DOI: 10.15439/2018KM04
ISSN 2300-5963 ACSIS, Vol. 14

Detection of Malicious Executables Using Rule Based Classification Algorithms

Dr. Neeraj Bhargava[1], Aakanksha Jain[2], Abhishek Kumar[3], Dr. Dac-Nhuong Le[4]

[1] School of system sc. &engg. Professor MDS University Ajmer, India
[2]M.TECH Scholar at MDS University Ajmer, India
[3]Assistant Professor, Computer Science Department ACERC, Ajmer
[4] Deputy-Head, Faculty of Information Technology, Haiphong University Haiphong, Vietnam
[1]profneerajbhargava@gmail.com, [2]Jain1994aakanksha@gmail.com,[3]Abhishekkmr812@gmail.com,
[4]Nhuongld@hus.edu.vn.com

***Abstract*—Machine Learning class rule has varied packages together with classification, clustering, will understand association rules furthermore and is capable of the method an enormous set of the information set as measure supervised or unsupervised learning data. The paper deals with statistics mining sort set of rules on virus dataset created records from varied anti-virus logs. The work deals with classifications of malicious code per their impact on user's system &amp; distinguishes threats on the muse in their connected severity; these threads are therefore named as malicious possible from varied sources, on various running structures. During this paper, the generated output is that the listing of records summarizing however because it ought to be the classifier algorithms are ready to predict the authentic magnificence of the days at a lower place the chosen take a look at module. The operating model deals with predicting the outliers of the threat datasets and predicts the optimum results supported analysis victimization the chosen rule. The work illustrates implementation of the algorithms corresponding to half, JRIP and RIDOR in additional economical manner because it relies on virus-log datasets to come up with A level of accuracy to the classification results.**

***Index Terms*—Threats, Rule-based Classification, Prediction of Severity, Moderate, Malicious Executable, Danger, Normal.**

I. Introduction

(1) INTRODUCTION to JRIP Algorithm (Rule-based Classification algorithm): JRIP sometimes called as RIPPER is one of popular classifier algorithm [7][5]. In JRIP instances of the dataset are evaluated in increasing order, for given dataset of threat a set of rules are generated. JRIP (RIPPER) algorithm treats each dataset of given databaseand generates a set of rules including all the attributes of the class. Then next class will get evaluated and does the same process as previous class, this process continues until all the classes have been covered.

(2) Introduction to PART Algorithm (Rule-based Classification algorithm): Full form of PART is Projective Adaptive Resonance Theory [4].PART is refined method of rule generation [6]. After rule generation entire tree generated, the best tree is selected and its leaves are translated into rules.-PART support all type of classes like Binary and Nominal classand supports all type of attributes.

(3) Introduction to RIDOR Algorithm (Rule-based Classification algorithm): This algorithm is an implementation of a Ripple-Down Rule learner. The RIPPER algorithm directly extracts best rules from the provided dataset.

The Ripper algorithm completes its process in following phases:

a) Growth. b) Pruning. c) Optimization. d) Selection.

While the generation of Rules in Growth PARTs the manslayer algorithmic rule will usually decision as greedy algorithmic rule i.e. it avariciously adds attributes in rules being generated till the stopping criteria of the rule.

Incremental pruning is finished in Pruning PART, the i.e. algorithmic rule permits pruning of attribute sequences until fulfillment of pruning metrics.

The third PART suggests that the improvement stage optimizes every rule by followings a pair of steps:

1) Greedy addition of attributes in original rule

2) Grow a replacement rule severally with growing and prune PART as mentioned within the paragraph. When growing new rule victimization choice (last PART of Ripper Algorithm) PART best rule is chosen, and different not chosen rules are deleted.

II. Literature survey

Automated analysis operates on vast solicitation of detected malware threats and reduces the human effort in analysis of anti-malware [8] [16]. Another works are able to discuss malicious codes are dynamically analyzed by any machine driven system then analyzes some cognition performed classification system that is generated by analysis. SVM (Support Vector Machine) classifier uses these samples to coach itself so SVM will proactively notice malicious threats. Once we value the results on basis of quality of classification [15] and speed of experimental execution, it demonstrates smart results on the machine driven system. As per the author, this machine-driven analysis is enough ready to notice whether or not a file is infected or not infected, as a result of supported classification system footprints the file are often mechanically blacklisted.

The planned classifier will discover previously undiscovered malware. However, it cannot discriminate between safe and malicious threat files.

