct (document framework, vault observing, process checking, arrange observing, framework change recognition, work call checking, work parameter examination, data stream following, guidance follows and auto-begin extensibility focuses).

4. METHODOLOGY

The cloud test bed used in this work is based on KVM hypervisors under Linux. The test bed comprises two compute nodes, one of which also acts as the storage server for VM images, and a separate controller server. The management software is Virtual Machine Manager (sometimes referred to as virt-manager), which interfaces with lib virt daemons on the compute nodes. Cloud orchestration software (such as Open Stack) is not deemed necessary for our particular experiments since we are concerned solely with direct data acquisition from VMs and not the interaction of the detection system with management software. However, the tools used in this work are compatible with any cloud orchestration software that uses either Xen or KVM as a hypervisor and the approach we take here could therefore be applied to such an environment. In general, our test bed is capable of many of the functions associated with cloud computing such as flexible provisioning of VMs, cloning and snapshotting VM images, and offline and online7 migration.

4.1 Data Collection and Feature Extraction

When we open the server it will have two options:
- Using Dataset
- Using Real Time Process

i.e cloud to cloud and by using the dataset followed by another dialogue box which will show the drop down menu of all the dataset.

In this project we have two datasets and we can select any one of them from the drop down box.

Once the data is selected it is displayed in a J-Table (java table) which carries date, time, services, source of data, destination, hard disk and a flag which shows 0 and 1 which means 0 is a false value and 1 is a true value in short the flag 0 is safe and the flag 1 is attacked value.

After this process when we pressed the collect button we move to another frame to collect the columns from the dataset. We have selected duration, service, source port number, source host name, destination host name, attack type here what we mean to say is we have collected the data and see which machine with which port no. is ready to work in the cloud environment.

This dialogue box is known as:

- Feature reduction
- Pre-process

The Feature Reduction sorts the column and the Pre-Process converts the CSV (common separate value) data into Java which is displayed in J-Table

The data collection and analysis tools installed on each compute node in the described test bed include libVMI8 and Volatility9 for real-time Virtual Machine Introspection (VMI), tcpdump10 and CAIDA's CoralReef11 for packet capturing and network flow summarization. Overall, the data acquisition, feature extraction and anomaly detection performed by both the SAE and NAE components of our resilience architecture are achieved through custom software that operates on VMs in real-time at the hypervisor level of the cloud node. Based on the monitoring and measurement tools described above, the collection of training data into a training dataset is achieved through the monitoring of a VM that has been created from a known-to-be-clean disk image. Each VM snapshot that is collected is stored in a single file that represents the normal behavior of that VM image. At 8 second intervals the Volatility tool is invoked with our custom plugin that crawls VM memory for every resident process structure. From each process we extract the following raw features per process:

- memory usage (i.e. actual size of the process in memory)
- peak memory usage (i.e. the requested memory allocation)
- number of threads

- number of handles (resources the process has open, e.g. files)

As mentioned, the raw features are per process, which is not useful if we are to consider each sample, or snapshot, as a single feature vector. Therefore a subsequent step is dedicated to building statistical meta-features such as the mean, variance and standard deviation of each feature across all processes. This results in a final feature vector for the snapshot of the form x = (x1; x2; : : : ; x_n 1; x_n), where n = 12 due to the three groups of four meta-features. At this stage, the snapshot feature vector is either appended to a file that represents the training dataset for normal operation, or is classified through online anomaly detection. At the network level the NAE gathers data through TCP dump, which separates packets into 8 second time bins. Features are then extracted using the CAIDA Coral Reef suite of tools, which provides the capability to generate statistics per uni-directional TCP and UDP flow. The raw features include:

- packets per address pair
- bytes per address pair
- flows per address pair

The raw features are then used to produce meta-features in a similar manner to the functionality of the SAE. The resulting feature vector therefore has dimension n = 9 and, in experiments where the NAE and SAE feature sets are combined into one, the resulting feature vector has dimensionality n = 21

