2017 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT), May 19-20, 2017, India

DETECTION OF MALICIOUS URLs IN BIG DATA USING RIPPER ALGORITHM

Sonika Thakur,
Centre for Computer Science & Technology,
Central University of Punjab,
sonikathakur1693@gmail.com,

Er.Meenakshi,
Centre for Computer Science & Technology,
Central University of Punjab,
meenakshi.cup@gmail.com

AkanshaPriya,
Centre for Computer Science & Technology,
Central University of Punjab,
akanshpriya25@gmail.com

Abstract— 'Big Data' is the term that describes a large amount of datasets. Datasets like web logs, call records, medical records, military surveillance, photography archives, etc. are often so large and complex, and as the data is stored in Big Data in the form of both structured and unstructured therefore, big data cannot be processed using database queries like SQL queries. In big data, malicious URLs have become a station for internet criminal activities such as drive- by-download,information warfare, spamming and phishing. Malicious URLs detection techniques can be classified into Non-Machine Learning (e.g. blacklisting) and Machine learning approach (e.g. data mining techniques). Data mining helps in the analysis of large and complex datasets in order to detect common patterns or learn new things.Big data is the collection of large and complex datasets and the processing of these datasets can be done either by using tool like Hadoop or data mining algorithms. Data mining techniques can generate classification models which is used to manage data, modelling of data that helps to make prediction about whether it is malicious or legitimate. In this paper analysis of RIPPER i.e. JRip data mining algorithm has been done using WEKA tool. A training dataset of 6000 URLs has been made to train the JRip algorithm which is an implementation of RIPPER algorithm in WEKA. Training dataset will generate a model which is used to predict the testing dataset of 1050 URLs. Accuracy are calculated after testing process. Result shows JRip has an accuracy of 82%.

Keywords—*Big Data, Data Mining, JRip, Weka, True positive rate, True negative rate, False positive rate, False negative rate, Accuracy.*

I. INTRODUCTION

Big data is the term that describes a large amount of datasets. Datasets like web logs, call records, medical records, military surveillance, photographyarchives, etc. are often so large and complex, and as the data is stored in Big Data in the form of both structured and unstructured therefore, big data cannot be processed using database queries like SQL queries[1]. The amount of data generated every day in the world is massive. The increasing volume of digital and social media andthe internet of thingsis fueling it even further. The rate of data growth is surprising and this growth rate is really very fast, with variety(not necessarily structured) and contains a wealth of information that can be a key togain the valuable knowledge in businesses. "Big data" is the term for a collection of data sets so large and complex that it becomes difficult to process it using traditional database management tools such as Relational Database Management System(RDBMS)[2]. RDBMS can't handle, huge, unstructured and complex data. The processing of large amount of dataset in RDBMS takes time as it is generally designed for fixed amount of data. So a different tools and techniques is needed to process the big datasets.

Now days everyone uses internet. Increase in internet usage results into increase in the number of websites linked with one particular organization. For example Wikipedia is an organization having thousands of webpages. Every webpages has different URL. Thus Wikipedia has its own big data of URLs, and it is increasing day by day. Similarly, there are many other organization like Google, Bing, yahoo, etc. which maintains big data database of URLs.

Some organization does not concern about legitimacy websites that are present in their big data database. The organization does not have intension to add malicious URL in there database but there are intruder which can add such pages in the organization's database. So there is a need for a model, which can be directly apply on the big data database of URLs of these organization and check the legitimacy of every URLs[3].

Some models can be generated using datamining techniques. For generating such models analysis of big data of URLs is also needed. This analysis is helpful in determining the feature of URLs. How these features helped for analysis of big data database of URLs, these features are utilized by data mining. Data mining first generate a model for prediction of big data database of URLs whether it is malicious URL or legitimate URL[4].

II. RELATED WORK

Many algorithm like PRISM, ONE, C4.5, etc. has been proposed by the researchers to analyze the malicious URLs. Until now, these algorithms haven't performed much well, the accuracy was much less, and the error rate was higher in real time. Thesealgorithms are designed for generation of classification rules[5],[6]. Different algorithms have different efficiency or accuracy issues. Ripper (JRIP) algorithmcan also be used to detect malicious URLs detection. It is based on Sequential covered algorithm. In Sequential covering algorithm, the rules are extracted directly from the training dataset. JRIP calculated the FOIL's Gain for every attribute. The attribute with the highest FOIL's Gain is selected.Again FOIL's Gain for rest of the attributes are calculated and ruleset further grow according to the FOIL's Gain after this ruleset is created. This JRIP algorithm needs to be compared with existing.

