Malware Detection Using Markov Blanket Based on Opcode Sequences

Hamid Divandari
Islamic Azad University
Faculty of computer engineering
Mashhad, Iran
hamiddivandari@ mshdiau.ac.ir

Bassir Pechaz
Imam Reza University
Faculty of computer engineering
Mashhad, Iran
bpechaz@gmail.com

Majid Vafaie Jahan
Islamic Azad University
Faculty of computer engineering
Mashhad, Iran
vafaeijahan@mshdiau.ac.ir

***Abstract* — Generally malwares are defined as security threat for computing systems and computer networks. Today's common and most used method for detecting malwares is signature-based method. There is also another way that focuses on file behavior. More over, recent researches used data mining and machine learning and other heuristic solutions for malware detection. In the current research, new method based on opcode sequences extraction has been introduced. Markov Blanket Algorithm has been used as a feature selection method to reduce the number of features. Using Markov Blanket reduced features size about 99%. Finally, a Hidden Markov Model has been trained based on the best features. Experimental results revealed that the trained model detects malwares with about 99% precision and 98% sensitivity.**

Keywords: Malware Detection, Markov Blanket, Hidden Markov Model

I. INTRODUCTION

Malware, considered as a computer program that is created meaningfully to harm data or computer's operations[1]. Recently, malwares have grown very quickly in both number and power. According to the report of security laboratory of MacAfee, at the last three month of 2014, more than 387 new security threat had been detected every minute[2]. Contrary to the past that malwares were created for fun, nowadays they are being created for economic benefits. Generally recent malwares are more complex, organized and dangerous than the past ones[3].

There are three main methods used for malware detection: Signature based, Behavioral based and Heuristic ones[4]. Most of today's antivirus programs use signature based methods as detection techniques. Malware's signature is a unique sequence of bytes that can absolutely detect malware. Signature based methods have acceptable results in practice[5], but they can't handle new types or unknown malwares. Behavioral methods collect and control set of behavioral properties of a computer program and make decision about its type. These types of methods have good performance for detecting new type of malwares, but they need more time in practice[6]. Heuristic methods use data mining and machine learning techniques to select unique features of the file and then decide about its type based on the selected features. Features comprise opcodes, API method recall, control flow graph and other kinds of program features.

One of the most important research about malware detection based on opcodes, had been done by Bilar[7]. Bilar reported opcodes with fewer redundancy are better for malware detection. Santos et al[8] extracted opcode sequences and assigned particular weight to each extracted sequence. Assigned weight was related to frequency appearance of opcode sequences and mutual information of opcodes in the file.

Santos improved his work by using opcode sequences and data mining methods. Santos used k-nearest neighborhood, SVM, decision tree and Bayesian network as data mining method in his work and finally reported that the best result was relevant to SVM method[9]. Zoletkin et al[10] disassembled file to sequence of opcodes in length of 1 and 2 and used RELIFE method as feature selection method. Zoletkin used Iterative Support Vector Machine for creating his model. Zoletkin reported that the best result was 97.09% when he used opcode sequences of length 2. Bigloo et al[11] disassembled malware and benign files to their opcodes sequence and introduce the new method of Role-Opcode for reducing sample volume. According to Role-Opcode method, opcodes were categorized to different groups based on their assembly operation role. Bigloo determined 10 different role domains and categorized them into relevant group and used data mining methods like k-nearest neighborhood and SVM for create the model. Pechaz et al[12] gathered malwares from different kinds of family and used Markov Blanket as feature selection method in order to reducing sample volume. Pechaz applied Hidden Markov Model to distinguish malwares from benign files.

In this research, opcode sequences of files have been extracted as file features. Then Markov Blanket method was used to reduce sample size. Finally trained Hidden Markov Model based on selected features has been developed and used to detect malwares from benign files.