III. Related Work

(1) Adware: Adware stands for Advertising-Supported software system. Adware generates by itself on websites once a user needs to access any video audio or the other quite data, it seems as advertising material and pop-ups.It quickly generates a commercial. Adware keeps track [9] on user activity and steals their browsing and different data. Most of the Adware's don't seem to be thought-about as a dangerous threat; it usually comes underneath low-risk threats.

(2) Trojan Horse: Also known as “Trojan”, itinterprets itself as any simple file to end users, so if they download it they are actually downloading [14] a malware. Trojan produces the effect of repudiation and Elevation of privileges to end-user. Any malicious [8] PARTy can remotely access that infected computer. Once access is obtained from a contaminative computer, the attacker can possibly to steal data like end-user login-detail; financial transactions can also access victim’s electronic money. Further modification in files, installation [14] of other malware, keep track on user's activity, keylogging can also be done.

(3) Virus: A virus is a type of threat which is can itself and also can spread itself to other systems in a network. They can attach to programs and executable codes, when a system user accesses any of infected programs, whenever the user of that network access that program or code attachment infect those systems also. [11] Viruses spread through vulnerabilities in web apps. Viruses can spoof the information, harm client computer may be the whole network. Viruses some time generate botnets and infer user account information.

(4) Worm: Computer worms explore operating system vulnerabilities. They generate Payloads which are actually programming code to produce a harmful effect on host computer [10]. Basically, the worm can be defined as a replicating computer threat which produces a harmful effect on the system by slowing it down and many other annoying effects. Computer worm can be viewed as computer virus but it distinguishes itself with its self-replication characteristics [12] and spread independently means it needs not to be activated or access by running a program, opening a file, etc). Worms mostly spread by email attachments

IV. METHODOLOGY AND IMPLEMENTATION

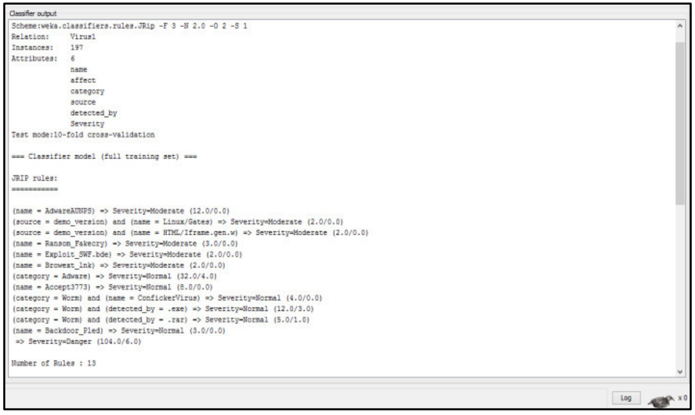
(1) Implementation using JRIP: Here below figure 1 shows the practical implementation of JRIP algorithm in weka . Implemented results show that there could be 13 rule constructed by using JRIP, which are as follows

Severity will be Moderate if malicious threat name is AdwareAUNPS and if source isdemo_version

And threat names are Linux/Gates or HTML/Iframe.gen.w, and if threat name is Ransom_Fakecry, Exploit_SWF.bde, Browext_lnk. If category is Adware then Severity will be Normal

Severity will be Normal if category is Worm and threat name is Conficker virus

Otherwise, Severity will be Danger

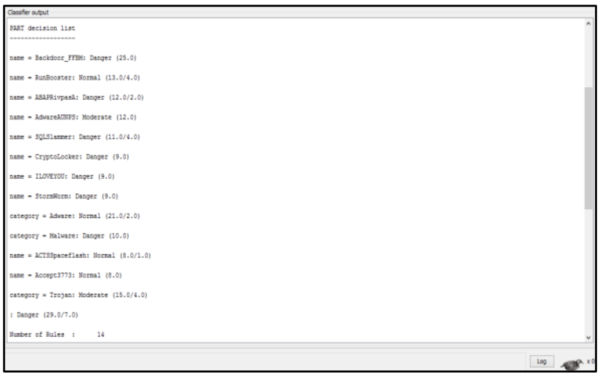


```
Scheme:weka.classifiers.rules.JRip -F 3 -N 2.0 -O 2 -S 1
Relation:     Virus1
Instances:    197
Attributes:   6
              name
              affect
              category
              source
              detected_by
              Severity
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

JRIP rules:
===========

(name = AdwareAUNPS) => Severity=Moderate (12.0/0.0)
(source = demo_version) and (name = Linux/Gates) => Severity=Moderate (2.0/0.0)
(source = demo_version) and (name = HTML/Iframe.gen.w) => Severity=Moderate (2.0/0.0)
(name = Ransom_Fakecry) => Severity=Moderate (3.0/0.0)
(name = Exploit_SWF.bde) => Severity=Moderate (2.0/0.0)
(name = Browext_lnk) => Severity=Moderate (2.0/0.0)
(category = Adware) => Severity=Normal (32.0/4.0)
(name = Accept3773) => Severity=Normal (8.0/0.0)
(category = Worm) and (name = ConfickerVirus) => Severity=Normal (4.0/0.0)
(category = Worm) and (detected_by = .exe) => Severity=Normal (12.0/3.0)
(category = Worm) and (detected_by = .rar) => Severity=Normal (5.0/1.0)
(name = Backdoor_Pled) => Severity=Normal (3.0/0.0)
 => Severity=Danger (104.0/6.0)

Number of Rules : 13
```