978-1-5090-3704-9/17/$31.00 © 2017 IEEE

III. MALICIOUS URLS DETECTION TECHNIQUES

URLs have become a common channel to facilitate Internet criminal activities such as drive-by-download, spamming, phishing and information warfare[7]. Many attackers use fake web sites for spreading malicious programs or stealing identities. In this research data mining technique has been used to detect malicious URLs from the big data. Malicious URLs detection techniques can be classified into Non-Machine Learning and Machine Learning approach[8].

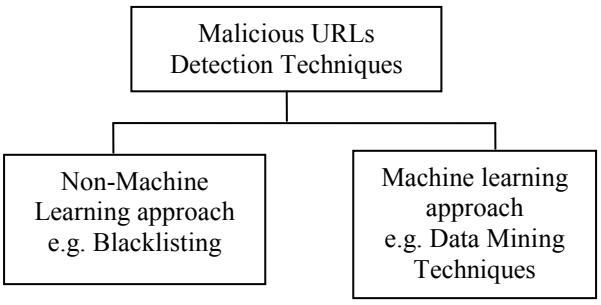


Figure 1: Malicious URLs Detection Technique

The non-machine learning approach, suffer from poor generalization to new malicious URLs and unseen malicious patterns. One of the examples of non-machine learning approach is blacklisting. Many web browser uses blacklisting method (e.g. google, yahoo, etc.)[9]. Web browsers maintains a list of malicious URLs, this list is known as blacklist. Web browsers update their blacklist regularly. But some malicious URLs are online for only few hours. Because of this short life of malicious URLs, they may not get updated in blacklist and because of this web browsers may not be able to detect malicious URLs. This is the main disadvantage of blacklist approach[8].

Most used machine learning method is data mining techniques. Data mining algorithms can be used to generate aclassification model that can identify the malicious URLs[8]. Malicious URLs generally have some common features like host name, favicon, etc. Data mining technique can utilize these feature in order to check URLs whether they are malicious or not. Data mining techniques generate a classification model from a training dataset (in this set URLs category are already defined i.e. malicious or legitimate) and then this model is applied on the testing dataset (in this set URLs category is not defined i.e. malicious or legitimate) to predict whether a URL is legitimate or malicious. How these features helped for analysis of big data database of URLs, these features are utilized by data mining[10]. Data mining first generate a model for prediction of big data database of URLs whether it is malicious URL or legitimate URL. Data mining model has the ability to analyze the big data. Data mining can take a very large dataset for analysis purpose. A good model can only be generated by datamining if it is trained with large and relevant datasets[10].

IV. RIPPER ALGORITHM FOR MALICIOUS URLS DETECTION

RIPPER algorithmwas designed by cohen in 1995[4].The RIPPER algorithmis one of the rule-based classification algorithm that generates rule-based classifier model which is a set of IF-THEN rules and these rules are extracted directly from the training dataset that's why it is called direct method. It is especially more efficient on large noisy datasets like Big data database[4]. The algorithm progresses through four phases[4]:

1) Growth: In the growth phase, one rule is generated by greedily adding attributes to the rule until the rule meets stopping criteria.
2) Pruning: In the pruning phase, each rule is pruned and made shorter by removing redundancy and reducing length of earlier rule which allows the rule to become better.
3) Optimization: The first growth and prune phase generates rules from empty rule set. Optimization step utilize the rules generated in first growth and prune stage and tries to generate new rules from ruleset. Rules is further optimized by:-
   a) Adding attributes to the original rule using greedy method (i.e. depth first search).
   b) New ruleset is generated after a growing and pruning phase.
4) Selection: In the selection phase, the best rules are kept and the other rules are deleted from the model.

V. METHODOLOGY

Data mining algorithm have been proved to be beneficial for detecting malicious URLs. After analyzing various URLs,their common features like host name, path length, etc. are extracted.