II. FEATURE EXTRACTION

Operation Codes are operational part of assembly codes. Because opcodes are significant by themselves to detect

malwares[7], we can consider a computer program as finite sequence of opcodes. The sequence of opcodes is illustrated by p where $p = (o_1, o_2, o_3, \dots, o_{l-1}, o_l)$ in which l is the number of opcodes in the computer program. We use opcode sequences in 2, 3 and 4 gram length for demonstrating files. Figure 1 shows the assembly code related to an executable file. opcode sequences of length 2 that can be generated from this sample are $s_1 = (push, call), s_2 = (call, pop), s_3 = (pop, test), s_4 = (test, jnz), s_5 = (jnz, push), s_6 = (push, call), and\ s_7 = (call, pop)$.

```
push esi
call SUB_L0100D36F
pop ecx
test eax,eax
jnz L0100D281
push 0000001Ch
call SUB_L0100D330
pop ecx
```

Figure 1. Assembly code example

III. FEATURE SELECTION

Because of large set of features that were extracted from files, Markov Blanket method was our candidate for feature selection and reduction of sample size. Markov blanket method used normalized mutual information value for selecting features[13]. First, we calculate entropy of each feature (1).

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \qquad (1)$$

Entropy is a criterion for defining disordering rate. Then we calculate mutual information of each feature in the predefined class (2).

$$I(X,Y) = H(X) - H(X|Y) \qquad (2)$$

Mutual information is used for measuring the amount of information that each feature contains about the class or another feature. Mutual information would be in high value if the conditional entropy $H(X|Y)$ is low. If we categorize values of feature into specific ranges so that much less value for conditional entropy is obtained, then it might have high value for mutual information for that class. To achieve this, Best Split Algorithm[14] is being used. Best Spit algorithm is demonstrated in figure 2.

Because features with high mutual information values have upper chance for being selected as best features, mutual information values should be normalized with related entropy[13]. For this reason, we use symmetric uncertainty method.

We illustrate normalized value of feature and class by symbol $SU_{i,c}$ and normalized value of features i and j by symbol $SU_{i,j}$.

```
input   V_list      // list of values of feature F
output  bestSplit   // best splitter for a feature

1  begin
2  V'_list = get distinct values (V_list)
3  order V'_list in Ascending values
4  v_i = getFirstElement(V'_list)
5  bestSplit = find Conditional Entropy using v_i
6  do begin
7     v_j = getNextElement (V'_list , v_i);
8     if (v_j <> Null)
9        mid = (v_i + v_j)/2;
10       c = find Conditional Entropy using mid;
11       bestSplit = Min(bestSplit, c);
12    end if
13    v_i = v_j
14    end until (v_i == Null)
15 end.
```

Figure 2. Best split algorithm

Definition 1- Approximate Markov Blanket: For the two features F_i and F_j $(i \neq j)$, F_i formed approximate markov blanket for F_j if and only if $SU_{i,c} \geq SU_{j,c}$ and $SU_{i,j} \geq SU_{j,c}$. We compute $SU_{i,c}$ for each feature of i. Then all features are arranged in descending order based on $SU_{i,c}$ to create set S. Finally, all features that have Approximate Markov blanket are removed. this process is demonstrated in figure 3.

