

# Decision Tree based Learning Approach for Identification of Operating System Processes

AMIT KUMAR, SHISHIR KUMAR

Department of Computer Science and Engineering  
Jaypee University of Engineering and Technology

A.B. Road, Raghogarh, Guna

INDIA

amitrathi10@yahoo.co.in <http://www.juet.ac.in/Department/faculty.php?id=42483894&dep=cse>,  
dr.shishir@yahoo.com <http://www.juet.ac.in/Department/faculty.php?id=45652778&dep=cse>

*Abstract:* - In present scenario various tools like firewalls, anti-virus tool, network security tools, malware removal tools, monitoring tools etc, are being used for providing security to computer systems. Computer security tools available in present era need to be updated and monitored regularly. If any computer users do not regularly update the security tools, such systems will be vulnerable to virus and other attacks. Through this paper a learning system is being proposed to identify the operating system processes as Self and Non-Self, using the concepts of Decision Tree Learning. ID3 algorithm has been used to construct a Decision Tree after calculating the Entropy and Information Gain. Initially Decision Trees are generated using training examples and then these constructed Decision Trees are tested with test data. Further, it has been inferred through experimental results that the Decision Tree Learning approach will provide better security through effective identification of Self and Non-Self processes.

*Key-words:* - Self and Non Self Process, Process-Parameters, Decision Tree, ID3 (Iterative Dichotomiser 3), Entropy, Information Gain.

## 1 Introduction

In the present era of computer information security, Cyber Security and Computer Security are vital issue [4]. For providing the highest possible extent of computer security, implementation of an efficient and secure operating system is a necessity [5]. Some operating system developers provide a secure operating system and security tools which works to identify the unauthorized access of the system. Lots of hardware and software based security tools are made available by various vendors as Computer Security Tool [6].

Many operating systems and computer security tools cannot provide the maximal level of computer security due to its design constraints. For providing the maximum security for a computer system major change in the design of an operating system is required. Through this paper a methodology will be proposed for providing the maximum security by identification of Self and Non-Self process [1,2,3] using concepts of the Decision Tree and machine

learning [7,9,10]. The operating system processes can be categorized into two parts, Self and Non-Self. Self are those processes which is not harmful to the system like system process, Microsoft application's processes etc. The processes which are generated by viruses, worms etc can be classified as Non-Self. Main Objective of this paper is to identify these Non-Self processes.

Machine learning is a fast growing field of Artificial Intelligence and Computer Science. Tom M. Mitchell [7] has provided a widely quoted, formal definition: "A computer program is said to learn from experience 'E' with respect to any class of tasks 'T' and performance measure 'P', if its performance of tasks in 'T', as measured by 'P', improves with experience E". Machine learning deals with the development of such computer programs which automatically improves their performance and gain experience.

There are various learning concepts which can be used to provide security like Concept Learning, Decision Tree Learning, Learning through ANN,

Bayesian Learning, Instance-Based Learning, Genetic Algorithm, and Analytical Learning etc [7]. Through this paper a methodology is being proposed in which Decision Tree Learning will be used to provide the security of a computer system.

In a Decision Tree Learning the learned target function is specified by Decision Tree which provides learning, decision through root to a leaf node. Decision Tree Learning [14] method is used for approximation of discrete-valued target function. ID3, ASSISTANT and C4.5 algorithms [12, 13] are used to provide learning in the Decision Tree. Decision Tree categorizes the examples of sorting from top root node to bottom leaf node. Each node in the tree is a test of the attributes of the example. In the proposed approach, the ID3 algorithm has been used along with the concept of Information Gain and Entropy.

## 2 Proposed Methodology

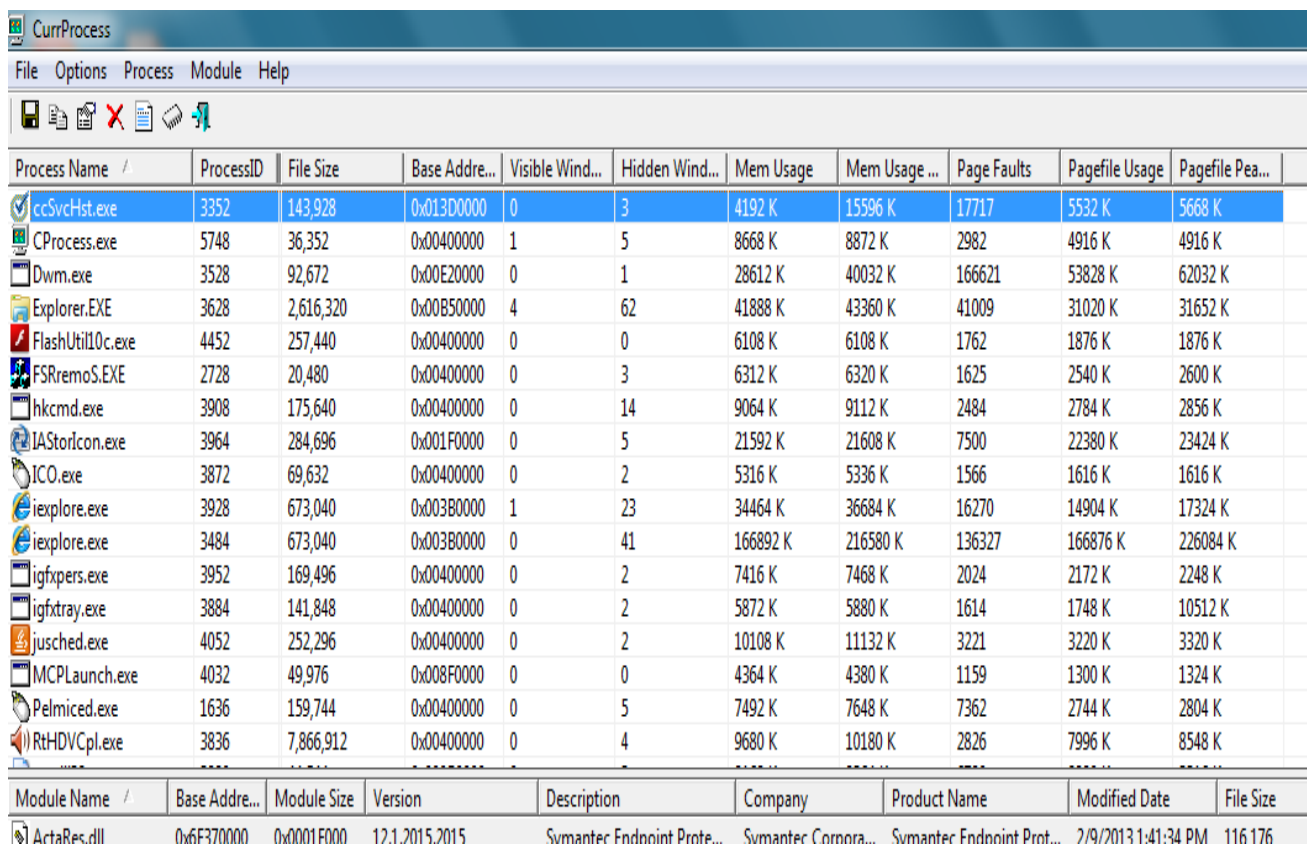
A process is the basic unit of execution in an operating system. During the execution of a program the Operating System generates many processes to complete the task. If a computer system is affected by

