Network Intrusion Data Analysis via Consistency Subset Evaluator with ID3, C4.5 and Best-First Trees

***Shih Yin Ooi*[†]*, Yew Meng Leong*[††]*, Meng Foh Lim*[†††]*, Hong Kuan Tiew*[††††]*, and Ying Han Pang*[†††††]**

[†]*Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia*

Summary

Intrusion Detection System (IDS) is widely used to verify the incoming traffic whether it is malicious or benign connection, but traditional IDS requires a lot of human efforts and costs vast amount of computational overhead to build the set of rules in order to distinguish the intruders connection (from suspicious traffic). In view of this limitation, many researchers are adopting and researching the potential data mining and machine learning techniques to assist the stated tasks in a quicker and semi-automated manner. One of the popular statistical models would be the decision tree. It builds a simpler and straightforward tree model based on existing pre-classified network traffic database. Through the tree generation and rule discovery from the tree (rules to classify normal and malicious traffic), it is able to predict the unknown network anomalies. This prediction is meaningful to supplement the honey pot analysis. In this paper, ID3, C4.5 and Best-First trees are tested and compared on the NSL-KDD dataset. Data engineering process (including data preprocessing and feature selection) is very important in data mining, so that the rightful data can be retained for building the hypothesis, while the meaningless data should be removed. Thus, numerous feature selection techniques are explored, tested and compared in this paper. Performances are represented by using Receiver Operating Characteristic (ROC) curve, and compared through McNemar tests.

Key words:

Network Intrusion Analysis, Data Mining, Decision Tree, IDS, C4.5, Best-First Tree, Consistency Subset Evaluator

1. Introduction

Intrusion Detection System (IDS) has been developed over the past twenty years. However, traditional intrusion detection requires a significant amount of human effort to maintain and improve the performance. From the research study, we found that the quantity and quality of the alert generated by IDS are the two fundamental problems which have not been solved.

In this paper, three widely used decision tree algorithms: (i) ID3 [1], (ii) C4.5 [2] and (iii) Best-First Tree [3] are explored, tested and compared against each other. Over the past decade, decision tree is widely used in building classification models for intrusion detection. Decision tree is an automated method from the fields of statistics and machine learning. It builds simple and straightforward tree structure. This generated tree will be then used to generate the decision rules and predict the future unknown data.

Research study [4] suggested that decision tree is suitable for inductive learning due to the following three main reasons:

a) Computation methods are proportional to the number of observed training instances are efficient
b) Decision tree is a good generalization for unobserved instances unless the instances are described in the terms of features that interrelated with the concept of target
c) Resulting decision tree will provide a representation of the concept to the lap peals human in the view of it renders the classification process with self-evident.

Decision tree is more efficient in handling low dimensional data, thus feature selection methods are adopted to reduce the data dimensionality. In this paper, Consistency Subset Evaluator appears to be the most compatible feature selection method. Consistency Subset Evaluator evaluates the goodness of a subset attributes based on the level of consistency in the class value while the training instances are forecast onto the subset of attributes.

2. Literature

Many types of decision tree algorithms have been formulated with different cost effectiveness and accuracy. Those popular algorithms are including ID3 [1], C4.5 [2], CART [5], Regression [6], SPRINT [7], SLIQ [8] and Best First Tree [3]. The basic concept to build decision tree is the adoption of entropy to measure the amount of the information. Entropy is an important information theory to measure impurity of data sets:

Manuscript received February 5, 2013
Manuscript revised February 20, 2013

$$Entropy, H(X) = \sum_{x \in c_x} P(x) \log \frac{1}{P(x)} . \qquad (1)$$

In communication theory, the entropy defines the average number of bits in encoding and transmitting the data item. In anomaly detection, entropy is used to measure the regularity of audit data. A low entropy data contains a fewer number of different classes or records, which mean a high-regularity data can be expected. In other words, high-regularity data is helpful in predicting future events where redundant records in current datasets will have higher probability of occurrence. Therefore, dataset with minimum entropy is ideal to build a simpler decision tree with better intrusion detection performance.

