Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2016, Article ID 8069672, 9 pages
http://dx.doi.org/10.1155/2016/8069672

Research Article

A Data Mining Classification Approach for Behavioral Malware Detection

Monire Norouzi,[1] Alireza Souri,[2] and Majid Samad Zamini[3]

[1] *Young Researchers and Elite Club, Islamic Azad University, Hadishahr Branch, Hadishahr, Iran*
[2] *Department of Computer Engineering, Islamic Azad University, Hadishahr Branch, Hadishahr, Iran*
[3] *Department of Computer Engineering, Islamic Azad University, Sardroud Branch, Sardroud, Iran*

Correspondence should be addressed to Alireza Souri; alirezasouri.research@gmail.com

Received 29 November 2015; Revised 21 June 2016; Accepted 28 June 2016

Academic Editor: Zhiyong Xu

Data mining techniques have numerous applications in malware detection. Classification method is one of the most popular data mining techniques. In this paper we present a data mining classification approach to detect malware behavior. We proposed different classification methods in order to detect malware based on the feature and behavior of each malware. A dynamic analysis method has been presented for identifying the malware features. A suggested program has been presented for converting a malware behavior executive history XML file to a suitable WEKA tool input. To illustrate the performance efficiency as well as training data and test, we apply the proposed approaches to a real case study data set using WEKA tool. The evaluation results demonstrated the availability of the proposed data mining approach. Also our proposed data mining approach is more efficient for detecting malware and behavioral classification of malware can be useful to detect malware in a behavioral antivirus.

1. Introduction

Malicious code is one of the serious threats on the internet platform that is called malware [1]. Malware is known as a malicious application that has been obviously considered to damage the networks and computers [2]. The malware detection design depends on a signature database [3, 4]. For example, a file can be examined with comparison of its bytes using signatures database. If there is an equal specification in the bytes, the suspicious file will be recognized as a malicious file [5, 6]. Some subjects concentrate the signature-based malware detection less than dependable entirely which cannot handle the dynamic modification of malware behavior and cannot identify the hidden malware. In contrast, the behavior based malware detection can find the real behavior of a malicious file [7, 8].

The data mining objectives contain refining advertising abilities, irregular patterns detection, and the upcoming based experiences prediction [9] which can be influenced to identify the suspicious programs which have a destructive content for computer systems such as Virus, Worm, and Trojan [10]. The malware word is assigned to [11, 12] as a destructive file. Data mining techniques rely on data sets that contain some individual configurations for the malicious files and benign software to construct the classification methods for malware detection [13, 14].

Because of the growing malware in the technology, the knowledge of unknown malware protection is an essential topic in the malware detection according to the machine learning methods. Generally, the data mining approaches specified both malicious executable and benign software programs as set of malware programs in the wild [13, 15, 16]. Usually, the data mining algorithms can be categorized into two various forms: supervised and unsupervised learning procedures. The supervised learning methods are called classification algorithms that are needed to the exercise for data set [13, 17]. In contrast, the unsupervised learning methods are called clustering algorithms that are attempted to evaluate organizing data into different clusters [18, 19].

Usually, the malware programs are classified into some parts such as Worm, Virus, Trojan, Spyware, Backdoor, and Rootkit [10, 20–22]. The base of typical and traditional

approaches to identify the malware is using signature-based techniques. In recent years, the disappointment of old methods in unrecognized malware detection or polymorphic malicious files exasperated researchers and they attempted to present more dependable approaches for malware detection with behavior of the malware [23]. The procedure of detecting and finding malware has been done by two types of analysis: static analysis and dynamic analysis. In the software analyzing methods, analyzing without running the codes is called static analysis which can detect the malicious code and put it in one of the available collections based on different learning methods [24]. In the static analysis, malicious files and malware are detected based on binary codes. The main disadvantage of static analysis is unavailability of the source codes of the program. It is valuable to declare that extracting binary codes is a relatively complex and complicated work.

In contrast, the dynamic analysis detects malicious codes according to the runtime behavior [10]. The runtime code analyzing is called dynamic analysis which also denoted behavior analyzing and observing behavior and system interaction [23]. Dynamic analysis mechanism needs to execute the infested files in a virtual machine [21]. Dynamic analysis can be used with classification and clustering methods to navigate the increasing volume and range of malware. The malware classification methods help to assign unknown malware to recognized families [7, 20]. Therefore, malware classification is used to filter unknown cases and thus decreases the costs of analysis [8, 25–29].

The contributions of this paper are included as follows:

(i) Proposing a behavioral analysis mechanism for malware detection.

(ii) Presenting a converter program for transforming a malware behavior executive history XML file to a suitable WEKA input.

(iii) Discussing some classification methods on a real case study of malware.

(iv) Comparing the experimental results such as Correctly Classified Instances, mean absolute error, and accurate optimistic ratio in the real data set by WEKA tool.

(v) Testing the best classification method based on the important features in the malware detection in order to develop a behavioral antivirus.

