WWJMRD 2017; 3(11): 363-367
www.wwjmrd.com
International Journal
Peer Reviewed Journal
Refereed Journal
Indexed Journal
UGC Approved Journal
Impact Factor MJIF: 4.25
e-ISSN: 2454-6615

Sonali Goyal
Student, Giani Zial Singh
College of Technology,
Bathinda, Punjab, India

ParamJeet
Professor, Giani Zial Singh
College of Technology,
Bathinda, Punjab, India

Classification of Malicious URLS for Web Using Ripper Algorithm

Sonali Goyal, ParamJeet

Abstract
Now a days Web data is the most discussed topic. In various fields related to internet produces data of thousands of gigabytes every minute. Various applications use multimedia data sharing procedure. So data will automatically be of bulk amount. This bulk amount of data is hard to process, takes longer time of search this much large data. RIPPER (Repeated Incremental Pruning to Produce Error Reduction) is one of the Classification rule algorithm.

Keywords: Ripper, Web Data

Introduction
Web Data
Web related data is the application of specialized tools through which large amount of data will be processed. This data otherwise will be very difficult to process without the automated tool.
The amount of data generated in the different mediums is enormous. Various social media sites which are producing the data of large nature. This type of data requires large amount of data processing abilities. So that after analysis the data can be represented in graphical way. This graphically represented data will helps in having better and fast data point of view. So that system understanding regarding the system will be better.
As we know the data produced will be enormous. This data belongs unstructured category. Because data produced in different mediums like audio, videos, text etc. This type of data is produced in billions of bytes every hour. Once this whole data will be produced and stored at the server. Now requires various levels of processing. So that system of understanding regarding the data can be developed. This data requires various levels of processing. Structuring etc.

Malicious URLs
Malicious URLs are the main tool to arrive somewhere. It is a necessity component required by the legal user to arrive at the specific server and access the resources. Any malicious user tries to access the resources in illegal way. So URLs are the way for both the persons that is legitimate and malicious. Any user authenticity of the username or password stands for only legal URLs. But when user has less knowledge about the system resources and illegal URLs can be scoffed, any illegal user can prepare the duplicate copy of the username and password. Later use that identity to arrive at the web server where legal users legal contents are stored, but malicious user access them illegally.

Web data URLs Challenges
1. Large scale: several million URLs are being produced every hour.
2. Extremely imbalanced data set: The lists of Malicious URLs are in very small amount compared to the total no. of URLs. It is 0.01% of total URLs list.

Ripper
Ripper is one of the classification rule algorithm. It basically extracts the rules directly from the data. This algorithm progresses through the given four phases: Growth phase,

Correspondence:
Sonali Goyal
Student, Giani Zial Singh
College of Technology,
Bathinda, Punjab, India

pruning, optimization, selection. In first phase that is growth phase first rule is generated and various attributes are added incrementally till certain stopping criteria arise. Each rule is incrementally pruned for any final sequence of the attributes. This procedure will go on till the final step is achieved. Finally those attributes are selected which are best suitable for the situation. Ripper is a rule based learner that build a set of rules that identify the classes while minimizing the amount of error.

The Ripper algorithm builds a single rule in the following steps:

- Split the dataset with growing and pruning set.
- In growth phase, start the things with empty set.
- Add the new rule and also provide gain criteria.
- Repeat step 3 till negative example or dataset is not found.
- Prune the new rule (attribute) based on new prune rules.

In a multi-class situation, the rules generated from the RIPPER algorithm are ranked in ascending order based on the number of examples in the class.

The RIPPER algorithm for multi-class classification is described in the following steps:

1. Ripper arranges the class based on ascending or descending order.
2. It identifies the short class as positive class and long class as negative class.
3. Only positive class for rules is too identified.
4. Repeat the steps 2 and 3 until short class finding stops.