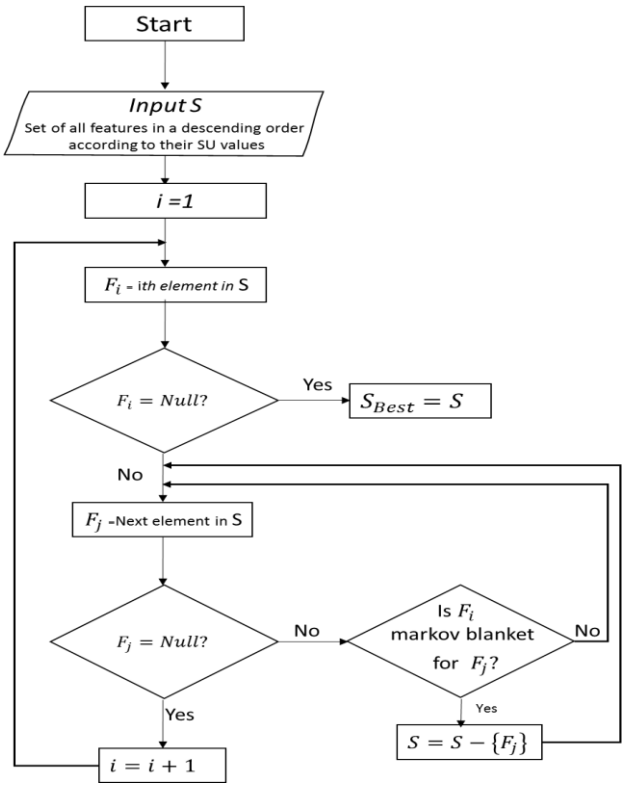

Figure 3. Feature selection process

IV. MODEL CREATION

After all files were presented with their selected features, all selected features in each malware group were appended together to create a long length sequence as training sequence. We trained one Hidden Markov Model with each training sequence for each malware group. To train Hidden Markov Model, we have used Machine Learning toolbox of MATLAB (2015)[1]. Finally, we tested our reserved files saved as test files with trained model.

V. VALIDATION METHODS

In this research, three methods have been chosen for validating results: precision, sensitivity and accuracy.
TP is the number of malwares correctly classified and FP is the number of benign files that wrongly determined as malwares.
TN is the number of benign files correctly classified that is negative. Finally, FN is the number of malwares that wrongly classified as benign files.
Precision measures the performance of used method for determining positive class (3).

$$precision = \frac{TP}{TP + FP} \quad (3)$$

Sensitivity measures the number of positive instances classified correctly (4).

$$Sensitivity = \frac{TP}{TP + FN} \quad (4)$$

Accuracy measures the total number of correct prediction in total number of instances (5).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (5)$$

In malware detection process, sensitivity is more important because a wrong detection of malware is more dangerous than a wrong detection of a benign file.

VI. EXPERIMENTS AND RESULTS

Five groups of malware family that have been selected in this research were Worms, Backdoors, Trojan horses, Viruses and Rootkits. All malware files were collected from VxHeaven[2]. Benign files gathered from three common operating systems, Microsoft windows 7 and 8 and Linux Ubuntu. For assuring that each file is labeled correctly within each malware group, all files were scanned with Norton Internet Security (2015)[3]. To make our work more accurate, in current research we used k-fold cross validation method. According to [15] we considered 10 for k. Then all files in both groups of malware and benign were disassembled with IDA pro[4]. Operation codes were collected from Intel web site[5]. All files were saved as sequence of opcodes. Table 1 shows number, minimum, maximum, average and total volume of selected files.

TABLE I. NUMBER, MINIMUM, MAXIMUM, AVERAGE AND TOTAL VOLUME OF SELECTED FILES

class	Number of files	Min size	Max size	Average size	Total size of selected files
backdoors	95	92.9 KB	757 KB	280.25 KB	26 MB
rootkits	451	6.85 KB	185 KB	15.89 KB	7 MB
Trojan horses	235	18.9 KB	797 KB	113.29 KB	26 MB
viruses	165	29.8 KB	1.01 MB	161.35 KB	26 MB
worms	261	19.3 KB	4.57 MB	102 KB	26 MB
Benign file (non-rootkit class)	128	62.5 KB	705 KB	208 KB	26 MB
Benign file (rootkit class)	66	62.5 KB	199 KB	108.6 KB	26 MB

Diagram 1 demonstrates number of extracted features from each malware group. It's obvious that increase in features length results in increased number of extracted features.