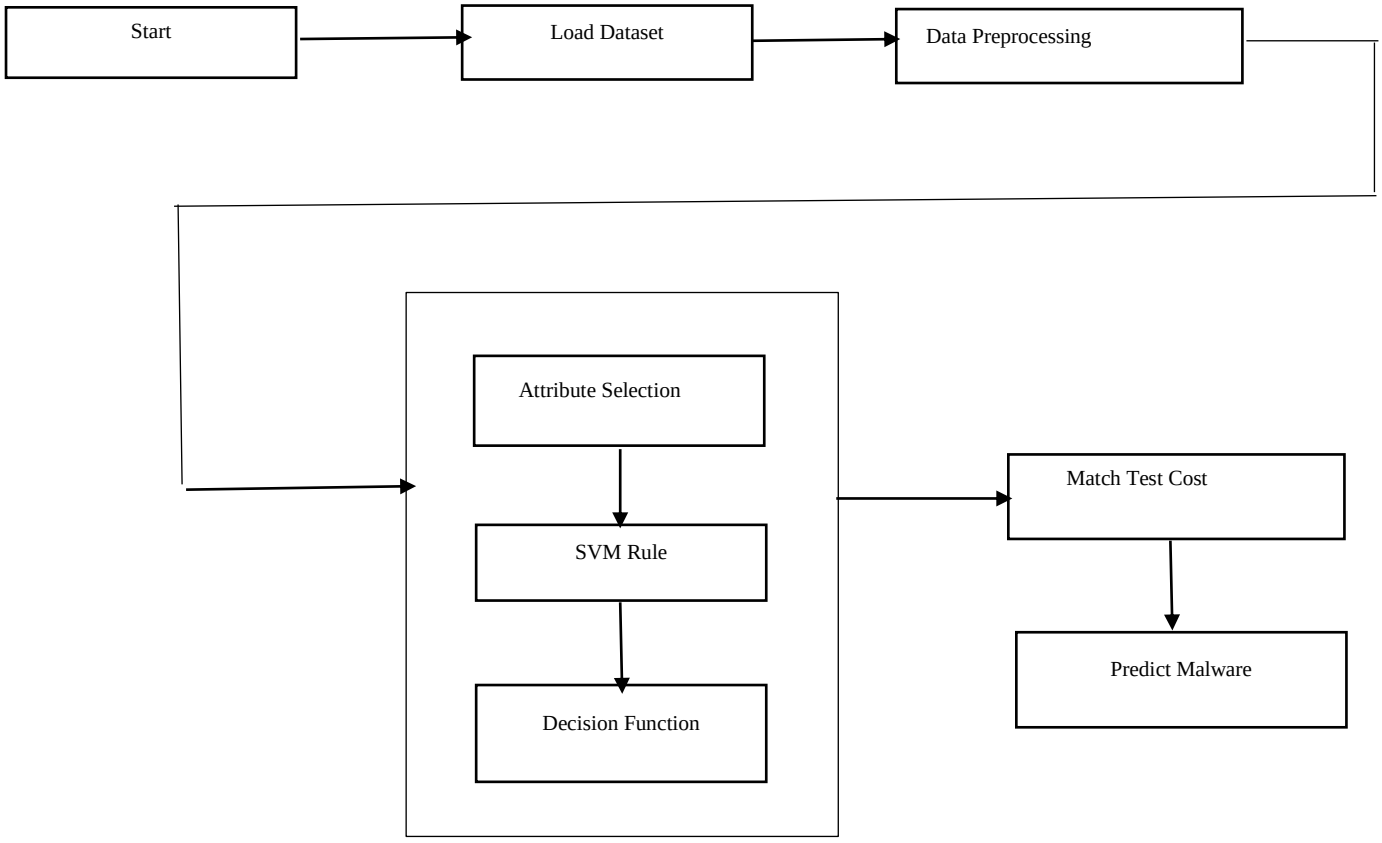

Fig 1 : System Architecture

4.2 One Class SVM

Bolster Vector Machines (SVMs) are regulated learning models that break down information and perceive designs, and that can be utilized for both order and relapse assignments. The SVM calculation is given a lot of preparing models marked as having a place with one of to classes. A SVM model depends on partitioning the preparation test focuses into independent classes by as wide a hole as could reasonably be expected, while punishing preparing tests the fall on an inappropriate side of the hole. The SVM model at that point makes forecasts by doling out focuses to the other side of the hole or the other. Now and then oversampling is utilized to repeat the current examples with the goal that you can make a two-class model, however it is difficult to foresee all the new examples of misrepresentation or framework flaws from restricted models can be costly. Therefore, in one class SVM, the support vector model is trained on data that has only one

class, which is the normal class. It infers the properties of normal cases and from these properties can predict which examples are unlike the normal examples.

After the above process we convert the data into contextual format which will display the fuzzy value for each data.

Fuzzy Value: The Fuzzy value class allows one to create a specific fuzzy concept for a given Fuzzy Variable, say temperature, For example, one might want to represent the concept temperature is very cold. Assuming that we have a Fuzzy Variable for the temperature with the term cold defined, we simply create the Fuzzy Value by specifying the temperature Fuzzy Variable and a linguistic expression. The expression is parsed and a Fuzzy set that mathematically defines the shape of this concept is created and stored with the Fuzzy Value. So a Fuzzy Value is an association of a Fuzzy Variable and a linguistic expression to describe a fuzzy concept.