The idea of implementing decision tree in conjunction with intrusion detection has been raised in recent years. The intention was initiated by few motivations and reasons. Gregio et al. [9] reduced the number of attributes on some honeypot data log sets to build decision tree. Sangkatsanee et al. [10] proposed the real-time intrusion detection system by using the decision tree technique to classify online network data. Revathi and Ramesh [11] from Bharathiar University in India used the best-first method to reduce the feature from 41 attributes to 14 and 7 potential attributes for classification using information obtained from KDD Cup 99 data set. The result of identifying type of attack with this approach yields more accurate result compared to purely random one.

From the previous study, the decision tree can be seen to create a less complicated tree for selected dataset to identify malicious activity. The generated tree is helpful in learning adversary trends and creating rules to predict the unknown new data and detect malicious activity.

3. Overview of Work

WEKA [12] has been used throughout the project. The dataset is stored in Attribute Relation File Format (ARFF), the default file format accepted by WEKA. The overall architectures of ID3, C4.5 and Best-First tree are shown in Figure 1, 2, and 3 respectively. ID3 can only deal with nominal attributes, thus, the continuous attributes need to be discretized in pre-processing stage.

As shown in the figures, the ArffLoader imports the selected intrusion dataset in .arff file format. By doing this, the intrusion dataset is built into WEKA and ready for further data processing. The feature selection can be optionally used to filter out irrelevant attribute in order to increase the classification performance. In this study, 10-fold cross validation is used.

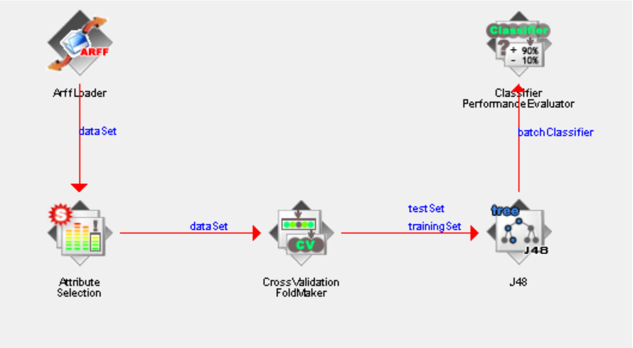


Fig. 1 Overall architecture of C4.5 (J48).

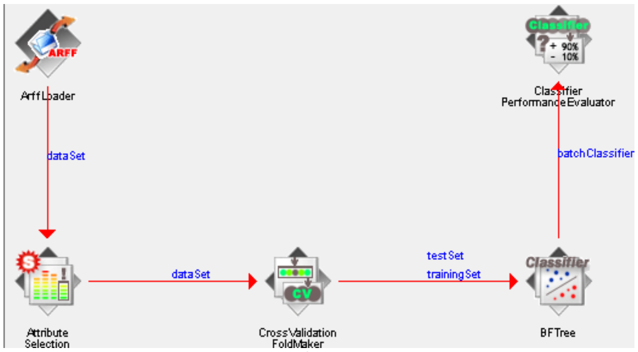


Fig. 2 Overall architecture of Best-First Tree (BFTree).

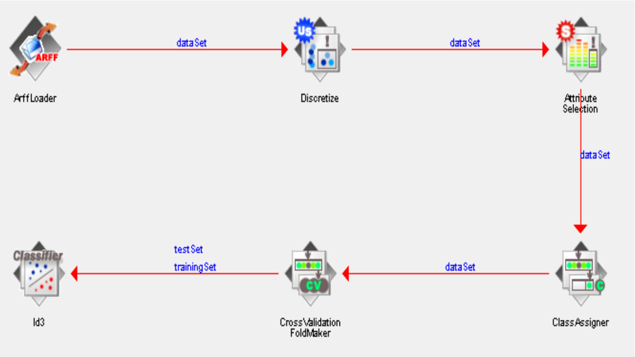


Fig. 3 Overall architecture of ID3.

4. Feature Extraction

In this paper, Consistency Subset Evaluator is recommended as the most compatible feature selection method for decision tree. Consistency Subset Evaluator is used in this paper to reduce the data dimensionality. The dimensionality is defined as the number of features which make up as a stumbling block against the algorithm in classification [13]. The intention of dimensionality reduction is due to the "Curse of Dimensionality" phenomena where the sample space needs to enlarge exponentially when level of dimensionality increases for effective estimate of multivariate densities [14]. Consistency Subset Evaluator is used to remove the useless attribute that are irrelevant or duplicated.