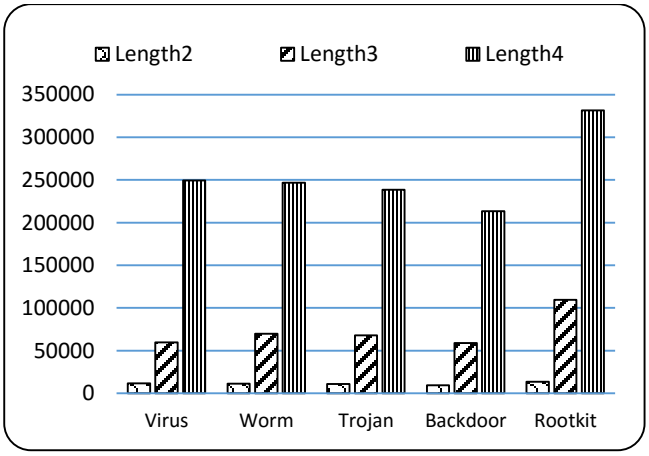

Diagram 1. Number of extracted features from each group of malware

Next, best features were selected from preliminary features with Markov Blanket Algorithm. Diagram 2 illustrates selected features in each malware group. Comparison of diagrams shows that for features with length 2, number of features have been reduced about 99.66%. Also for features with length 3, obviously the number of features that have been delimited is about 99.9%. This reduction is about 99.96% for features with length 4.

[1] http://*www.mathworks.com*
[2] http://*www.vxheaven.com*
[3] *http://us.norton.com/internet-security*
[4] *htpp://www.hex-rays.com*
[5] *http://Ref.x86asm.net/coder32.html*

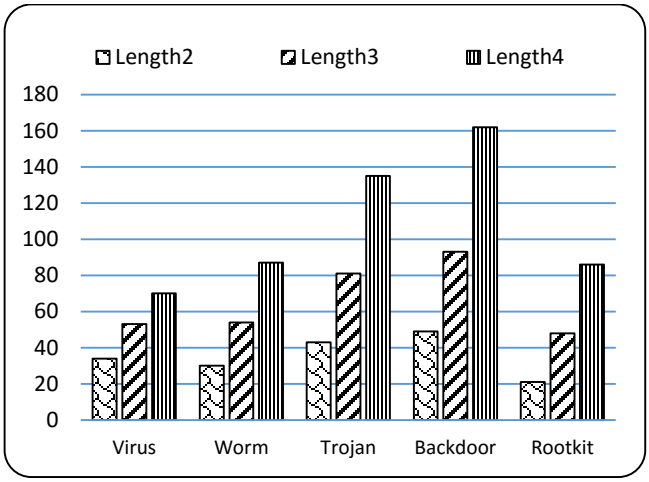

Diagram 2. Number of selected features from each group of malware

Next, we created a sequence of selected features from training set in each group. Diagram 3 demonstrates the final length of training sequence for each group of malwares.

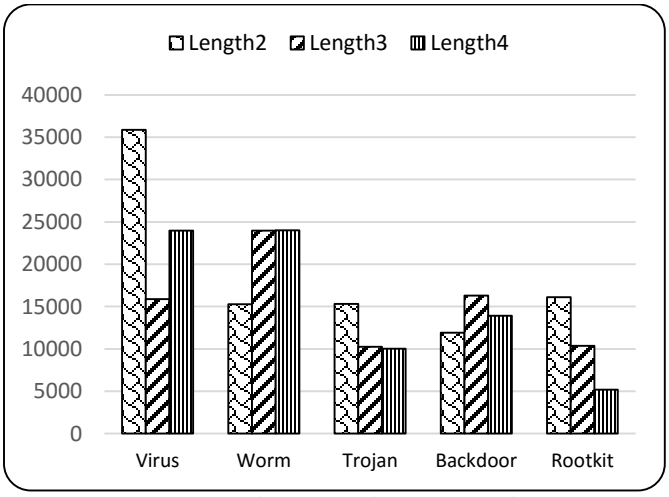

Diagram 3. Length of training set for each group of malware

It's definitely clear that growth in features length does not necessarily results in reduction of training sequence length. Probably it's because of the existence of one major feature with high frequency in file sequence. After the creation of training sequence, we developed one Hidden Markov Model for each malware group. According to our experiments, using HMM with 10 states contained better results. We used a computer system with Intel core i3 2.4 GHz processor and 8 GB RAM and MATLAB software for training HMM. Each trained HMM was evaluated with corresponding malware group test series of files and all values for precision, sensitivity and accuracy were registered. Diagrams 4-8 illustrate developed HMMs validation results.