The core of our online detection methodology within the SAE and NAE lies with the implementation of the supervised one-class SVM algorithm, which is an extension of traditional two-class SVM, and was proposed by Scholkopf et al. in [35]. In practice, the one-class SVM formulation handles cases using unlabeled data (i.e. novelty detection), the main goal of which is to produce a decision function that is able to return a class vector y given an input matrix x based on the distribution of a training dataset. The class y is a binary class where one outcome is the known class, which in our case is the normal VM behavior, and the other is the novel class, which represents any testing instances that are unknown to the classifier. If we let x = (x1; x2; : : : ; xn−1; xn) represent a feature vector, which contains all of the VM-related features described earlier (section 3.1), then the decision function f(x) takes the form:

$$f(x) = \sum_{i=1}^{N} \alpha \text{i k (x, xi)} - \rho$$

However, in order to achieve f(x) and attain the α_i multiplier over the kernel function k(x; x_i) it is firstly required to solve the optimization problem in Equation 2 using Lagrange multipliers, as follows:

$$\min_{\omega,\varepsilon,\rho} \frac{1}{2} \| \omega \|^2 + \frac{1}{\upsilon n} \sum_{i=1}^{n} \varepsilon_n - \rho$$

The parameter υ is extremely critical and characterizes the solution by setting an upper bound on the fraction of outliers, and a lower bound on the number of support vectors. Increasing υ results in a wider soft margin, meaning there is a higher probability that the training data will fall outside the normal frontier, thus identifying legitimate VM behavior as anomalous in our case. With reference to Equation 1, the function k(x; x_i) denotes the kernel function and can be chosen to suit a particular problem. In our implementation we employed the Radial Basis Function
(RBF) kernel function, which is defined as:

$$k(x, x_i) = \exp(-\gamma \| x - x_i \|^2)$$

The kernel parameter γ is sometimes expressed as $\frac{1}{\sigma^2}$ and a reduction in σresults in an decrease in the smoothness of the frontier between normal data and outliers. It is therefore possible to produce a decision function which approximates a nearest neighbour classifier by increasing the value of γ. As we explain next, both γ and υ parameters are quite critical and require some tuning in order to avoid miss classifications of abnormal behaviour to normal and vice versa.

