Hybrid permission and Signature based technique for Malware Detection using KNN Naïve Baye's Algorithm

[1]Priya Sharma,[2] Jyoti Arora
[1] Research Scholar,[2] Assistant Professor
[1]Department of Computer Science & Engineering,
[1]Desh Bhagat University,Mandi Gobindgarh, India

Abstract - Malware Classification has been a challenging problem in the recent past and several researchers have attempted to solve this problem using various tools. It is security threat which can break machine operation while not knowing user's data and it's tough to spot its behavior. This paper proposes a novel technique using DBSCAN (Density based Kmeans) algorithmic rule to spot the behavior of malware. After classification from DBSCAN, pattern matching is applied using the instructions pattern in the generated reports. Among of these techniques a pattern based mostly technique is well famed for the detection of malware. For the moderation and improvement of the present system the signature based mostly technique is most popular. The results are found to be quite accurate and better than the existing ones in terms of accuracy.

IndexTerms - **Malware, DBSCAN Algorithm, Pattern matching, Polymorphism, Metamorphism.**

I. INTRODUCTION

Nowadays, there are numbers of computer security threat that cause by the malware attack have extraordinarily enlarged. It includes seventeen categories like worms, viruses, spyware, bug and together several malicious packages that cause a billion of losses to the operation worldwide. However, all types of malware have their specific objective , the foremost purpose is to interrupt the computer operation.

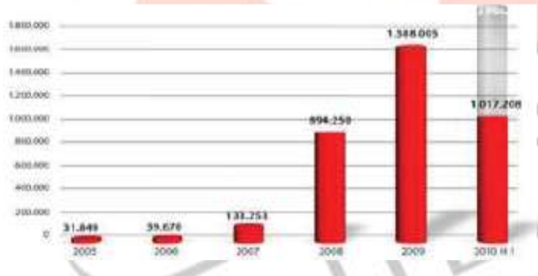


Figure. 1: Malware growth since 2005 to 2010

Malware is defined by its malicious intent, acting against the requirements of the computer user, and does not include software that causes unintentional harm due to some deficiency. Malware is sometimes used broadly against government or corporate websites to gather guarded information, or to disrupt their operation in general. However, malware is often used against individuals to gain information such as personal identification numbers or details, bank or credit card numbers, and passwords. Left unguarded, personal and networked computers can be at considerable risk against these threats. The computer programs designed to infiltrate and damage computers without the users consent . "Malware" is the general term covering all the different types of threats to computer safely such as viruses, spyware, worms, Trojans, rootkits and so on.

However, attributable to the technology advancement many malware writers try to use higher concealment techniques to avoid detection. The concealment technique is created with the combination of previous behavior therefore on attack and at identical time to avoid the signature-based detection. In this, several common techniques that are commonly used like as polymorphic and metamorphic. Additionally analysis ought to be tired order to hunt out possible resolution on avoiding current technique of malware attacks.

There are four main generations in the gradual development of the stealing methodologies: secret writing, Polymorphism, Oligomorphism, and natural action. It displays the evolution timeline of camouflage techniques in malware.

2. Primitive Malware

When the story began, virus writing has a kind of programming fun for portable computer specialists to point off their technical skills, but it step by step became as a tool for several functions, like swiping the people's information, like mastercard numbers, passwords, or bank account numbers, or for avenges functions, and so on.

3. Stealth Malware

Malware creators' initial tries therefore on turn tail from redounded to appear of stealing techniques. Stealth virus is prepared to cover its signs and traces. Virus normally changes and modifies info resources on the system. For example, a file-hosted virus would possibly append its own code to the tip of Associate in possible file. If Associate in application examines the infected file, it'll discover the being code inside the file and catch the virus.

4. Camouflage Evolution

4.1 Encryption

Malware authors invariably try to improve the program to escape from the code analyzer technicians. Consequently, they will get longer for his or her created malware to live inside the wild and blow their own horns extra. The simplest and earliest methodology utilized by the malware programmers to achieve this goal was secret writing. Encrypted virus contains two basic sections: a secret writing loop and main body. Decrypt, or secret writing loop, might be a brief piece of code, that's responsible to cipher and rewrite code of main body. The foremost body is that particular code of malware, encrypted, and is not purposeful before its being decrypted by secret writing loop. Once the virus starts to run on host portable computer, initial the decrypt or loop ought to decipher the foremost body into machine possible code and purposeful info.

