Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN ANALYSIS OF SYSTEM CALLS USING J48 AND JRIP FOR MALWARE DETECTION

***FAIZAL M. A, WARUSIA YASSIN, NUR HIDAYAH M. S, SR SELAMAT, RAIHANA SYAHIRAH ABDULLAH**

Department of System and Computer Communication, Faculty of Information and Communications Technology, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

E-mail: *faizalabdollah@utem.edu.my, s.m.warusia@utem.edu.my, nurhidayahmohdsaudi@gmail.com, sitirahayu@utem.edu.my, raihana.syahirah@utem.edu.my

ABSTRACT

The evolution of malware possesses serious threat ever since the concept of malware took root in the technology industry. The malicious software which is specifically designed to disrupt, damage, or gain authorized access to a computer system has made a lot of researchers try to develop a new and better technique to detect malware but it is still inaccurate in distinguishing the malware activities and ineffective. To solve the problem, this paper proposed the integrated machine learning methods consist of J48 and JRip in detecting the malware accurately. The integrated classifier algorithm applied to examine, classify and generate rules of the pattern and program behaviour of system call information. The outcome then revealed the integrated classifier of J48 and JRip outperforming the other classifier with 100% detection of attack rate.

Keywords: *Malware Detection, System Call, Machine Learning, Classifier, J48 and JRip*

1. INTRODUCTION

According to [1], malware is a program that has a malicious intention, whereas [2] has defined it as a generic term that encompasses viruses, Trojans, spyware and other intrusive codes. Malware refers to viruses, worms, ransomware, Trojan horses, key-loggers, root-kits, spyware, adware and malicious programs [3]. As the networks develop massively in size and complexity, many researchers propose techniques for classification and detection of malware especially in system call as it is the most important event of being traced for recognizing malware behaviour. System calls are defined as the fundamental interface between an application and the operating system kernel [4]. In the operating system, all the tasks and services required by the malware to execute malicious action through system calls. As a result, any execute malware activity can be monitored by observing the patterns in the system calls include opening a file, running a thread, writing to the registry or opening a network connection [21]. Thus, it is important to track the activity of the system call through malware execution in order to characterize the malware behaviour.

The various research focused on malware detection based on machine learning and classification method stated that the input data for statistical approach can use the features of the behaviour of malware such as system call. Another researcher also tries to use advanced methods of machine learning in the probabilistic model in detecting malicious system call sequences [5]. Even though various classifier e.g. Naive Bayes, Decision Tree, Support Vector Machine, Neural Network and so forth have been proposed, to derived information form these classifier required a deep knowledge and understanding within the subject matter such as altering the parameter [6][7] and adding new features that might influence the output. Thus, an algorithm that will facilitate the user, comprehensive and less complex is most preferred. This motivate researcher to come with the solution by introducing rule-based classification models in which the rules produced by analyses the logic of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the classifier decision and eliminates the uncertain consequences represented by classifiers [6].

However, the current detection method has a drawback in differentiating the anomalous behaviours in system call more precisely. Efficiency and accuracy are two essential aspects of performing malware classification and detection in the system call. In order to classify the behaviour of malware the frequency of system calls made by each malware activity, a method consisting of data collection, feature extraction and integrated learning approach are considered. Therefore, the J48 and Jrip classifier with association rules mining will be used in this paper to select significant features of a dataset. The rules are built by using this two classifier and then the rules have been used to assign ranks of the feature. To this effect, we have achieved an acceptable level of accuracy in classifying the activity as malicious or benign using the J48 and JRip algorithm.

The remainder of this paper is presented as follows: Section 2 discusses the related study and section 3 presents the methodology used for this paper. Section 4 presents the discussion of the results. Section 5 concludes the paper and presents future work directions.

2. LITERATURE REVIEW

Recently, many malware researchers have focused on data mining to distinguish unknown malware. Data mining is known as the process of discovering patterns in data [8]. Meanwhile, classification of malware is one of machine learning approach that has been used broadly for different data mining problems [9]. Many of the researchers have proposed the method of malware classification and detection by using several of the classifiers in order to obtain high accuracy. Yet, selecting an appropriate classification algorithm is very essential since it influences the detection accuracy and performance. There is some previous work for classification algorithm that has been done for several researchers as shown in Table 1.