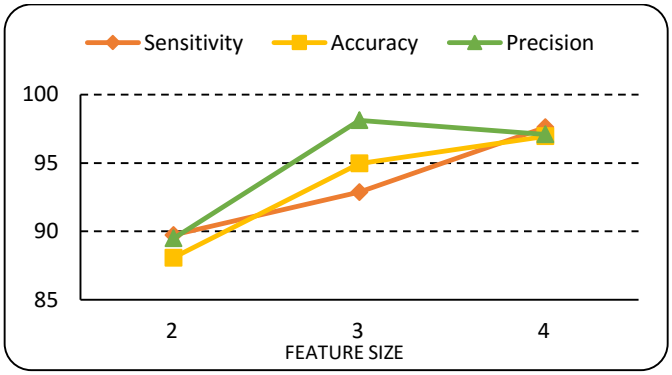

Diagram 4. Sensitivity, accuracy and precision for virus

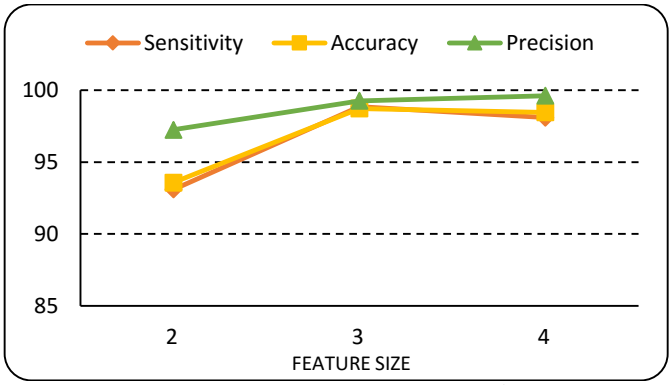

Diagram 5. Sensitivity, accuracy and precision for worm

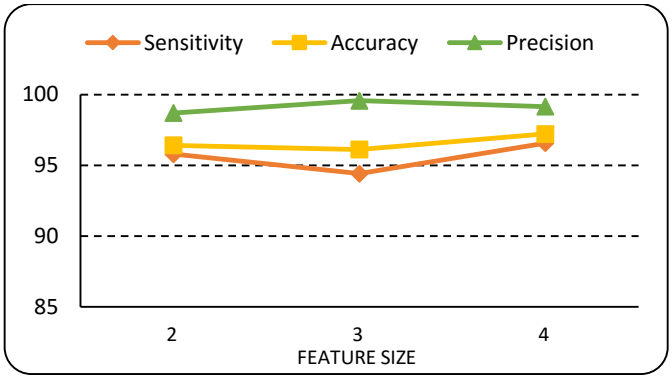

Diagram 6. Sensitivity, accuracy and precision for trojan

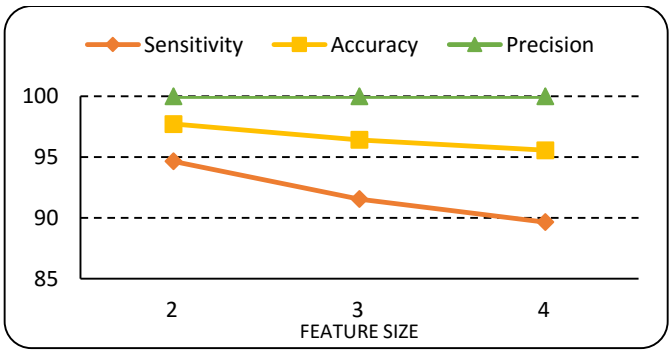

Diagram 7. Sensitivity, accuracy and precision for backdoor

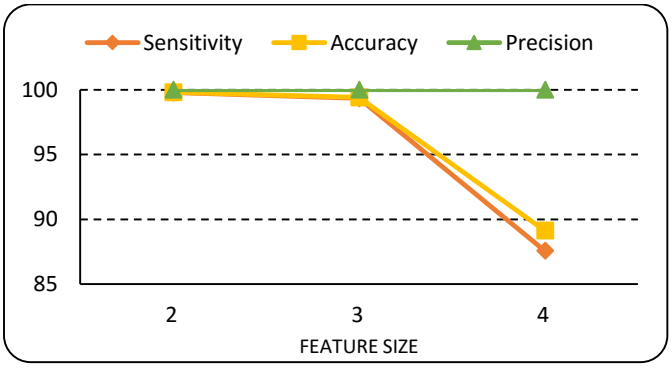

Diagram 8. Sensitivity, accuracy and precision for rootkit

Comparing the results of the current research with similar research done by Pechaz[12] shows that we had an effective progress in results. Diagrams 9-11 demonstrate current research results and previous work conducted by Pechaz[12]. Diagram 9 shows that sensitivity progressed in all