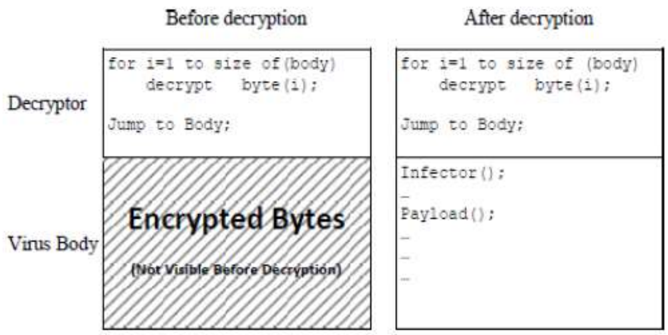


Figure.2: Structure of encrypted virus

4.2 Oligomorphism

It contains a collection of assorted decryptors, that are haphazardly selected for the replacement victim. Oligomorphism is not a heavy drawback for antivirus package as a result of it entirely makes a malware slightly harder to seem at. In distinction to encrypted virus, antivirus engine ought to check all possible decryptor instances instead of looking for only 1 decryptor, and it wishes extended time decryptors.

4.3 Polymorphic

Polymorphic techniques try to create analysis of the virus additional sturdy by ever-changing its look. The principal rule is to change appearance of code constantly, from a reproduction to a unique [7]. It ought to be administered in such the best method so no permanent common string keeps among variants of a deadly illness to be exploited by the antivirus scanner engine for detection purpose. Polymorphic techniques are rather hard to implement and manage.

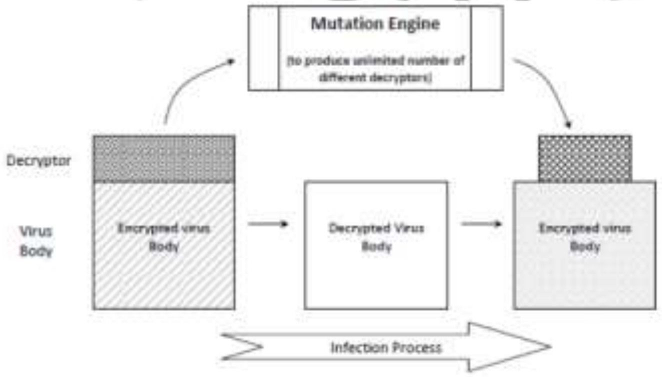


Figure 3: Polymorphic virus structure

4.4 Metamorphism

Unlike the three camouflage generations, metamorphic virus has no encrypted. Therefore, it does not would really like decryptor, but like polymorphic virus, it works a mutation engine, as well, instead of modifying the decryptor loop entirely, it mutates all its body. The foremost sententious definition of metamorphic viruses is introduced by Igor Muttik "Metamorphics are body-polymorphics". Each new copy may need completely totally different structure, code sequence, size and descriptive linguistics properties, but behaviour of the virus does not modification.

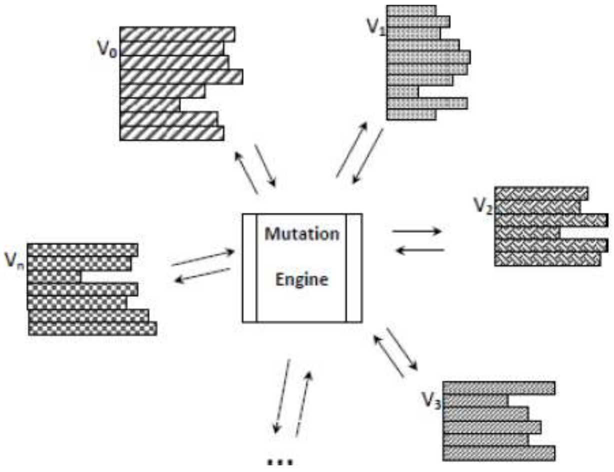


Figure.4: Metamorphic virus propagation scheme

Components of mutation engine square measure displayed. Once the virus finds things of its own code, it's to convert code into assembly instruction, that's finished by an inside dis-assembler. The code analyzer is responsible to give information for code device module. The code device could have some information like structure and flow chart of the program, subroutines, life quantity of variables and registers, and so on.