Figure (1): Prediction Rule Generated Using JRIP Rule Classifier

(2) Implementation using PART: Here below figure (2) shows the sensible implementation of the half algorithmic program in Weka. Enforced results show that there can be 14rule created by victimization JRIP, that area unit as follows If threat names areBackdoor_FFBMand ABAPRivpasA, SQLSlammer, ILOVEYOU, StormWorm then severity can Danger. If threat name is RunBooster then severity is traditional. If threat class is Adware then severity is traditional. If threat class is Malware then severity is Danger

If threat class is Trojan then severity is Moderate



```
PART decision list
------------------

name = Backdoor_FFBM: Danger (25.0)

name = RunBooster: Normal (13.0/4.0)

name = ABAPRivpasA: Danger (12.0/2.0)

name = AdwareAUNPS: Moderate (12.0)

name = SQLSlammer: Danger (11.0/4.0)

name = CryptoLocker: Danger (9.0)

name = ILOVEYOU: Danger (9.0)

name = StormWorm: Danger (9.0)

category = Adware: Normal (21.0/2.0)

category = Malware: Danger (10.0)

name = ACTSSpaceflash: Normal (8.0/1.0)

name = Accept3773: Normal (8.0)

category = Trojan: Moderate (15.0/4.0)

: Danger (29.0/7.0)

Number of Rules  :      14
```

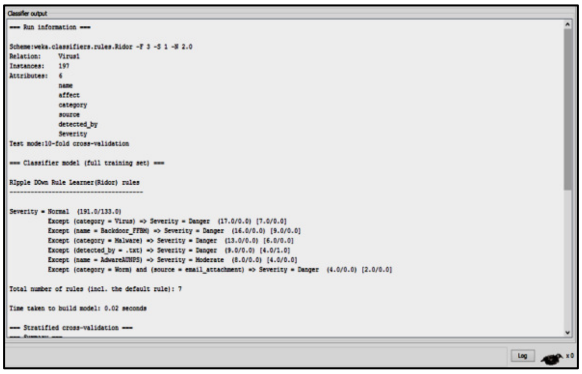
Figure (2): Prediction Rule Generated Using PART Rule Classifier

(3) Implementation using RIDOR:

Here below figure 3 shows the practical implementation of RIDOR algorithm in weka. Implemented results show that there could be 7rule constructed by using JRIP, which are as follows

Severity will be Normal Except if category is Virus then Severity will be Danger

and if name is Backdoor_FFBM then Severity will be Danger and if category is Malware then Severity will be Danger and if detected_by .txt) then Severity will be Danger and if name is AdwareAUNPS) then Severity will be Moderate and if category is Worm and source is email_attachment) then Severity will be Danger



```
=== Run information ===

Scheme:weka.classifiers.rules.Ridor -F 3 -S 1 -N 2.0
Relation:     Virus1
Instances:    197
Attributes:   6
              name
              affect
              category
              source
              detected_by
              Severity
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

RIpple DOwn Rule Learner(Ridor) rules
--------------------------------------

Severity = Normal  (191.0/133.0)
           Except (category = Virus) => Severity = Danger  (17.0/0.0) [7.0/0.0]
           Except (name = Backdoor_FFBM) => Severity = Danger  (16.0/0.0) [9.0/0.0]
           Except (category = Malware) => Severity = Danger  (13.0/0.0) [6.0/0.0]
           Except (detected_by = .txt) => Severity = Danger  (9.0/0.0) [4.0/1.0]
           Except (name = AdwareAUNPS) => Severity = Moderate  (8.0/0.0) [4.0/0.0]
           Except (category = Worm) and (source = email_attachment) => Severity = Danger  (4.0/0.0) [2.0/0.0]

Total number of rules (incl. the default rule): 7

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===
```