Problem in ripper algorithm

1. As there will be growth in the knowledge of the attributes. This over knowledge will generates the over fitting of the rules, which may leads to the misclassification.
2. The major disadvantage is the noisy data. This noisy data can leads to MIS classification.
3. The major drawback of RIPPER is the over fitting of the rules. Such that wrong justification is performed at.

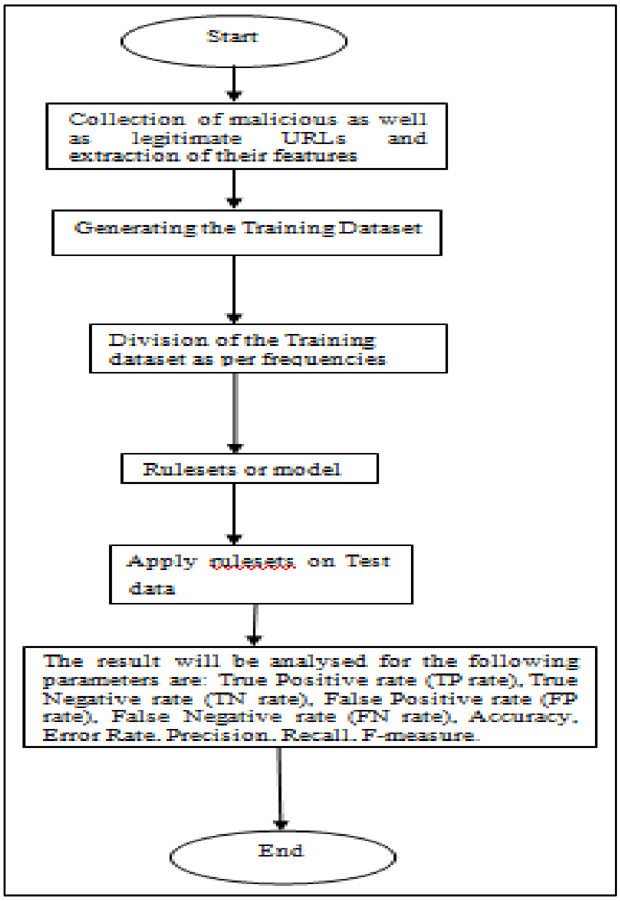
In RIPPER algorithm the normalization and balancing follows the common procedure. The rules developed are based on training dataset. The ruleset covers the rules based on various attributes.

1. The algorithm is designed to be fast and accurate. So that the improved proficiency is shown such that detecting malicious URLs can be identified.
2. If the rule set length is more and attributes are less then activity is performed using loop. The RIPPER algorithm with normalization is fast and effective way of doing the activity...
3. Each rule's attributes are checked against the initial seven rules. Then aggregation of the rules is taken place. Only those rules are selected which are based on high rank value.

Related Work

URLs have now days become a way to hack the resources belings to other. Attacker using malicious URLs distributes the malicious programs all around. Kaspersky La. b Author has reported that the browser bsed attacks have grown substantially. URLs are the Gateway for arriving at somewhere. Through URLs malicious person breech the security, and enter into the existing so called secured system in malicious way. Once enter into the secured system can destroy the existing information. So we need to protect the system from such breeches. It simply downloads the contents and checks the authenticity of the contents. Analyze how much time it has taken to download what is the download time. But contents based detection is not the base for identifying the attack. As new URLs are being produced every hour. It proposes the content based description to identify the malicious nodes. So that list of malicious and legitimate URL can be identified. Those URLs which fails the conditions will be put into the malicious list. And those which pass the contents description will be put into the legitimate list of URLs.

Flowchart



Ripper Algorithm

Ripper algorithm builds a single rule in the following steps:

- Split the dataset with growing and pruning set.
- In growth phase, start the things with empty set.
- Add the new rule and also provide gain criteria.
- Repeat step 3 till negative example or dataset is not found.
- Prune the new rule (attribute) based on new prune rules.

In a multi-class situation, the rules generated from the RIPPER algorithm are ranked in ascending order based on the number of examples in the class.