Waikato Environment for Knowledge Analysis (WEKA)tool[11]has been used for the analysis of RIPPER algorithm. JRip is an optimized implementation of RIPPER in WEKA tool which generates rulesets after the evaluation over the training dataset. The different steps involved in the analysis of JRip algorithm is as follows:

1) Collection of both malicious URLs and legitimate URLs from Wiktionary_en_2012-07-21.hdt[12]which is a big data database.
2) Extract the features of URLs to detect the malicious and legitimate URLs.
3) Creation of training dataset and testing dataset on the basis of features extracted.
4) Training of RIPPER using training dataset and generation of rule-based classifier model.
5) Rule-based Classifier model is used to predict the missing values of testing dataset.
6) URLs from testing dataset are predicted by using rule-based classifier model on the basis of different parameters such as True Positive Rate (TP rate), False Positive Rate (FP rate), True Negative (TN Rate), False Negative (FN Rate) and Accuracy.

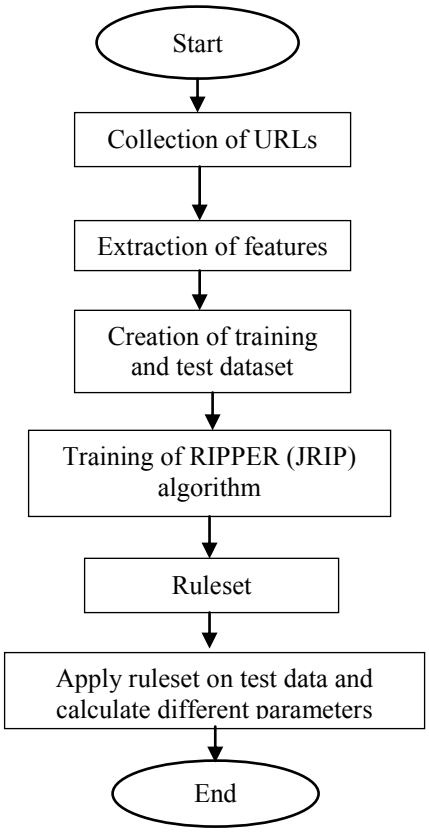


Figure 2: Methodology

A. Collection and Extraction of URLs

Collection of bothlegitimate as well as malicious URLs is done for the extraction of their features. The Malicious as well as legitimate URLs are collected from "Wiktionary_en_2012-07-21.hdt"[12] shown in figure 2. The extracted features of URLs are the basis for determining that whether a URLs is malicious or not. Twenty-five URL's features have been extracted which is used in training and testing dataset for categorizing the URLs.

	Subject	Predicate
1	http://wiktionary.dbpedia.org/resource/	http://usefulinc.com/ns/doap#creator
2	http://wiktionary.dbpedia.org/resource/	http://wiktionary.dbpe...org/terms/hasLangUsag
3	http://wiktionary.dbpedia.org/resource/	http://wiktionary.dbpe...org/terms/hasLangUsag
4	http://wiktionary.dbpedia.org/resource/	http://wiktionary.dbpe...org/terms/hasLangUsag
5	http://wiktionary.dbpedia.org/resource/	http://wiktionary.dbpe...org/terms/hasLangUsag
6	http://wiktionary.dbpedia.org/resource/	http://wiktionary.dbpe...org/terms/hasLangUsag
7	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
8	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
9	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
10	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
11	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
12	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
13	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
14	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
15	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
16	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
17	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense
18	http://wiktionary.dbpedia.org/resource/	http://www.monnet-project.eu/lemon#sense

Figure 3: Wiktionary database

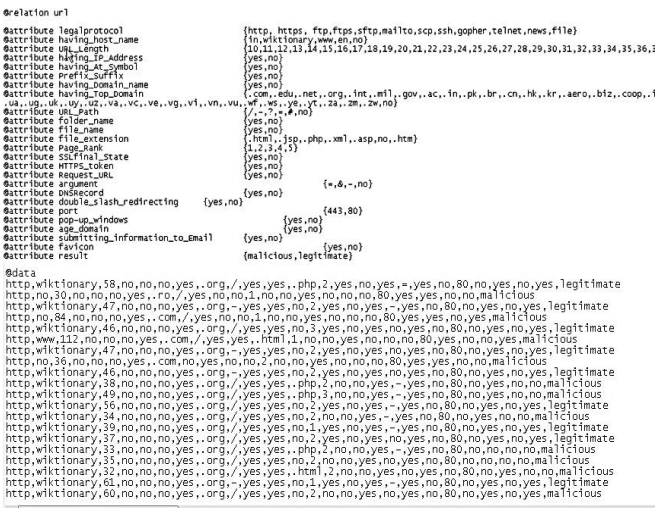
B. Creating dataset

After extracting the features of various URLs, training and testing dataset is created. Both datasets are in .arff fileformat which is supported by WEKA[11]. Training and testing datasets are different from one another. In training dataset the category (i.e. malicious or legitimate) of URLs are already known. This category is represented as 'Result' attribute. Whereas in testing dataset the 'Result' attribute contains missing values.