worms and virus or has been attacked, then operating system also generates its processes. Proposed approach works on the processes and its parameters to identify the process generated by viruses or attacks. These processes will be identified as Non-Self by using the concepts of Decision Tree Learning.

A process has many attributes like ProcessID, Priority, Product name, Version, Description, Company, Window Title, File size, File Created Date, File Modified Date, File Name, Base Address, Created On, Visible Windows, Hidden Windows, User Name, Memory Usage, Memory Usage Peak, Page Faults, Pagefile Usage, Pagefile Peak Usage and File Attributes etc.

Initially the parameters of a process which has NOT NULL values will be identified. By using the Currprocess tool [8] initially five process attributes Process ID, File size, Memory Peak Usage, Page Faults and Page File Peak Usage will be used. More attributes can be added to get better security.

Figure 1 shows the screen shoot of CurrProcess tool window.



Process Name /	ProcessID	File Size	Base Addr...	Visible Wind...	Hidden Wind...	Mem Usage	Mem Usage ...	Page Faults	Pagefile Usage	Pagefile Pea...
ccSvcHst.exe	3352	143,928	0x013D0000	0	3	4192 K	15596 K	17717	5532 K	5668 K
CProcess.exe	5748	36,352	0x00400000	1	5	8668 K	8872 K	2982	4916 K	4916 K
Dwm.exe	3528	92,672	0x00E20000	0	1	28612 K	40032 K	166621	53828 K	62032 K
Explorer.EXE	3628	2,616,320	0x00B50000	4	62	41888 K	43360 K	41009	31020 K	31652 K
FlashUtil10c.exe	4452	257,440	0x00400000	0	0	6108 K	6108 K	1762	1876 K	1876 K
FSRremoS.EXE	2728	20,480	0x00400000	0	3	6312 K	6320 K	1625	2540 K	2600 K
hkcmd.exe	3908	175,640	0x00400000	0	14	9064 K	9112 K	2484	2784 K	2856 K
IAStorIcon.exe	3964	284,696	0x001F0000	0	5	21592 K	21608 K	7500	22380 K	23424 K
ICO.exe	3872	69,632	0x00400000	0	2	5316 K	5336 K	1566	1616 K	1616 K
ieexplore.exe	3928	673,040	0x003B0000	1	23	34464 K	36684 K	16270	14904 K	17324 K
ieexplore.exe	3484	673,040	0x003B0000	0	41	166892 K	216580 K	136327	166876 K	226084 K
igfxpers.exe	3952	169,496	0x00400000	0	2	7416 K	7468 K	2024	2172 K	2248 K
igfxtray.exe	3884	141,848	0x00400000	0	2	5872 K	5880 K	1614	1748 K	10512 K
jusched.exe	4052	252,296	0x00400000	0	2	10108 K	11132 K	3221	3220 K	3320 K
MCPLaunch.exe	4032	49,976	0x008F0000	0	0	4364 K	4380 K	1159	1300 K	1324 K
Pelmedic.exe	1636	159,744	0x00400000	0	5	7492 K	7648 K	7362	2744 K	2804 K
RtHDVCpl.exe	3836	7,866,912	0x00400000	0	4	9680 K	10180 K	2826	7996 K	8548 K

Module Name /	Base Addr...	Module Size	Version	Description	Company	Product Name	Modified Date	File Size
ActaRec.dll	0x6F370000	0x0001F000	12.1.2015.2015	Symantec Endpoint Prote...	Symantec Corpra...	Symantec Endpoint Prot...	2/9/2013 1:41:34 PM	116 176

Fig.1: Screenshot of CurrProcess tool

## 3 Range of the parameters

By using the Currprocess tool process parameter's range has been identified as shown in Table 1. The process parameters which have their minimum and

maximum range as shown in Table 1. The range of these parameters is according to their measure unit. (i.e. Bytes, K Bytes, a natural number, etc).

Table 1: Process Parameters and its minimum and maximum range.

Parameter	Range Min - Max
Process ID	000-9999
File Size	00000 – 9999999 (Bytes)
Base Address	0x00000000 – 0x99900000
Hidden Windows	0 -999
Memory Usage	000 – 999999 (K)
Memory Peak Usage	000 – 999999 (K)
Page Faults	0000 – 9999999
Page File Usage	000 – 999999 (K)
Page File Peak Usage	000 – 999999 (K)

### 3.1 Range for Learning

Initially for the proposed approach, five parameters of processes are used to identify the Self and Non-Self processes. Better security can be achieved by using more parameters and dividing parameter ranges into small parts. After analyzing various processes initially, five parameters Process ID (divided into three ranges, low, medium and high), File Size (divided into three ranges, low, medium and high), Memory Peak Usage (divided into five ranges very low, low, medium, high and very high), Page Fault (divided into five ranges very low, low, medium, high and very high), Page File Peak Usage divided into three ranges low, medium and high. These ranges have been divided into different parts as mentioned below-

Process ID /DID range has been divided into three parts -

Low	– 1938 and Below
Medium	- 1939 to 3163
High	- 3164 and Above

File Size /DFS range has been divided into three parts -

Low	– 314688 and Below
Medium	- 314689 to 4375625
High	- 4375626 and Above

Memory Peak Usage / DMPU range has been divided into five parts -

Very Low	- 10490 and below
Low	– 10491 to 31302
Medium	- 31303 to 78172
High	- 78173 to 109391
Very High	- 109392 and above

Page Faults /DPF range has been divided into five parts -

Very Low	- 2274 and below
----------	------------------

Low	– 2275 to 5358
Medium	- 5359 to 25001
High	- 25002 to 43750
Very High	- 43751 and above

Page File Peak Usage /DPFPU range has been divided into three parts -

Low	– 5008 and below
Medium	- 5009 to 31269
High	- 31270 and above

The range of these process parameters can be changed according to the system architecture & organization along with operating system running on the computer system.