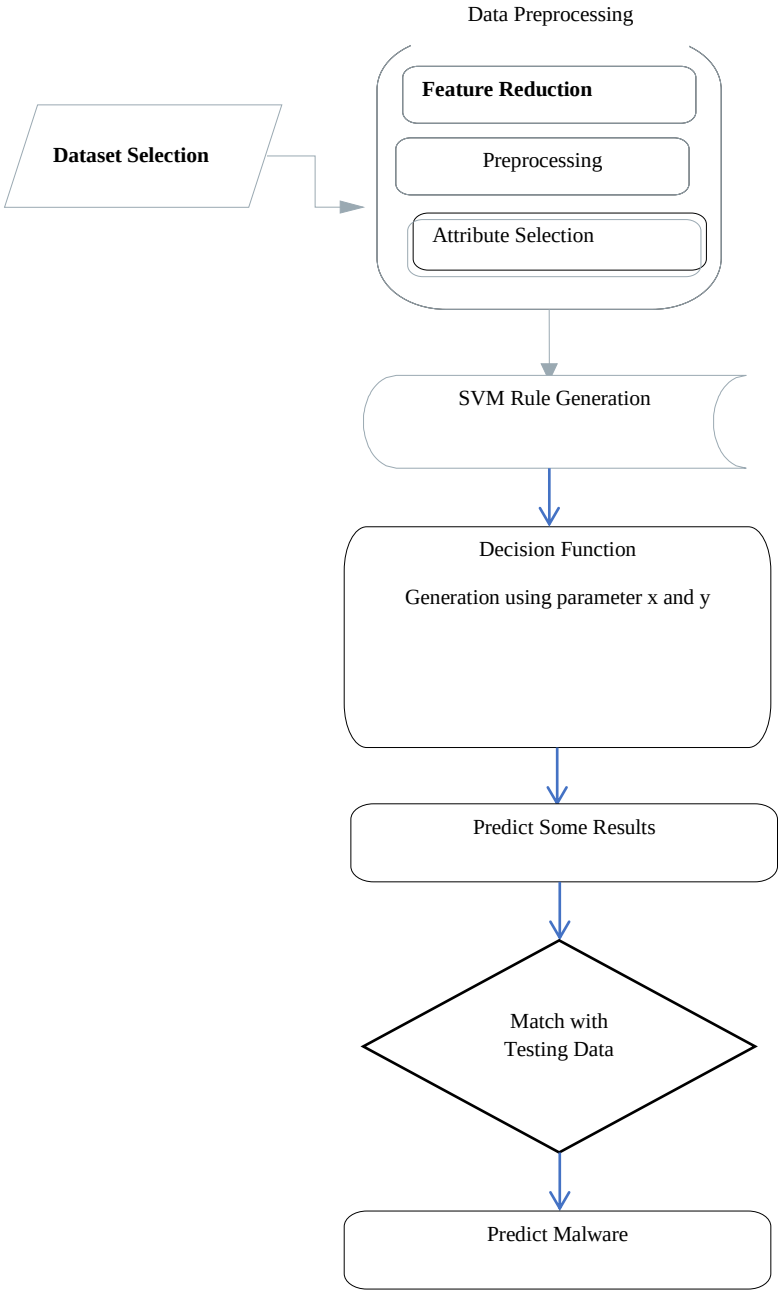

Fig 2: Flow Diagram of Dataset Selection

4.3 One Class SVM Tuning

Prior to the training process, the SAE & NAE engines automatically transform the initial gathered dataset by scaling them towards a Gaussian distribution. This is due to a requirement of the RBF kernel that the data be centered on zero and have unit variance. Thus the tuning process embedded in the SAE and NAE removes the mean from each feature and divides the feature vector by the standard deviation. The training process subsequently involves passing the scaled training dataset as an input to the one-class SVM algorithm, which produces a decision function that is able to classify new feature vectors. In general, the training process is determined by four factor: the size and content of the training dataset and the two parameters υ and γ. The training dataset size is determined by the length of time over which VM monitoring is conducted, after which it is possible to select subsets of the available data resulting in a refinement of training data and a reduction in dataset size if required. Dataset content is determined by the behavior of the processes in the VM and is not accurately controllable, hence the only influence that can be imposed on the data is by varying the applications and the loads on each of them. In contrast, the parameters υ and γ can be finely controlled and are chosen at training time to alter the accuracy of the classifier with respect to the available training data. The choice of algorithm parameters is not obvious

a priori and a small change of υ or γ either way can result in a less accurate detector. However, by choosing the parameters based on how accurately the classifier classifies its own training dataset it is possible to optimize the detector for a particular server profile. The process of parameter selection is conducted in an incremental manner by selecting the lowest reasonable values for υ and γ and incrementing the values of first υ and γ then in a pair of nested loops[12]. The increment for γ need not be as fine as υbecause, within our experimentation, we have found it to have much less influence on the accuracy of the detector. At each step the training data is reclassified using the new values of υ and γ and the False Positive Rate (FPR) is calculated for the pair of parameter values according to the formula in Equation 4. This search allows us to select the values that produce a minimum FPR. Overall, by conducting this iterative process we have found that once a minimum is reached there may be some parameter pairs that yield the same minimum, after which the FPR will rise again for all subsequent pairs of values. This is to be expected due to the fact that increasing both parameters past a certain point results in a frontier that fits too tightly to close neighbors in the training data and does not generalize well. Thus, a compromise needs to be reached between fitting the training data loosely with low values of the algorithm parameters, and being too restrictive with high values. Hence, with empirical experience of search times it is possible to stop the procedure long before the end of the exhaustive search and therefore reach an optimized set of parameters in reasonable time[13].

4.4 SAE & NAE Online Detection Process

As described in the previous subsections, the one-class SVM classifier within our SAE and NAE implementations is trained to identify anomalies by training it on a dataset of normal VM behavior. This is embodied in a dataset comprising features obtained during normal operation and is used to generate a decision function that is capable of classifying novel samples (i.e. anomalous behavior). Once trained, the classifier operates on feature vectors in an online capacity in order to produce a classification in real-time. The evaluation of the classifier within the SAE is conducted experimentally through the following procedure:

- A clean VM is created from a known-to-be-clean disk image.
- The VM is monitored for a period of 10 minutes in what we refer to as the “normal phase”
- Malware is injected and a further 10 minutes of monitoring follows in what we refer to as the “anomalous phase”

The output of the detector component is a vector y with an n dimension equal to the m dimension of the input matrix, which in the case of online detection yields a single value of y $\epsilon \{1, -1\}$; 1g for each snapshot vector x. This means it is possible to infer the success of the detector from its output relative to the phase in which the output was produced.

