ISSN 2286-4822
www.euacademic.org

EUROPEAN ACADEMIC RESEARCH
Vol. IX, Issue 6/ September 2021

Impact Factor: 3.4546 (UIF)
DRJI Value: 5.9 (B+)

Automatic Malware Analysis Based on Machine Learning: a Comparative Study of the WEKA and Rapid Miner Tools

CÉSAR AUGUSTO BORGES DE ANDRADE[1]
JOÃO PAULO ABREU MARANHÃO[2]
GILDÁSIO ANTONIO DE OLIVEIRA JÚNIOR[3]
RAFAEL T. DE SOUSA JR.[4]

Abstract

Malware detection is an important cybersecurity research area to provide several security technologies and techniques applied to counter at-tacks targeting information systems and networks. Different data extrac-tion tools apply classification algorithms in order to

[1] **César Augusto Borges de Andrade** received his bachelor degree in data processing in 1997 from the Mackenzie Presbiterian University, São Paulo, Brazil, and his M.Sc. degree in systems and computing in 2013 from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil. Currently, he is a Ph.D. student at the Graduate Program in Electrical Engineering at the University of Brasilia (UnB), Brazil, researching on machine learning applied to malicious software detection systems. ORCID: 0000-0001-5776-2119

[2] **João Paulo Abreu Maranhão** received his bachelor degree in telecommunications engineering in 2003 and his M.Sc. degree in systems and computing in 2014 both from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil. Currently, he is a Ph.D. in Electrical Engineering at the University of Brasilia (UnB), Brazil, researching on multidimensional signal processing and machine learning applied to network intrusion detection systems. **ORCID: 0000-0003-0632-6434**

[3] **Gildásio Antonio de Oliveira** Júnior received his Bachelor Degree in Computer Science (2006) from University Center of Bahia - FIB (Brazil). Has a Specialization Course in Computer Networks with an Emphasis on Security (2015) from University Center of Brasilia - UniCEUB (Brazil) and a Master Degree in Electrical Engineering (2016), area of computer forensic concentration and information security from University of Brasilia - UnB (Brazil). He is PhD in Electrical Engineering and a researcher at the University of Brasília (Brazil). His main research interests are cyber security, information security, computer networks, text mining, sentiment analysis, data visualization and big data. ORCID: 0000-0001-6198-4945

[4] **Rafael T. de Sousa Jr.** (Senior Member, IEEE) received the bachelor's degree in electrical engineering from the Federal University of Paraíba (UFPB), Campina Grande, Brazil, in 1984, the master's degree in computing and information systems from the Ecole Supérieure d'Electricité–Supélec, Rennes, France, in 1985, and the Ph.D. degree in telecommunications and signal processing from the University of Rennes 1, Rennes, in 1988. He was a Visiting Researcher with the Group for Security of Information Systems and Networks (SSIR), Ecole Supérieure d'Electricité–Supélec, from 2006 to 2007. He has worked in the private sector from 1988 to 1996. Since 1996, he has been a Network Engineering Associate Professor with the Electrical Engineering Department, University of Brasília (UnB), Brazil, where he is currently the Coordinator of the Professional Post-Graduate Program on Electrical Engineering–Cybersecurity (PPEE) and supervises the Decision Technologies Laboratory (LATITUDE). He is Chair of the IEEE VTS Centro-Norte Brasil Chapter (IEEE VTS Chapter of the Year 2019) and of the IEEE Centro-Norte Brasil Blockchain Group. He is currently a Researcher with the Productivity Fellowship Level 2 (PQ-2) granted by the Brazilian National Council for Scientific and Technological Development (CNPq). His professional experience includes research projects with Dell Computers, HP, IBM, Cisco, and Siemens. He has coordinated research, development, and technology transfer projects with the Brazilian Ministries of Planning, Economy, and Justice, as well as with the Institutional Security Office of the Presidency of Brazil, the Administrative Council for Economic Defense, the General Attorney of the Union and the Brazilian Union Public Defender. He has received research grants from the Brazilian research and innovation agencies CNPq, CAPES, FINEP, RNP, and FAPDF. He has developed research in cyber, information and network security, distributed data services and machine learning for intrusion and fraud detection, as well as signal processing, energy harvesting and security at the physical layer.
ORCID: 0000-0003-1101-3029

identify malicious codes. In this work, we compare the WEKA and the Rapid Miner data mining tools in regard to their use of state-of-the-art classification al-gorithms Naive Bayes and Random Forest for malware detection. The analysis is performed by using a dataset generated by submitting mal-ware artifacts to a SandBox. Results show that WEKA presents better performance with the Random Forest algorithm, whereas for Naive Bayes the best tool is Rapid Miner.