malware groups except backdoors. Diagram 10 illustrates that we can detect more files accurately comparing to the last work. Generally, in this research we can improve accuracy about 21% and sensitivity about 24% in comparison with the last similar work.

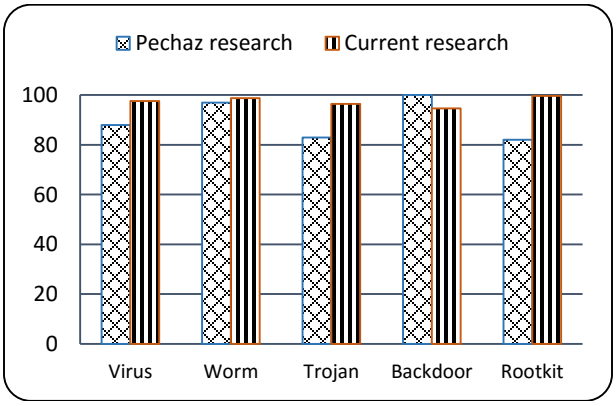

Diagram 9. Comparison of results in terms of sensitivity

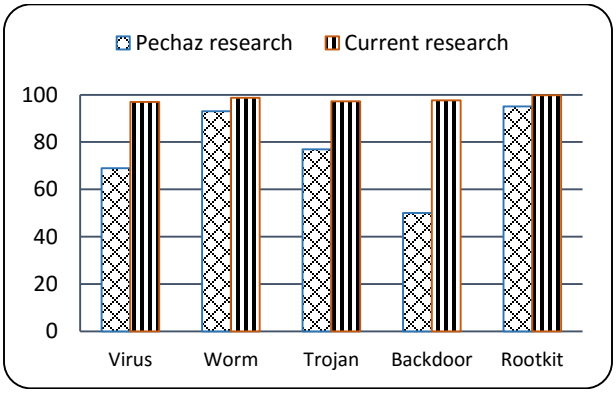

Diagram 10. Comparison of results in terms of accuracy

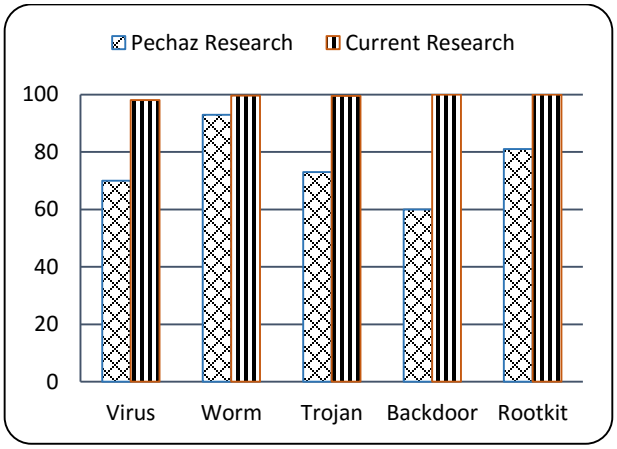

Diagram 11. Comparison of results in terms of precision

VII. CONCLUSION

In this research, we introduce a new heuristic method for malware detection. Todays, malwares categorized into number of family based on its behavior and type of attack and their effects on computer or network of computers.