The structure of this paper is organized as follows: in Section 2 we have discussed some backgrounds and related works in the malware detection and data mining techniques. Section 3 depicts the malware behavioral analysis. In this section we propose a new approach for analyzing the malware behavior and translating the malicious files to data mining files by using a real case study. Also this section describes the classification and prediction approaches using data mining platform. Then, we apply some of the popular classification methods on our real case study using WEKA tool. The evaluation and experimental results are reported in Section 4. Section 5 concludes discussion and the future work.

2. Related Works

This section discusses a brief background and some related works for malware detection in data mining methods. Firstly, we review data mining approach briefly based on classification methods in malware and other systems. Recently, some researchers presented the different approaches in malware analysis. Schultz et al. [30] proposed a data mining method to recognize the new malicious files in runtime execution. Their method was based on three types of DLL calls such as the list of DLLs used by the binary; the list of DLL function calls; and number of different system calls used within each DLL. Also they examine byte orders extracted from the hex-dump (a hexadecimal schema of computer data) of an executable file using signature methods. The main structure of this method is based on Naive-Bayes (NB) algorithm. They compared the experimental results by traditional signature-based methods.

Also Kolter and Maloof [31] presented a data mining approach and n-gram analysis to identify malicious executable files based on signature approach. They presented a hex-dump utility for translating each executable file to hexadecimal code in an ASCII format. Their main data set consisted of the clean programs and the malicious programs. They analyzed the proposed approach by some popular classification methods such as instance-based learner, TFIDF, Naive-Bayes, support vector machines, decision tree, boosted Naive-Bayes, and boosted decision tree. In the other research, Siddiqui et al. [32] proposed data mining techniques for recognition some malware programs such as Worms. They considered variable length instruction sequence for their approach. Their main data set includes some Windows files and Worms. As experimental results, sequence reduction was executed, 97% of the sequences were removed, and random forest decision tree model was performed slightly better than the others.

Also some research work presented the data mining methodologies for different approach. For example, in [33] the researchers presented various data mining methods that have been developed for cancer diagnosis. Consequently, this research focused on captivating the clinical information which can be found without surgery to exchange the pathology report. They used to discover the association between the clinical information and the pathology report in order to maintain lung cancer pathologic staging diagnosis using data mining techniques. In the other research [1, 34], the authors proposed a data mining approach to analyze the students careers. Their approach is based on clustering and sequential methods with the aim of categorizing strategies for refining the performance of the exams scheduling and students. They analyzed a real case study using K-mean cluster techniques in WEKA tool. Likewise [26] presented a new data mining method for the problem of detecting the phishing websites using a developed associative classification method called multilabel classifier that generates multiple labels rules. They analyzed the experimental results by various patterns in WEKA software. Also the researchers in [35] analyzed the several decision tree models to classify patients of the hospital surveillance data as a real case study. The experimental results of their analysis showed that their

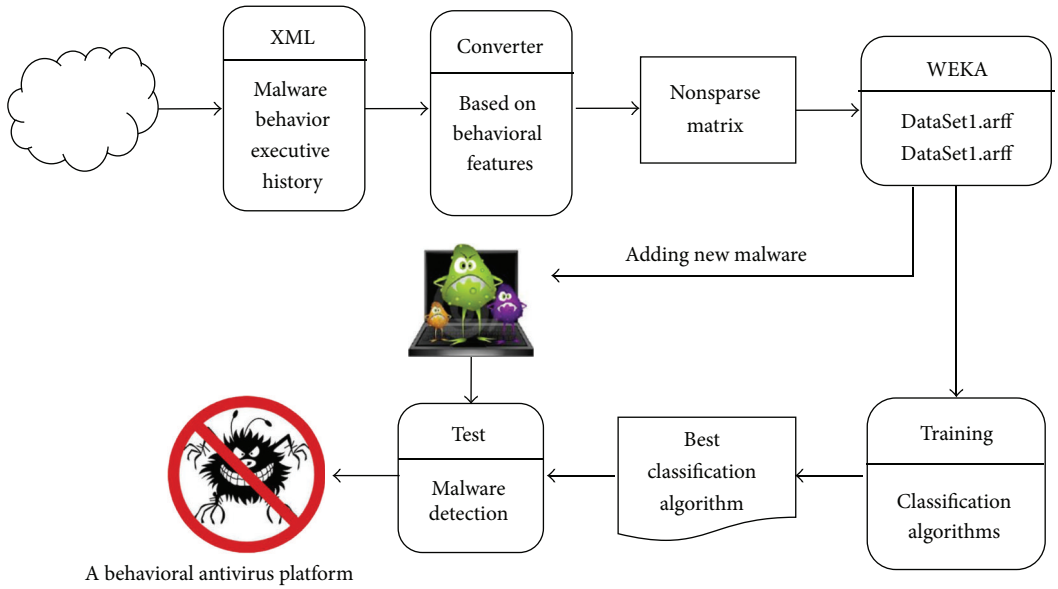

Figure 1: The behavioral analysis of malware detection mechanism.

approach improved identical dissemination of instances in each class. Other related work [36] used a neurofuzzy data mining approach for classification of generalized bell-shaped membership functions. They applied the proposed technique to ten real standard data sets from the UCI machine learning repository for classification using Kappa statistic. They simulated proposed technique in MATLAB. Also some researches focused on the other approaches that consist of the host behavior classification methods [37–40]. For example, [29] presented a novel managed discretization technique for analyzing multivariate time series which uses frequent temporal patterns as features for classification of time chain for geared near improvement of classification correctness. This paper used temporal abstraction classification approach and time intervals mining for the presented multivariate time series. Also [38] presented novel Artificial Neural Networks (ANN) based mechanism for discovering the computer Worms based on the behavioral computer events. According to estimation of different parameters of the infected computers, the ANN, decision tree, and K-nearest neighbors classification techniques are compared. The other research is [41] where the authors presented computer measurements extracted mechanism for identifying unknown computer Worm activity in the operating system using support vector approaches. This paper separates a series of trials to check the new technique by retaining several computer configuration activities.