Keywords: Malware Classification, Machine Learning, Naive Bayes, Random Forest, WEKA, Rapid Miner.

1 INTRODUCTION

The growth of information technology increased the number of cyber attacks around the world. Such attacks can be executed by using malicious software, which are programs designed to perform harmful actions on a computer. Since new malware variants are regularly created with new evasive skills, anti-malware products (for example, antivirus software) are not able to keep up with the creation and dissemination of several artifacts, which makes malware analysis techniques inefficient. The number of unique samples of malware has increased dramatically over the last 10 years, probably exceeding 1,2 billion in the end of 2021, as it can be seen in Figure 1[5].

Therefore, manual analysis for signature generation becomes impractical, since it takes a lot of time compared to the speed of creation and spreading of new malware. In this scenario, automatic analysis is a more efficient option. However, one of the great problems of automatic analysis is that the interpreta-tion of the large reports generated by sandboxes (i.e., restricted and controlled environments for the execution of artifacts, usually suspicious software) is left to the user. To overcome this issue, such reports can be treated and submitted to machine learning algorithms to generate classifiers with good performance in order to perform malware detection and classification tasks [1].

[5] https://www.av-test.org/en/statistics/malware/

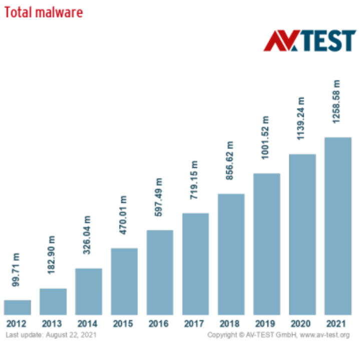


Figure 1: Total number of malware (in millions) from 2012 to 2021.

There are several works in the literature about comparative studies of data mining tools applied on classification. In [3], the authors proposed the use of data mining tools to perform intrusion detection in wireless local area networks (WLANs). They presented theoretical details about WEKA, SPSS and Tana-gra, providing a brief description of each tool.

Sharma et al. [4] described the main features of WEKA and provided a quick start on other data mining tools. The authors evaluated WEKA with different classification algorithms, such as Naive Bayes and Classification Trees.

In 2018, Bisht et al [2] performed a comparative study of several data mining tools for intrusion detection including WEKA and Rapid Miner. They have done a preliminary analysis of the results of three different data mining tools using the KDD'99 attack dataset [16], obtaining promising results.

Finally, in 2019, Kawelah and Abdala [17] did a comparative study between WEKA and Rapid Miner with Random Tree and Random Forest classification algorithms for network intrusion detection. The authors used the KDD'99 [16] attack dataset and concluded that WEKA plus Random Forest outperformed the other tools. In the same year, the authors performed a similar study using the C4.5 and Decision Stump algorithms [20].

The main contribution of this work is to compare WEKA and Rapid Miner tools by applying Naive Bayes and Random Forest algorithms for automatic malware analysis and consequently determine which one is the best data mining tool in this specific case.

The remainder of this paper is organized as follows. Section 2 presents a brief description about the data mining tools and classification algorithms used in this work. In Section 3 we compare the performance of WEKA and Rapid Mine tools with Naive Bayes and Random Forest classification algorithms and present the simulation results. Finally, Section 4 concludes the paper.

2 DATA MINING TOOLS AND CLASSIFICATION ALGORITHMS

In this section we present a brief description about the WEKA and Rapid Miner data mining tools, as well as the Naive Bayes and Random Forest state-of-the-art classification algorithms.

2.1 WEKA

WEKA is a data mining system developed by the University of Waikato, New Zealand, in 1992 [5]. It is a collection of different machine learning algorithms that can be used with data mining [6] and contains several tools for data pre-processing, sorting, regression, clustering, association and visualization rules. WEKA is well suited for developing new machine learning schemes [7] and is considered an independent platform because the program is written in the JavaTM language, with a graphical interface for interacting with data and producing visual results. Since WEKA contains a generic API, we can include it in our applications for several tasks such as automatic server-side data mining [8].

2.2 Rapid Miner

Rapid Miner was developed in Java by Klinkenberg et al in 2001 [9]. It is used for commercial applications, as well as for research, education and training. The development of applications supports all stages of data mining processes, including data preparation, visualization results, validation, and model opti-mization. Rapid Miner is one of the most commonly used analytical tools for tasks that requires rapid predictive recognition and was considered a leader in the 2016 Gartner Magic Quadrant for Advanced Analytics Platforms [9].