Table 1. Previous Work (Classification Algorithm)

Author	Purpose	Technique/Classifier	Result
(Norouzi, M. et al., 2016) [10]	Detecting the malware behavior in system call by using data mining approach	Naive Bayes, BayesNet, IBI, J48, Regression and SVM.	J48 classifier has great accuracy with 96.07%.
(Intan N. A. et al., 2017) [11]	Tokenization approach based on system call has been used for mobile attacks.	SVM, Random Forest, Naive Bayes, and J48.	J48 classifier has 95.38% of accuracy.
(Sanya Chaba et al., 2017) [12]	Detecting the malicious of system call running on a host system.	Naive Bayes, Random Forest and Stochastic Gradient Descent (SGD).	SGD achieve 95.5% rate of accuracy.
(M. Mazhar Rathore et al., 2016) [13]	Hadoop has been used in real-time IDS for ultra-high speed big data scenario.	J48, REPTree, Random Forest, Conjunctive rule, SVM, and Naive Bayes.	J48 and REPTree are the best classifiers in terms of 99.9% accuracy.

For the approach used in [10], there is five classification approach that has been used to identify changes made by malware in the in the system call. The detection of malware features and behaviour will be distinguished by using dynamic analysis. The evaluation results of this research show that J48 classifier provides better accuracy. The In addition, in research [11], a new classification model produces unique behaviour patterns based on system call sequence pattern. The result of this approach producing a higher result in accuracy. Besides, the author detecting the behaviour of the malicious that running on a host system based on system call sequences [12]. The dataset using in this experiment are based on generated from system call log and the quality of dataset has been improved using filtering algorithm.

Furthermore, [13] proposed a system using Hadoop implementation to detect real-time IDS for ultra-high speed big data environment. The four layer of IDS architecture such as capturing layer, filtration and load balancing layer, processing and decision-making has been used in his research.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Lastly, based on literature review, it concludes that J48 decision tree classifier gives the highest detection accuracy rate with 99%. Therefore, this classifier will be used to examine the occurrence of system calls and unknown registry made by each malware activity.

On another hand, the rule induction approaches discover the rule on the basis of greedy and per class label, whereas these classifier algorithm build the trained model from training set which comprises the part of available class labels. This attract various researcher attention in proposed a number of rule-induction and combined algorithm in a decades. Table 2 shows previous work for rule-induction and combined algorithm.

Table 2: Previous Work (Rule-Induction and Combined Algorithm)

Author	Purpose	Technique/Classifier	Result
(Shahzad & Lavesson, 2012) [14]	The author has proposed a malware identification scheme based on machine learning approach called VETO consisting an ensemble learning and preprocessing techniques. The prediction is conducted through a set of combined algorithm while the result is finalized based on the veto voting classifier.	VETO (Combined Algorithm)	The JRip achieved 35 false positive and Veto at 111 while false negative at 41 and at 8, respectively.
(Zakeri, Faraji Daneshgar, & Abbaspour, 2015) [15]	The author proposed a method which is significantly focused on crucial heuristic features as well as fuzzy classifier namely FURIA. Moreover a pre-processing approach also considered to increase the prediction accuracy via avoiding the suspicious exceptions in legitimate files.	Information Gain & Fuzzy Rule Induction Classifier (Combined Algorithm)	FURIA, JRip and J48 has recorded similar accuracy rate at 99% and for false positive rate FURIA achieved better result at 0.008% while RIPPER and J48 at 0.02%.
(Joshi, 2016) [16]	The author proposed a framework comprises a set of rules for intrusion detection in live traffic. First, the algorithm applied to construct behavior pattern of system audit data and extract the important features using improved apriori algorithm. Later, a set of rules generated based on the definition of these features using JRip.	Apriori Algorithm and JRip (Combine Algorithm)	JRip generate the most perfect rules.
(Bhaya & Ali, 2017) [17]	An author has reviewed a series of current data mining algorithm which applied for unknown and known malware identification.	Method reviewed are RIPPER, Naive Bayes, Neural Network, Decision Tree, Support Vector Machine, N-Gram,	Author has concluded that data mining approach can be applied to detect and classify malware behavior. However, the selection of algorithm must be based on few consideration such as scalable, fast and flexible.

Based on Table 2, an author [14] has proposed a combined algorithm based on machine learning approach between ensemble learning and preprocessing techniques called VETO. Using

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

VETO, an arrangements of functional codes are extracted from the malware and legitimate files as a features into several dataset. Later, a series of learning algorithm used to evaluate these dataset while the prediction output generate based on veto voting using ensemble learning. Apart of other available voting approach, the VETO proposed to detect malware activity more accurately.