LITERATURE REVIEW

Malware could be a program with the malicious intent designed to interrupt computer on it executes the network on that it communicates **[1].** However all kinds of malware have specific objective, the foremost purpose is to interrupt computer operation. As results, the protection management should be enforced therefore on safeguard all code and knowledge against the replacement, modification or sub versioned. Signature-based matching technique is told the modish approaches to malware detection **[2].** This scheme was commercially applied by the anti-virus or anti-spyware product among market. The foremost limitation of signature based technique is detection new malware. This technique is used distinctive bytes string forever fails to take unseen malware that typically known as zero day attacks.In this, a flexible and automatic approach to extract the malware behavior by the perceptive systems that operate calls performed during virtualized execution in surroundings. Similarities and distances between the malware behaviors unit of the measurement computed that lets classifying malware **[3]** behaviors. The foremost choices of approach reside in coupling a sequence alignment technique to figure out similarities and leverage the distance to figure out associated distances. **[4]** The classification technique projected by work is pattern tree. However, this method still encompasses the limitation because of incorrectly classified number of malware behaviour. It applied machine learning rule to their malware detection method. **[5]** That technique has pattern adaptive info compression thus on counter the limitation of signature-based technique in present industrial anti-malware tool. The two limitations are signature-based technique. First, not all malicious programs have bit patterns that unit of the measurement proof of their malicious nature or collectively not recorded among the virus book of facts. Second, obfuscated malware that take many types of bit patterns will not engage on signature-based technique. The projected **[6]** technique used adaptive info compression model and prediction by the partial matching as learning engine to form two compression models. This method works on unstructured input, that is, raw executables, with Associate in underlying math compression model.In this, consider the structural analysis and comparison of computer code supported by management flows. They used graph to enhance the results of comparison. The difficulties **[7]** of disassembling the code placed at specific locations within network traffic aren't absolutely self-addressed; however, the methodology is computationally valuable. There is a tendency to found that geological process and obfuscation modification the management flow in such some way that easy comparison on the structural characteristics is probably going to fail. The malware detection and classification tools area unit didn't address variants that may be created mechanically victimization newest polymorphic and metamorphic transformation engines that may manufacture variants that area unit shut to every different.Chouchane and Lakhotia**[8]** proposed using “engine signatures” to assist in detecting metamorphic malware. Basically, this technique evaluates collected forensic evidence from x86 code segments through a code scoring function. This score is a measure of how likely it is that the code has been generated by a known instruction substituting metamorphic engine.In lately the pc systems could be a part of our life and have various applications in our society the Signature-based strategies are accustomed discover the Malware **[9].**These strategies are used wide by industrial detectors. However assault techniques of malwares progressed reciprocally. New malware uses several techniques like polymorphism and geological process to run off from detection strategies. These techniques modify the look of malicious codes by encrypting the codes or commutation them with the different codes or mix of them in such way that the malicious origin of codes remains unchanging. Bayer et al. was used an (able to be made bigger or smaller) clustering approach to identify and group harmful programs or apps samples that show almost the same behavior **[10].** This approach also (does/completes) energetic/changing analysis to get the execution traces of harmful programs or apps programs using automated tools.This approach is used to boost the efficiency of dynamic malware analysis systems **[11].** It is a large sort of latest malicious files presently appears. It’s because of mutations of only variety of malware programs. The projected system avoids analyzing malware binaries that just represent mutated instances of already analyzed polymorphic malware. It drastically decrease the quantity of some time required for analyzing a set of malware programs. Seyedet al. proposed harmful programs or apps detection that applied data mining which is based on the analysis of byte level file content **[12].** This way of doing things also designed to provide protection against first day launched harmful programs or apps.

PROBLEM FORMULATION

Understanding malware behaviors is useful information for computer system. But it is difficult to identify its behavior and classification. The aim of this thesis is to design a automatic malware classification system based on the reports generated by the executable files. The malwares will be classified into three classifications. First task is to separate the clean files from the malwares and then a text mining and pattern matching algorithm needs to be applied for classifying these malwares into normal malwares and poly/metamorphic malwares. An improved Hybrid DBSCAN algorithm needs to be designed which will take into account the various parameters like icmp counts, number of host calls, affected system part, binary registry key etc.