2.3 Naive Bayes

Naive Bayes algorithm is a probabilistic classifier based on the Bayes' Theorem [10]. Such algorithm can be used classify texts based on the frequency of words and consequently identify if An e-mail is a SPAM [11]. Since Naive Bayes is very fast and simple, it presents a relatively higher performance than other classifiers. In addition, the algorithm needs a small number of test data to complete classifications with a good accuracy. Since it has a relatively high speed and only needs a few data to perform classification, Naive Bayes can be used for real-time predictions.

2.4 Random Forest

Random Forest algorithm is a decision tree based classifier which recognizes patterns of several classes at the same time [12]. According to Breiman [12], Random Forest is considered as an extension of the decision tree algorithm, since it makes use of resampling methods in order to improve the accuracy of constructed models. The algorithm was initially proposed by Ho in 1995 [14] and subsequently developed by Breiman in 2001 [12]. Such an algorithm combines the concepts of bagging [13] and random selection for the construction of a set of trees with controlled variance. Each tree votes a decision on the class of a given object and the class with the highest number of votes is selected.

3 AUTOMATIC MALWARE ANALYSIS BASED ON MACHINE LEARNING

In this section we discuss the performance of WEKA and Rapid Miner data min-ing tools when applying Naive Bayes and Random Forest algorithms on malware classification. All experiments were executed on a computer with Intel® Core™ i3 2.10 GHz, with 8 GB of RAM and operational system Linux Ubuntu. The data mining tool versions were WEKA 3.6.13 and Rapid Miner Studio 9.3.001. The artifacts are applied on the Cuckoo Sandbox 2.0.6, which is an advanced open source automated malware analysis system. Figure 2 illustrates the mal-ware analysis methodology, which is composed by six blocks described in the following subsections.

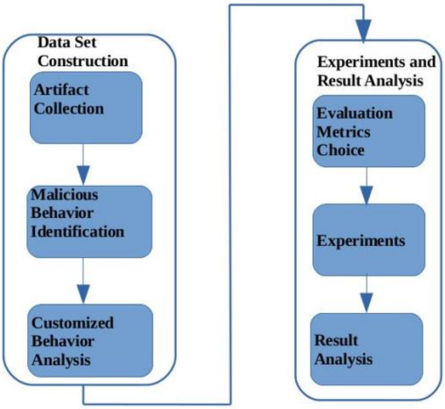

Data Set Construction
Artifact Collection
Malicious Behavior Identification
Customized Behavior Analysis
Experiments and Result Analysis
Evaluation Metrics Choice
Experiments
Result Analysis

Figure 2: Malware analysis methodology.

3.1 Artifact Collection

The first block of Figure 2 corresponds to the artifact collection. For per-formance analysis, we consider the dataset generated from the submission of artifacts in a sandbox, according to the distribution shown in table 1. Two sets of examples are considered, malware and not malware, both in PE (Portable Executable) format. The malware dataset was extracted from the VX Heaven Windows Virus Collection repository [18]. Non-malware or benign programs were collected from clean Windows machines.

3.2 Malicious Behavior Identification

The automatic artifact analysis is performed in the second block of Figure 2. Each sample is automatically applied on the Cuckoo Sandbox, which generates an artifact activity report in csv (comma-separated values) format. This process is briefly described in Figure 3.

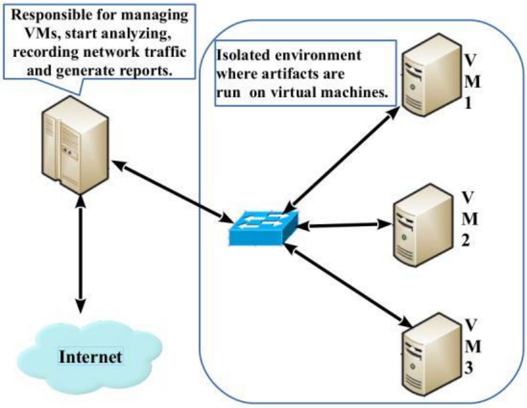

Responsible for managing VMs, start analyzing, recording network traffic and generate reports.
Isolated environment where artifacts are run on virtual machines.
VM 1
VM 2
VM 3
Internet

Figure 3: Malware analysis in Cuckoo Sandbox

The submission of the artifacts was performed automatically through a script implemented in shell script language. The overall process for each artifact, from submission to generation of the Cuckoo Sandbox report, is finished in approximately 6 minutes.