Moreover, [15] has intention to improve the detection of malware files. The author proposed a combination of learning method between Information Gain (IG) for pre-processing stage and Fuzzy Rule Induction for classifier stage which known as Fuzzy Unordered Rule Induction Algorithm (FURIA). The IG can construct a ratio for every single related feature and estimate its significance in a process to determine either the subject is malicious or not. In addition, the FURIA algorithm has considered as this algorithm adequate to learn fuzzy rules and rule sets as compare to traditional rules. The author [16] has proposed a framework comprises Apriori and JRip algorithm for live-traffic intrusion detection. Using an improved Apriori algorithm, the relevant features extracted after the behavior pattern of system audit data derived. In another word, this algorithm has applied for generate association rules which acquired via frequent item sets data. Subsequently, JRip employed to construct a series of rules based on earlier feature definition.

Additionally, author [17] has presented a survey based article regarding the ability of data mining (DM) approach on malware behavior detection. DM approach usually used in analyzing and disclose concealed knowledge within the data before predict its behaviors. Therefore, DM has gain popularity and still being used in various field including cyber security in identifying and classify malware activities. The author also recommended a combined algorithm could be effective for certain problem such as detection of malware. Even though, various research has been conducted yet there is challenges exist within the field of malware identification more accurately. Thus, a combined learning algorithm has been proposed in this work.

Briefly, from the related works many alternative techniques have been proposed by the researcher in distinguishing malware detection in the system call. Nonetheless, the method still lacks in differentiate the behaviors of malware in the system call and affect the rate of false negatives. The benefit of using system call is the behavioral characteristics of malware detection can be obtained as it gathered in real-time on development hosts. Other than that, by tracing the sequences of the system calls the malicious activity underlying operating system through system calls which cause permanent damage can be detected. Hence,

Currently, the type of malware attack has encountered significant changes and seems the malware attack can be evaded by relying on legitimate system call sequences but the evasion is possible as all available features of system calls do not take into account. Therefore, this research is focused on detecting the behavior of malware in the system call. This research is supported by the author [23] which recommended a combined algorithm could be effective for certain problem such as detection of malware. Thus, a combined learning which is J48 and the Jrip algorithm has been proposed in this work.

3. RESEARCH METHODOLOGY

In this research, there are three phase in the proposed method which is data collection, feature extraction and integrated classifier.

3.1 Data Collection

The process of data collection shows in Figure 1. The data of system call which involve malware and benign dataset are collected using Drakvuf [20] where the frequency rate of data collection is about 4 minute due to the detection of the malware behaviour factor. Firstly, Dom0 will be created in XenServer which is used in Window 7 as its virtual medium. After running the Windows 7 Operating System, the malware that has been selected will be copied inside the DomU. Before injecting the malware inside the Drakvuf, There are a few step need to be considered: 1) Identify the PID process inside kernel32.dll that can be used as a process to inject the malware. 2) Activate screen log utilities to capture all log produce by Drakvuf. The next step is injecting the malware using the PID process that has been chosen from the previous step. The screen log utilities will capture the Drakvuf log for 4 minute. The Drakvuf log will be filtered using the grep command to capture only system call log produced by the Drakvuf. All the log will be save into a file and will be used to the next process to generate the frequency for the system call. The procedure will be repeated for capturing the normal application log except injecting the malware inside the DomU.

Journal of Theoretical and Applied Information Technology
15th July 2018. Vol.96. No 13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

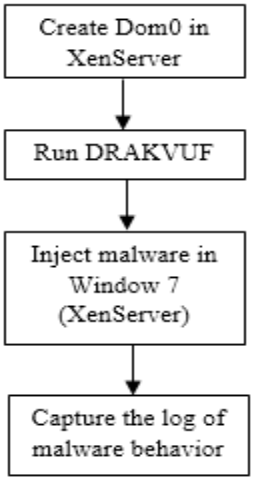

Figure 1. Process of Data Collection

3.2 Feature Extraction

Figure 2 illustrate the process of feature extraction. In this phase, the system call log from the malware application will be filtered. Then, the data frequency selected from the log will be map to the windows 7 system call which has 473 attributes. Next, the data frequency selected from the log will generate and convert into csv file. After that, the frequency csv file will generate into excel and the data will be analyses to produce the output result based on a system call.