Figure (3): Prediction Rule Generated Using Rule Classifier

V. RESULT ANALYSIS

Algorithm	Correctly Classified Instances (%)	Incorrectly Classified Instances (%)	Kappa statistic	Mean absolute error	Root mean squared error	Relative absolute error (%)	Root relative squared error
JRIP	82.20	17.8	0.6888	0.1509	0.314	37.65	70.30
PART	87.43	12.56	0.7848	0.1179	0.256	29.41	57.26
RIDOR	75.39	24.60	0.5644	0.164	0.405	40.92	90.54

Table 1: Summary of result

Algorith m	TP Rate	FP Rate	Precisio n	Recall	F-Measure	ROC Area	No. of Rules	Time taken to build model
JRIP	0.96	0.22	0.84	0.96	0.87	0.88	13	0.7sec
PART	0.99	0.12	0.91	0.99	0.94	0.98v	14v	0.5sec
RIDOR	0.97	0.31	0.79	0.97	0.87	0.84	7	0.2sec **v**

Table 2: Detail Accuracy of result by class

Here symbol ‘**v’** means “better result”
Symbol ‘*’ means “worse result”
And blank symbol means can't say whether a result is better or worse than the base algorithm.

- *Here we take **JRIP as base Algorithm.***

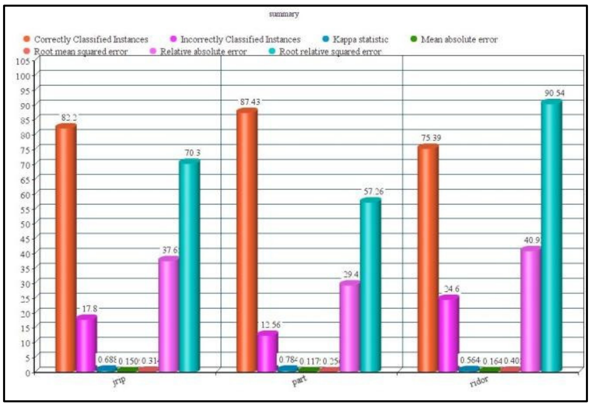


Figure (4): Summary of result

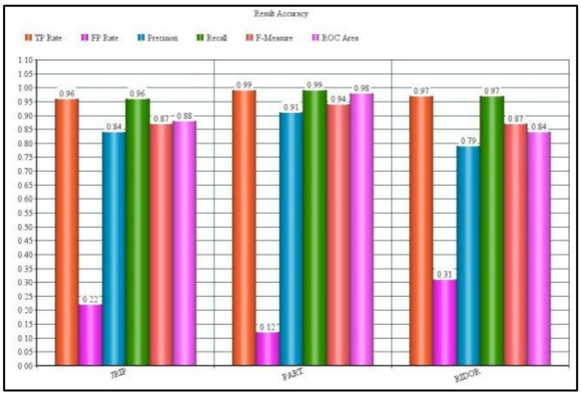


Figure (5): Comparison of results obtained through JRIP, PART and RIDOR algorithm

VI. CONTRIBUTION

We propose a strategy that performs well on chosen dataset of malicious threats; on the premise of this experiment we will extend the scale of info i.e. as well as multiple networks knowledge log, a module will be created victimization this system i.e. analysis victimization 3 formula RIDOR, JRIP and half as whole to perform prediction of laptop threats in a very dynamic manner. The planned module can generate set of rules once process given dataset of threats detected that were detected In last decades, these rules will be wont to produce virus signatures to be wont to predict malicious threat samples in period of time. Rules generated by RIDOR half and JRIP,includes all potential rules, which will be generated for deleting malicious behavior of suspicious threat and outliers are shown here that shows that severity is considerably low i.e. those code of line will be treated as non-malicious class codes, and module can keep eye on those line of code that is underneath suspicious class. The planned model is going to be able to observe malicious behavior, intrusive advertisements, spying tools, phishing activities, and speedy replication of bound code in addition.