3.3 Customized Behavior Analysis

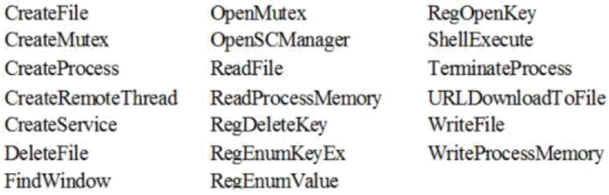
The next block of Figure 2 corresponds to the customized behavior analysis. This phase is divided into two steps: (i) Attribute Engineering, and (ii) Learning and Evaluation.

3.3.1 Attribute Engineering

In this step three activities are performed: selection of relevant attributes from the dataset, creation of the dictionary of terms and creation of the vector model.

1. selection of relevant attributes: the most relevant attributes are iden-tified from all csv reports. In our experiments, 121 attributes were initially selected, which are some APIs commonly used by malware activities [19].
2. creation of the dictionary of terms: from the 121 initial APIs, the 20 most relevant were selected to compose our dictionary of terms, as shown in Figure 4. In this set, two more attributes were added: number of processes created and number of downloads performed during the analysis.

3. creation of the vector model: each csv report is compared to the dic-tionary of terms and the frequency of each term is recorded. Next, the csv reports are converted into an attribute vector. This is done automatically through a script, implemented in the shell script language, which crosses the artifact with the most suspicious features.



CreateFile
CreateMutex
CreateProcess
CreateRemoteThread
CreateService
DeleteFile
FindWindow
OpenMutex
OpenSCManager
ReadFile
ReadProcessMemory
RegDeleteKey
RegEnumKeyEx
RegEnumValue
RegOpenKey
ShellExecute
TerminateProcess
URLDownloadToFile
WriteFile
WriteProcessMemory

Figure 4: The most relevant APIs selected from the dataset.

3.3.2 Learning and Evaluation

In this phase, we apply machine learning techniques on the attribute vector files for learning and evaluating malware. As data was represented in vector form, several classification algorithms can be chosen and compared with each other. We verify the performance of the methodology through several parameters such as accuracy, sensitivity and precision.

3.4 Evaluation Metrics Choice

The evaluation metrics are chosen in the fourth block of Figure 2. In this work, we considered accuracy, precision and recall as the performance metrics, which are defined as follows [2]:

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}}, \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2}$$

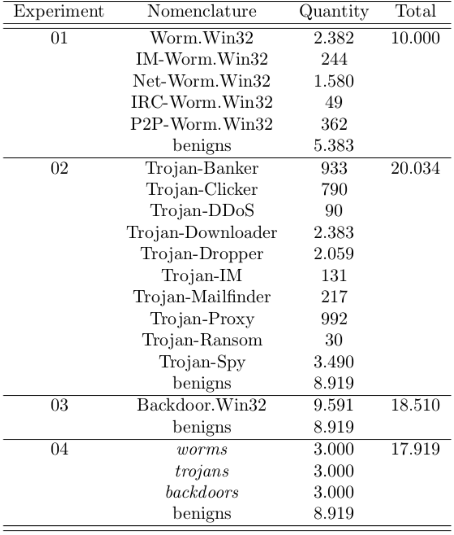
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3}$$

where TP, TN, FP and FN denote respectively the number of True Positives, True Negatives, False Positives and False Negatives.

3.5 Experiments and Result Analysis

Finally, the fifth and sixth blocks in Figure 2 correspond to the experiment conduct and result analysis, respectively. In this work we performed four experiments: procedures 1 to 3 evaluated different classes of malware, while a set composed by malware samples extracted from each previous experiment was analyzed in procedure 4. All experiments are described in Table 1.

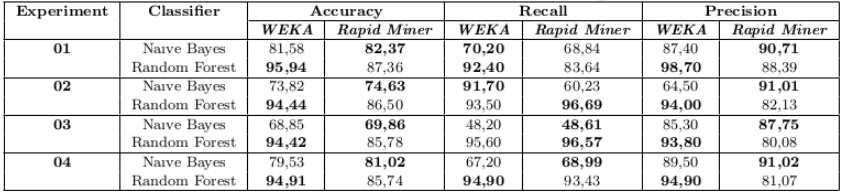
Table 1: Artifact distribution on each experiment.