4.5 Classification Performance Matrix

The detection performance of the classifier can be assessed by determining the difference between the class it produces for a given input and the class it should produce. For example, if a sample of data contains no anomalies due to a malware strain, and the classifier produces an output of 1 for that data point, it is a correct classification. In order to quantify the classification performance.

$$\text{FPR} = \frac{FP}{FP+TN}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+EP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F Score} = 2\ \frac{Precision\ Recall}{Precision+Recall}$$

$$\text{G mean} = \rho \frac{}{Precision\ Recall}$$

Accuracy is the degree to which the detector classifies any newly tested data samples correctly whereas precision is a measure of how many of the positive classifications are correct, i.e. the probability that a detected anomaly has been correctly classified. The recall metric is a measure of the detector’s ability to correctly identify an anomaly, i.e. the probability that an anomalous sample will be correctly detected. The final two metrics are the

harmonic mean (F score) and geometric mean (G mean), rounded measure of the performance of a particular detector by accounting for all of the outcomes to some degree.

Once the Fuzzy value process is done, we are ready to play with the data. Here we have generated some rules using SVM for normal data and intruded data (means we have identified the rules at which data is corrupted, and which data is safe by using 3 parameters i.e.

- Parameter X
- Parameter Y
- Decision Function

These 3 Parameters are part of the SVM architecture. Once we press the extraction rule button, it displays all the rule. If we press the parameter 1 button i.e (X), the data is introduced in the dataset of our database, if we press the parameter i.e (Y) it will helps in inserting the data in the database. Then we introduce a detection function that is used by the SVM architecture to know the set of actual position of the data in the database. By prssinf the next button we will get two sets of data i.e