The RIPPER algorithm for multi-class classification is described in the following steps. Ripper arranges the class based on ascending or descending order. It identifies the short class as positive class and long class as negative class. Only positive class for rules is too identified. Repeat the steps 2 and 3 until short class finding stops. (Pan & Ding, 2006).

Analysis

Ripper (JRip) is a direct method i.e. is often used to extract rules directly from data. In WEKA tool RIPPER is implemented as JRip, generates rules set after theevaluation over the Training dataset. This rules set is the classifier model for JRip algorithm which can further be used to predicting the unknown URLs. Here, the output rules set of used to predict the data of the testing set after which all the parameters listed in table 5.2 is calculated.

Figure 5.1 shows the ruleset generated by the RIPPER algorithm. There are a total of 25 attributes and RIPPER algorithm make rulesets using these attributes. The rules are: -

Table 4.3: Rulesets of RIPPER Algorithm

NO.	Rules
1	(Favicon=yes)∧(SSL_final_state=yes)→ Legitimate
2	(Favicon=yes)∧(having_host_name=yes)→ Legitimate
3	(Page_Rank=2)∧(Favicon=yes)∧(URL_Length=56)→Legitimate
4	(double_slash_redirecting=yes)∧(folder_name=no)→Legitimate
5	(URL_Length=55)∧(Favicon=yes)→Legitimate
6	(Favicon=yes)∧(URL_Length=54)→Legitimate
7	Otherwise→Malicious

A rule-based is a technique for classifying record using a collection of “if…then…”rules. Table 4.3 ensures that every record is covered by exactly one rule.

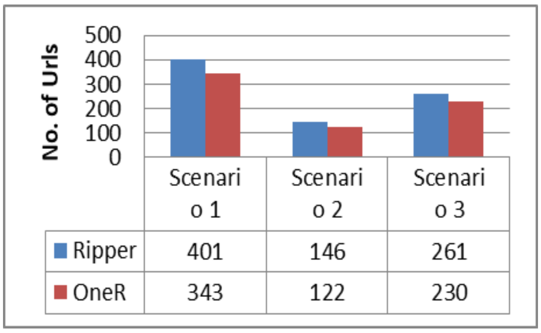
1. The first rule is interpreted asif a URL has the value yes for both favicon and SSL final state then the result shows that it is a legitimate URL.
2. The second rule is interpreted as if a URL have the value yes for favicon and en for having_host_name then the result shows that it is a legitimate URL.
3. The third rule is interpreted as if a URL has the value yes for favicon and has the value 2 for Page_Rank and also have the value 56 for URL_Length then the result shows that it is legitimate.
4. The fourth rule interpreted as if a URL have the value yes for double_slash_redirecting and have the value no for folder_name then the result shows that it is legitimate URL.
5. The fifth rule interpreted as if URL have the value yes for favicon and have the value 55 for URL_Length then the result shows that it is legitimate URL.
6. The sixth rule interpreted as if URL have the value yes for favicon and have the value 54 for URL_Length then the result shows that it is legitimate URL.
7. If all the previous rules are not satisfied by the URL of dataset then it will go to seventh rule which interpret that URL is malicious.

According to RIPPER algorithm, it is clear from the confusion matrix of the true positive rate of this algorithms proportion of examples which were classified the last rule, among all examples which truly have rules, i.e., how much of the rules was captured correctly (the number of malicious executable examples classified as malicious executables). True Negatives rate is proportion of examples which were classified above mentioned six rules was capture correctly the number of legitimate URLs classified as legitimate. False positive are those URLs which are actually legitimate but predicted malicious. False Negatives are those URLs which are actually malicious but predicted legitimate. So after each and every URLs data is checked with these rule sets the total number of True positive, true negative, false positive, false negative are calculated. After that accuracy of URLs is calculated from number of true positive and true negative by total number of URLs data. Error rate of URLs is calculated from number of false positive and false negative by total number of URLs data. Precision of the URLs is calculated from the number of exactly classified instance of a target URL, i.e., positive URL, over the number of instance classified as view to those URLs. It is also known as positive predicted value. Recall of the URLs is calculated from the number of exactly classified instance of a URL, i.e., positive URL, over the number of instance of that URL. The F-measure of URLs is calculated from the compromise between recall and precision.