Experiment	Nomenclature	Quantity	Total
01	Worm.Win32	2.382	10.000
	IM-Worm.Win32	244	
	Net-Worm.Win32	1.580	
	IRC-Worm.Win32	49	
	P2P-Worm.Win32	362	
	benigns	5.383	
02	Trojan-Banker	933	20.034
	Trojan-Clicker	790	
	Trojan-DDoS	90	
	Trojan-Downloader	2.383	
	Trojan-Dropper	2.059	
	Trojan-IM	131	
	Trojan-Mailfinder	217	
	Trojan-Proxy	992	
	Trojan-Ransom	30	
	Trojan-Spy	3.490	
	benigns	8.919	
03	Backdoor.Win32	9.591	18.510
	benigns	8.919	
04	*worms*	3.000	17.919
	trojans	3.000	
	backdoors	3.000	
	benigns	8.919	

We performed experimental analysis on WEKA and Rapid Miner tools using Naive Bayes and Random Forest classification algorithms. The table 2 shows the obtained results. Each table column describes, respectively, the identification number for each experiment, type of classification algorithm, accuracy, recall and precision. In each line, the best performance of each tool is in bold.

Table 2: Evaluation metrics for malware detection in Rapid Miner and WEKA.



Experiment	Classifier	Accuracy		Recall		Precision	
		WEKA	*Rapid Miner*	*WEKA*	*Rapid Miner*	*WEKA*	*Rapid Miner*
01	Naıve Bayes	81,58	**82,37**	**70,20**	68,84	87,40	**90,71**
	Random Forest	**95,94**	87,36	**92,40**	83,64	**98,70**	88,39
02	Naıve Bayes	73,82	**74,63**	**91,70**	60,23	64,50	**91,01**
	Random Forest	**94,44**	86,50	93,50	**96,69**	**94,00**	82,13
03	Naıve Bayes	68,85	**69,86**	48,20	**48,61**	85,30	**87,75**
	Random Forest	**94,42**	85,78	95,60	**96,57**	**93,80**	80,08
04	Naıve Bayes	79,53	**81,02**	67,20	**68,99**	89,50	**91,02**
	Random Forest	**94,91**	85,74	**94,90**	93,43	**94,90**	81,07

From the results shown in table 2, we observe that Rapid Miner, in most of the cases, provides better results with Naive Bayes classifier. Otherwise,

Random Forest, in most of the cases, provides better results in the experiments in WEKA. Such observation is corroborated by Figures 5 and 6, which illustrate the comparison between both tools when considering the results obtained in experiment 4.

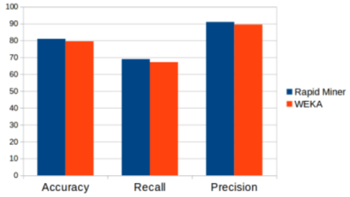


Figure 5: Comparative Results of Tools using Naïve Bayes.

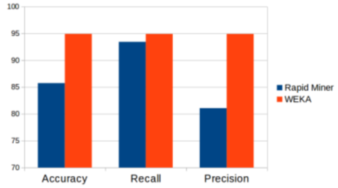


Figure 6: Comparative Results of Tools using Random Forest.

4 CONCLUSION

Data mining tools play a fundamental role in malware analysis and detection. In this work, we build a dataset from the submission of thousands of artifacts into a sandbox. Based on this preprocessed dataset, a comparative study between WEKA and Rapid Miner tools was performed. We observed that Naive Bayes is outperformed by Random Forest algorithm when Rapid Miner tool is applied, while the inverse performance occurred with WEKA. As a future work, we intend to provide an in-depth analysis of the results for several data mining tools applied to automatic malware detection and classification. In addition, we intend to apply this same methodology to other platforms, such as Android and Linux.

Acknowledgments

This work was developed in a laboratory with computational and staff resources supported in part by CNPq - Brazilian National Research Council, Grant 312180/2019-5 PQ-2, Grant BRICS 2017-591 LargEWiN, and Grant 465741/2014-2 INCT in Cybersecurity, in part by CAPES - Brazilian Higher Education Personnel Improvement Coordination, Grant 23038.007604/2014-69 FORTE and Grant 88887.144009/2017-00 PROBRAL, in part by the Brazilian Ministry of the Economy, Grant 005/2016 DIPLA and Grant 083/2016 ENAP, in part by the Administrative Council for Economic Defense, Grant CADE 08700.000047/2019-14, in part by the General Attorney of the Union, Grant AGU 697.935/2019, in part by the Ministry of Health, Grant DENASUS 04/2021, and in part by the Attorney General of the National Treasury, Grant PGFN 01/2021.