Consistency Subset Evaluator is a technique that evaluates the value of a subset of attributes by measuring the goodness of a subset, an optimal subset will be chosen using one evaluation function [15]. Consistency of the subset should never lower than the full set of attributes, for the reason that the practice of this subset evaluator is relevance with the exhaustive or random search which looks for the tiniest subset with consistency equals to the full set of attributes [16]. It is used in conjunction with exhaustive search method or Greedy Stepwise search method to identify best subset. The Greedy Stepwise search method is the method that can move either forward or backward through the search space.

5. ID3, C4.5 and Best-First Tree Classifiers

ID3 (or Iterative Dichotomiser 3) is a simple decision tree algorithm developed by Ross Quinlan in 1986. ID3 use information theory to build the decision tree that applies a top-down, greedy search through given set. Information gain was the metric given to select the most useful information for classifying given sets. Information gain is such as function that can measure the most balanced of splitting and minimizing the depth of the tree is the optimal way to classify a learning set. The disadvantage is ID3 cannot handle discrete and continuous variables.

Quinlan developed the C4.5 algorithm (better known as J48 in WEKA) in 1993. C4.5 is the evolution and refinement of ID3 algorithm. It better handles the unavailable of values, pruning of decision trees, continuous attribute value range, and rule derivation [17]. However, the processing speed is slightly slower than others.

Table 1: Advantages and disadvantages of three classifiers (ID3, C4.5 and Best-First Tree)

Decision Tree Classifiers	Advantages	Disadvantages
ID3	Strong classifier with lower error rate, can builds the fastest tree and short tree	Unable to handle both continuous and discrete variables
C4.5 (J48 in WEKA)	Handle continues attribute and can classifies the data with missing attributes	Slower to classify than other techniques
Best-First Tree	Able to build the simpler and understandable tree	Unstable that the accuracy might not be guaranteed

Shi [3] proposed a simpler and less complicated binary tree which named as Best-First tree. The tree node will expand according to the best-first order. This approach add “best” node to the tree that reduces impurity maximally in each step. However this technique is less stable if compared to C4.5, where the accuracy is still arguable.

6. Experiment

6.1 Database Setup

NSL-KDD by Information Security Centre of eXcellence [18] is selected as our database in this study. The derivative NSL-KDD dataset is created to address few issues related to the use of original KDD Cup 99 dataset. The original KDD Cup 99 dataset has been created by Lincoln Lab for Defense Advanced Research Projects Agency (DARPA) IDS to process TCP dump in 1998. Shortly afterward, it has been adopted for the 1999 KDD Cup Challenge. Since then, the dataset becomes the most diffusely used network intrusion dataset by many researchers over years of IDS evolution. However, the issue has been raised when McHugh criticized that the data rate is high since it contains unnecessary TCP dump header data [19]. Furthermore, the KDD Cup 99 is also been criticized for inheriting performance measurement issues.

The NSL-KDD solved a number of problems lay beneath the original KDD Cup 99 dataset. NSL-KDD eliminates duplicated record in the train set, thus classifier will not tends toward frequent record. Unbiased learner will perform better on distinct records with higher detection rates. Last but not least, the total amount of records in both train and test sets have been reduced. Original KDD train set has reduction rate of 78.05% (shrink from 4,898,431 train set records to 1,074,992 records), while 311,072 of original KDD test set record was reduced to 77,289 distinct records (75.15% reduction rate).

The NSL-KDD data set is provided with binary labelled class (normal or anomaly) which comprises the train set and test set in ARFF file format, KDDTrain+.arff with 125973 instances. On the other hand, KDDTest+.arff is the test set with 22544 instances supplied to examine how the generated predictive model perform after learning from the train set.