II. PROPOSED METHODOLOGY

The thesis proposes a novel algorithm to classify malwares as clean, normal malwares and polymorphic/metamorphic malwares. The approach is to generate pydasm report of each malware and then collect the other parameters like binary count,icmp,opcode and machine cycles. Internet Control Message Protocol (**ICMP**) is one of the main protocols of the Internet Protocol Suite. It is used by network devices, like routers, to send error messages indicating.It is a computer file that is not a text file. Many binary file formats contain parts that can be interpreted as text; for example some computer document files containing formatted text, such as older Microsoft Word document files, contain the text of the document but also contain formatting information in binary form

The instruction sets will be extracted from the report via text mining and text preprocessing will be done for various processes like comment removal, function extraction etc. Once the features are extracted then a database corresponding to various files and function list will be created. The similarity score will be calculated between any two files using signature based technique and then a Hybrid DBSCAN algorithm will be applied on the selected feature for classification.

Pattern Matching

The text report of the remaining files is generated and instruction sets are fetched from it for pattern extraction. A pattern matching algorithm is designed on the instruction sets by calculating the scores of matching between two documents. The matching criteria is set by creating a moving window of the instruction set and sliding the instruction set of one document against the other. All the instructions are given a numerical value and the moving window is subtracted from each other. The Scores are calculated on the basis of number of zeros in the result of subtraction.

Flowchart

The process includes the decoding of exe files and extraction of instructions. Assign no. to each instructions in the dictionary. Scores are calculated in the end by adding the no. of zeroes. If the no .of 0’s are more, it means high score and high accuracy but if no. of 0’s are less then it means file contains malware and there is less accuracy.

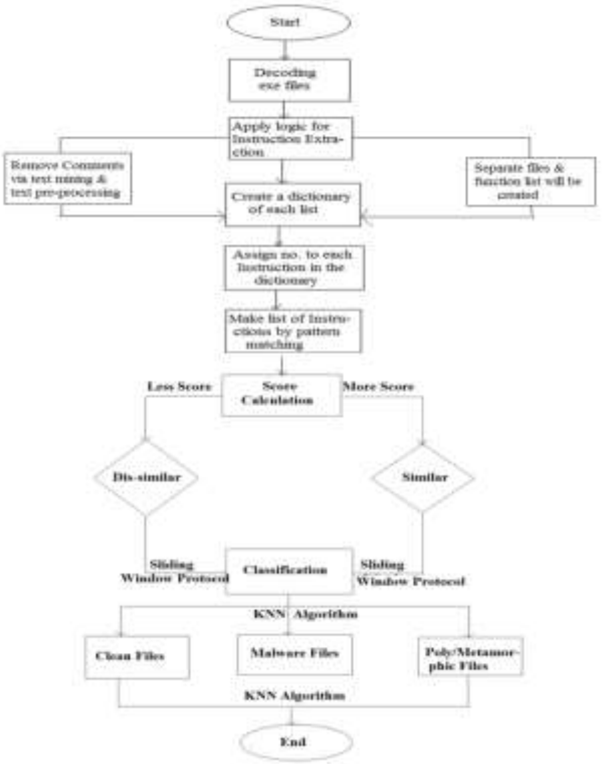
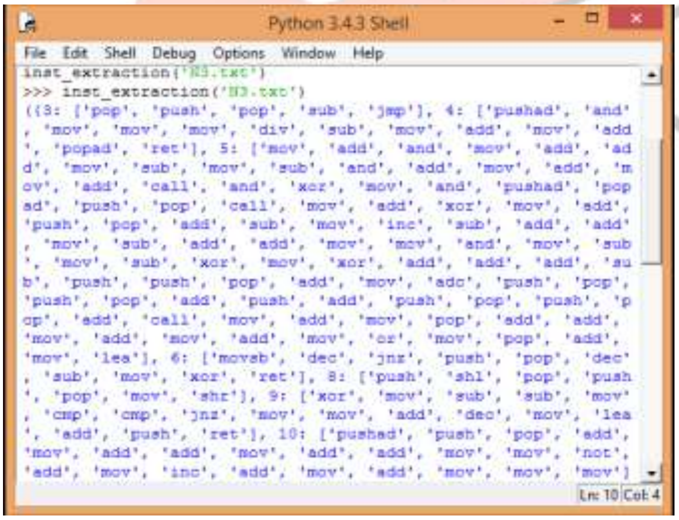

Start
Decoding exe files
Apply logic for Instruction Extraction
Remove Comments via text mining & text pre-processing
Separate files & function list will be created
Create a dictionary of each list
Assign no. to each Instruction in the dictionary
Make list of Instrucitons by pattern matching
Less Score
Score Calculation
More Score
Dis-similar
Similar
Sliding Window Protocol
Classification
Sliding Window Protocol
KNN Algorithm
Clean Files
Malware Files
Poly/Metamorphic Files
KNN Algorithm
End

Figure.7: Flowchart representation

III. SIMULATION RESULTS AND DISCUSSION

All the simulation has been done in a computer with 4 GB of RAM and 1.7 GHz processor. The language utilized for scripting is Python. Various Natural Language processing tool kits have been utilized.