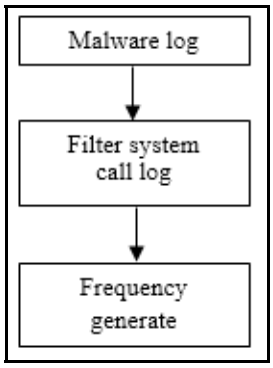

Figure 2. Process of Feature Extraction

3.3 Integrated Classifier

In integrated classifier phase, J48 and JRip classifier will be used in this research as shown in Figure 3. J48 classifier is a Java implementation based on C4.5 algorithm which uses the technique of decision tree to organize the data classification. The classifier needs to create a decision tree based on the attribute values of the available training data and classify the attributes when a set of the feature in training data is identified [18]. The classification criterion of the selected attribute is based on the calculation of entropy and information gain as it is the best attributes to separate the data. Entropy specifies the amount of information that is held where the higher the entropy the more information it contain while information gain emphasis on the significant of the feature or attribute which help in selecting the best split of the data. Thus, J48 decision tree can be divided in three stage namely feature selection, entropy stage and gain information stage. The algorithm of the J48 as Figure 4:

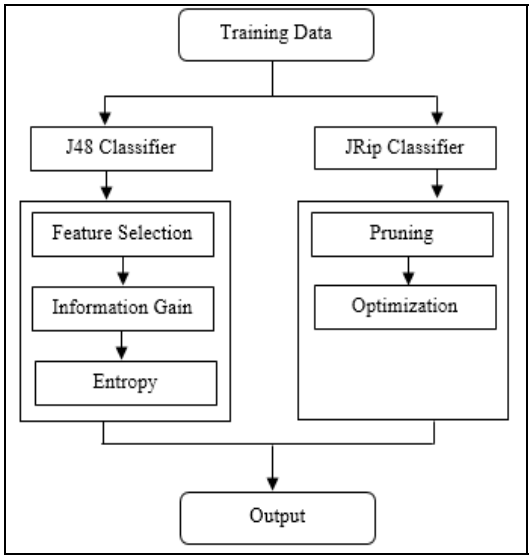

Figure 3. Process of Integrated Classifier

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
Algorithm
INPUT:
DataSet//Training data
OUTPUT
Tree//Decision tree

BUILD(*DataSet)
{
Tree = Ø;
Tree = Create node as root and label with splitting attribute;
Tree = Add arc to node which is root and for each split predicate and label are assigned;

For each arc do

DataSet = Database created by applying splitting predicate to DataSet;

If stopping point reached to this path, then
Tree = create leaf node and label with appropriate class;