## 4 Training Examples

For application of Decision Tree Learning approach of Machine Learning a set of training examples has been used. This set has both positive and negative examples as shown in Table 2. There are 14 training examples in which 9 are positive and 5 are negative. The values of parameters are converted as per the above section into Very Low (VL), Low (L), Medium (M), High (H) and very High (VH) to make the easy calculation and understanding. The abbreviations VL, L, M, H and VH are used instead of actual values. These training examples are taken after various running conditions on various workloads of a Computer System. The system was virus infected during these observations. Fourteen different processes are identified as training set. In these 14 training examples, nine examples are positive examples and treated as Self processes (system and some application processes) and five examples are negative examples and treated as Non- Self processes (generated by viruses and worms).

Table 2: Training Examples

Process	Process ID / DID	File Size / DFS	Memory Peak Usage /DMPU	Page Faults /DPF	Page File Peak Usage /DPFPU	SELF
P1	L	L	VL	L	M	Yes
P2	L	M	VL	L	M	Yes
P3	M	L	VL	VL	L	No
P4	L	M	M	L	H	Yes
P5	H	H	M	H	H	No
P6	H	L	L	M	M	Yes
P7	H	H	L	M	M	Yes
P8	L	M	L	M	M	Yes
P9	H	M	M	L	M	No
P10	M	M	M	H	H	Yes
P11	L	L	VL	L	L	No
P12	M	M	M	M	H	Yes
P13	H	L	M	H	H	Yes
P14	H	M	VH	VH	H	No

## 5 Decision Tree Learning

Decision Tree Learning is a technique for reminiscent of discrete-valued target functions, in which the learned function, has been represented by a Decision Tree. In general, Decision Tree characterize a disjunction and conjunction of constraints on the attribute values of instances. Algorithms such as ID3, ASSISTANT and C4.5 are generally used in Decision Tree Learning [12, 13].

Decision Tree categorizes instances by sorting them down the tree from the root to any leaf node, which provides the categorization of the instances. Each and every node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by sorting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch consequent to the value of the attribute in the given example. This method is then repeated for the sub-tree rooted at the new node.

For providing the learning to the proposed approach, using the Decision Tree Method ID3 algorithm has been used. The vital choice in the ID3 algorithm is selection of attribute to test at each node in the tree. The attribute which is most helpful in classifying examples will be selected. For construct a Decision Tree a statistical property called Information Gain [7,15] and Entropy [7] is used to

classify the attributes. By using these concepts, it becomes easy to select the root node and the nodes comes under the root node. ID3 uses this Information Gain for selection among the candidate attributes at each step while mounting the tree.

### 5.1 Information Gain and Entropy

Information Gain is clearly related to a measure commonly used in information theory, called Entropy [7], that characterizes the purity (and impurity) of a random collection of examples. Given a collection S, having some positive and some negative examples of some target perception, the Entropy S relative to this Boolean classification is-

$$\text{Entropy}(S) = -(P_+ \log_2 P_+) - (P_- \log_2 P_-) \dots \dots \dots (1)$$

Where  $P_+$  is the proportion of positive examples in S and  $P_-$  is the proportion of negative examples in a given collection S. In all calculation involving Entropy  $0 \log 0$  has been assumed as 0.

As in Table 2 there are a collection of 14 examples of processes of operating system. These examples are two types, Self or Non-Self. So these examples are satisfied the Boolean concepts. Boolean concept which have 9 positive and 5 negative examples (adopt the notation [9+, 5-]). Then the Entropy of S (given examples in Table 2) relative to this Boolean classification by using equation (1) is:

$$\begin{aligned} \text{Entropy}([9+, 5-]) &= -(9/14) \log_2 (9/14) - \\ &\quad (5/14) \log_2 (5/14) \\ &= 0.940 \end{aligned}$$

Entropy is 0 if all the members of given collection S belong to same class i.e. having all positive or all negative examples. Entropy is 1 when the collection

contains the equal number of positive and negative examples. Figure 2 shows the how the Entropy function varies between 0 and 1.

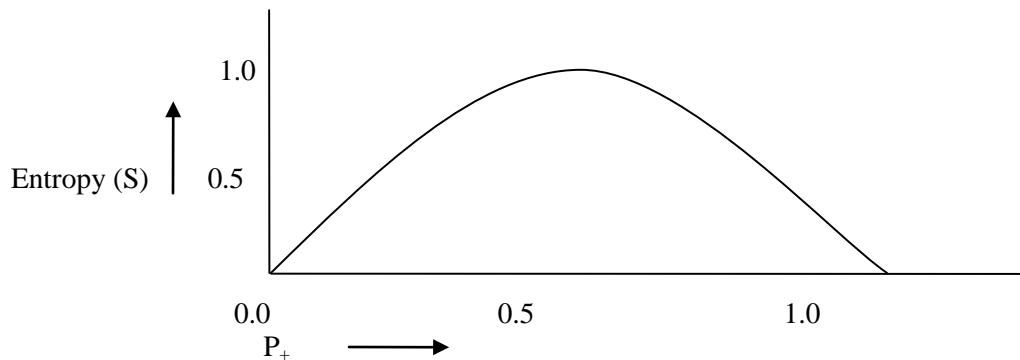


Fig.2: The Entropy function relative to a Boolean classification, as the proportion, P<sub>+</sub>, of positive example varies between 0 and 1.