A training dataset contains three parts[11]:

1) Relation:It is just a name for the training dataset.
2) List of attribute: This contains the features of URLs and their values i.e. attribute value which can be either nominal or numeric
3) Data: This contains the URLs' data according to the attributes.

Training dataset of 6000 URLs has been created out of which 4000 URLs are malicious and 2000 URLs are legitimate.



```
@relation url

@attribute legalprotocol                {http, https, ftp,ftps,sftp,mailto,scp,ssh,gopher,telnet,news,file}
@attribute having_host_name             {tn,wiktionary,www,en,no}
@attribute URL_Length                   {10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,3
@attribute having_IP_Address            {yes,no}
@attribute having_at_Symbol             {yes,no}
@attribute Prefix_Suffix                {yes,no}
@attribute having_Domain_name           {yes,no}
@attribute having_Top_Domain            {.com,.edu,.net,.org,.int,.mil,.gov,.ac,.in,.pk,.br,.cn,.hk,.kr,.aero,.biz,.coop,.i
.ua,.ug,.uk,.uy,.uz,.va,.vc,.ve,.vg,.vi,.vn,.vu,.wf,.ws,.ye,.yt,.za,.zm,.zw,no}
@attribute URL_Path                     {/,-,?,=,#,no}
@attribute Folder_name                  {yes,no}
@attribute File_name                    {yes,no}
@attribute File_extension               {.html,.jsp,.php,.xml,.asp,no,.htm}
@attribute Page_Rank                    {1,2,3,4,5}
@attribute SSLfinal_State               {yes,no}
@attribute HTTPS_token                  {yes,no}
@attribute Request_URL                  {yes,no}
@attribute argument                                 {=,&,-,no}
@attribute DNSRecord                    {yes,no}
@attribute double_slash_redirecting     {yes,no}
@attribute port                                     {443,80}
@attribute pop-up_windows                       {yes,no}
@attribute age_domain                           {yes,no}
@attribute submitting_information_to_Email      {yes,no}
@attribute favicon                                  {yes,no}
@attribute result                       {malicious,legitimate}

@data
http,wiktionary,58,no,no,no,yes,.org,/,yes,yes,.php,2,yes,no,yes,=,yes,no,80,no,yes,no,yes,legitimate
http,no,30,no,no,no,yes,.ro,/,yes,no,no,1,no,no,yes,no,no,no,80,yes,yes,no,no,malicious
http,wiktionary,47,no,no,no,yes,.org,-,yes,yes,no,2,yes,no,yes,-,yes,no,80,no,yes,no,yes,legitimate
http,no,84,no,no,no,yes,.com,/,yes,no,no,1,no,no,yes,no,no,no,80,yes,yes,no,yes,malicious
http,wiktionary,46,no,no,no,yes,.org,/,yes,yes,no,3,yes,no,yes,no,yes,no,80,no,yes,no,yes,legitimate
http,www,112,no,no,no,yes,.com,/,yes,yes,.html,1,no,no,yes,no,no,no,80,yes,no,no,yes,malicious
http,wiktionary,47,no,no,no,yes,.org,-,yes,yes,no,2,yes,no,yes,no,yes,no,80,no,yes,no,yes,legitimate
http,no,36,no,no,no,yes,.com,no,yes,no,no,2,no,no,yes,no,no,no,80,yes,yes,no,no,malicious
http,wiktionary,46,no,no,no,yes,.org,-,yes,yes,no,2,yes,no,yes,no,yes,no,80,no,yes,no,yes,legitimate
http,wiktionary,38,no,no,no,yes,.org,/,yes,yes,.php,2,no,no,yes,-,yes,no,80,no,yes,no,no,malicious
http,wiktionary,49,no,no,no,yes,.org,/,yes,yes,.php,3,no,no,yes,-,yes,no,80,no,yes,no,no,malicious
http,wiktionary,56,no,no,no,yes,.org,/,yes,yes,no,2,yes,no,yes,-,yes,no,80,no,yes,no,yes,legitimate
http,wiktionary,34,no,no,no,yes,.org,/,yes,yes,no,2,no,no,yes,-,yes,no,80,no,yes,no,no,malicious
http,wiktionary,39,no,no,no,yes,.org,/,yes,yes,no,1,yes,no,yes,-,yes,no,80,no,yes,no,yes,legitimate
http,wiktionary,37,no,no,no,yes,.org,/,yes,yes,no,2,yes,no,yes,no,yes,no,80,no,yes,no,yes,legitimate
http,wiktionary,33,no,no,no,yes,.org,/,yes,yes,.php,2,no,no,yes,-,yes,no,80,no,no,no,no,malicious
http,wiktionary,35,no,no,no,yes,.org,/,yes,yes,no,2,no,no,yes,no,yes,no,80,no,no,no,no,malicious
http,wiktionary,32,no,no,no,yes,.org,/,yes,yes,.html,2,no,no,yes,no,yes,no,80,no,yes,no,no,malicious
http,wiktionary,61,no,no,no,yes,.org,-,yes,yes,no,1,yes,no,yes,-,yes,no,80,no,yes,no,yes,legitimate
http,wiktionary,60,no,no,no,yes,.org,/,yes,yes,no,2,no,no,yes,no,yes,no,80,no,yes,no,yes,malicious
```