To the best of our knowledge, there is no any approach that analyzes the malware behavior in data mining platform exactly and also there is no any approach to convert malware behavior XML executive history file to a suitable WEKA tool input. Our approach can be used in base of a behavioral antivirus. For improving this defect, we present a new approach to translate a malicious file to the data mining platform. Then we consider some classification methods for evaluating our approach based on malware behavior.

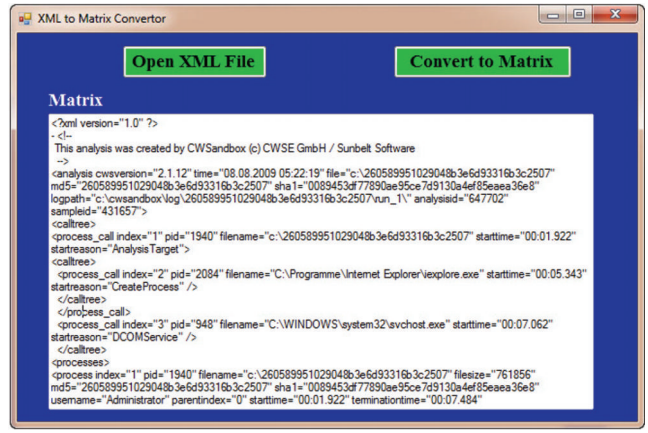

Figure 2: A snapshot of XML convertor to nonsparse matrix.

3. Malware Behavior Analysis

In this section, we proposed a malware behavioral analysis mechanism as shown in Figure 1. In this mechanism, a XML file of malware behavior executive history will be converted to a nonsparse matrix using a suggested application. This application is produced with VB.Net language. Figure 2 shows a snapshot of XML convertor to a nonsparse matrix using our suggested application. The procedure of converting each XML file to a suitable WEKA input includes two elements: the number of library file calls which are attacked by malware and their volume. For example, in Box 1 the XML library file ntdll.dll has been called 16 times by the malware which are between (0, 2). Then, we translate this matrix to WEKA input data set. The training methods will be proceeded by some classification algorithms. Each classification that has best performance will be chosen for test platform by new data set

```
<?xml version="1.0" ?>
- <!- -
This analysis was created by CWSandbox (c) CWSE GmbH/Sunbelt Software
- ->
<analysis cwsversion="2.1.12" time="08.08.2009 05:22:19"
file="c:\260589951029048b3e6d93316b3c2507"
md5="260589951029048b3e6d93316b3c2507"
sha1="0089453df77890ae95ce7d9130a4ef85eaea36e8"
logpath="c:\cwsandbox\log\260589951029048b3e6d93316b3c2507\run_1\"
analysisid="647702" sampleid="431657">
<calltree>
<process_call index="1" pid="1940"
filename="c:\260589951029048b3e6d93316b3c2507" starttime="00:01.922"
startreason="AnalysisTarget">
<calltree>
  <process_call index="2" pid="2084" filename="C:\Programme\Internet
Explorer\iexplore.exe" starttime="00:05.343" startreason="CreateProcess" />
  </calltree>
  </process_call>
  <process_call index="3" pid="948"
filename="C:\WINDOWS\system32\svchost.exe" starttime="00:07.062"
startreason="DCOMService" />
  </calltree>
<processes>
<process index="1" pid="1940"
filename="c:\260589951029048b3e6d93316b3c2507" filesize="761856"
md5="260589951029048b3e6d93316b3c2507"
sha1="0089453df77890ae95ce7d9130a4ef85eaea36e8" username="Administrator"
parentindex="0" starttime="00:01.922" terminationtime="00:07.484"
startreason="AnalysisTarget" terminationreason="NormalTermination"
executionstatus="OK" applicationtype="Win32Application">
<dll_handling_section>
  <load_image filename="c:\260589951029048b3e6d93316b3c2507" successful="1"
address="$400000" end_address="$4C1000" size="790528" />
  <load_dll filename="C:\WINDOWS\system32\ntdll.dll" successful="1"
address="$7C910000" end_address="$7C9C9000" size="757760" quantity="16"/>
  <load_dll filename="C:\WINDOWS\system32\kernel32.dll" successful="1"
address="$7C800000" end_address="$7C908000" size="1081344" quantity="2" />
  <load_dll filename="C:\WINDOWS\system32\gdi32.dll" successful="1"
address="$77EF0000" end_address="$77F39000" size="299008" quantity="2" />
  <load_dll filename="C:\WINDOWS\system32\USER32.dll" successful="1"
address="$7E360000" end_address="$7E3F1000" size="593920" quantity="2" />
  </dll_handling_section>
<filesystem_section>
```

Box 1: A sample part of XML file contains a malware behavior.

malware. Finally, this procedure can be used for developing a behavioral antivirus. For describing the behavioral model of malware we should download the XML file which is available in PIL (http://dws.informatik.uni-mannheim.de) as an XML file [38–40]. We use 7155 XML files as data set 1 and data set 2. Our first data set contains 4024 XML file and data set 2 has 3131 XML files too. Data set 1 has 89 properties and data set 2 has 91 properties for each malware.