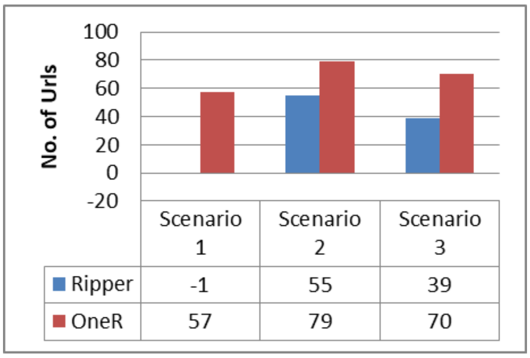
Resuls and Discusions

In web malicious URLs are most dangerous process to control. The malicious URL will be the most researched process now a days. In current research as well we have taken up WEKA tool to classify the malicious URLs. So that those which are most responsible as malicious URLs can be identified. In current research two algorithms are being compared Ripper and oneR. Both are classifiers. Ripper performs pruning and filtering. Performs better than the oneR in all the respects like true positive, false positive, true negative, false negative.

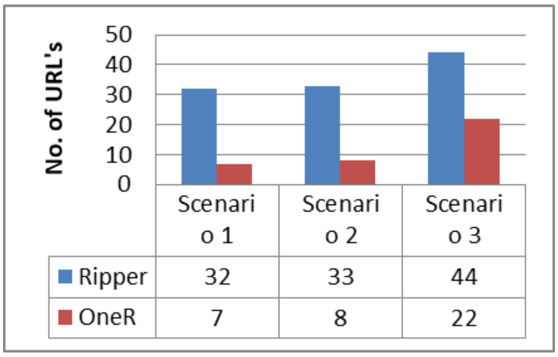
True Positive



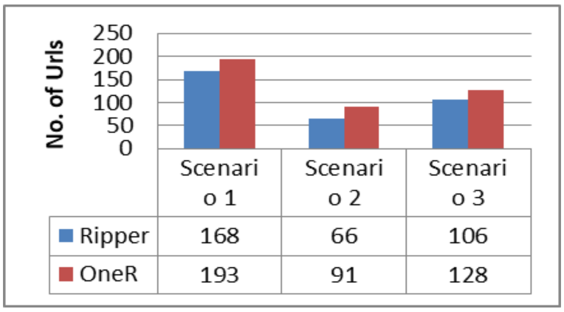
False Positives



True Negatives



False Negative



True Positive			
	Scenario 1	**Scenario 2**	**Scenario 3**
Ripper	401	146	261
OneR	343	122	230
True Negative			
	Scenario 1	Scenario 2	Scenario 3
Ripper	-1	55	39
OneR	57	79	70
False Positive			
	Scenario 1	Scenario 2	Scenario 3
Ripper	32	33	44
OneR	22	75	30
False Negative			
	Scenario 1	Scenario 2	Scenario 3
Ripper	168	66	106
OneR	193	91	128

Scenario 1		
	Ripper	OneR
No. of Urls Classified Correctly	569	536
No. of Urls Classified Incorrectly	31	64
Percentage of Correctness	94.8333 %	89.3 %
Percentage of Incorrectness	5.1667 %	10.7%
Scenario 2		
	Ripper	OneR
No. of Urls Classified Correctly	212	213
No. of Urls Classified Incorrectly	88	87
Percentage of Correctness	70.667%	71%
Percentage of Incorrectness	29.333%	29%
Scenario 3		
	Ripper	OneR
No. of Urls Classified Correctly	367	358
No. of Urls Classified Incorrectly	83	92
Percentage of Correctness	81.556%	79.556%
Percentage of Incorrectness	18.444%	20.444%