Figure 4: Training dataset

C. Simulation environment of RIPPER (JRIP) algorithm

Table I shows the simulation environment to predict whether a URL in the testing dataset is malicious or legitimate using rule-based classifier model generated by JRip algorithm with training dataset:

Table I: Simulation Environment

S.no	Algorithm	Training dataset	Testing dataset	Parameters
1.	JRIP (RIPPER) Algorithm	Malicious URLs-400 Legitimate URLs-200	Malicious URLs-300 Legitimate URLs-150	TP Rate TN Rate FP Rate FN Rate Accuracy

1) Training Dataset: Training dataset is of legitimate as well as malicious URLs. Training dataset will contain the 200 legitimate URLs and 400 malicious URLs.
2) Ripper algorithm will generate Rule-based classifier model.

3) Testing Dataset: Testing dataset of 450 URLs is created in which 150 are legitimate URLs and 300are malicious URLs.

D. Preprocessing of training dataset and training of JRIP algorithm

When a dataset is loaded in WEKA, it is first preprocessed. After preprocessing a list of all the attributes with their values is shown. For every attribute, calculation is done and number of instances having nominal value (like strings, characters) and numeric value (use of numbers i.e. 0 and 1) of attribute is determined[11].

JRip algorithm is selected in WEKA and it is trained with the training dataset. After this, the rule-based classifier modelis saved for prediction of malicious as well as legitimate URLs.Ripper algorithm uses FOIL's information gain[13] in order to choose the attributes that can generate best rule. Every rules in RIPPER algorithm is created in the form of disjunctive normal form i.e. $R = r_1 \vee r_2 \vee \ldots . r_k$, where R is rule set and r_i's are rules. Attribute 'Favicon' is chosen first as it has highest FOIL's information gain. In the first rule attribute 'SSL' is added with and ('∧') conjunction as it improves the rule's quality. Attribute 'SSL' is chosen as when it is added with attribute 'favicon' it gives larger FOIL's information gain. The other rules can be find by the same procedure[13].

After training of the JRip algorithm, the accuracy obtained for the algorithm is 98.1667%.