Then, we convert this XML file to a nonsparse matrix by using our suggested program. The nonsparse matrix includes two numbers: the first number shows the number of properties and the second number shows their importance. The first row of this matrix is shown as follows:

(0 1.068, 2 0.534, 8 0.534, 11 0.534, 12 0.534, 23 0.534, 32 0.534, 33 0.534, 35 0.534, 36 0.534, 40 0.534, 45 1.068, 46 1.603, 47 1.068, 48 1.068, 49 1.068, 50 1.068, 51 1.068, 52 1.068, 53 1.068, 54 2.137, 55 1.068, 56 0.534, 57 1.068, 58 2.137, 61 0.534, 62 0.534, 63 2.137, 65 0.534, 66 0.534, 73 1.603, 83 22, 84 16, 85 4, 86 8, 87 6, 88 T1).

The last number of this row is 88 T1 that shows the kind of malware.

Finally we analyze the executive history of malware in WEKA environment. The malware executive history can be developed by some applications such as SandBox tool and virtual machine for safe execution of malware in computer

```
@RELATION TEST              file name
@ATTRIBUTE dll1             numeric          property
@ATTRIBUTE dll2             numeric          property
@ATTRIBUTE dll3             numeric          property
@ATTRIBUTE dll4             numeric          property
. . . . . .                                  property
@ATTRIBUTE param88          numeric          property
@ATTRIBUTE class              Answer - property
@DATA
0 1.068, 2 0.534, 8 0.534, 11 0.534, 12 0.534, 23 0.534, 32 0.534, 33 0.534, 35 . . . . . ..
```

Box 2: An example of standard form for WEKA input.

systems and preventing malware spread [28, 38–41]. The XML file includes useful information such as system library files calls, creating, searching, and change of files, modifying registry, main processes information, creating the mutex (a mutex is an application object which permits the multiple program threads to share the same resource), modifying virtual memory, sending email, registry operations, and switches communications. By using the suggested program all of the information is read and saved as a nonsparse matrix.

Now, the matrix has been converted to a standard form of WEKA tool input as .arff file for data set 1 and data set 2. This standard form is shown in Box 2.

3.1. Classification and Prediction Approaches. This section describes the classification methods in two real case studies as data set 1 and data set 2. At first, we analyze the data mining result on data set 1 and data set 2 by WEKA classification algorithms. For specifying the performance of classification methods in WEKA, we describe some effective features briefly [27]. The Correctly Classified Instances (CCI) depict the test cases percentages that were correctly classified. Also the Incorrectly Classified Instances (ICI) represent the test cases percentages that were incorrectly classified.

The relative absolute error (RAE) is qualified to a simple predictor error which is objective for the typical real values. In the RAE, the error is only the total absolute error rather than the total squared error.

Definition 1. A relative absolute error is a 3-tuple $\mathrm{RAE}_i = (F_{(i,j)}, V_j, \overline{\mathsf{T}})$ in formula (1), where $F_{(i,j)}$ is the value predicted by the individual program i for sample case j (out of k sample cases); V_j is the objective value for sample case j; and $\overline{\mathsf{T}}$ is given by the following formulas

$$\mathrm{RAE}_i = \frac{\sum_{j=1}^{k} \left|F_{(i,j)} - V_j\right|}{\sum_{j=1}^{k} \left|V_j - \overline{\mathsf{T}}\right|}, \tag{1}$$

$$\overline{\mathsf{T}} = \frac{1}{k}\sum_{j=1}^{k} V_j. \tag{2}$$

Also the mean absolute error (MAE) shows the mean average greatness of the errors in a set of predictions, without allowing for their course. The MAE depicts the correctness of incessant variables in prediction procedure. The MAE specifies and verifies an average on the absolute values between forecast and the corresponding statement. The MAE is a linear score which means that all the individual differences are weighted equally in the average [42–44].