Conclusion and Future Work

Web data have various challenges related to security like-computation in distributed programming, security of data storage. For tackling with such security challenges we used different security methods like Type Based keyword search for security of Web data, use of hybrid cloud to provide privacy in Web data. Various techniques have been implemented in order to control the malicious attacks. Different tools and software are there to determine such sites. Most of the browsers are built with phishing alert functionality for these cases. Another functionality of Blacklisting has come out to be a promising approach in past but with its dynamic nature of malicious URLs demanding more and more efficient methods. Different systems such as Phish Tank and Wiktionary are provided in order to determine URLs that are malicious and pose threat to the users in real time. Data mining techniques are utilized in order to detect such malicious URLs on a regular basis. Data mining methods use algorithms that first extract the features of the suspected site and check it with the provided classifier. Classifiers are the rules generated using data mining algorithms for determining the legitimate from the illegitimate malicious ones. In this research work there is use of JRip i.e. Ripper algorithm.

References

1. Due, B., Kristiansen, M., Colomo-Palacios, R., & Hien, D. H. T. Introducing Web Data Topics: A Multicourse Experience Report from Norway. Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, 2015, 8(2), 565–569.
2. Shirudkar, K., &Motwani, D. Web-Data Security, 2015, 5(3), 1100–1109.
3. J. Ma, L. Saul, S. Savage and G. Voelker, “Learning to Detect Malicious URLs”, ACM Transactions on Intelligent Systems and Technology, (2011), 1(1), 30:1-30:24.
4. H. S. Choi, B. B. Zhu and H. J. Lee, “Detecting Malicious Web Links and Identifying Their Attack Types”, Proceedings of the 2nd USENIX Conference on Web application development (WebApps), USENIX Association Berkeley, (2011), 1(3), 1-12.
5. B. Eshete, A. Villafiorita and K. Weldemariam, “BINSPECT: Holistic Analysis and Detection of Malicious Web Pages”, Proceedings of the 8th International ICST Conference, SecureComm,(2012), 3(6), 1544-1562.
6. W. Tao, S. Z. Yu and B. L. Xie, “A Novel Framework for Learning to Detect Malicious Web Pages”, Proceedings of the International Forum on Information