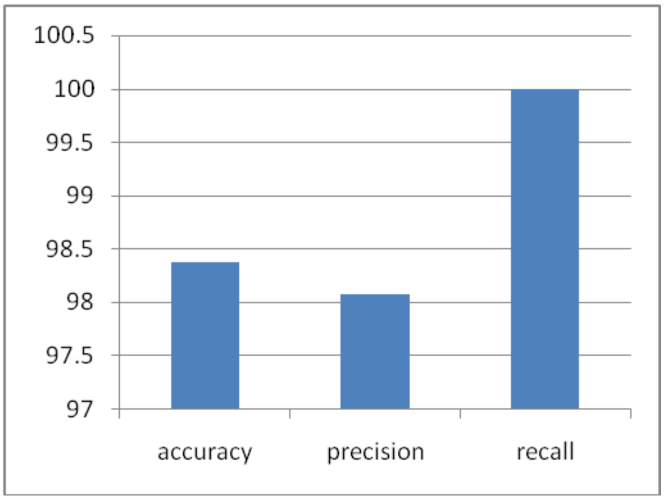
Total number of files taken for testing purpose is 25 and by using DBSCAN algorithm for malware classification the 23 files are correct means malware free out of 25 files and 2 files are wrong as shown in below figure. The value of K is taken as 3.



```
inst_extraction('N3.txt')
>>> inst_extraction('N3.txt')
({3: ['pop', 'push', 'pop', 'sub', 'jmp'], 4: ['pushad', 'and'
, 'mov', 'mov', 'mov', 'div', 'sub', 'mov', 'add', 'mov', 'add
', 'popad', 'ret'], 5: ['mov', 'add', 'and', 'mov', 'add', 'ad
d', 'mov', 'sub', 'mov', 'sub', 'and', 'add', 'mov', 'add', 'm
ov', 'add', 'call', 'and', 'xor', 'mov', 'and', 'pushad', 'pop
ad', 'push', 'pop', 'call', 'mov', 'add', 'xor', 'mov', 'add',
'push', 'pop', 'add', 'sub', 'mov', 'inc', 'sub', 'add', 'add'
, 'mov', 'sub', 'add', 'add', 'mov', 'mov', 'and', 'mov', 'sub
', 'mov', 'sub', 'xor', 'mov', 'xor', 'add', 'add', 'add', 'su
b', 'push', 'push', 'pop', 'add', 'mov', 'adc', 'push', 'pop',
'push', 'pop', 'add', 'push', 'add', 'push', 'pop', 'push', 'p
op', 'add', 'call', 'mov', 'add', 'mov', 'pop', 'add', 'add',
'mov', 'add', 'mov', 'add', 'mov', 'or', 'mov', 'pop', 'add',
'mov', 'lea'], 6: ['movsb', 'dec', 'jnz', 'push', 'pop', 'dec'
, 'sub', 'mov', 'xor', 'ret'], 8: ['push', 'shl', 'pop', 'push
', 'pop', 'mov', 'shr'], 9: ['xor', 'mov', 'sub', 'sub', 'mov'
, 'cmp', 'cmp', 'jnz', 'mov', 'mov', 'add', 'dec', 'mov', 'lea
', 'add', 'push', 'ret'], 10: ['pushad', 'push', 'pop', 'add',
'mov', 'add', 'add', 'mov', 'add', 'add', 'mov', 'mov', 'not',
'add', 'mov', 'inc', 'add', 'mov', 'add', 'mov', 'mov', 'mov']
```