Given Entropy computes the impurity in a collection of training examples. Further, the usefulness of an attribute in classifying the training data called Information Gain has been calculated. Information Gain is just the expected decrease in Entropy caused by partitioning the example according to this attribute. The Information Gain “Gain (S, A)” of an attribute A, relative to a collection of example S, has been defined as-

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \dots \dots \dots (2)$$

Where Values (A) are the set of all possible values for attribute A, and S<sub>v</sub> is the subset of S for which attribute A has value v (i.e. S<sub>v</sub> = { s ∈ S | A(s) = v }). Gain (S, A) is the expected reduction in Entropy caused by knowing the value of attribute A.

For the given training example (S) there is an attribute Process ID (DID) which have the values Low (L), Medium (M) and High (H). For the 14 examples [9+,5-] given in Table 2, five Process Id values are Low (L), three are Medium (M) and six are High (H).

For Low(L) value of Process ID there are 4 positive and 1 negative examples.

For Medium(M) value of Process ID there are 2 positive and 1 negative examples.

For High(H) value of Process ID there are 3 positive and three negative.

The Information Gain due to sorting the original 14 examples by the attribute Process ID-

Values (Process ID) = Low, Medium, High

- S = [9+,5-]
- S<sub>low</sub> = [4+,1-]
- S<sub>Medium</sub> = [2+,1-]
- S<sub>High</sub> = [3+,3-]

Using the equation (2) the Information Gain has been calculated by the Process ID as:

$$\begin{aligned} \text{Gain}(S, \text{Process ID}) &= \text{Entropy}(S) - (5/14) \text{Entropy}(S_{\text{low}}) - (3/14) \\ &\text{Entropy}(S_{\text{Medium}}) - (6/14) \text{Entropy}(S_{\text{High}}) \\ &= 0.940 - (5/14) * 0.721 - (3/14) * 0.918 - (6/14) * 1 \\ &= .059 \end{aligned}$$

Using the equation (2) the Information Gain by the other parameters has also been calculated and valued are mentioned below:

- Gain (S, Process ID) = 0.059
- Gain (S, File Size) = 0.021
- Gain (S, MPU) = 0.263
- Gain (S, Page Faults) = 0.401
- Gain (S, PFPU) = 0.272

It is clear from the above values of the Information Gain that the “Page Fault” attribute has the maximum value, so it provides the greatest prediction of the target learn function. So “Page Fault” is chosen as the decision attribute of the root node To build a Decision Tree following ID3 algorithm [7] has been used:

**5.2 ID3 Algorithm**

ID3 (*Examples, Target\_attribute, Attributes*)

[*Examples* are the training examples. *Traget\_attribute* is the attribute whose value is to be predicated by the tree. *Attributes* are a listing of attributes that may be tested by the learned Decision Tree. Returns a Decision Tree that correctly classifies the given examples.]

- Create a ROOT node in the Decision Tree.
- If all the *Examples* are positive, Return the single-node tree ROOT, with label = +

- If all the *Examples* are negative, Return the single-node tree ROOT, with label = -
- If *Attributes* are empty, Return the single-node tree ROOT, with label = most common value of *Target\_Attribute* in *Examples*
- Otherwise Begin
  - $A \leftarrow$  the attribute from *Attributes* that best\* classifies *Examples*.
  - The decisionattribute for ROOT  $\leftarrow A$
  - For each possible value,  $V_i$  of  $A$ .
    - » Add a new tree branch below ROOT, corresponding to the test  $A = V_i$
    - » Let *Examples\_Vi* be the subset of *Examples* that have value  $V_i$  for  $A$

- » If *Example\_Vi* is empty
  - Then, under this new branch add a leaf node with label = most common value of *Target\_Attribute* in *Examples*
  - Else below this new branch add the subtree.

ID3(*Examples*, *Target\_attribute*, *Attributes* - { $A$ })

- End
- Return Tree

On the basis of above algorithm a Decision Tree has been constructed.

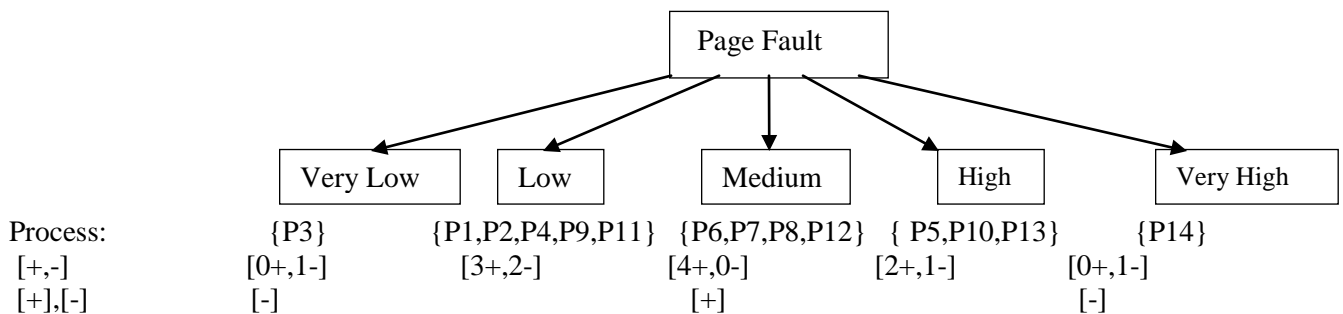


Fig.3: Partial Decision Tree after Applying ID3 on the training example given in Table 2.

### 5.3 Construction of the Decision Tree

To construct the Decision Tree, Information Gain calculated in section 5.1 is used. Page Fault has the maximum value of Information Gain so “Page Fault” attribute becomes the root node as shown in Figure 3. The “Page Fault” parameter has been divided into five parts as very low, low, medium, high and very high. These are becomes the branches of root node.

In Table 2 there is only one training example P3 which has the ‘very low’ value of Page Fault. As according to the ID3 algorithm if any node has only positive (or all negative) examples then this node will become the leaf node. This shows that if any process, has very low “Page Fault” this process may be non-Self process (shown as [-] as shown in Figure 3). In Table 2 there are five training example P1,P2,P4,P9 and P11 which has the ‘low’ value of Page Fault. P1, P2 and P4 are positive example and P9 and P11 are negative example. As according to the ID3 algorithm if any node has both positive and negative examples then for this node again apply the

ID3 on P1,P2,P4,P9 and P11 with remaining parameters.