Definition 2. A mean absolute error is a 2-tuple $\mathrm{MAE}_i = (P_i, T_i)$ in formula (3), where P_i is the prediction of value and T_i is the true value. This feature specifies the average error in the classification procedure in

$$\mathrm{MAE}_i = \frac{1}{k}\sum_{j=1}^{k} \left|P_j - T_j\right|. \tag{3}$$

Also we can measure the classifiers proficiency using a true optimistic ratio (TOR), where NC is the number of correctly detected malware programs and NI is the number of incorrectly detected malware programs in (4). The AOR creates the cost of estimated classification that is significant to setting the cost of malware classification [45]:

$$\mathrm{TOR} = \frac{\mathrm{NC}}{\mathrm{NC} + \mathrm{NI}}. \tag{4}$$

Also there are two error rates for measuring the classification performance. The False Acceptance Rate (FAR) is the ratio of the number of test cases that are incorrectly accepted by a given model to the total number of cases. This means that this ratio shows the percentage of invalid inputs which are incorrectly accepted. The False Rejection Rate (FRR) is the ratio of the number of test cases that are incorrectly rejected by a given model to the total number of cases. This means that this ratio shows the percentage of valid inputs which are incorrectly rejected [46]. By using these factors we can calculate the Total Error Rate (TER) as follows [47]:

$$\mathrm{TER} = \frac{\mathrm{FAR} + \mathrm{FRR}}{\mathrm{NC} + \mathrm{NI}}. \tag{5}$$

In the classification process, we use NaiveBayse, BayseNet, IB1, J48, and classification via regression algorithms. The NaiveBayes and BayesNet are a probabilistic learning algorithms based on supervised learning method which require a small number of training data to estimate the constraints. The IB1 data mining algorithm is based on lazy approaches.

Table 1: The statistical analysis of data set 1 for specified classification methods.

Algorithms	Results							
	Correctly Classified Instances Number, %	Incorrectly Classified Instances Number, %	Mean absolute error	Relative absolute error	Kappa statistic	Root mean squared error	Root relative squared error	Total number of instances
NaiveBayes	1107, 27.5099%	2917, 72.4901%	0.0069	90.0871%	0.2526	0.0754	122.8107%	4024
BayesNet	2662, 66.1531%	1362, 33.8469%	0.0032	42.4047%	0.5979	0.0479	78.1282%	4024
IB1	2802, 69.6322%	1222, 30.3678%	0.0028	37.2325%	0.6199	0.0533	86.8274%	4024
J48	2908, 72.2664%	1116, 27.7336%	0.0032	41.6312%	0.6379	0.0454	73.9957%	4024
Regression	3051, 75.8201%	973, 24.1799%	0.0011	21.0201%	0.6859	0.0392	63.9686%	4024
SVM	2251, 64.1571%	1773, 35.8429%	0.0039	42.0019%	0.5743	0.4758	84.9596%	4024

Also J48 data mining algorithm is based on decision tree methods. Finally, classification via regression algorithm is based on Meta approach that is the new approach in data mining methods. In other words regression analysis is a statistical method which is used to achieve data analysis. Regression is applied with correlation analysis usually. The correlation analysis evaluates the association degree between two quantitative data sets [37]. For example, Figure 3 shows the classification result of NaiveBayse algorithm in WEKA tool. The following section describes the experimental results of classification algorithms in WEKA. Some effective features such as Correctly Classified Instances, Incorrectly Classified Instances, mean absolute error, and relative absolute error are compared with each other in order to achieve the best classification algorithm for developing a behavioral antivirus.

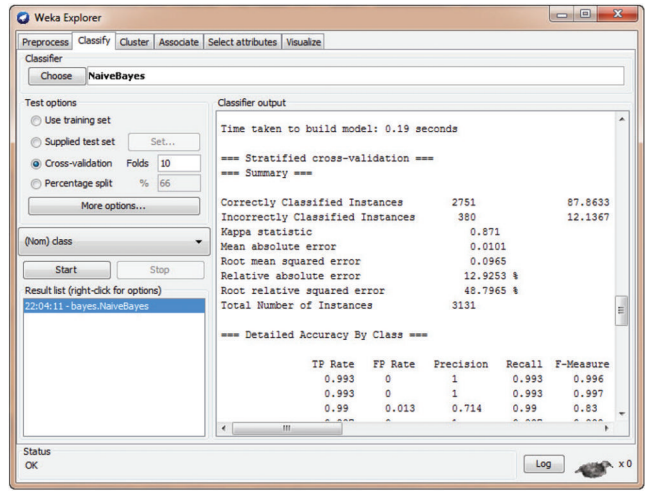

Figure 3: The snapshot of NaiveBayse classification algorithm in WEKA.

4. Experimental Results and Discussion

In this section, we implemented our approach using WEKA tool. We use a system by Intel Core i3 2.13 GHz CPU, 4 GB RAM, for the classification methods. This analysis has been done by some classification algorithms such as NaiveBayse, BayseNet, IB1, J48, System Vector Machine (SVM), and logistic regression method. We compared performance of classification methods in two malware data sets.

In Tables 1 and 2, the statistical analysis of data sets 1 and 2 is specified for proposed classification methods. The compared factors in the classification methods are Correctly Classified Instances, Incorrectly Classified Instances, Kappa statistic, mean absolute error, relative absolute error, root mean squared error, and root relative squared error. In this comparison, we show that the classification via regression method has best performance in malware detection. For example, in data set 1, the number of correctly classified malware programs is 3051 from total 4024 malware programs. Also in data set 2, the number of correctly classified malware programs is 3069 from total 3131 malware programs.

According to Tables 1 and 2, the percentage of Correctly Classified Instances of the logistic regression algorithm is higher than the other classification methods in each of data sets 1 and 2. Also the percentage of Incorrectly Classified Instances of the logistic regression algorithm is lower than the other classification methods in each of data sets 1 and 2.