VII. Conclusion and Future Work

Here in work model, we have a tendency to predict the severity of threats by imposing a information of threats in rule-based classification formula, here as established in Table two PART set of rules manufacture the upper lead to the period of rules as we have a tendency to take JRIP because the base formula. Here JRIP manufacture thirteen rules and half manufacture fourteen policies and RIDOR manufacture seven policies. Here the results of half formula proves conclusion over JRIP and RIDOR formula. Thus in term of rule generation, half represents the most effective result. currently come back to our next motive The Prediction of severity is as follows: if the class is malware is that the threat in ten times, if the class is malware it's miles sometimes risk and completely different policies like if the decision of threat is backdoor_ffbm is consistently hazarding. Severity may well be regular except class is virus i.e. severity may well be each moderate and danger and if decision = adwareaunps then severity could be moderate. Assessment on basis of consequences PART of half manufacture higher effects than JRIP and RIDOR within the period of mythical monster space enclosed and vary of rules.

As elite methodology perform well on the chosen dataset of malicious threats, in future we will extend the scale of information i.e. together with multiple networks knowledge log, a module may be made mistreatment this technique i.e analysis mistreatment 3 formula RIDOR,JRIP and half during a whole to perform prediction of pc threats during a dynamic manner.

References

[1] W. Nor Haizan W. Mohamed, Mohd Najib Mohd Salleh, Abdul Halim Omar "A Comparative Study of Reduced Error Pruning Method in Decision Tree Algorithms,2012 IEEE International Conference on Control System, Computing and Engineering", 23–25 Nov. 2012, Penang, Malaysia.

[2] Quinlan R. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers; 1993.

[3] Thorsten Lehr, Jing Yuan, Dirk Zeumer, SupriyaJayadev, and Marylyn D RitchiRule based classifier for the analysis of gene-gene and gene-environment interactions in genetic association studies Published online 2011 Mar 1. doi: 10.1186/1756-0381-4-4

[4] Pinto C M A, Machado J A T. Fractional Dynamics of Computer Virus Propagation. Mathematical Problems in Engineering, 2014: 259-305.

[5] Ripple Down Rule learner (RIDOR) Classifier for IRIS Dataset V. Veeralakshmi et al. / International Journal of Computer Science Engineering (IJCSE) ISSN : 2319-7323 Vol. 4 No.03 May 2015

[6] Himadri Chauhan, Vipin Kumar, Sumit Pundir and Emmanuel S. Pilli 2013 International Symposium on Computational and Business Intelligence "A Comparative Study of Classification Techniques for Intrusion Detection", department of Computer Science and Engineering Graphic Era University Dehradun India DOI: 10.1109/ISCBI.2013.16

[7] Neeraj Bhargava, Sonia Dayma, Abhishek Kumar, Pramod Singh IEEE Sponsored 3rd International Conference on Electronics and Communication Systems (ICECS 2016) An Approach for Classification using Simple CART Algorithm in Weka, MDS University Ajmer India.

[8] George Cabau, Magda Buhu, CiprianOpris: "Malware Classification Using Filesystem Footprints" a Bitdefender Technical University of Cluj-Napoca

[9] Hengli Zhao, Ming Xu, Ning Zhong, Jingjing Yao, and Q. Ho, "Malicious Executables Classification

[10] Based on Behavioral Factor Analysis," presented at the 2010 International Conference on e-Education, e-Business, e-Management and e-Learning, Sanya, China, 2010

[11] "Malware Behavioral Analysis System:" TWMAN

[12] F. Cohen, "Computational aspects of computer viruses" Computers & Security, vol. 8, no. 4, pp. 297–298, 1989.

[13] J. Stewart, "Behavioural malware analysis using Sandnets," Computer Fraud & Security, vol. 2006, no.Issue, pp. 4-6, December 2006.

[14] Microsoft, "File system minifilter drivers," 2016. Available: https://msdn.microsoft.com/enus/library/windows/hardware/ff540402 (v=vs.85).aspx

[15] Hi-Juan Jia, Yan-yan Yang, Na Guo Zhengzhou: "Research on Computer Virus Source Modeling with Immune Characteristic" Normal University, Zhengzhou Henan 450044

[16] Muroya Y, Enatsu Y, Li H. Global stability of a delayed IRS computer virus propagation model. International Journal of Computer Mathematics, 2014, 91(3):347-367.

[17] C. Developers, "Cuckoo sandbox - open source automated malwareanalysis," 2016. [Online]. Available: https://media.blackhat.com/us-13/US-13-Bremer-Mo-Malware-Mo-Problems-Cuckoo-Sandbox-WP.pdf