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

(favicon = yes) and (SSLfinal_State = yes) and (file_name = yes) => result=legitimate (410.0/0.0)
(favicon = yes) and (file_name = no) and (URL_Path = =) => result=legitimate (380.0/0.0)
(favicon = yes) and (file_name = no) and (Page_Rank = 5) => result=legitimate (180.0/0.0)
(favicon = yes) and (having_host_name = en) => result=legitimate (290.0/0.0)
(favicon = yes) and (file_name = no) and (Page_Rank = 4) => result=legitimate (100.0/0.0)
(favicon = yes) and (file_name = no) and (URL_Path = /) and (Page_Rank = 1) and (SSLfinal_State = yes) => result=legitimate (100.0/0.0)
(favicon = yes) and (Page_Rank = 2) and (URL_Length = 54) => result=legitimate (30.0/0.0)
(favicon = yes) and (Page_Rank = 2) and (URL_Length = 55) => result=legitimate (40.0/0.0)
(favicon = yes) and (URL_Length = 56) => result=legitimate (40.0/0.0)
(favicon = yes) and (file_name = no) and (legalprotocol = http) and (DNSRecord = yes) => result=legitimate (40.0/0.0)
(double_slash_redirecting = yes) and (folder_name = no) => result=legitimate (190.0/20.0)
(favicon = yes) and (Page_Rank = 2) and (argument = =) => result=legitimate (20.0/0.0)
(favicon = yes) and (Page_Rank = 2) and (URL_Length = 49) => result=legitimate (20.0/0.0)
(favicon = yes) and (URL_Length = 59) => result=legitimate (20.0/0.0)
(favicon = yes) and (Page_Rank = 2) and (URL_Length = 64) => result=legitimate (10.0/0.0)
(favicon = yes) and (URL_Length = 48) => result=legitimate (10.0/0.0)
 => result=malicious (4120.0/140.0)

Number of Rules : 17


Time taken to build model: 1.34 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        5840               97.3333 %
Incorrectly Classified Instances       160                2.6667 %
```

Figure 5: Rule-Based Classifier Model

Figure 5 shows that JRIP generates 17 rulesets which helps to identify whether a URL is malicious or legitimate in big data database of URLs.

E. Testing of JRIP algorithm

Some samples of URLs are taken from "Wiktionary_en_2012-07-21.hdt"[12] to check the accuracy of the classifier model like 1000, 1050, 2000, etc. These URLs are made as testing dataset. The testing dataset is different from training dataset. In the testing dataset, the category (i.e. malicious and legitimate) of the URLs is unknown. It contains missing values. This missing attribute value is filled by the rule-based classifier which was saved as earlier.The testing dataset contains 1050 URLs in which 700 are malicious URLs and 350 are legitimate URLs.

```
@attribute port                                 {443,80}
@attribute pop-up_windows                       {yes,no}
@attribute age_domain                           {yes,no}
@attribute submitting_information_to_Email      {yes,no}
@attribute favicon                              {yes,no}
@attribute result                               {malicious,legitimate}