Conflicts of Interest

The authors declare no conflict of interest.

REFERENCES

1. C. A. B. Andrade, C. G. Mello and J. C. Duarte; "Malware Automatic Analysis"; 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence; pp. 681-686; Brazil; Ipojuca; 2013; ISSN: 2377-0589; doi: 10.1109/BRICS-CCI-CBIC.2013.119.
2. P. Bisht, N. Eeraj, M. Preeti and C. Pushpanjali; "A Comparative Study on Various Data Mining Tools for Intrusion Detection"; International Journal of Scientific & Engineering Research; Volume 9; Issue 5; ISSN: 2229-5518; May-2018.
3. A. M. Patel, D. A. Patel and M. H. R. Patel; "A Comparative Analysis of Data Mining Tools for Performance Mapping of WLAN Data"; International Journal of Computer Engineering & Technology (IJCET); vol. 4; no. 2; pp. 241{251, ISSN: 0976-6375; 2013.
4. A. Sharma and B. Kaur; "A Research Review on Comparative Analysis of Data Mining Tools, Techniques and Parameters"; International Journal of Advanced Research in Computer Science; vol. 8; no. 7; ISSN: 0976-5697; DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4255; July {August 2017.
5. S. S. Aksenova; WEKA Explorer Tutorial; California State University; 2004. accessed: Jun. 20, 2019. [Online]. Available: http://people.sabanciuniv.edu/berrin/cs512/lectures/WEKA/WEKA
6. H. Solanki; "Comparative Study of Data Mining Tools and Analysis with Unified Data Mining Theory"; International Journal of Computer Applica-tions; vol. 75; no. 16; pp. 23-28; ISSN: 0975- 8887; doi: 10.5120/13195-0862; 2013.
7. Laboratory Module 1 - Description of WEKA (Java-implemented machine learning tool); accessed: Jun. 20, 2019; [Online].Available: http://software.ucv.ro/cmihaescu/ro/teaching/AIR/docs/Lab2-DescriptionOfWEKA.pdf

8. R. R. Bouckaert, E. Frank, M. Hall, R.Kirkby, P. Reutemann, A. Seewald et al; "WEKA Manual for Version 3-7-8"; Hamilton; New Zealand; 2013.
9. Rapid Miner;"Gartner Magic Quadrant for Data Science Platforms"; accessed: Jun. 20; 2019 [Online]; Available: https://rapidminer.com/resource/gartnermagicquadrant-data-science-platforms//.
10. F. Carani; "The Naïve Bayes Learning Algorithm"; 2019, doi: 10.13140/RG.2.2.18248.37120
11. V. Metsis, I. Androutsopoulos and G. Paliouras; "Spam Filtering with Naïve Bayes - Which Naïve Bayes?"; in Third Conference on Email and Anti-Spam (CEAS); 2006.
12. L. Breiman; "Random Forests"; Machine Learning; vol. 45; no. 1; pp. 5{32; 2001.
13. L. Breiman; "Bagging predictors"; Machine Learning; vol. 24; no. 2; pp. 123{140; 1996.
14. T. K. Ho; "Random Decision Tree"; Proceedings of the 3rd International Conference on Document Analysis and Recognition; Montreal; pp. 278{282; 1995.
15. U.S. trademark registration number 3185828, registered 2006/12/19.
16. KDD'99; KDD Cup 1999 Data; 1999; accessed: Jun. 10; 2019 [Online]; Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
17. W. Kawelah and A. Abdala; "A Comparative Study for Machine Learning Tools Using WEKA and Rapid Miner with Classifier Algorithms Random Tree and Random Forest for Network Intrusion Detection"; International Journal of Innovative Science and Research Technology (IJISRT); vol. 4; no. 4; pp 749-752; ISSN:-2456-2165; April { 2019.
18. VXNETLUX; accessed: Abr, 5, 2012; [Online]. Available: http://vx.netlux.org/.
19. M. Sikorski and A. Honig; "Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software"; San Francisco, CA, USA; 1st edition; ISBN: 1593272901, 9781593272906; 2012.
20. W. Kawelah and A. Abdala; "A Comparative Study on Machine Learn-ing Tools Using WEKA a nd Rapid Miner with Classifier Algorithms C4.5 and Decision Stump for Network Intrusion Detection"; EUROPEAN ACA-DEMIC RESEARCH; Vol. VII; Issue 2; ISSN 2286-4822; May 2019.