Else
Tree' = BUILD(DataSet);
Tree = add Tree' to arc;
}
```

Figure 4. J48 Algorithm [19]

Meanwhile, the JRip known as repeated incremental pruning has proposed by [22]. These algorithm has been selected as it possess several capabilities include strategy in revising and replacing the generated rules which could increase the accuracy for detection, able to handle noisy data and overcome over fitting issues as well as suitable for imbalanced class distributions such as for system calls data. JRip also called as RIPPER as for learning its uses an information gain other than simplify every single rule shortly upon it is learned. Moreover, this algorithm contain a stage for optimizing a rule set by removing learned rule and re-learn it in different perspective or conditions that is not similar with learned rules. This will lead in increasing the accuracy as the rule dynamically updated. The JRip which considered in this work can be divided in four principle stages namely initialization stage, building stage that involving growing and pruning steps, optimization stage and deletion stage. The algorithm procedures work as Figure 5:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

START PROCEDURES

//Begin initialization stage
Initialize a set of Rule R={} for every single Class of C of the less and most frequent

//1. Begin building stage
Re-iterate the growing and pruning step until meet one of the below criteria:

a) error rate surpass 50% *or*
b) the description length (DL) of the rule higher than the lowest description length found so far *or*
c) no more discovered instances of C

//1.1 Begin growing step
proceed to grow rule via adding condition terms to the rule greedily until the rule is accurately perfect 100%, in which every single potential value of every single attribute have to test and select the highest condition in term of Information Gain value: p(log(p/t)-log(P/T)).

//1.2 Begin pruning step
Every single rule is growingly pruned and the formula 2p/(p+n)-1 considered to measure pruned value.

//2. Optimization stage
The beginning rule set {R_i} discovered in this stage. By employing procedures of growing and pruning steps, two variants of every single rule R_i from random data generated. One of the variant generated via empty rule while the later through the original rule which is added with antecedents. The pruning metric applied over here is (TP+TN)/(P+N). Consequently, the lowest DL for entire variant is calculated and only the minimum DL are considered as the final representative of R_i in the ruleset. The building stages will be run again to generate more rules if residual positives still exist after observing {R_i} rules.

//2. Deletion stage
Delete the entire rules from ruleset which have maximum DL and added the remaining rules into resultant set.

END PROCEDURES

Figure 5. JRip Algorithm

4. RESULTS AND DISCUSSION

In this section, the accuracy result will be displayed and documented in brief a detailed analysis to draw the conclusion and the findings. The performance of classification using selected features were measure in the term of accuracy which is equal to (TN+TP)/TP+FP+TN+FN). Table 3 illustrate the classification table.

Table 3. *Classification table*

		Predicted	
		Normal	Attack
Observed	Normal	TN	FP
	Attack	FN	TP

Where:

i) TP: True positive, the number of malware correctly classified
ii) TN: True negative, the number of benign correctly classified.
iii) FP: False positive, the number of benign detected as malware.
iv) FN: False negative, the number of malware detected as benign.

By using the confusion matrix from Table 3, the infer parameters:

i) False Positive = FP / (FP+TP)
ii) False Negative = FN / (FN+TN)
iii) Detection Attack Rate = TP / (FN+TP)
iv) Overall Detection Rate = (TN + TP) / (TN + FP + FN + TP)

The results of the system call classification are presented and discussed. The system call data was extracted and has been analysed by using two different classifiers (J48 and Jrip). The integrated of these two classifiers also has been analysed to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

obtain high detection of malware based on classification table. Table 4 show the result of classification Jrip, J48 and the integrated classifier between Jrip and J48.

Table 4: Classification Result

Algorithm	False Positive (FP)	False Negative (FN)	Detection Attack Rate (DAR)
Jrip	23.07%	50%	95.24%
J48	10.52%	28.57%	90.47%
Jrip + J48	4.54%	0%	100%

Table 4 demonstrates the result of classifier Jrip produces 23.07% false positive as six of the data in the sample has been misclassified as malware which is 'iexplore_syscall', 'firefox_syscall', 'putty_syscall', 'notepad_syscall2', 'calc_syscall' and 'iexplore_syscall'. Meanwhile, in Table 4, the FPR of classifier J48 was reduced to 10.52% but 'putty_syscall' and 'notepad_syscall2' are two data has been misclassified as malware. Contrast with the result of the integrated classifier Jrip+J48 which has 4.54% of false positive because only one data system calls have been misclassified as a malware which is 'firefox_syscall'. In this case, the difficulty of a classifier to distinguish between malware and benign resulted in misclassified all the data as malware. Thus, the combination of Jrip+J48 proves that this classifier has an ability to detect malware or attack in the system call as it results lower false positive compared with Jrip and J48 classifier.

Moreover, the integrated classifier of Jrip+J48 yields lower false negative with 0% because there is no attack found in the system. In the meantime, classifier J48 and Jrip produce false negative with 28.57% and 50% respectively. Jrip has highest false negative compared with J48 as Jrip has been misclassified 'w17_syscall' and 'w20_syscall' as normal while classifier J48 only misclassified 'w20_syscall' as normal. The result shows that it will be risky for an organization if there is attack or data that has been misclassified and not discovered in the system network. Hence, the integrated classifier of Jrip+J48 is the greatest choice as it has lower false negative and yielded rules for classification.

Furthermore, integrated classifier Jrip+J48 has the highest detection attack rate with 100% compared with J48 and Jrip which 90.47% and 95.24%. This shows that integrated classifier Jrip+J48 have abilities to identify categorizing the attack accurately. Since the result of detection attack have 100% abilities to detect the malware, it shows that the classifier has capable to differentiate the classification of normal and attack since it has the better expectation. Therefore, integrated classifier Jrip+J48 fits and good in expecting the outcome variable since it indicates the increase in of the correct percentage for the classification of the attack compared with another classifier.