After data mining process, we test a new malware case by the regression classification algorithm. 100 binary malware programs are downloaded from NetLux (http://vxheaven.org/) and we analyzed their behaviors by using CW-Sandbox tool and we get its XML file [38]. Then, we add these 100 malware programs to the new data set and compute the quality of their classification as true optimistic ratio. As we expect, by classification via regression 88 malware programs are detected. So we can use the classification via regression to develop a behavaioral antivirus.

Table 2: The statistical analysis of data set 2 for specified classification methods.

Algorithms	Results							
	Correctly Classified Instances Number, %	Incorrectly Classified Instances Number, %	Mean absolute error	Relative absolute error	Kappa statistic	Root mean squared error	Root relative squared error	Total number of instances
NaiveBayes	2678, 85.5318%	453, 14.4682%	0.012	15.3329%	0.8459	0.1026	51.8792%	3131
BayesNet	2874, 91.7918%	257, 8.2082%	0.0073	9.3575%	0.9127	0.0747	37.7504%	3131
IB1	3028, 96.7103%	103, 3.2897%	0.0027	3.5032%	0.965	0.0524	26.472%	3131
J48	3008, 96.0715%	123, 3.9285%	0.0043	5.5353%	0.9581	0.0527	26.652%	3131
Regression	3069, 98.321%	62, 1.679%	0.0021	2.2102%	0.9578	0.0543	27.4333%	3131
SVM	1698, 54.2319%	1433, 45.7681%	0.0046	5.7993%	0.5011	0.1942	98.1954	3131

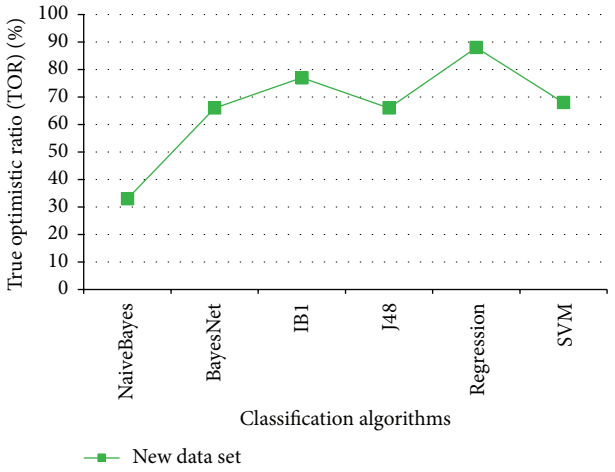

Figure 4: The true optimistic ratio for the classifications test in the new data set.

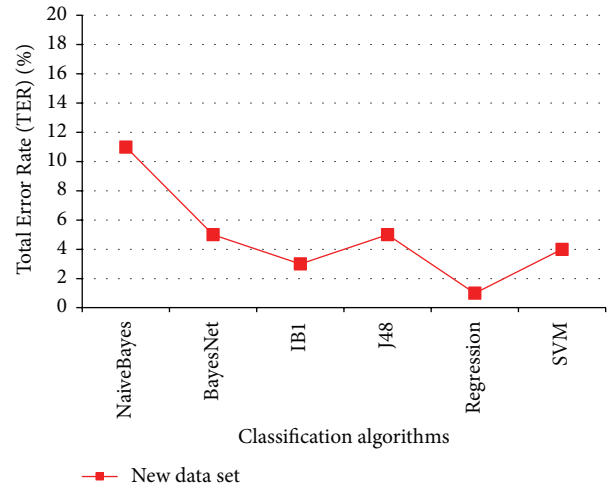

Figure 5: The Total Error Rate (TER) for the classifications test in the new data set.

Figure 4 depicts the true optimistic ratio percentage for malware detection in the new data sets. The true optimistic ratio percentage of regression method is higher than the other classification methods in the new data set.

After testing our new case study by 100 malware programs, Table 3 describes a statistical result for the False Acceptance Rate (FAR) number of cases and the False Rejection Rate (FRR) number of cases. Of course, there are some platforms such as STAC (http://tec.citius.usc.es/stac/) [48] for statistical comparison of the tested algorithms. But we use the WEKA tool for statistical and experimental results for our data sets.

According to Table 3, there is no valid input which is incorrectly rejected using our approach by regression method. Also NaiveBayes method rejected 6 valid inputs incorrectly.

Also in this test case we find one FAR incorrectly accepted as a malware. So, Figure 5 shows the Total Error Rate (TER) for our new test case using our approach by the regression method.

Table 3: The statistical analysis of the FAR and FRR number of cases in the new test case study.

Algorithms	Statistical analysis		
	Number of FAR cases	Number of FRR cases	Total number of instances
NaiveBayes	5	6	100
BayesNet	4	2	100
IB1	2	1	100
J48	3	2	100
Regression	1	0	100
SVM	3	2	100

5. Conclusion and Future Work

In this paper, we proposed a new data mining approach based on classification methodologies for detecting malware behavior. Firstly, a malware behavior executive history XML file is converted to a nonsparse matrix using our suggested application. Then, this matrix was translated to WEKA input data set. To illustrate the performance efficiency, we applied the proposed approaches to a real case study data set using WEKA tool. The training methods proceeded using some classification algorithms such as NaiveBayse, BayseNet, IB1, J48, and regression algorithms. The regression classification method had best performance for classification of malware detection. Also we analyzed the new data set by the regression classification method. The evaluation results demonstrated the availability of the proposed data mining approach. Also our proposed data mining mechanism is more efficient for detecting malware. By notice to the experimental results, classification of malware behavioral features can be a convenient method in developing a behavioral antivirus. In the future work, we will try to develop and analyze a real behavioral antivirus platform based on classification via regression algorithm.

