Artificial intelligence in antivirus detection system computer science

Abstract- Artificial intelligence (AI) techniques have played increasingly important role in antivirus detection. At present, some principal artificial intelligence techniques applied in antivirus detection are proposed, including heuristic technique, data mining, agent technique, artificial immune, and artificial neural network. It believes that it will improve the performance of antivirus detection systems, and promote the production of new artificial intelligence algorithm and the application in antivirus detection to integrate antivirus detection with artificial intelligence. This paper introduces the main artificial intelligence technologies, especially Heuristic which have been applied in antivirus system. Meanwhile, it also points out a fact that combining all kinds of artificial intelligence technologies will become the main development trend in the field of antivirus.

Keywords- Anti-virus, Artificial Intelligence, Data mining, Heuristic, Neural network

Introduction

Artificial Intelligence (AI) is the branch of computer science which deals with intelligence of machines where an intelligent agent is a system that perceives its environment and takes actions which maximize its chances of success. It has numerous applications like robotics, medicine, Finance, Space.

One of the most recent one is antivirus softwares.

Here we give details regarding heuristic method used in antivirus software.

Malware and its types

Malware (malicious software) is software designed to infiltrate or damage a computer system without the owner’s informed consent.

Malware types

We can distinguish quite few malicious software types. It is important to be aware that nevertheless all of them have similar purpose, each one behave differently.

Viruses

Worms

Wabbits

Trojan horses

Exploits/Backdoors

Spyware

Due to different behaviour, each malware group uses alternative ways of being undetected. This forces anti-virus software producers to develop numerous solutions and countermeasures for computer protection. This paper focuses on methods used especially for virus detection, not necessarily effective against other types of malicious software.

Infection Strategies

To better understand how viruses are detected and recognized, it is essential to divide them by their infection ways.

A. Non Resident Viruses

The simplest form of viruses which don’t stay in memory, but infect founded executable file and search for another to replicate.

Resident viruses

More complex and efficient type of viruses which stay in memory and hide their presence from other processes. Kind of TSR apps.

Fast infectors type which is designed to infect as many files as possible.

Slow infectors using stealth and encryption techniques to stay undetected outlast.

Methods Used

A. Metaheuristic

Metaheuristic is a heuristic method for solving a very general class of computational problems by combining user-given black-box procedures in a hopefully efficient way. Metaheuristics are generally applied to problems for which there is no satisfactory problem-specific algorithm or heuristic.

B. Heuristic

Heuristic is a method to help solve a problem, commonly an informal method. It is particularly used to rapidly come to a solution that is reasonably close to the best possible answer.

General Heuristics

It is important to remember that metaheuristics are only ‘ ideas’ to solve a problem not a specific way to do that. List below shows main metaheuristics used for virus detection and recognition:

Pattern matching

Automatic learning

Environment emulation

Neural networks

Data mining

Bayes networks

Hidden Markov models

Concrete Heuristics

Specific heuristics practically used in virus detection and recognition, are naturally inherited from metaheuristics.

And so, for example concrete method for virus detection using neural networks can be implementation of SOM (Self Organizing Map). Neural Networks (metaheuristic) A? aˆ aˆ™ SOM (heuristic).

The most popular, and one of most efficient heuristic used by

anti-virus software is technique called Heuristic Scanning.

Lacks in Specific Detection

Great deal of modern viruses are only slightly changed versions of few conceptions developed years ago. Specific detection methods like signature scanning became very efficient ways of detecting known threats. Finding specific signature in code allows scanner to recognize every virus which signature has been stored in built-in database.

BB ? 2 B9 10 01 81 37 ? 2 81 77 02 ? 2 83 C3 04 E2 F2

FireFly virus signature(hexadecimal)

Problem occurs when virus source is changed by a programmer or mutation engine. Signature is being malformed due to even minor changes. Virus may behave in an exactly same way but is undetectable due to new, unique signature.

BB ? 2 B9 10 01 81 37 ? 2 81 A1 D3 ? 2 01 C3 04 E2 F2

Malformed signature(hexadecimal)

Heuristic Scanning

We can recognise a virus without examining its

structure by its behaviour and characteristics. Heuristic scanning in its basic form is implementation of three metaheuristics:

Pattern matching

Automatic learning

Environment emulation

The basic idea of heuristic scanning is to examine assembly language instruction sequences(step-by-step) and qualify them by their potential harmfulness. If there are sequences behaving suspiciously, program can be qualified as a virus. The phenomenon of this method is that it actually detects threats that aren’t yet known!