In addition, the integrated classifier between J48 and Jrip significantly improved the result despite the data being imbalanced. Other than that, we identified that the integrated classifier of these two classifiers turned out to produce the excellent result among the other approach. Besides, the integrated classifier yielded the best outcomes in identifying malware when classified against the benign samples. Finally, we conclude that, the integrated classifier between J48 and Jrip as a viable option for the malware detection in the system call.

5. TESTING AND VALIDATION

In this section, the outcomes of the system call classification are presented and discussed. The system call data was extracted and has been analyzed by using different classifiers such as J48, JRip, Naïve Bayes, OneR, PART, Random Forest, Support Vector Machine (SVM) and Neural Network The integrated classifiers (J48 and JRip) also has been analysed in order to obtain high detection of malware based on classification table. Table 5 show the result of classification while Table 6 shows the result of the misclassified data by the algorithm.

Based on Table 5, it concludes that the JRip+J48 classifier and PART classifier has the higher value of accuracy with 96.42% compared to another classifier. However, the PART classifier has the high value of false negative compared to JRip+J48 classifier with 12.5% as only one data system calls have been misclassified as a malware which is 'w17_syscall'.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Meanwhile, ONeR classifier has the lowest value of accuracy among the other classifier with 25%. The value of false negative for OneR is very high which 75% as it has been misclassified 21 data of system call as a normal data. The data are 'w24_syscall', 'w89_syscall', 'w12_syscall', 'w54_syscall', 'w22_syscall', 'w34_syscall', 'w17_syscall', 'w58_syscall', 'w20_syscall', 'w70_syscall', 'w21_syscall', 'w30_syscall', 'w66_syscall', 'w55_syscall', 'w46_syscall', 'w50_syscall', 'w87_syscall', 'w18_syscall', 'w39_syscall', 'w94_syscall' and 'w84_syscall' as shown in Table 6.

Moreover, PART and OneR does not generate any false alarm as the value of positive rate is 0% but the classifier still can detect malware in the system call. This showed that the organization can be very dangerous since of lots of attacks were not discovered and the false alarm does not produce. Hence, it concludes that the classifier of PART and OneR are not the variable option for malware detection in system call as the classifier cannot accurately distinguish the malware.

Table 5. Classification Result

Algorithm	False Positive (FP)	False Negative (FN)	Detection Attack Rate (DAR)
JRip	23.07%	50%	95.24%
J48	10.52%	28.57%	90.47%
JRip + J48	4.54%	0%	100%
Naïve Bayes	13.04%	20%	95.23%
OneR	0%	75%	0%
PART	0%	12.5%	95.23%
Random Forest	19.23%	0%	100%
SMO	4.76%	14.28%	95.23%
Nueral Network	25%	0%	100%

Table 6. *Result of Misclassified Data by Algorithm*

Algorithm	False Positive (FP)	False Negative (FN)
JRip	iexplore_syscall, firefox_syscall putty_syscall, notepad_syscall2, calc_syscall, iexplore_syscall2	w20_syscall
J48	putty_syscall, notepad_syscall2	w17_syscall, w20_syscall
Jrip + J48	firefox_syscall	-
Naïve Bayes	putty_syscall, notepad_syscall2, calc_syscall	w17_syscall
OneR	-	w24_syscall, w89_syscall, w12_syscall, w54_syscall, w22_syscall, w34_syscall, w17_syscall, w58_syscall, w20_syscall, w70_syscall, w21_syscall, w30_syscall, w66_syscall, w55_syscall, w46_syscall, w50_syscall, w87_syscall, w18_syscall, w39_syscall, w94_syscall, w84_syscall
PART	w17_syscall	-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Random Forest	iexplore_syscall, firefox_syscall putty_syscall, notepad_syscall2, iexplore_syscall2	-
SVM	putty_syscall	w17_syscall
Neural Network	putty_syscall, firefox_syscall, iexplore_syscall, iexplore_syscall2, notepad_syscall2, winrar_syscall, calc_syscall	-

In addition, JRip+J48, Random Forest and Neural Network classifier has the higher value for detection attack rate which is 100%. Yet, Neural Network and Random Forest still has the high value of false positive rate with 25% and 19.23%. In this case, the classifier of Neural Network has been misclassified 'putty_syscall', 'firefox_syscall', iexplore_syscall', 'iexplore_syscall2' and 'notepad_syscall2', 'winrar_syscall' and 'calc_syscall' as a malware data. Nonetheless, Random Forest not classified 'winrar_syscall' and 'calc_syscall' as a malware data while JRip+J48 only classified 'firefox_syscall' as a malware data. Thus, it shows that JRip+J48 classifier is the best in detecting malware attack in the system call compared to another classifier.

Finally, based on the results achieved, it concludes that JRip+J48 classifier still triumphs over another approach for the malware detection in the system call. Besides, the JRip+J48 classifier has the capability to recognizing malware attack without misclassified data and produce the best efficient outcome. Hence, using the JRip+J48 classifier is more suitable for expecting the outcome variable since it indicates the increasing number of correct percentage for the classification of the attack compared to another classifier. The result of JRip+J48 classifier has been outperforming other as shown in Table 5. Figure 6 shows a set of rules generated based on the results obtained (Table 3) from the usage of JRip+J48 classifier. The set of rules consists of four rules and stated that if only all the rules are fulfilled, then it will classify the data as normal and respectively as malware if the rules are not met. Therefore, these rules are good for future reference to detect malware with reduced false negative as the main purpose is to detect malware in the system call accurately.

In addition, the proposed approach (J48+JRip) significantly improved the result despite the data being imbalanced yet yielded the best outcomes in identifying malware when classified against the benign samples. Other than that, the integrated of these two classifiers turned out to produce the excellent result among the other approach. Finally, it is concluded that the proposed approach (J48+JRip) as a viable option for the malware detection in the system call.