- Malware Data
- Normal Data

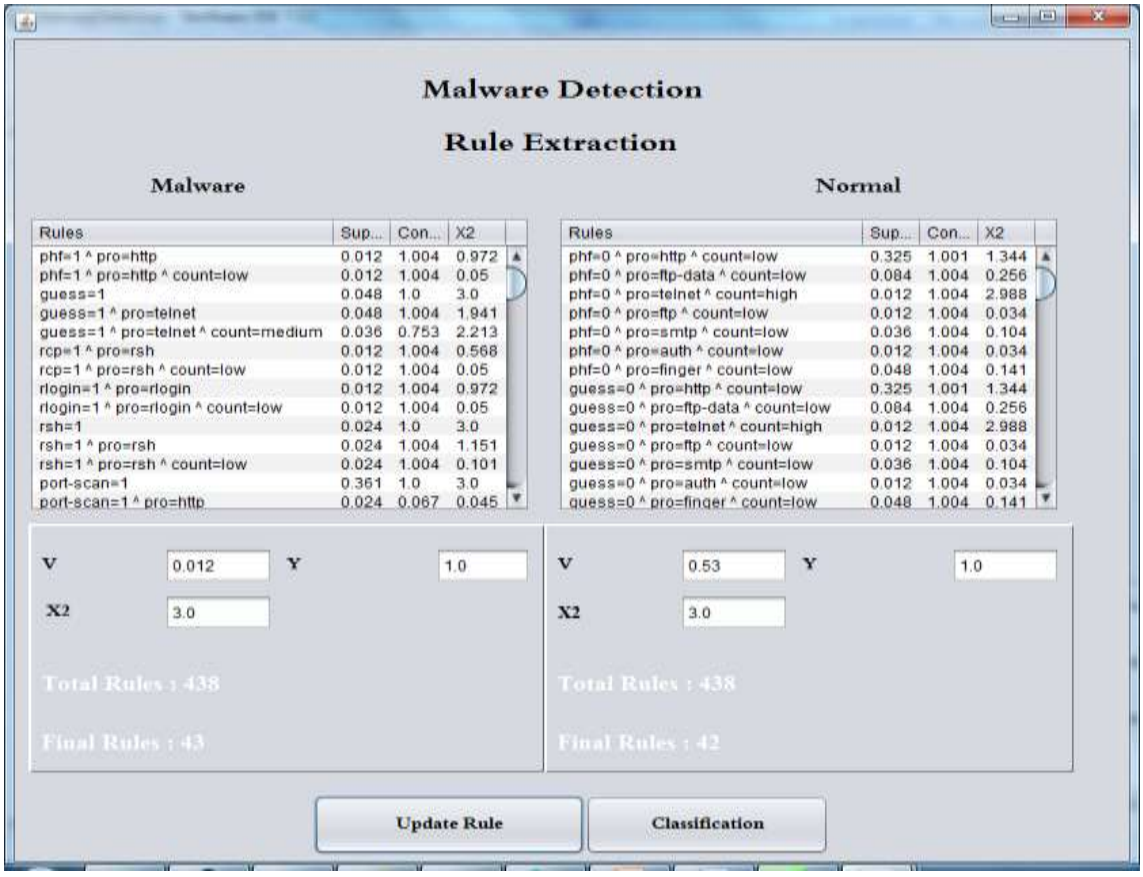

Fig. 3: Rule Extraction of Malware & Normal Data

Now we change the paramater 1 (X) , parameter 2 (Y) value and Decision function according to our choice for any row once we specify the values to know the extent of damage done on the set or not by using the specific parameters given.

Once it is done now we have to identify how much is the real value data & how much is the corrupted value data i.e why we have made a code to classify the level of intrusion to a data.

We now identify the level of intrusion with attack which will show “1” for attack data and “0” for non attack data. When we get this value we calculate the extraction level whether it can be cleaned or not using the classification.

After classification we get two set of data one where the elimination of malware can be done and other the attack which cannot be eliminated. Then according to SVM we see the priority of the data that the attack is genuine or not.

- Predict: In the predict button we have the figures from the normal algorithm of SVM.
- Evaluate: It shows our evaluation with the prototype data.

4.6 Malware Analysis On Static VMs

The major concern of any cloud provider is the VM screening the process of profiling the system and network features of a running VM and subsequently confirming that it is not infected with malware. The VM in our experimentation hosts a simple web server that provides an HTTP service to multiple client requests. The experiment lasted for 20 minutes, with malware injection.