Fig1. Examination of assembly language sequence

A. Recognising Potential Threat

In real anti-virus software, heuristic scanning is implemented to recognize threats by following built-in rules, e. g. if program tries to format hard drive its behaviour is highly suspicious but it can be only simple disk utility. Singular suspicion is never a reason to trigger the alarm. But if the same program also tries to stay resident and contains routine tosearch for executables, it is highly probable that it’s a real virus. AV software very often classifies sequences by their behaviour granting them a flag. Every flag has its weight, if total values for one program exceeds a predefined threshold, scanner regards it as virus.

Fig. 2. Single-layer classifier with threshold

Heuristics Flags

Some scanners set a flag for each suspected ability which has been found in the file being analyzed. This makes it easier to explain to the user what has been found. TbScan for instance recognizes many suspected instruction sequences. Every suspected instruction sequence has a flag assigned to it.

A. Flag Description:

F = Suspicious file access. Might be able to infect a file.

R = Relocator. Program code will be relocated in a suspicious way.

A = Suspicious Memory Allocation. The program uses a non-standard way to search for, and/or allocate memory.

N = Wrong name extension. Extension conflicts with program structure.

S = Contains a routine to search for executable (. COM or . EXE) files.

= Found an instruction decryption routine. This is common for viruses but also for some protected software.

E = Flexible Entry-point. The code seems to be designed to be linked on any location within an executable file. Common for viruses.

L = The program traps the loading of software. Might be a virus that intercepts program load to infect the software.

D = Disk write access. The program writes to disk without using DOS.

M = Memory resident code. This program is designed to stay in memory.

! = Invalid opcode (non-8088 instructions) or out-of-range branch.

T = Incorrect timestamp. Some viruses use this to mark infected files.

J = Suspicious jump construct. Entry point via chained or indirect jumps. This is unusual for normal software but common for viruses.

? = Inconsistent exe-header. Might be a virus but can also be a bug.

G = Garbage instructions. Contains code that seems to have no purpose other than encryption or avoiding recognition by virus scanners.

U = Undocumented interrupt/DOS call. The program might be just tricky but can also be a virus using a non-standard way to detect itself.

Z = EXE/COM determination. The program tries to check whether a file is a COM or EXE file. Viruses need to do this to infect a program.

O = Found code that can be used to overwrite/move a program in memory.

B = Back to entry point. Contains code to re-start the program after modifications at the entry-point are made. Very usual for viruses.

K = Unusual stack. The program has a suspicious stack or an odd stack.

Avoiding False Positives

Just like all other generic detection techniques, heuristic scanners sometimes blame innocent programs for being contaminated by a virus. This is called a “ false positive” or “ False Alarm”. The reason for this is simple. Some programs happen to have several suspected abilities.

If a heuristic scanner pops up with a message saying: “ This program is able to format a disk and it stays resident in memory”, and the program is a resident disk format utility, is this really a false alarm? Actually, the scanner is right. A resident format utility obviously contains code to format a disk, and it contains code to stay resident in memory.

The heuristic scanner is therefore completely right! You could name it a false suspicion, but not a false positive. The only problem here is that the scanner says that it might be a virus. If you think the scanner tells you it has found a virus, it turns out to be a false alarm. However, if you take this information as is, saying ‘ ok, the facts you reported are true for this program, I can verify this so it is not a virus’, I wouldn’t count it as a false alarm. The scanner just tells the truth. The main problem here is the person who has to make decisions with the information supplied by the scanner. If it is a novice user, it is a problem.

Whether we call it a false positive or a false suspicion doesn’t matter. We do not like the scanner to yell every time we scan. So we need to avoid this situation. How do we achieve this?

Definition of (combinations of) suspicious abilities

Recognition of common program codes

Recognition of specific programs

Assumption that the machine is initially not infected

Performance of Heuristics Scanning

Heuristics is a relatively new technique and still under development. It is however gaining importance rapidly. This is not surprising as heuristic scanners are able to detect over 90% of the viruses without using any predefined information like signatures or checksum values. The amount of false positives depends on the scanner, but a figure as low as 0. 1% can be

reached easily. A false positive test however is more difficult to perform so there are no independent results available.

Pros and Cons

A. Advantages

Can detect future viruses. User is less dependent on product updates.

B. Disadvantages

False positives are possible. Judgment of the result requires some basic knowledge.

Conclusions

Thus, artificial intelligence technique helps improving the performance of antivirus softwares.

This detection-avoiding method makes detection by conventional anti-virus products easier because it means that the programmer can not use very tight and straight code. The virus writer will be forced to write more complex viruses. Thus artificial intelligence increases the threat to virus writers.

Acknowledgment

I hereby thank Ms. Padmapriya for encouraging and helping us for the submission of this paper