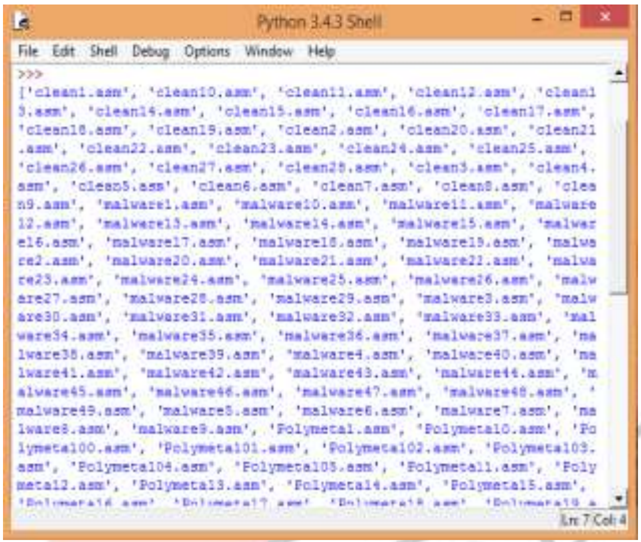
Figure. 5: Representation of malware classification by using Hybrid algorithm

Figure below represents the correctly classified documents vs the incorrectly classified documents. As observed the number of correctly classified documents forms a high percentage of the incorrectly classified ones.



.

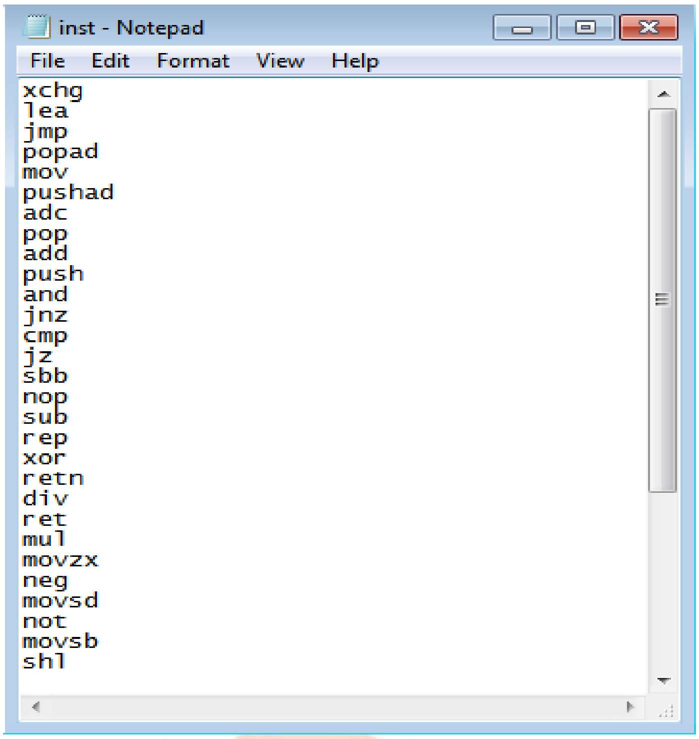
Figure 6: Graphical representation of malware classification by using DBSCAN algorithm



```
>>>
['clean1.asm', 'clean10.asm', 'clean11.asm', 'clean12.asm', 'clean1
3.asm', 'clean14.asm', 'clean15.asm', 'clean16.asm', 'clean17.asm',
'clean18.asm', 'clean19.asm', 'clean2.asm', 'clean20.asm', 'clean21
.asm', 'clean22.asm', 'clean23.asm', 'clean24.asm', 'clean25.asm',
'clean26.asm', 'clean27.asm', 'clean28.asm', 'clean3.asm', 'clean4.
asm', 'clean5.asm', 'clean6.asm', 'clean7.asm', 'clean8.asm', 'clea
n9.asm', 'malware1.asm', 'malware10.asm', 'malware11.asm', 'malware
12.asm', 'malware13.asm', 'malware14.asm', 'malware15.asm', 'malwar
e16.asm', 'malware17.asm', 'malware18.asm', 'malware19.asm', 'malwa
re2.asm', 'malware20.asm', 'malware21.asm', 'malware22.asm', 'malwa
re23.asm', 'malware24.asm', 'malware25.asm', 'malware26.asm', 'malw
are27.asm', 'malware28.asm', 'malware29.asm', 'malware3.asm', 'malw
are30.asm', 'malware31.asm', 'malware32.asm', 'malware33.asm', 'mal
ware34.asm', 'malware35.asm', 'malware36.asm', 'malware37.asm', 'ma
lware38.asm', 'malware39.asm', 'malware4.asm', 'malware40.asm', 'ma
lware41.asm', 'malware42.asm', 'malware43.asm', 'malware44.asm', 'm
alware45.asm', 'malware46.asm', 'malware47.asm', 'malware48.asm', '
malware49.asm', 'malware5.asm', 'malware6.asm', 'malware7.asm', 'ma
lware8.asm', 'malware9.asm', 'Polymeta1.asm', 'Polymeta10.asm', 'Po
lymeta100.asm', 'Polymeta101.asm', 'Polymeta102.asm', 'Polymeta103.
asm', 'Polymeta104.asm', 'Polymeta105.asm', 'Polymeta11.asm', 'Poly
meta12.asm', 'Polymeta13.asm', 'Polymeta14.asm', 'Polymeta15.asm',
'Polymeta16.asm', 'Polymeta17.asm', 'Polymeta18.asm', 'Polymeta19.a
```