Competing Interests

The authors declare that they have no competing interests.

References

[1] D. B. Ekta Gandotra and S. Sofat, "Malware analysis and classification: a survey," *Journal of Information Security*, vol. 5, pp. 56–64, 2014.

[2] P. Wang and Y.-S. Wang, "Malware behavioural detection and vaccine development by using a support vector model classifier," *Journal of Computer and System Sciences*, vol. 81, no. 6, pp. 1012–1026, 2015.

[3] G. Ollmann, "The evolution of commercial malware development kits and colour-by-numbers custom malware," *Computer Fraud and Security*, vol. 2008, no. 9, pp. 4–7, 2008.

[4] M. Ghiasi, A. Sami, and Z. Salehi, "Dynamic VSA: a framework for malware detection based on register contents," *Engineering Applications of Artificial Intelligence*, vol. 44, pp. 111–122, 2015.

[5] D. Bruschi, L. Martignoni, and M. Monga, "Detecting self-mutating malware using control-flow graph matching," in *Detection of Intrusions and Malware & Vulnerability Assessment*, R. Büschkes and P. Laskov, Eds., vol. 4064, pp. 129–143, Springer, Berlin, Germany, 2006.

[6] M. R. Chouchane and A. Lakhotia, "Using engine signature to detect metamorphic malware," in *Proceedings of the 4th ACM Workshop on Recurring Malcode*, Alexandria, Va, USA, 2006.

[7] N. Kuzurin, A. Shokurov, N. Varnovsky, and V. Zakharov, "On the concept of software obfuscation in computer security," in *Information Security*, J. Garay, A. Lenstra, M. Mambo, and R. Peralta, Eds., vol. 4779, pp. 281–298, Springer, Berlin, Germany, 2007.

[8] M. Christodorescu and S. Jha, "Testing malware detectors," *SIGSOFT Software Engineering Notes*, vol. 29, no. 4, pp. 34–44, 2004.

[9] L. K. Mehedy Masud and B. Thuraisingham, *Data Mining Tools for Malware Detection*, vol. 1, CRC Press, 2012.

[10] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, "A survey on automated dynamic malware-analysis techniques and tools," *ACM Computing Surveys*, vol. 44, pp. 1–42, 2008.

[11] S. P. Monire Norouzi and A. Mahjur, "A new approach for formal behavioral modeling of protection services in antivirus systems," *International Journal in Foundations of Computer Science & Technology*, vol. 4, pp. 57–67, 2014.

[12] A. Safarkhanlou, A. Souri, M. Norouzi, and S. E. H. Sardroud, "Formalizing and verification of an antivirus protection service using model checking," *Procedia Computer Science*, vol. 57, pp. 1324–1331, 2015.

[13] 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*, vol. 231, pp. 64–82, 2013.

[14] N. Abdelhamid, A. Ayesh, and F. Thabtah, "Phishing detection based associative classification data mining," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5948–5959, 2014.

[15] G. Jacob, H. Debar, and E. Filiol, "Behavioral detection of malware: from a survey towards an established taxonomy," *Journal in Computer Virology*, vol. 4, no. 3, pp. 251–266, 2008.

[16] A. Shabtai, R. Moskovitch, Y. Elovici, and C. Glezer, "Detection of malicious code by applying machine learning classifiers on static features: a state-of-the-art survey," *Information Security Technical Report*, vol. 14, no. 1, pp. 16–29, 2009.

[17] S. B. Kotsiantis, "Supervised machine learning: a review of classification techniques," in *Proceedings of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pp. 3–24, IOS Press, 2007.

[18] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[19] L. Chen, Q. Jiang, and S. Wang, "Model-based method for projective clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 7, pp. 1291–1305, 2012.

[20] C. Ravi and R. Manoharan, "Malware detection using Windows Api sequence and machine learning," *International Journal of Computer Applications*, vol. 43, no. 17, pp. 12–16, 2012.

[21] R. Rizwan, G. C. Hazarika, and G. Chetia, "Malware threats and mitigation strategies: a survey," *Journal of Theoretical and Applied Information Technology*, vol. 29, no. 2, pp. 69–73, 2011.

[22] K. Mathur and H. Saroj, "A survey on techniques in detection and analyzing malware executables," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 44, no. 2, 2012.

[23] N. F. Doherty, L. Anastasakis, and H. Fulford, "The information security policy unpacked: a critical study of the content of university policies," *International Journal of Information Management*, vol. 29, no. 6, pp. 449–457, 2009.

[24] G. Tahan, L. Rokach, and Y. Shahar, "Automatic malware detection using common segment analysis and meta-features," *Journal of Machine Learning Research*, vol. 13, pp. 949–979, 2012.