@data
http,no,256,no,no,no,yes,.br,/,yes,no,no,1,no,no,yes,no,yes,no,80,no,no,no,no,?
http,no,100,no,no,no,yes,.ru,/,yes,yes,no,1,no,no,yes,no,yes,no,80,no,no,no,no,?
http,www,137,no,no,no,yes,.com,/,yes,yes,.html,1,no,no,yes,no,yes,no,80,no,no,no,no,?
http,no,67,no,no,no,yes,.gov,/,yes,yes,.php,4,no,no,yes,no,yes,no,80,no,no,no,no,?
http,no,71,no,no,no,yes,.dz,/,yes,yes,.php,3,no,no,yes,no,yes,no,80,no,no,no,no,?
http,no,130,no,no,no,yes,.net,/,yes,yes,.html,1,no,no,yes,no,yes,no,80,no,no,no,no,?
http,no,130,no,no,no,yes,.net,/,yes,yes,.html,1,no,no,yes,no,yes,no,80,no,no,no,no,?
http,no,130,no,no,no,yes,.net,/,yes,yes,.html,1,no,no,yes,no,yes,no,80,no,no,no,no,?
http,no,130,no,no,no,yes,.net,/,yes,yes,.html,1,no,no,yes,no,yes,no,80,no,no,no,no,?
http,no,93,no,no,no,yes,.il,/,yes,yes,.php,2,no,no,yes,no,yes,no,80,no,no,no,no,?
http,no,26,no,no,no,yes,.com,/,yes,no,no,1,no,no,yes,no,no,no,80,no,no,no,no,?
http,www,37,no,no,no,yes,.com,/,yes,yes,.php,1,no,no,yes,no,no,no,80,no,yes,no,no,?
http,www,39,no,no,no,yes,.by,/,yes,yes,.html,1,no,no,yes,no,no,no,80,no,no,no,no,?
http,no,39,no,no,no,yes,.ua,/,yes,yes,.html,1,no,no,yes,no,no,no,80,no,yes,no,no,?
```

Figure 6: Testing dataset

Figure 6 shows the testing dataset in which the 'result' attribute is set missing (?) for every URL. The rule-based classifier model of JRip algorithm thus predicts the value of 'result' attribute.

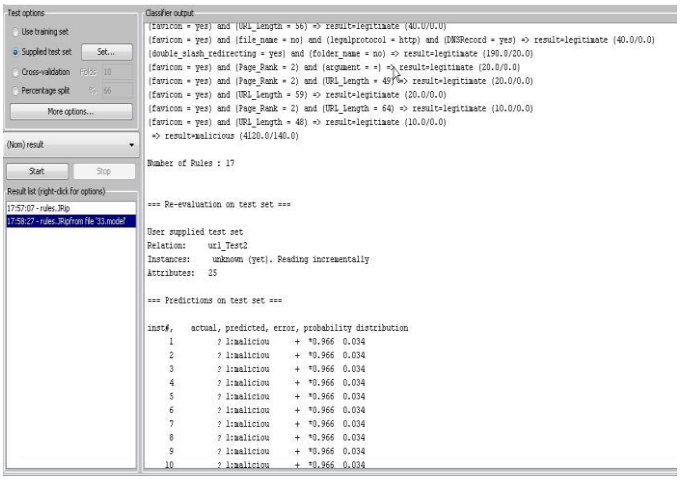


Figure 7: Result of testing dataset

In WEKA from the "Test Options" tab, "Supplied test set" option is selected and testing dataset file is loaded. Finally, right click on the loaded model and run "Re-evaluate model on current test set"[11]. The results are shown in the"Classifier output" panel, under "Predictions on test data" as shown in figure 7.

The symbol "+" occurs only for those items where error is encountered, that is the actual value for 'Result' attribute is different from its predicted value. As in testing dataset the values given for 'Result' attribute is "?" and after prediction 'Result' attribute has value either "malicious" or "legitimate", the symbol "+" occurs under the error column[11].

VI. RESULT

After the prediction of testing dataset using rule-based classifier modela confusion matrix is generated. The confusion matrix is useful for analyzing how well the rule-based classifiers can predict the unknown URLs. Each column of the confusion matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The

true positive, true negative, false positive and false negative counts are used as a metrics and the performance of the algorithm's model is measured in terms of the accuracy.

The confusion matrix obtained is shown in table II.

Table II: Confusion matrix

	Predict malicious	Predict legitimate
Actual malicious	542 (True Positive)	158 (False Negative)
Actual legitimate	25 (False Positive)	325 (True Negative)

The different parameters are calculated from the confusion matrix. The above Table II has two rows and two columns that showsthe number offalse positives (FP), false negatives (FN), true positives (TP), and true negatives (TN).

The parameters that is to be calculated are explained below[13]:

1) **True Positive (TP):** If the outcome from a prediction is malicious and the actual value is also malicious, then it is called a true positive[13].

True positive rate (TPR) =TP/ (TP+FN)

TP is the True Positive, FN is the False Negative.

2) **True negative (TN):**A true negative (TN) has occurred when both the prediction outcome and the actual value are legitimate[13]. True negative rate is calculated as

True Negative Rate =TN/ (TN+FN)

TN is the True Negative, FN is the False Negative.

3) **False Positive (FP):** False Positive (FP) is when the prediction outcome is malicious while the actual value is legitimate[13].

False Positive Rate= FP/ (FP+TN)

4) **False Negative (FN):** False negative (FN) is when the prediction outcome is legitimate while the actual value is malicious[13].

False Negative Rate= FN/ (TP+FN)

5) **Accuracy:** It is also referred as "correct classification rate" and is measured by taking the ratio of correctly prediction from the total URLs[13].

Accuracy= (TP+TN)/ (TP+ FN+FP+ +TN)

Table III: Values of different parameters

Parameter Table for Ripper Algorithm		
Sl. No.	Parameters	Values
1	True Positive Rate	0.802
2	True Negative Rate	0.8714
3	False Positive Rate	0.1285
4	False Negative Rate	0.1971
5	Accuracy	82%

Table III contains the values of parameters after calculation based on the formulas explained above. The value of different parameters are as follows: false positives (FP) is 0.1285, false negatives (FN) is 0.1971, true positives (TP) is 0.802, and true negatives (TN) is 0.8716, accuracy is 82%.