Technology and Applications (IFITA), (2010), 1(9), 212-220.
7. W. Zhang, Y. X. Ding, Y. Tang and B. Zhao, "Malicious web page detection based on online Learning algorithm", Proceedings of the International Conference on Machine Learning, (2011), 17(6), 1914-1919.
8. V. L. Le, I. Welch, X. Y. Gao and P. Komisarczuk, "Two-Stage Classification Model to Detect Malicious Web Pages", Proceedings of the International Conference on Advanced Information Networking and Application (AINA), (2011), 15(11), 113-120.
9. M. Cova, C. Kruegel and G. Vigna, "Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code", Proceedings of the International World Wide Web Conference Committee (IW3C2), WWW, (2010), 44(1), 48-58.
10. Y. H. Choi, T. G. Kim and S. J. Choi, "Automatic Detection for JavaScript Obfuscation Attacks in Web Pages through String Pattern Analysis", International Journal of Security and Its Applications, (2010), 22(7), 13-26.
11. R. B. Basnet and A. H. Sung, "Classifying Phishing Emails Using Confidence-Weighted Linear
12. Classifiers", Proceedings of the International Conference on Information Security and Artificial Intelligence (ISAI), (2010), 4(3), 108-112.
13. R. B. Basnet and A. H. Sung, "Learning to Detect Phishing Webpages", Journal of Internet Services and Information Security (JISIS), (2014), 4(1), 21-39.
14. K. Rieck, T. Krueger and A. Dewald, "Cujo: Efficient Detection and Prevention of Drive-by-
15. DownloadAttacks", Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC), (2010), 3(5), 31-39.
16. Eldardiry, H., Bart, E., Liu, J., Hanley, J., Price, B., &Brdiczka, O. Multi-domain information fusion for insider threat detection. In Security and Privacy Workshops (SPW), IEEE (2013),7(6) 45-51
17. Young, W. T., Goldberg, H. G., Memory, A., Sartain, J. F., & Senator, T. E. Use of domain knowledge to detect insider threats in computer activities. In Security and Privacy Workshops (SPW), IEEE, (2013), 11(3), 60-67.
18. Gupta, N., Aggarwal, A., &Kumaraguru, P. bit. ly/malicious: Deep Dive into Short URL based e-Crime Detection. In 2014 APWGSymposium on Electronic Crime Research (eCrime) IEEE, (2014), 5(8) 14-24.
19. Invernizzi, L.,&Comparetti, P. M. Evilseed: A guided approach to finding malicious web pages. In 2012 IEEESymposium on Security and Privacy IEEE, (2012), 80(4), 428-442.
20. Huang, D., Xu, K., & Pei, J. Malicious URL detection by dynamically mining patterns without pre-defined elements. World Wide Web, (2014), 17(6), 1375-1394.
21. Cao, J., Li, Q., Ji, Y., He, Y., &Guo, D. Detection of forwarding-based malicious urls in online social networks. International Journal of Parallel Programming, (2016), 44(1), 163-180.
22. Mekky, H., Torres, R., Zhang, Z. L., Saha, S., &Nucci, A. Detecting malicious http redirections using trees of user browsing activity. InIEEE INFOCOM 2014-IEEE Conference on Computer CommunicationsIEEE, (2014), 55(2), 1159-1167.
23. R. Dhamija, J.D. Tygar, Hearst M, Why phishing works, In: Proc. Int. Conf. Human-Computer Interaction, CHI‟06, Montreal, Quebec, Canada,(2006), 3(9) 581-590.
24. Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang. Knowing your enemy: understanding and detecting malicious web advertising. In T. Yu, G. Danezis, and V. D. Gligor, editors, ACMConference on Computer and Communications Security, ACM, (2012), 15(4) 674–686.
25. H. K. Pao, Y. L. Chou, and Y. J. Lee. Malicious URL detection based on Kolmogorov complexity estimation. In The 2012 IEEE/WIC/ACMInternational Conference on Web Intelligence,IEEE, (2012), 11(7), 541-556.
26. K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time URL spam filtering service. In Proceedings of the 2011 IEEESymposium on Security and Privacy, SP '11, Washington, DC, USA, IEEE (2012),14(5), 202-212.
27. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. Journal of Machine Learning Research, (2006), 7(2), 551–585.
28. S. Garera, N. Provos, M. Chew, and A. D. Rubin. A framework for detection and measurement of phishing attacks. In Proceedings of the 2007 ACM workshop on Recurring malcode, WORM '07, New York, NY, USA, ACM, (2007), 1(7), 1–8.
29. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explorations, (2009), 11(2), 10-18.
30. S. Sheng, B. Wardman, G. Warner, L.F. Cranor, J. Hong, C. Zhang, An empirical analysis of phishing blacklists, In: Proc. 6th Int. Conf. Email and Anti-Spam, CEAS‟09, Mountain View, California, USA, (2009), 7(1), 83-88.
31. H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transaction on Pattern Analysis and Machine Intelligence IEEE (2005), 27(5), 1226-1238.
32. J. Ma, L.K. Saul, S. Savage, G.M. Voelker, Identifying suspicious URLs: an application of large-scale online learning, In: Proc. 26th Annual Int. Conf. Machine Learning, ICML‟09, Montreal, Quebec, Canada, (2009), 30(6), 681-688.
33. J. Ma, L.K. Saul, S. Savage, and G.M. Voelker, beyond blacklists: Learning to detect malicious web sites from suspicious URLs, In: Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, Paris, France, (2009), 17(9), 1245-1254.
34. Y. Zhang, J. Hong, L. Cranor, and CANTINA: a content-based approach to detecting phishing web sites, In: Proc. 16th Int. Conf. World Wide Web, WWW‟07, Banff, Alberta, Canada, (2007), 14(3), 639-648.