Now, the training examples P6, P7, P8, P12 of Table 2 has the ‘medium’ value of “Page Fault”. It has been observed these four examples are positive example. This shows that if the process has the medium “Page Fault” it may be a Self process. Now ensuring the training example of Table 2 for the high value of “Page Fault” 2 positive (P10, P13) and 1 negative (P5) example has been obtained. By evaluating the training example of Table 2 for the very high value of “Page Fault” 1 negative (P14) example has been obtained.

In Figure 3 very low (Non-Self), medium (Self) and very high (Non-Self) becomes the leaf node of the Decision Tree. Other low and high value will have a sub-tree. Now the Information Gain of the remaining parameters Process Id, File Size, Memory Peak Usage and Page File Peak Usage of training examples of P1, P2, P4, P9, P11 for low values of “Page Fault” and P5, P10, P13 for high values of “Page Fault” have been calculated using equation 2.

It has been observed that “Page File Peak Usage” will become the node of the low value of “Page Fault” and “File Size” will become the node of the high value of “Page Fault” as shown in Figure 4.

Page File Peak Usage has three possible values, Low, Medium and High. According to the training example of Table 2, P11 has the Low value of Page File Peak Usage and it is a negative example (Non-Self). P4 has the High value of Page File Peak Usage

and it is a negative example (Non-Self), P1, P2, P9 [2+, 1-] has the Medium value of Page File Peak Usage.

File Size has three possible values Low, Medium and High from the training examples of Table 2, P13 has the Low value of File Size and it is a positive example (Self), P10 has the Medium value of File Size and it is a positive example (Self), P5 has the High value of File Size positive example (Self).

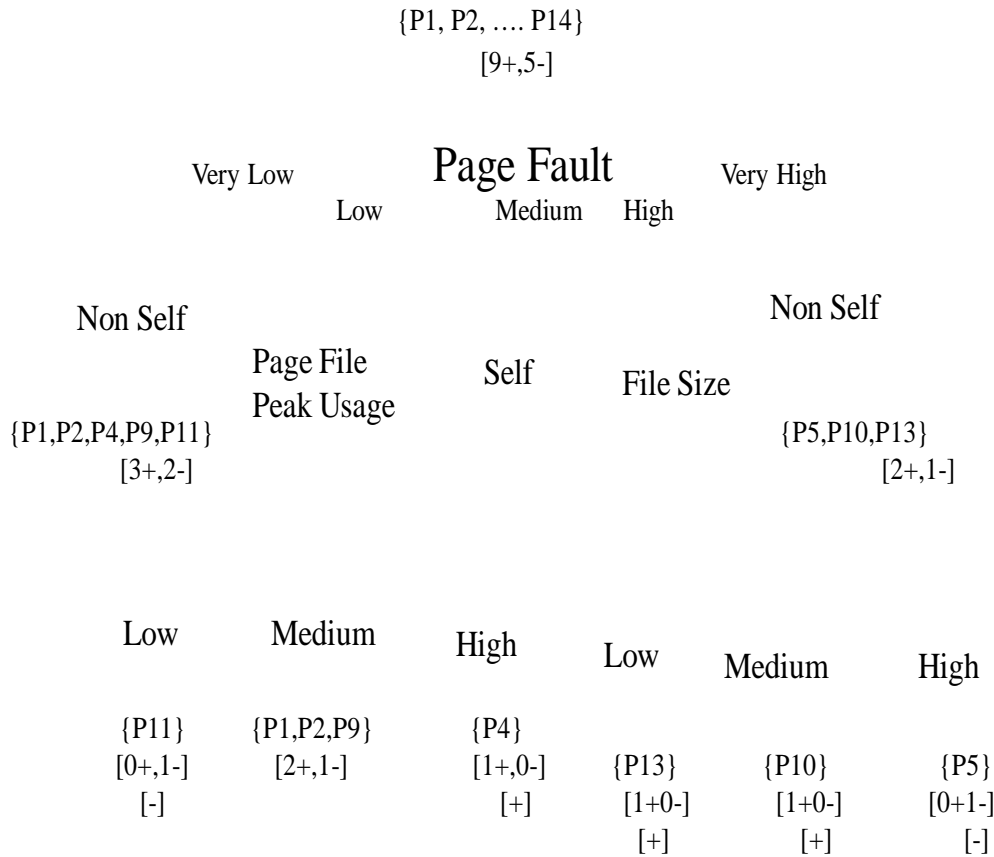


Fig.4: After Applying ID3 on the training example P1, P2, P4, P9, P11 for low value and P5, P10, P13 for high value on training examples of Table 2.

After the execution of ID3 Algorithm and performing all the calculations, the final Decision Tree has been generated on the basics of the training example of Table 2.

Figure 5 show the final decision tree constructed by implementing the ID3 Algorithm on the training examples of Table 2. Different Decision Trees are constructed by using different training examples. Between these different Decision Trees, select one which will give better results on test data.

The final Decision Tree shown in Figure 5 has been tested with test data of Table 3. Table 3 shows the test result also in the last column. Final Decision Tree of Figure 5 identifies all the Self process correctly, but processes P12 is identified as a Self (incorrectly).

The final Decision Tree of Figure 5 now is tested with test data of Table 4. Table 4 shows the test result also in the last column. Final Decision Tree of Figure 5 identifies process P3 and P14 incorrectly.