[25] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated classification and analysis of internet malware," in *Recent Advances in Intrusion Detection*, C. Kruegel, R. Lippmann, and A. Clark, Eds., vol. 4637, pp. 178–197, Springer, Berlin, Germany, 2007.

[26] U. Bayer, A. Moser, C. Kruegel, and E. Kirda, "Dynamic analysis of malicious code," *Journal in Computer Virology*, vol. 2, no. 1, pp. 67–77, 2006.

[27] J. Z. Kolter and M. A. Maloof, "Learning to detect and classify malicious executables in the wild," *Journal of Machine Learning Research*, vol. 7, pp. 2721–2744, 2006.

[28] P. Trinius, C. Willems, T. Holz, and K. Rieck, *A Malware Instruction Set for Behavior-Based Analysis*, 2009.

[29] R. Moskovitch and Y. Shahar, "Classification-driven temporal discretization of multivariate time series," *Data Mining and Knowledge Discovery*, vol. 29, no. 4, pp. 871–913, 2015.

[30] M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," in *Proceedings of the IEEE Symposium on Security and Privacy, S&P*, pp. 38–49, Oakland, Calif, USA, 2001.

[31] J. Z. Kolter and M. A. Maloof, "Learning to detect malicious executables in the wild," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*, pp. 470–478, ACM, Seattle, Wash, USA, August 2004.

[32] M. Siddiqui, M. C. Wang, and J. Lee, "Detecting internet worms using data mining techniques," *Journal of Systemics, Cybernetics and Informatics*, vol. 6, pp. 48–53, 2008.

[33] H. Yang and Y.-P. P. Chen, "Data mining in lung cancer pathologic staging diagnosis: correlation between clinical and pathology information," *Expert Systems with Applications*, vol. 42, no. 15-16, pp. 6168–6176, 2015.

[34] R. Campagni, D. Merlini, R. Sprugnoli, and M. C. Verri, "Data mining models for student careers," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5508–5521, 2015.

[35] R. M. Rahman and F. R. Md Hasan, "Using and comparing different decision tree classification techniques for mining ICDDR,B Hospital Surveillance data," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11421–11436, 2011.

[36] S. Ghosh, S. Biswas, D. Sarkar, and P. P. Sarkar, "A novel Neuro-fuzzy classification technique for data mining," *Egyptian Informatics Journal*, vol. 15, no. 3, pp. 129–147, 2014.

[37] R. Moskovitch and Y. Shahar, "Fast time intervals mining using the transitivity of temporal relations," *Knowledge and Information Systems*, vol. 42, no. 1, pp. 21–48, 2015.

[38] D. Stopel, Z. Boger, R. Moskovitch, Y. Shahar, and Y. Elovici, "Application of artificial neural networks techniques to computer worm detection," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '06)*, pp. 2362–2369, July 2006.

[39] D. Stopel, R. Moskovitch, Z. Boger, Y. Shahar, and Y. Elovici, "Using artificial neural networks to detect unknown computer worms," *Neural Computing and Applications*, vol. 18, no. 7, pp. 663–674, 2009.

[40] R. Moskovitch, I. Gus, S. Pluderman et al., "Detection of unknown computer worms activity based on computer behavior using data mining," in *Proceedings of the 1st IEEE Symposium on Computational Intelligence and Data Mining (CIDM '07)*, pp. 202–209, IEEE, Honolulu, Hawaii, USA, April 2007.

[41] N. Nissim, R. Moskovitch, L. Rokach, and Y. Elovici, "Detecting unknown computer worm activity via support vector machines and active learning," *Pattern Analysis and Applications*, vol. 15, no. 4, pp. 459–475, 2012.

[42] N. Karthik, R. Arul, and M. J. H. Prasad, "Modeling of wind turbine power curves using firefly algorithm," in *Power Electronics and Renewable Energy Systems*, C. Kamalakannan, L. P. Suresh, S. S. Dash, and B. K. Panigrahi, Eds., vol. 326, pp. 1407–1414, Springer, New Delhi, India, 2015.

[43] F. Galton, *Finger Prints*, Macmillan and Company, 1892.

[44] B. D. Eugenio and M. Glass, "The kappa statistic: a second look," *Computational Linguistics*, vol. 30, no. 1, pp. 95–101, 2004.

[45] M. N. Mohammad, N. Sulaiman, and O. A. Muhsin, "A novel intrusion detection system by using intelligent data mining in weka environment," *Procedia Computer Science*, vol. 3, pp. 1237–1242, 2011.

[46] M. Kantardzic, *Data Mining: Concepts, Models, Methods and Algorithms*, John Wiley & Sons, 2002.

[47] M. Deshmukh and M. N. K. Prasad, "Partial segmentation and matching technique for iris recognition," in *Computational Intelligence in Data Mining—Volume 1*, L. C. Jain, H. S. Behera, J. K. Mandal, and D. P. Mohapatra, Eds., vol. 31, pp. 77–86, Springer India, 2015.

[48] I. Rodríguez-Fdez, A. Canosa, M. Mucientes, and A. Bugarín, "STAC: a web platform for the comparison of algorithms using statistical tests," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 1–8, Istanbul, Turkey, August 2015.