```
JRip.J48 rules:
===========

IF (NtIsUILanguageComitted = 0) AND (NtTerminateProcess = 0)
   THEN Normal
IF (NtMapCMFModule = 6)
   THEN Normal
IF (NtAlpcCreatePort = 9)
   THEN Normal
ELSE Malware
```

Figure 6. JRip+J48 Rule

6. OPEN RESEARCH ISSUE

There are many techniques that have been used by researchers in order to detect malware activity. Using only one detection method to detect the overall malware activities becomes a new challenge due to the difference attack behaviour that will affect the feature selection on the detection technique. Besides, identify the most significant feature and classification algorithm in malware

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

detection is an important factor to increase the accuracy of malware detection. The classification algorithm must fit with the dataset and produce high accuracy rate of detection. Moreover, constraints on malware detection (i.e. cannot differentiate and recognize the new malware activity precisely) need to be improved. Therefore, the main contribution of this work is proposed method which consists of three phase and generates the rule of integrated J48 and JRip classifier to distinguish malware behavior in the system call.

7. CONCLUSION

As a conclusion, from the related works shows that mostly researcher concludes that J48 and Jrip classifier provide the better accuracy. Some of them also suggest that the combined algorithm are effective for malware detection. The result from our proposed method shows that the integrated classifier between J48 and JRip is the best approach and more efficient to distinguish malware as it gives high accuracy in this research. Therefore, it proves that the integrated algorithm that has been proposed in this paper produces high accuracy rate of detection. The limitation of this study is the feature extracted from the system call and only use two type of malware with the different variant. For future works, it is recommended to develop and analyse a real behavioural antivirus platform based on classification via the integrated classifier algorithm

ACKNOWLEDGEMENTS

This work has been supported under Universiti Teknikal Malysia Melaka research grant Gluar/CSM/2016/FTMK-CACT/100013 and KPT MyBrain15. The authors would like to thank to Universiti Teknikal Malaysia Melaka, Cybersecurity Malaysia and all members of CMERP INSFORNET research group for their incredible supports in this project.

REFERENCES

[1] Liao Grini, L.S. “Feature extraction and static analysis for large-scale detection of malware types and families”, *Master's thesis*, Gjovik University College, 2016.

[2] Egele, M.; Scholte, T.; Kirda, E.; and Kruegel, C. “A Survey on Automated Dynamic Malware-Analysis Techniques and Tools”, *ACM Computing Surveys*, Vol. 44, No. 2, Article 6, pp. 1-42, 2012.

[3] Sweeney, A. M. “Malware Analysis and Antivirus Signature Creation”, Master Thesis, Letterkenny Institute of Technology, 2015.

[4] Kerrisk, M. “The Linux Programming Interface (2nd ed.)”, San Francisco: No Starch Press, 2016.

[5] Xiao, H.; and Stibor, T. “A Supervised Topic Transition Model for Detecting Malicious System Call Sequences”, *In Workshop on Knowledge Discovery, Modelling and Simulation* on San Diego, California, USA, pp. 23-30, 2011.

[6] Dubitzky, W.; Wolkenhauer, O.; Cho, K.-H.; and Yokota, H. “Encyclopedia of Systems Biology (3rd ed.”, New York: Springer, 2013.

[7] Juma, S.; Muda, Z.; and Yassin, W. “Machine Learning Techniques for Intrusion Detection System: A Review”, *Journal of Theoretical and Applied Information Technology*, Vol. 72, Issues 3, pp. 422-429, 2015.