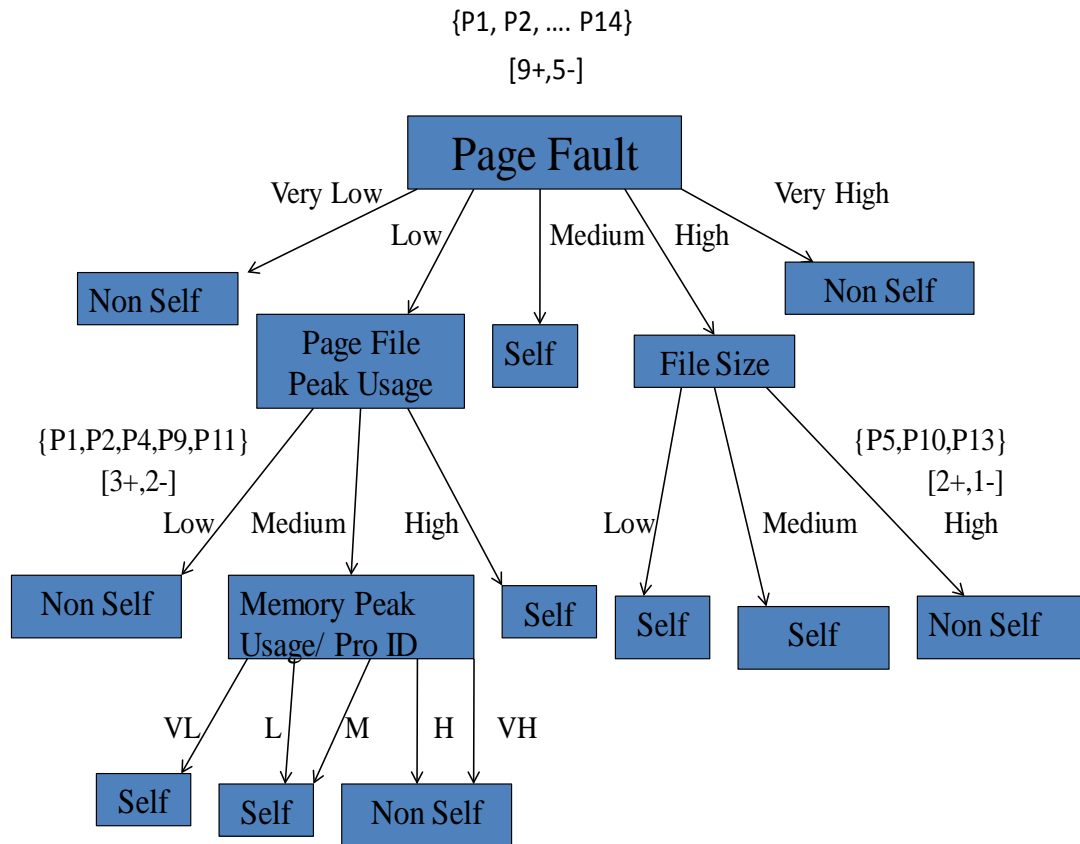


Fig.5: Final Decision Tree generated by ID3 Algorithm on training data of Table 2

Table 3: Test data and result by final Decision Tree of Figure 3.

	Process ID	File Size	Mem Usage Peak	Page Faults	Pagefile Peak Usage	Self/Non-Self	Identified As
P1	H	M	M	M	M	Yes	Self
P2	M	M	L	M	M	Yes	Self
P3	M	H	M	M	M	Yes	Self
P4	M	M	M	M	M	Yes	Self
P5	M	H	M	M	M	Yes	Self
P6	H	H	L	M	M	Yes	Self
P7	H	M	L	M	M	Yes	Self
P8	H	M	VL	L	M	Yes	Self
P9	H	H	M	M	H	Yes	Self
P10	M	M	VL	VL	L	No	Non-Self
P11	M	H	VH	L	M	No	Non-Self
P12	L	L	L	M	M	No	Self



Table 4: Test data and result of final Decision Tree of Figure 3.

	Process ID	File Size	Mem Usage Peak	Page Faults	Pagefile Peak Usage	Self	Identified As
P1	L	M	M	M	H	Yes	Self
P2	H	M	VL	L	M	Yes	Self
P3	M	L	VL	VL	L	Yes	Non-Self
P4	M	M	M	M	M	Yes	Self
P5	M	M	L	M	M	Yes	Self
P6	M	L	H	H	H	Yes	Self
P7	L	M	L	L	M	Yes	Self
P8	H	L	H	M	H	Yes	Self
P9	H	L	VL	H	L	Yes	Self
P10	H	L	L	L	M	Yes	Self
P11	H	L	L	VH	M	No	Non-Self
P12	L	M	M	VH	M	No	Non-Self
P13	H	M	VL	VL	L	No	Non-Self

## 6 Experimental Results and Comparisons

After analyzing the result of Table 3 and Table 4, it has been observed that Decision Tree can play an important role to provide the security to a computer system.

As Decision Tree is constructed by adding new nodes by ID3 Algorithm, the accuracy of the tree measured over the training examples increases monotonically. However, when measured over a set of test examples independent of the training examples, accuracy first increases, then decreases. It has been clear from the Figure 6 when to stop the tree growing in other words how many process parameters are sufficient to provide a decision on Self and Non-Self processes. It has been observed from the graph of Figure 6, that near about ten to fifteen parameters will be sufficient to get a better result.

It has been observed from the Figure 7, as the numbers of nodes (parameters) in the Decision Tree increase the security also increases. It has been observed from the graph of Figure 7 that when the number of nodes increases in the Decision tree after fifteen nodes, the security remain constant. It has been observed from the Figure 8, as the numbers of nodes (parameters) in the Decision Tree increases there is a degradation in system performance. It is has

been observed from the graph of Figure 8 that when the nodes increases after fifteen then the system performance decreases very rapidly. As processes and its parameter are used in the proposed approach for making the decision. During the security check by the Decision Tree the process remains ideal. So the overall system performance has been decreased by the proposed approach. If a process is identified as Non-Self by the proposed approach, user's action will be required. User can suspend its execution or delete this process from the system.

The proposed approach is compared some existing security approach as shown in Table 5. As shown in table it is clear that the proposed approach is better. Existing security approaches scan all the files and folders of the system but the proposed approach scan only the processes. It takes very less time to find out the Non-Self processes as other approaches scan all the files and data. Accuracy and detection rate is very high in comparison to existing anti-virus tool. As the proposed approach scans all processes so the system will become slow. The proposed approach is free from signature as required in existing approaches, proposed approach works on parameter's value not on any signature. No regular update is required in the proposed approach as it is required in anti-virus tools; new Decision Trees may be generated by new training examples. The disadvantage of the proposed

approach is that the Decision Tree constructed by various training example may be different, but these different trees may give the same result. To show the working of proposed approach five process parameters has been used, more parameters can be

added to improve the security. By adding more parameters different Decision Tree will be constructed.