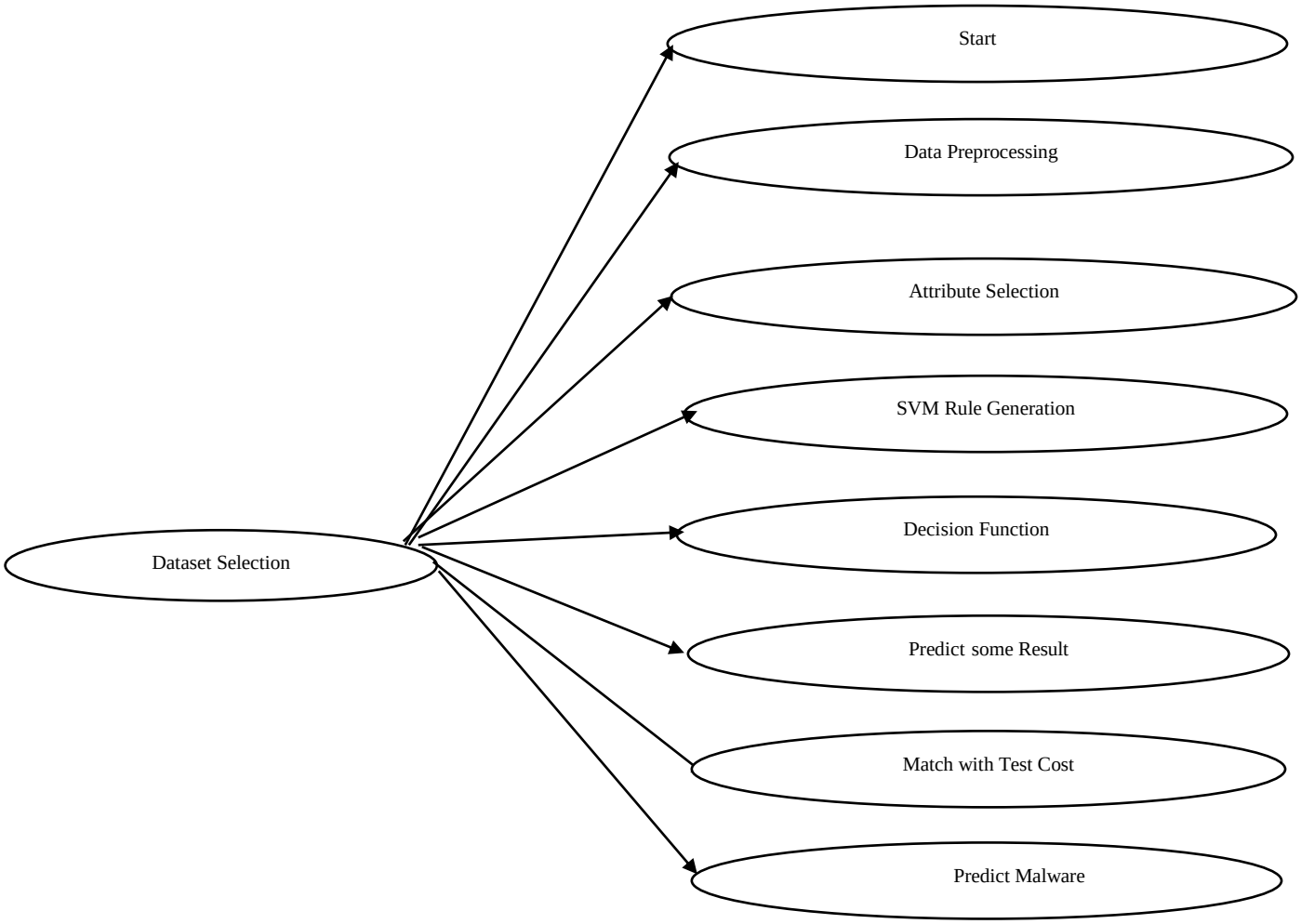

Fig 4: Use Case Diagram

4.7 Data Selection and Loading

Data selection is the process of selecting the appropriate data set for processing. Each of the record consists of 41 features and one marked as either normal or attack. The KDDCUP 99 Dataset is used for detecting the intrusion. All the data's are selected and loaded into the database for detecting the intrusion.

❖ **Data Preprocessing**

Pre-processing

The data is preprocessed to remove unwanted data that is presented in the dataset. The Incorrect data may provide the incorrect result so that all the data's are cleaned before processing. It is a process of cleaning the data for providing the better result.

❖ **Feature Selection**

Feature selection refers to the process of reducing the inputs for processing and analysis, or of finding the most meaningful inputs.

The meaningful data's are selected from the extracted features of KDD CUP 99 Dataset.

❖ **Classification**

Classification is a data mining function that assigns items in a collection to target categories or classes. A classification model could be used to identify the normal and attacks from the KDD Dataset. The goal of classification is to accurately predict the target class for each case in the data.

❖ **Prediction**

The goal of classification is to accurately predict the target class for each case in the data. The purpose of this module is to predict the attack from the KDD CUP 99 Dataset. Prediction in data mining is to identify data points purely on the description of another related data value. It is not necessarily related to future events but the used variables are unknown. It identify the closely related value. The attacks are predicted from the dataset. It increase the accuracy of the prediction result.

5. Results & Discussions

This model is acquainted with defeat every one of the hindrances that emerges in the current framework. It diminishes the data misfortune and the inclination of the induction because of the numerous evaluations.

To improve the exhibition of our proposed framework, we present the irregular woods arrangement calculation for expanding the characterization and forecast precision. This Algorithm will expand the exactness level up to 90%. Every one of the information's are viably ordered dependent on the distinguishing proof of typical and assaults.

❖ It keep away from Sparsity issues.

❖ Reduces the data Loss and the predisposition of the surmising because of the different appraisals.

❖ Increasing the characterization precision.

❖ Enhancing the exhibition of the expectation result

So as to demonstrate the figure in a diagram we have shown the substance of interruption information with the degree of extraction and non-interruption information as for extraction level.