In this research five groups of malware family have been selected as sample domain. First of all, all files are disassembled into their assembly codes and then pure operation codes (without any information about operands or registers) are extracted from them. Then sequence of opcodes are created in length 2, 3 and 4. Markov Blanket algorithm is used as selection feature filter to reduce feature numbers. Markov blanket reduced size of sample about 99%.

Eventually, for each group of malware, one special Hidden Markov Model was developed and trained. Trained HMM showed good practical functionality for detecting malwares from benign files.

Future works could be done on malware programs and detecting malwares very accurately based on their functions as computer programs.

References

1. M. Siddiqui, M.C. Wang, and J. Lee, *Data mining methods for malware detection using instruction sequences*, in *Proceedings of the 26th IASTED International Conference on Artificial Intelligence and Applications*. 2008, ACTA Press: Innsbruck, Austria. p. 358-363.
2. E. Skoudis and L. Zeltser, *Malware: Fighting Malicious Code*. 2003: Prentice Hall PTR.
3. M. Alazab, *Profiling and classifying the behavior of malicious codes.* Journal of Systems and Software, 2015. **100**(0): p. 91-102.
4. Z. Bazrafshan, H. Hashemi, S.M.H. Fard, and A. Hamzeh. *A survey on heuristic malware detection techniques*. in *Information and Knowledge Technology (IKT), 2013 5th Conference on*. 2013.

5. S. Venkatraman. *Autonomic context-dependent architecture for malware detection*. in *Proceedings of*. 2009.
6. G. Jacob, H. Debar, and E. Filiol, *Behavioral detection of malware: from a survey towards an established taxonomy*. Journal in Computer Virology, 2008. **4**(3): p. 251-266.
7. D. Bilar, *Opcodes as predictor for malware*. Int. J. Electron. Secur. Digit. Forensic, 2007. **1**(2): p. 156-168.
8. I. Santos, et al., *Idea: opcode-sequence-based malware detection*, in *Proceedings of the Second international conference on Engineering Secure Software and Systems*. 2010, Springer-Verlag: Pisa, Italy. p. 35-43.
9. I. Santos, F. Brezo, X. Ugarte-Pedrero, and P.G. Bringas, *Opcode sequences as representation of executables for data-mining-based unknown malware detection.* Information Sciences, 2013. **231**: p. 64-82.
10. M. Zolotukhin and T. Hamalainen. *Detection of zero-day malware based on the analysis of opcode sequences*. in *Consumer Communications and Networking Conference (CCNC), 2014 IEEE 11th*. 2014. IEEE.
11. Z. Ghezelbigloo and M. VafaeiJahan. *Role-opcode vs. opcode: The new method in computer malware detection*. in *Technology, Communication and Knowledge (ICTCK), 2014 International Congress on*. 2014.
12. B. Pechaz, M.V. Jahan, and M. Jalali, *Malware Detection Using Hidden Markov Model based on Markov Blanket Feature Selection Method*, in *First International Congress on Technology, Communication and Knowledge (ICTCK 2014)*. 2014: Mashhad, Iran. p. 26-27.
13. L. Yu and H. Liu, *Efficient Feature Selection via Analysis of Relevance and Redundancy*. J. Mach. Learn. Res., 2004. **5**: p. 1205-1224.
14. P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)*. 2005: Addison-Wesley Longman Publishing Co., Inc.
15. R. Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*, in *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*. 1995, Morgan Kaufmann Publishers Inc.: Montreal, Quebec, Canada. p. 1137-1143.