A. Optimization of results

A training dataset of 12000 URLs has been created. If training of ripper algorithm in WEKA is done using this training dataset then a ruleset having 24number of rules are generated. If this new ruleset is used for making the prediction on previously created test dataset of 1050 URLs then it can be observed that prediction error is reduced and give better accuracy.

Table III: Confusion matrix

	Predict malicious	Predict legitimate
Actual malicious	561 (True positive)	139 (False Negative)
Actual legitimate	30 (False Positive)	320 (True Negative)

Table III shows the confusion matrix obtained when test dataset is given to the ruleset generated by the training dataset of 12000 URLs. As there are more number of rules in this ruleset, every URL of the test dataset is predicted in better way. Thus, the accuracy obtained in this case is 83%. It can be stated here that if a training dataset is of large dataset, optimized result is obtained.

CONCLUSION

Big data is a collection of large dataset. Big data is explained using four V's: volume, variety, velocity and veracity. In big data, malicious URLs have become a station for internet criminal activities such as drive- by-download,information warfare, spamming and phishing.

In this research work"Wiktionary_en_2012-07-21.hdt"[12]big data database has been taken for creating training and testing datasets. The URLs' features are analyzed first in order to create training dataset. Twenty-five features of URLs are extracted for generating the two training and testing dataset. A training dataset of 6000 URLs and 12000URLs; testing dataset of 1050 URLs are created by using "Wiktionary_en_2012-07-21.hdt"[12]. The created training datasets is used to train the JRIPalgorithm in WEKA which generates a rule-based classifier model.The 1050 URLs of testing dataset are predicted with rule-based classifier model which is generated by training dataset of 6000 URLs, out of which 542 URLs are detected as malicious and 325 URLs are detected as legitimate. After this accuracy of the generated rule-based classifier model is calculated. The result showsthat the rule-based classifier model of RIPPER algorithm can identify URLs with an accuracy of 82%.The 1050 URLs of testing dataset are predicted with rule-based classifier model which is generated by training dataset of 12000 URLs, out of which 561 URLs are detected as malicious and 320 URLs are detected as legitimate. After this accuracy of the generated rule-based classifier model is calculated. The result showsthat the rule-based classifier model of RIPPER algorithm can identify URLs with an accuracy of 83%. It can be stated here

that if a training dataset is of large dataset, optimized result is obtained.

REFERENCES

[1] M. Wessler, "Concept of Big Data," in *Big Data Analytics For Dummies*, Hoboken, Wiley, 2013, pp. 5-25.

[2] C. Snijders, U. Matzat and U. D. Reips, ""Big Data" : Big Gaps of Knowledge in the Field of Internet Science," *International Journal of Internet Science,* vol. 7, no. 1, pp. 1-5, 2012.

[3] F. Hu, Big data sharing,storage and security, crc, 2016.

[4] P. N. Tan, V. Kumar and M. Steinbach, Introduction to data mininig, Boston: pearson, 2006.

[5] R. Jabri and B. Ibrahim, "Phishing Websites Detection Using Data Mining Classification," *Society for Science and Education United Kingdom,* vol. 3, no. 4, pp. 42-51, 2015.

[6] E. B. Rajsingh and S. C. Jeeva, "Phishing URL detection-based feature selection to classifiers," *Inder Science Online,* vol. 9, no. 2, pp. 33-40, 2014.

[7] M. S. Lin, C. Y. Chiu, Y. J. Lee and H. K. Pao, "Malicious URL Filtering – A Big Data Application," in *2013 IEEE International Conference on Big Data*, Taipei, 2013.

[8] P. Zhao and S. C. Hoi, "Cost-Sensitive Online Active Learning with Application to Malicious URL Detection," in *ACM*, Chicago, 2013.

[9] J. Makey, "Blacklists Compared," 2010. [Online]. Available: https://www.sdsc.edu/~jeff/spam/Blacklists_Compared.html. [Accessed 17 august 2016].

[10] J. Han and M. Kamber, Data Mining: Concepts and Techniques, USA: Morgan Kaufmann, 2000.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA data mining software: an update," *Article,* vol. 11, no. 1, pp. 10-18, 2009.

[12] "Wikitionary_en_rdt.hdt," 21 july 2012. [Online]. Available: http://www.rdfhdt.org/datasets/. [Accessed 22 july 2016].

[13] I. H. witten and E. Frank, data mining practical machine learning tools and techniques, USA: Elsevier, 2005.