6.2 Classification with train set

Since WEKA has the limitation of handling large dataset, the NSL-KDD train set with 125973 instances has been chopped down into thirteen (13) chunks of subset while test set with 22544 instances has been chopped into three (3) chunks of subset. ID3, C4.5 and Best-First Tree classifiers in WEKA are able to classify the thirteen subsets of the train sets independently. The training results are reported as per following:

Table 2: Performance of classifiers with NSL-KDD train set

Train Set Instance	Correctly Classified			Incorrectly Classified			Unclassified
	ID3	C4.5	Best-First Tree	ID3	C4.5	Best-First Tree	ID3
#000001 to #010000	9758 (97.58%)	9943 (99.43%)	9948 (99.48%)	132 (1.32%)	57 (0.57%)	52 (0.52%)	110 (1.10%)
#010001 to #020000	9736 (97.36%)	9921 (99.21%)	9924 (99.24%)	174 (1.74%)	79 (0.79%)	76 (0.76%)	90 (0.90%)
#020001 to #030000	9751 (97.51%)	9937 (99.37%)	**9954 (99.54%)**	147 (1.47%)	63 (0.63%)	46 (0.46%)	102 (1.02%)
#030001 to #040000	9734 (97.34%)	9927 (99.27%)	9937 (99.37%)	146 (1.46%)	73 (0.73%)	63 (0.63%)	120 (1.20%)
#040001 to #050000	9739 (97.39%)	9936 (99.31%)	9931 (99.31%)	147 (1.47%)	64 (0.64%)	69 (0.69%)	114 (1.14%)
#050001 to #060000	9744 (97.44%)	9923 (99.23%)	9935 (99.35%)	156 (1.56%)	77 (0.77%)	65 (0.65%)	100 (1.00%)
#060001 to #070000	9749 (97.49%)	9919 (99.19%)	9931 (99.31%)	134 (1.34%)	81 (0.81%)	69 (0.69%)	117 (1.17%)
#070001 to #080000	**9785 (97.85%)**	9929 (99.29%)	9930 (99.30%)	133 (1.33%)	71 (0.71%)	70 (0.70%)	82 (0.82%)
#080001 to #090000	9757 (97.57%)	9913 (99.13%)	9930 (99.30%)	143 (1.43%)	87 (0.87%)	70 (0.70%)	100 (1.00%)
#090001 to #100000	9767 (97.67%)	9920 (99.20%)	9935 (99.35%)	130 (1.30%)	80 (0.80%)	65 (0.65%)	103 (1.03%)
#100001 to #110000	9772 (97.72%)	**9954 (99.54%)**	9950 (99.50%)	127 (1.27%)	46 (0.46%)	50 (0.50%)	101 (1.01%)
#110001 to #120000	9729 (97.29%)	9930 (99.30%)	9941 (99.41%)	166 (1.66%)	70 (0.70%)	59 (0.59%)	105 (1.05%)
#120001 to #125973	5774 (96.67%)	5909 (98.93%)	5932 (99.31%)	89 (1.49%)	64 (1.07%)	41 (0.69%)	110 (1.84%)
Total	122795 (97.48%)	125061 (99.28%)	125178 (99.37%)	1824 (1.45%)	912 (0.72%)	795 (0.63%)	1354 (1.07%)

Based on the results, ID3 performed the best in the 8th chunk of dataset (Instance #070001 to #080000) which yields the accuracy 97.85%. C4.5 classifier recorded the highest accuracy of 99.54% in the 11th chunk (Instance #100001 to #110000) of train subset. Best First Tree yielded the highest accuracy (99.54%) in the 3rd chunk of train set (Instance #020001 to #030000). These three subsets are pre-select to be used in the following experiments with various feature extraction methods and supplied test set for ID3, C4.5, and Best First Tree respectively.

6.3 Classification with supplied test set

Table 3: Performance of ID3 classifier with NSL-KDD test set

Test Set Instance	ID3 (Trained with 8th chunk of train set)		
	Correctly Classified	Incorrectly Classified	Unclassified
#00001 to #10000	7843 (78.43%)	1708 (17.08%)	449 (4.49%)
#10001 to #20000	7808 (78.08%)	1696 (16.96%)	496 (4.96%)
#20001 to #22544	1990 (78.22%)	428 (16.82%)	126 (4.95%)
Total	17641 (78.25%)	3832 (17.00%)	1071 (4.75%)

Table 4: Performance of C4.5 classifier with NSL-KDD test set

Test Set Instance	C4.5 (Trained with 11th chunk of train set)	
	Correctly Classified	Incorrectly Classified
#00001 to #10000	8054 (80.54%)	1946 (19.46%)
#10001 to #20000	8100(81.00%)	1900 (19.00%)
#20001 to #22544	2036(80.03%)	508 (19.97%)
Total	18190 (80.69%)	4354 (19.31%)