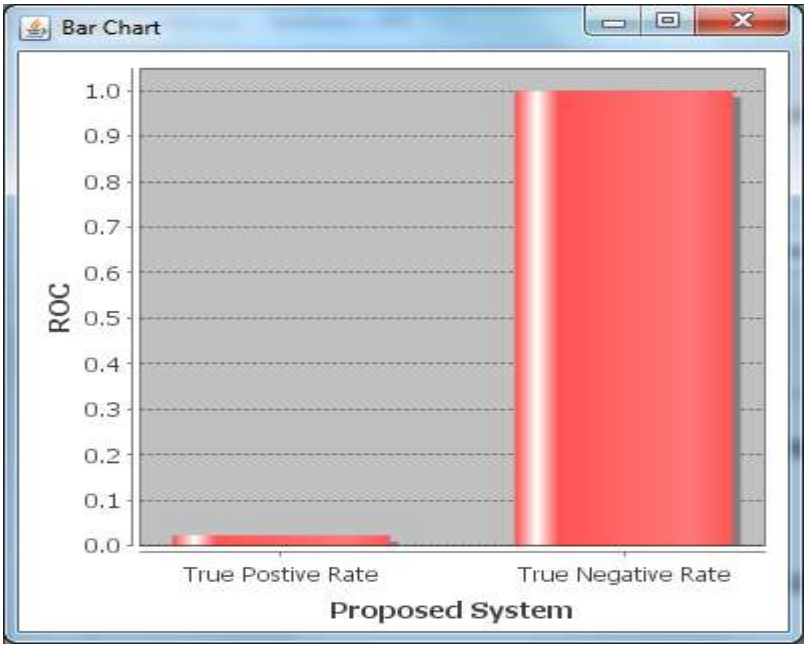

Chart 1: Bar Chart of True Positive Rate & True Negative Rate

6. CONCLUSION & SCOPE OF FURTHER WORK

In our proposed system, a secure, flexible and efficient data storage and retrieval system is designed based on the cloud computing techniques. The main challenges of this project we are detect the malware detection using malicious dataset. We are implementing the rule and decision function and match the data to detect the malware attack. The main purpose of this project to detect the malware attack. The whole summary is in case of cloud and there are three ways of using it.

Public cloud, Private cloud, Hybrid cloud. We have done the formulation with public cloud. Hence, the method is embodied by architecture that was initially defined in [9], and which comprises the System Analysis Engine (SAE) and Network Analysis Engine (NAE) components. Overall, this work performs online anomaly detection under two pragmatic cloud scenarios, based on suggestions by cloud operators, which emulate "static" detection as well as detection under the scenario of VM "live" migration. Hence, we have demonstrated that the extracted features for classifier training were appropriate for our purposes and aided towards the detection of the investigated anomalies under minimal time cost throughout the training and testing phase. Nonetheless, in order to further the investigation, this feature set can easily be expanded to include statistics derived from vCPU usage and a deeper introspection of process handles, which could be beneficial for the detection of highly stealthy malware. The results derived from the experiments based on network-level detection of DoS attacks have also justified that the network features used were sufficient for the detection of such challenges, since the detection accuracy rate also reached well above 90%. The future work of this project, we will analyze using this project. In future, we will implement the large amount of the data. It is helpful for the analyze person. In future we can store all the data in a Hadoop storage for increasing the processing speed.

7. REFERENCES

[1] Mahmoud Abdelsalam, Ram Krishnan and Ravi Sandhu, "Clustering-Based IaaS Cloud Monitoring", IEEE 10th International Conference on Cloud Computing", 2017

[2] Mahmoud Abdelsalam, Ram Krishnan, Yufei Huang and Ravi Sandhu, " Malware Detection in Cloud Infrastructures using Convolutional Neural Networks", IEEE 11th International Conference on Cloud Computing", 2018.

[3] N. Moses Babu and G. Murali, "Malware Detection for Multi Cloud Servers using Intermediate Monitoring Server", International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017)

[4] Bhavesh Borisaniya, Kevin Patel and Dr. Dhiren Patel, "Evaluation of Applicability of Modified Vector Space Representation for in-VM Malicious Activity Detection in Cloud", 2014 Annual IEEE India Conference (INDICON).

[5] Chia Tien Dan Lo, Odronez Pablo and Cepeda Carlos, "Feature Selection and Improving Classification Performance for Malware Detection", 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)

[6] H. S. Pannu, J. Liu, and S. Fu, "Aad: Adaptive anomaly detection system for cloud computing infrastructures," Reliable Distributed Systems, IEEE Symposium on, vol. 0, pp. 396–397, 2012.

[7] A. Marnerides, S. Malinowski, R. Morla, and H. Kim, "Fault diagnosis in fDSLg networks using support vector machines," Computer Communications, no. 0, pp.–, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366415000080

[8] W.-H. Chen, S.-H. Hsu, and H.-P. Shen, "Application of svm and ann for intrusion detection," Computers & Operations Research, vol. 32, no. 10, pp. 2617–2634, 2005.

[9] Y. Tang, Y.-Q. Zhang, N. Chawla, and S. Krasser, "Svms modeling for highly imbalanced classification," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 39, no. 1, pp. 281–288, Feb 2009.

[10] A. Marnerides, M. Watson, N. Shirazi, A. Mauthe, and D. Hutchison, "Malware analysis in cloud computing: Network and system characteristics," in GlobecomWorkshops (GC Wkshps), 2013 IEEE, Dec 2013, pp. 482–487.

[11] N.-U.-H. Shirazi, S. Simpson, A. Marnerides, M. Watson, A. Mauthe, and D. Hutchison, "Assessing the impact of intra-cloud live migration on anomaly detection," in Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on, Oct 2014, pp. 52–57.

[12] Michael R. Watson, Noor-ul-hassan Shirazi, Angelos K. Marnerides, Andreas Mauthe and David Hutchison "Malware Detection in Cloud Computing Infrastrucrures" IEEE Transactions on Dependable and Secure Computing 13 (2), 192-205, 2015.