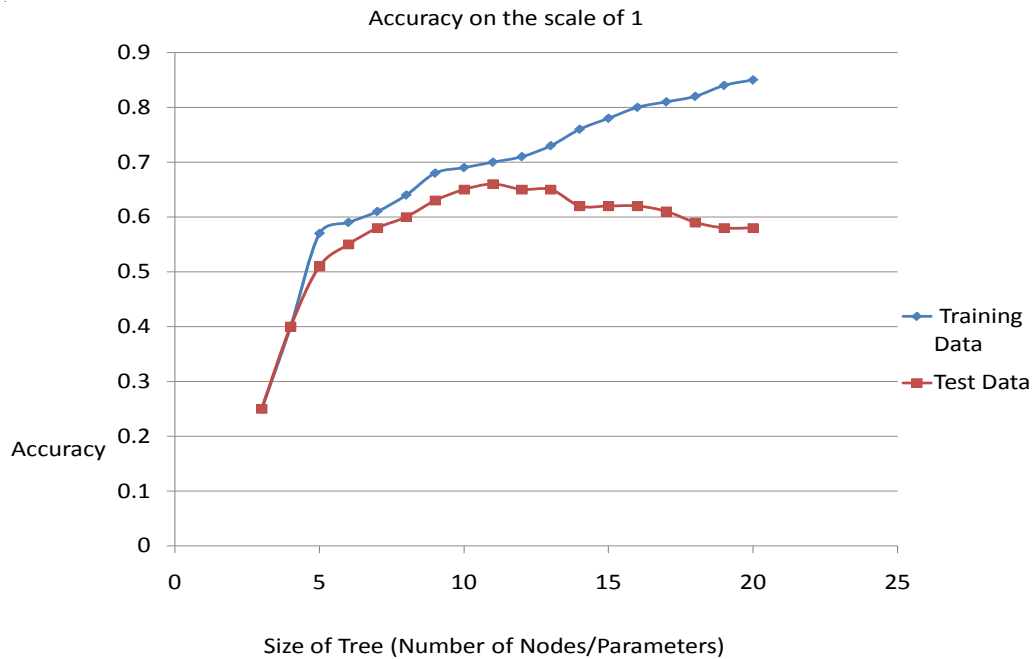


Fig.6: Accuracy of Decision Tree on training and test data

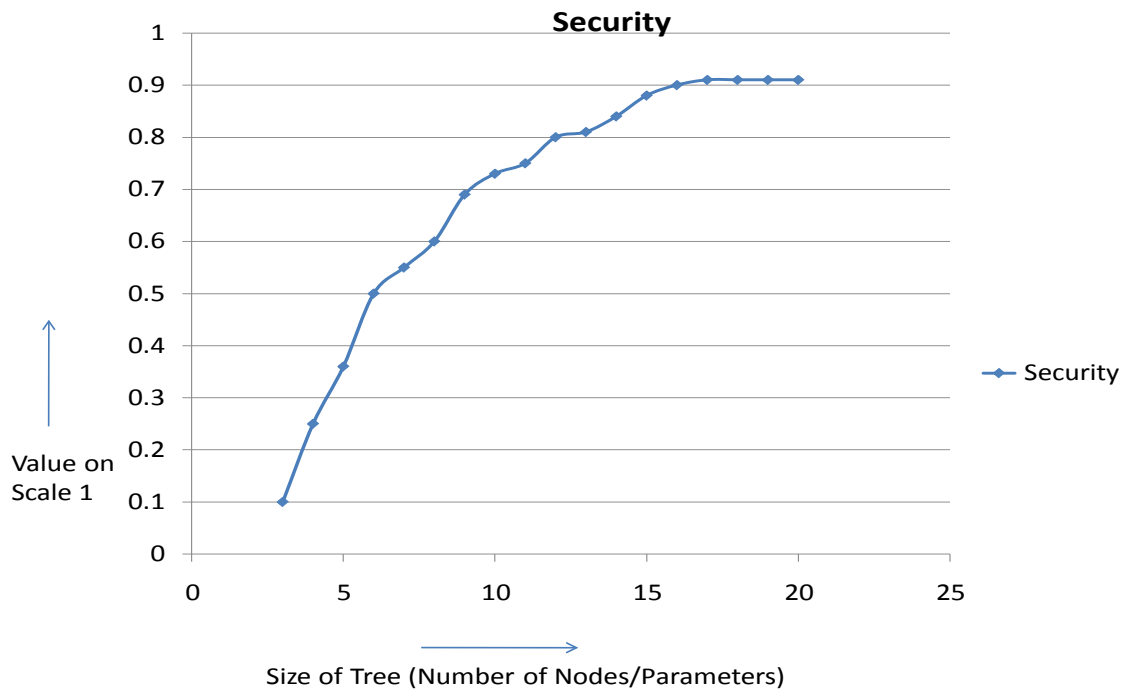


Figure 7: Security level of Decision Tree with increasing of number nodes.