Table 5: Performance of Best-First Tree classifier with NSL-KDD test set

Test Set Instance	Best-First Tree (Trained with 3rd chunk of train set)	
	Correctly Classified	Incorrectly Classified
#00001 to #10000	7679 (76.79%)	2321 (23.21%)
#10001 to #20000	7705 (77.05%)	2295 (22.95%)
#20001 to #22544	1947 (76.533%)	597 (23.467%)
Total	17331 (76.88%)	5213 (23.12%)

6.4 Feature selection

All of the feature extraction methods available in WEKA are tested. Three of the well-performed feature extraction methods with valid output were further analyzed. They are CFS (Correlation-based Feature Selection) Subset Evaluator, Consistency Subset Evaluator and Filtered Subset Evaluator.

In each feature extraction method, the unselected features are removed during the pre-processing phase. The result of before and after applying the feature extraction methods with three classifiers has been gathered and aggregated in the Table 6.

Table 6: Classification accuracy of various classifiers after employing different feature extraction

Algorithm/Feature Extraction		CFS Subset Evaluator		Consistency Subset Evaluator		Filtered Subset Evaluator	
		Before	After	Before	After	Before	After
ID3	Train Set (8th Chunk)	97.85%	97.09%	97.85%	97.81%	97.85%	97.12%
	Test Set	78.43%	76.90%	78.43%	**79.21%**	78.43%	76.73%
C4.5	Train Set (11th Chunk)	99.54%	97.99%	99.54%	99.31%	99.54%	98.01%
	Test Set	80.69%	78.93%	80.69%	**81.43%**	80.69%	78.37%
Best-First Tree	Train Set (Third Chunk)	99.54%	98.10%	99.54%	99.24%	99.54%	98.10%
	Test Set	76.88%	75.93%	76.88%	**80.87%**	76.88%	75.93%

Based on the result, Consistency Subset Evaluator is the most optimal feature extraction incorporated with all three decision trees to enhance the performance of the predictive model in anomaly detection. The detection rate has been improved by using only few features such as *service*, *src bytes*, *count*, *dst_host_count*, *dst_host_srv_count*, *dst_host_diff_srv_count* and *dst_host_serror_rate* out of forty-one (41) potential features. From here we can observed that the Consistency Subset Evaluator is able to look for tiniest subset of attribute with equivalent consistency in full set of attribute, removing all other unhelpful attributes is the key to aid the algorithm with higher detection rate.

6.5 ROC curves

Three different classifiers (Best-First Tree, ID3 and C4.5) are compared to find out the decision tree with best accuracy given the same train set (1st chunk of NSL-KDD train subset) and test set (1st chunk of NSL-KDD test subset). The Receiver Operating Characteristics (ROC) curve is used as the criteria to compare the accuracy of the model. Higher values in area under curve (AUC) of ROC denote that the classifier is able to assign a higher score to a randomly chosen instance.

All the statistical value of area under ROC as shown in Figure 4 and Figure 5 has been tabulated in Table 7. C4.5 has the highest value of area under curve which is 0.8079 while Best-First Tree obtains the lowest score. Therefore, it can be concluded that C4.5 performed better than the other two classifiers in the NSL-KDD dataset. ID3 has inconsistent value due to unclassified data.

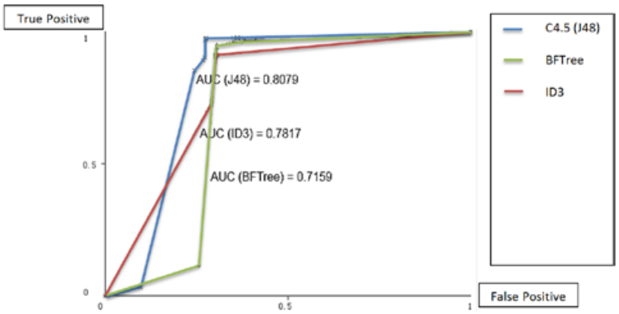


Fig. 4 ROC curves with normal class value for C4.5 (J48), Best First Tree (BFTree) and ID3.