[8] Witten, I.H.; Frank, E.; Hall, M.A.; and Pal, C.J. “Data Mining Practical Machine Learning Tools and Techniques Morgan Kaufmann Series in Data Management Systems (4th ed.)”, United States: Elsevier, 2016.

[9] Pilla, S.R.M., Kumar, R.K. and Sailaja, M. “A Novel Filter Based Partitioning Decision Tree Model For Real-Time Network Security”, *Journal of Theoretical and Applied Information Technology*, Vol. 83, No.2, pp. 165-172, 2016.

[10] Norouzi, M., Souri, A. and Samad Zamini, M. “A data mining classification approach for behavioral malware detection,” *Journal of Computer Networks and Communications*, 2016, pp. 1-9.

[11] Ahmad, I.N.; Ridzuan, F.; Saudi, M.M.; Pitchay, S.A.; Basir, N.; and Nabila, N.F. “Android Mobile Malware Classification using Tokenization Approach based on System Call Sequence”, *In Proceedings of the World Congress on Engineering and Computer Science* on San Francisco, USA, Vol. 1, 2017.

[12] Chaba, S.; Kumar, R.; Pant, R.; and Dave, M. “Malware Detection Approach for Android systems Using System Call Logs”, arXiv preprint arXiv:1709.08805, 2017.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[13] Rathore, M.M.; Ahmad, A.; and Paul, A. " Real time intrusion detection system for ultra-high-speed big data environments", *The Journal of Supercomputing*, Vol.72, Issue 9, pp. 3489-3510, 2016.

[14] Shahzad, R. K.; and Lavesson, N. "Veto-based Malware Detection", *In 2012 Seventh International Conference* on Availability, Reliability and Security on Prague, Czech Republic, pp. 47–54, 2012.

[15] Zakeri, M.; Faraji Daneshgar, F.; and Abbaspour, M. "A static heuristic approach to detecting malware targets", *Security and Communication Networks*, Vol. 8, Issue 17, pp. 3015–3027, 2015.

[16] Joshi, M. S. "Rule Based Classifier Models for Intrusion Detection System", *International Journal of Computer Science and Information Technologies*, Vol. 7 Issues 1, pp. 367–370, 2016.

[17] Bhaya, W. S.; and Ali, M. A. "Review on Malware and Malware Detection Using Data Mining Techniques" *Journal of University of Babylon*, Vol. 25, No. 5, pp. 1585–1601, 2017.

[18] Alazab, M.; Venkatraman, S.; Watters, P.; and Alazab, M. "Zero-day malware detection based on supervised learning algorithms of API call signatures", *In Proceedings of the Ninth Australasian Data Mining* Conference on Ballarat, Australia, Vol. 121, pp. 171-182, 2011.

[19] Bashir, U.; and Chachoo, M. "Performance Evaluation of J48 and Bayes Algorithms for Intrusion Detection System", *International Journal of Network Security & Its Applications (IJNSA),* Vol. 9, No. 4, 2017.

[20] Lengyel, T.K.; Maresca, S.; Payne, B.D.; Webster, G.D.; Vogl, S.; and Kiayias, A. "Scalability, Fidelity and Stealth in the DRAKVUF Dynamic Malware Analysis System*", Proceedings of the 30th Annual Computer Security Applications Conference* on New Orleans, Louisiana, USA, pp.386-395, 2014.

[21] Kolosnjaji, B.; Zarras, A.; Webster, G.; and Eckert, C. "Deep learning for classification of malware system call sequences", *In Australasian Joint Conference* on Artificial Intelligence, pp.137-149, 2016.

[22] B. Kolosnjaji, A. Zarras, T. Lengyel, G. Webster, and C. Eckert. "Adaptive Semantics-Aware Malware Classification", In *Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)* on San Sebastián, Spain, pp. 419-439, 2016.

[23] Bhaya, W. S., and Ali, M. A. (2017). Review on Malware and Malware Detection Using Data Mining Techniques. Journal of University of Babylon, Vol. 25, No. 5, pp. 1585–1601.