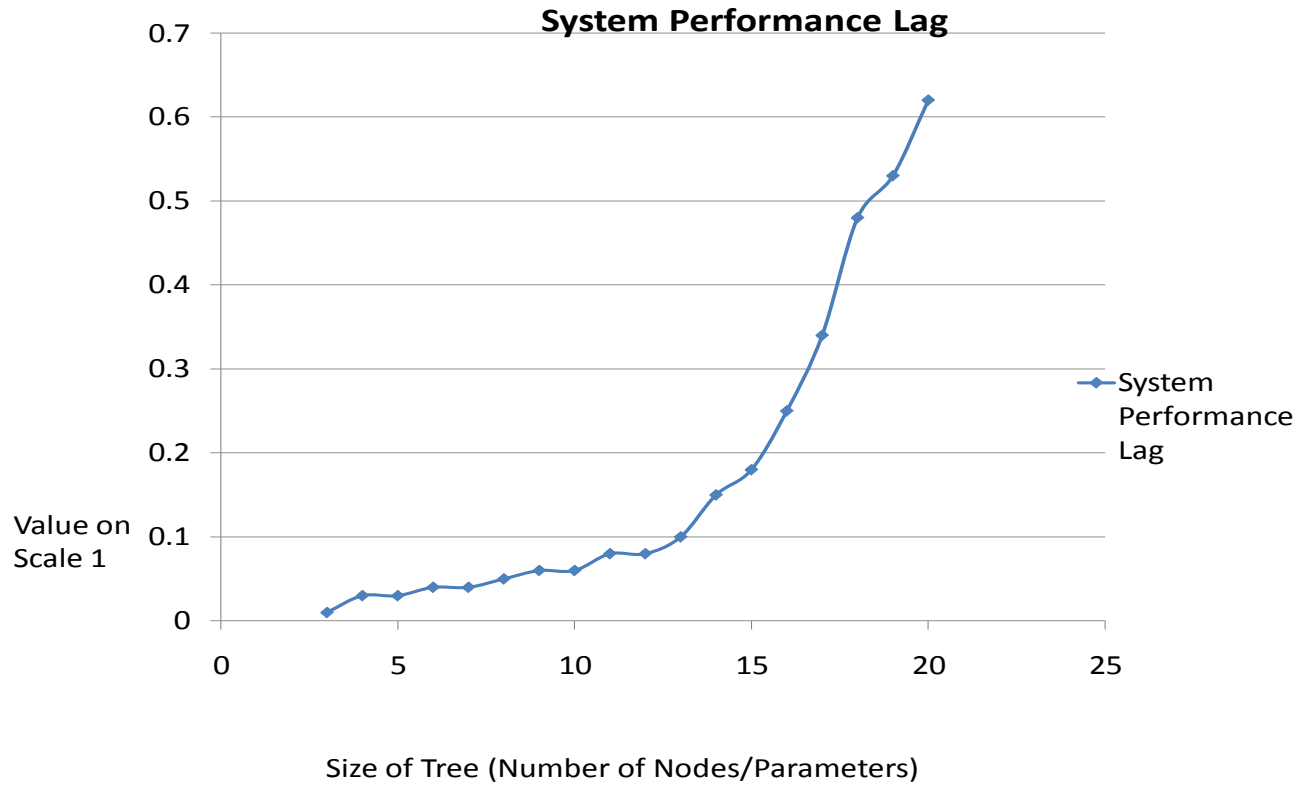


Fig.7: System Performance lag due to Decision Tree with respect to number node

Table 5: Comparison of Proposed Approach with Some Antivirus tools

<b>Parameters</b> <b>Antivirus</b>	<b>Scan Time</b>	<b>Accuracy</b>	<b>Detection Rate</b>	<b>Performance Lag</b>	<b>Signature based Detection</b>	<b>Regular Updating Required</b>
<b>AVG antivirus</b>	High	High	Normal	Yes	Yes	Yes
<b>Norton antivirus</b>	High	High	High	Yes	Yes	Yes
<b>Avast antivirus</b>	High	Medium	Normal	Yes	Yes	Yes
<b>Microsoft Security Essentials</b>	Average	High	High	Yes	Yes	Yes
<b>Proposed Approach</b>	Low	Very High	Very High	Yes	No	No

## 7. Conclusion and Future work

Various approaches and tools are used in the current scenario to provide security, but these approaches are not sufficient to provide best security level to a computer system. A better approach is required to get security.

Through this paper an approach based on Decision Tree is proposed to provide better security. Decision Tree learning plays an important role to provide better security to the computer system. The Decision Tree approach used in this paper provides better result over the current approach.

Decision Tree provides the best result to provide a learning system for identification of Non-Self processes. By using the various training and test data sets on ID3 algorithm a better learn system will be developed. Different tree can be constructed as according to training data. By using a large number of examples in training data a correct Decision Tree can be formed and it better works on test data.

### References:-

- [1] J. K. Percus, O. E. Percus and A. S. Perelson, "Probability of Self-Non-self discrimination" in Theoretical and Experimental Insights into Immunology, Volume 66, 1992, pp 63-70.
- [2] S. Forrest, S. A. Hofmeyr, A. B. Somayaji and T. A. Longstaff, "A sense of self for UNIX processes", in Proceedings of IEEE Symposium on Computer Security and Privacy, <http://www.cs.unm.edu/~immsec/publications/ieee-sp-96-unix.pdf>, 1996
- [3] Stephanie forrest, Alan S Perelson, 1994,"Self non-self discrimination in a computer", In Proceedings of the IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA: IEEE Computer Society Press, <http://www.cs.unm.edu/~immsec/publications/virus.pdf>
- [4] Rossouw von Solms, Johan Van Niekerk, "From information security to cyber security" Elsevier's Computer & Security, Volume 38, 2013, pp 97-102.
- [5] Cui-Qing Yang, "Operating System Security and Secure Operating Systems", version 1.4b, option for GSEC, Global Information Assurance Certification Paper.  
<http://www.giac.org/paper/gsec/2776/operating-system-security-secure-operating-systems/104723>, 2003
- [6] <http://www.cyberwarzone.com/massive-cyber-security-tools-list-2013>
- [7] Tom M. Mitchell, 1997, "Machine Learning", McGraw-Hill International Editions, Computer Science Series, 1997.
- [8] CurrProcess v1.13 - Freeware Process Viewer, Copyright (c) Nir Sofer, <http://www.nirsoft.net/utils/cprocess.html>, 2003-2008
- [9] Haoyong Lv, Hengyao Tang, "Machine Learning Methods And Their Application Research", IEEE International Symposium on Intelligence Information Processing and Trusted Computing, 2011, pp 108 – 110.
- [10] Wang Hua, MA Cuiqin, Zhou Lijuan, "A Brief Review of Machine Learning and its Application", IEEE Information Engineering and Computer Science, ICIECS, 2009, pp 1-4.
- [11] Olcay Taner Yıldız and Ethem Alpaydın, "Omnivariate Decision Trees" IEEE Transactions on Neural Networks, Volume 12, No. 6, 2001, pp 1539-1546.
- [12] Hua Ding, Xiu-Kun Wang, "Research on Algorithm of Decision tree induction", Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 2002, pp 1062-1065.
- [13] Chi Qingyun, "Research on Incremental Decision Tree Algorithm", International Conference on Electronic & Mechanical Engineering and Information Technology, 2011, pp 303-306.
- [14] Abdelhalim, A.; Traore, "A New Method for Learning Decision Trees from Rules" International Conference on Machine Learning and Applications, ICMLA. <http://www.uvic.ca/engineering/ece/isot/publications/by-area/rule-based-decision-tree/index.php>, 2009
- [15] Yang Yu-zhen ,Liu Pei-yu ,Zhu Zhen-fang ,QIU Ye , "The Research of an Improved Information Gain Method Using Distribution Information of Terms", IT in Medicine & Education, ITIME '09. IEEE International Symposium, Volume 1, 2009, pp 938-941.