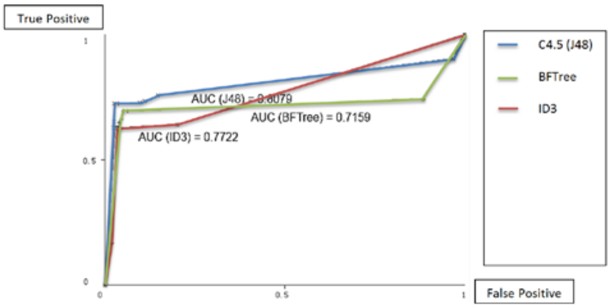


Fig. 5 ROC curves with anomaly class value for C4.5 (J48), Best First Tree (BFTree) and ID3.

Table 7: Summary of area under ROC curve for three classifiers

Classifier	Class	
	Normal	Anomaly
ID3	0.7817	0.7722
C4.5 (J48 in WEKA)	**0.8079**	**0.8079**
Best-First Tree	0.7159	0.7159

Table 8 compares the performance of ID3, C4.5 and Best-First Tree in terms of prediction accuracy before and after applying Consistency Subset Evaluator. For the given

NSL-KDD train and test subsets, C4.5 records the highest accuracy compared to the other two models.

Table 8: Accuracy of classifiers before and after applying feature extraction

Classifier	Feature Extraction	
	Before (%)	After (%)
ID3	75.07	75.13
C4.5	79.91	78.59
Best-First Tree	78.20	77.41

6.6 McNemar’s test

McNemar’s test as presented in Table 9 using matched pairs provides a 2x2 contingency table to identify the difference in training errors or misclassification.

Table 9: Data on two models using McNemar’s test (contingency table) from matched pairs

MODE L B	MODEL A	
	TRUE	FALSE
TRUE	n_{00} (misclassified by A and B)	n_{01} (misclassified by A but not B)
FALSE	n_{10} (misclassified by B but not A)	n_{11} (misclassified neither by A nor B)

If Model A classifies better than Model B then n01 would be less than n10.

Table 10: . Misclassification of ID3 and C4.5 after applying feature extraction

C4.5	ID3	
	TRUE	FALSE
TRUE	1705	830
FALSE	418	7047

Since C4.5 has lesser misclassification compared to ID3, thus C4.5 performs better.

Table 11: Misclassification of ID3 and Best-First Tree after applying feature extraction

Best-First Tree	ID3	
	TRUE	FALSE
TRUE	1874	661
FALSE	459	7006

Based on Table 11, Best-First Tree has lower training error compared to ID3, therefore Best-First Tree is able to classify better than ID3 in the given train and test subsets.

Table 12: Misclassification of C4.5 and Best-First Tree after applying feature extraction

Best-First Tree	C4.5	
	TRUE	FALSE
TRUE	1920	203
FALSE	413	7464

Lastly, C4.5 records lower misclassification compared to Best-First Tree based on results in Table 12. In conclusion, C4.5 performs better than Best-First Tree and ID3

7. Conclusion

In this paper, ID3, C4.5, and Best-First Tree are tested on NSL-KDD network intrusion dataset. The decision trees generated are used as a predictive model to detect anomaly connection for every unlabeled record in the test set depending on the selected features. The features are selected by using Consistency Subset Evaluator. From the experiment, it appears as the most optimal feature selection technique from WEKA to prepare the attributes for further classification through ID3, C4.5, and Best First Trees. Consistency Subset Evaluator filters unhelpful attributes while maintaining the consistency and hence able to increase the accuracy of algorithms by using lesser features. In this paper, 7 attributes are selected out of 41 attributes from the NSL-KDD network intrusion dataset, including *service*, *src bytes*, *count*, *dst_host_count*, *dst_host_srv_count*, *dst_host_diff_srv_count*, and *dst_host_error_rate* out of 41 attributes. The feature selection is done based on the data mining approach instead of pre-selected by the security expert.

References

[1] J.R. Quinlan, “Induction of decision trees,” Machine Learning, vol. 1, no. 1, pp 81-106, 1986.

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

[3] H. Shi, Best-First Decision Tree Learning, Master’s Thesis, University of Waikato, 2006.

[4] P.E. Utgoff, and C.E. Brodley, “An incremental method for finding multivariate splits for decision trees,” Machine Learning: Proceeding of the seventh International Conference, Palo Alto, CA: Morgan Kaufmann, pp 58, 1990.