Figure.7: Representation of Clean, Malware, Poly/Metamorphic files

Figure shows the results of pre-processing of the documents and as it is observed the pre-processing steps enable us to correctly classify the documents.



```
xchg
lea
jmp
popad
mov
pushad
adc
pop
add
push
and
jnz
cmp
jz
sbb
nop
sub
rep
xor
retn
div
ret
mul
movzx
neg
movsd
not
movsb
shl
```

Figure.8: Instruction list

Figure shows the list of instructions which are utilized for pattern matching. These instructions are stored in a text file and read in the code for pattern matching.

Table 1: Clean, Malware,Polydest files

	Clean	Malware	Polydest
Clean	26	0	0
Malware	3	49	0
Polydest	0	0	105

Table 2: Confusion Matrix

	True	False
Positive	154	3
Negative	26	0

$$Accuracy = \frac{154 + 29}{154 + 29 + 3}$$

Accuracy = 98.38%

$$Precision = \frac{Tp}{Tp + Fp}$$

$$Precision = \frac{154}{154 + 3}$$

=98.08%

$$Recall = \frac{Tp}{Tp + Fn}$$

IJEDR

$$Recall = \frac{154}{154}$$

$$= 100\ \%$$

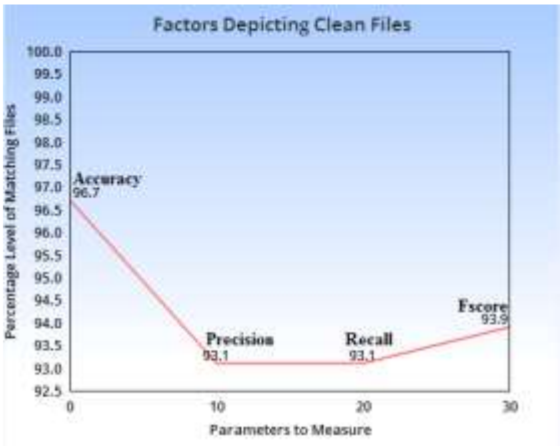


Figure.8: Graph representing Clean files

IV. CONCLUSION

The paper tries to detect different categories of malware viz. polymorphic and metamorphic malwares by using the concept of pattern matching. The KNN algorithm has been successfully implemented for detecting the malwares by classification. The discussed approach seems to work quite well and seems to yield encouraging results for betterment of PC's and protects our confidential data from viruses etc. Different folders are made for poly/metamorphic malwares and in the end it put them according to their class. This shows the efficiency of our algorithm. In future other algorithms can be utilized for the same and our proposed system can be used with other techniques also.

References

[1] MohamadFadliAmanJantan "An Approach for Malware Behavior Identification and Classification" ,IEEE 2011.

[2] Xufang Li, Peter K.K. Loh, Freddy Tan "Mechanisms of Polymorphic and Metamorphic Viruses" European Intelligence and Security Informatics Conference 2011.

[3] Nitesh Kumar Dixit1 ,Lokesh mishra2 , Mahendra Singh Charan3 and Bhabesh Kumar Dey4, "The new age of computer virus and their detection" International Journal of Network Security & Its Applications (IJNSA), May 2012.

[4] AbdellatifBerkat "Metamorphic computer virus detection by Case Based Reasoning (CBR) methods" International Journal of Software Engineering and Applications (IJSEA) octber 2011

[5] .M. Bauer, M.J.G. van Eeten "Obfuscation: The Hidden Malware" Security & Privacy IEEE 2011.

[6] BabakBashari Rad, Maslin Masrom,Suhaimi Ibrahim "Camouflage in Malware: from Encryption to Metamorphism" IJCSNS International Journal of Computer Science and Network security August 2012.