[5] L. Breiman, J. H. Friedman, R.A. Olshen, and C.J. Sotne, Classification and Regression Trees, Wadsworth, Belmont, 1984.

[6] L. Breiman, Classification and Regression Trees, New York, N.Y.: Chapman & Hall, 1993.

[7] J. Shafer, R. Agrawal, and M. Mehta, “SPRINT: a scalable parallel classifier for data mining,” Proceedings of the VLDB conference, Bombay, 1996.

[8] M. Mehta, R. Agrawal, and J. Riassnen, “SLIQ: a fast scalable classifier for data mining,” Extending database technology, Springer, Avignon, pp 18–32, 1996.

[9] A. Gregio, R. Santos, and A. Montes, “Evaluation of data mining techniques for suspicious network activity classification using honeypots data,” pp 1-10, 2007.

[10] P. Sangkatsanee, N. Wattanapongsakorn, and C. Charnsripinyo, “Real-time intrusion detection and classification,” pp 1-5, 2009.

[11] M. Revathi, and T. Ramesh, “Network intrusion detection system using reduced dimensionality,” Indian Journal of

Computer Science and Engineering (IJCSE), ISSN: 0976-5166, vol. 2, no.1, pp 61-67, 2011.

[12] M.A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten, “The WEKA data mining software: an update,” SIGKDD Explorations, vol. 11, Issue 1, 2009.

[13] R. Lior, and Z.M. Oded, Data Mining with Decision Trees: Theory and Application, World Scientific Publishing Co. Pte. Ltd., London, 2008.

[14] J. Hwang, S. Lay, and A. Lippman, “Nonparametric multivariate density estimation: a comparative study,” IEEE Transaction Signal Processing, vol. 42, pp 2795-2810, 1994.

[15] M. Dash, and H. Liu, “Consistency-based search in feature selection,” Artificial Intelligence, 151, pp 155-176, 2003.

[16] M.A. Hall, Correlation-Based Feature Selection for Machine Learning, Doctoral dissertation, University of Waikato, Hamilton, New Zealand, 1999.

[17] A.K. Sharma, S. Sahni, “A comparative study of classification algorithms for spam email data analysis,” pp 1890-1895, 2011.

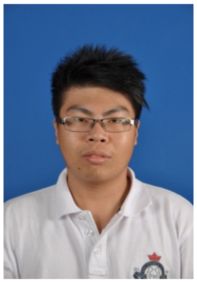
[18] ISCX (2009), The NSL-KDD data set [Online]. Available: http://nsl.cs.unb.ca/NSL-KDD/ [2013, January 6].

[19] J. McHugh, “Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory,” ACM Trans. Information System Security 3 (4), 262-294, 2000.

Ooi Shih Yin received her Bachelor of Information Technology (Hons) and Master of Science (Information Technology) from Multimedia University, Malaysia in 2004 and 2006 respectively. Shih-Yin joined the Faculty of Information Science and Technology in Multimedia University, Malaysia where she is the currently the Program Coordinator of B. IT (Hons) Security Technology. She has authored few indexing journals and conference papers, and served as paper reviewer in the field of biometrics, image processing, machine intelligence, computer vision, and data mining. She is a member of ISPA Malaysia.

Leong Yew Meng born at Kuantan 29 May 1990, graduated from SMK Air Putih, Pahang during secondary school. He was graduated from Multimedia University, Melaka Campus, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia in Bachelor of Information Technology (Hons) Security Technology.

Lim Meng Foh born at Muar 27 August 1988, He was graduated from Sekolah Menengah Kebangsaan Ledang, Negeri Johor during secondary school. He currently study in Multimedia University, Melaka Campus, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia in Bachelor of Information Technology (Hons) Security Technology.



Tiew Hong Kuan born at Melaka 1 July 1990, graduated from SMK Yok Bin, Melaka during secondary school on year 2007 and graduated from TARC in diploma of computer science and computer mathematics at KL on year 2010. And in the Jun 2010 enter into the Multimedia University, Melaka Campus, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia in Bachelor of Information Technology (Hons) Security Technology.

Pang Ying Han received her B.E. degree in Electronic Engineering in year 2002 and M.E. degree in year 2005 from Multimedia University. She is currently a PhD student at Multimedia University. Her research interests include face recognition, manifold learning, image processing and pattern recognition