Feature Grouping for Intrusion Detection System Based on Hierarchical Clustering

Jingping Song[1,2], Zhiliang Zhu[1], and Chris Price[2]

[1] Software College of Northeastern University, Shenyang, Liaoning, China, 110819
{songjp,zhuzl}@swc.neu.edu.cn
[2] Department of Computer Science, Aberystwyth University, United Kingdom, SY23 3DB
{jis17,cjp}@aber.ac.uk

Abstract. Intrusion detection is very important to solve an increasing number of security threats. With new types of attack appearing continually, traditional approaches for detecting hazardous contents are facing a severe challenge. In this work, a new feature grouping method is proposed to select features for intrusion detection. The method is based on agglomerative hierarchical clustering method and is tested against KDD CUP 99 dataset. Agglomerative hierarchical clustering method is used to construct a hierarchical tree and it is combined with mutual information theory. Groups are created from the hierarchical tree by a given number. The largest mutual information between each feature and a class label within a certain group is then selected. The performance evaluation results show that better classification performance can be attained from such selected features.

Keywords: Intrusion detection, Mutual information, Feature grouping, Hierarchical clustering.

1 Introduction

Network intrusion detection system is a tool for network operators to detect hazardous traffic and alert their existence in the networks [1]. Most intrusion detection systems adopt signature based methods to detect intrusion attacks [2]. A signature is a rule set that contains information regarding target patterns from exciting hazardous packet actions against the target patterns. A network intrusion detection system can obtain valuable information from ongoing or local traffic as well. An intrusion detection system is not a standalone system, but works with other systems as a firewall [3]. There are two types of intrusion detection methods, misuse detection and anomaly detection. Misuse detection specifically detects known attacks by using pattern matching approaches, which is the common drawback of this kind of detection method. On the other hand, anomaly detection builds profiles of normal behaviors by detecting attacks first, and then identifies potential attacks when their behaviors are obviously deviated from normal profiles [4].

Anomaly intrusion detection is a classification task, and it consists of building a predictive model which can identify attack instances [5]. Intrusion detection can be

S. Teufel et al. (Eds.): CD-ARES 2014, LNCS 8708, pp. 270–280, 2014.
© IFIP International Federation for Information Processing 2014

considered as a two class problem or a multiple class problem. A two class problem regards all attack types as anomaly patterns and the other class is a normal pattern [6]. A multiple class problem deals with the classification based on different attacks. Since there are too many features or attributes which may contain false correlation, classification of anomaly intrusion detection systems is complex work [7]. Moreover, many features may be irrelevant or redundant. For this reason, feature selection methods can be used to get rid of the irrelevant and redundant features without decreasing performance.

Feature selection based on mutual information was initially reported in [8] and subsequently modified in [9] and [10]. The present paper has implemented a feature selection method by grouping features based on the use of mutual information combined with a hierarchical clustering method. The selected features are then employed in the C4.5 classification method for intrusion detection [11]. The performance of the proposed approach is evaluated with respect to different numbers of features and compared with other work in applied feature selection for intrusion detection systems in [9]. [12] and [13] as well. [14] proposed an algorithm to use SVM and simulated annealing to find the best selected features to improve the accuracy of anomaly intrusion detection. [15] reported mutual information-based feature selection method results in detecting intrusions with higher accuracy.

2 Related Works

2.1 Hierarchical Clustering

Hierarchical clustering is a clustering method to build a hierarchy of clusters. There are two types of strategies for hierarchical clustering, agglomerative and divisive [16]. Agglomerative is a bottom-up approach where initially every data item constitutes its own cluster, and pairs of clusters are merged as one moves up the hierarchy. Divisive is a top-down approach and all data is part of the initial cluster and splits are performed recursively as one moves down the hierarchy [17].

In order to decide which clusters should be combined or split, a measure of dissimilarity between sets of observations is required [18]. In most methods of hierarchical clustering, this is achieved by use of an appropriate metric and a linkage criterion [19]. In this paper, an agglomerative hierarchical clustering algorithm is used based on linkage rule.

2.2 Mutual Information

Entropy is an important measurement for information in information theory. It is capable of quantifying the uncertainty of random variables and scaling the amount of information shared by them effectively.

Let X be a random variables with discrete values, its entropy is defined as

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \quad (1)$$

where $H(\cdot)$ is entropy, and $p(x)=Pr(X=x)$ is the probability density function of X. Note that entropy depends on the probability distribution of the random variable.

Conditional entropy refers to the uncertainty reduction of one variable when the other is known. Assume that variable Y is given, the conditional entropy $H(X|Y)$ of X with respect to Y is

$$H(X|Y) = -\sum_{y\in Y}\sum_{x\in X} p(x,y)\log p(x|y) \quad (2)$$

where $p(x,y)$ is the joint probability density function and $p(x|y)$ is the posterior probabilities of X given Y. Similarly, the joint entropy $H(X,Y)$ of X and Y is

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y) = -\sum_{y\in Y}\sum_{x\in X} p(x,y)\log p(x,y) \quad (3)$$

To quantify how much information is shared by two variables X and Y, a concept termed mutual information $I(X;Y)$ is defined as

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \quad (4)$$

$I(X;Y)$ will be very high when X and Y are closely related with each other. Otherwise, $I(X;Y)=0$ denotes that these two variables are totally unrelated. In this paper, the mutual information between two variables is calculated.

2.3 C4.5 Algorithm

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan and it is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.

In this paper, C4.5 will be used to do the classification in section 4. From the classification results we can compare our method with other feature selection methods. C4.5 uses the concept of information gain to make a tree of classificatory decisions with respect to a previously chosen target classification. The information gain can be described as the effective decrease in entropy resulting from making a choice as to which attribute to use and at what level. Compared to other classification algorithms, it is an effective method to deal with a dataset like KDD99 which has new class labels in the test dataset. The reason is C4.5 is a supervised learning method and based on information gain.

3 Implemented Work

In this section, our algorithm based on agglomerative hierarchical clustering is described in detail. The basic idea is grouping the features by agglomerative hierarchical clustering method, and then selecting features from the groups. As we used a clustering method to construct groups, cluster and group have the same meaning in the following formulation.

3.1 Selecting Strategy of Feature Grouping

Feature Grouping is highly beneficial in learning with high dimensional data. It reduces the variance in the estimation and improves the stability of feature selection [20]. Furthermore, it could help in data understanding and interpretation as well. The purpose of feature grouping is creating groups for candidate selecting features and selecting one feature or more features from certain groups to represent the group.

Clustering methods could be used to create groups since they select data in one cluster by specific metrics. Different clustering methods and metrics could compose different cluster constructions. Number of clusters affects how many features will be selected. For example, there are different strategies if we expect to select 8 features from a dataset. We could create 8 groups by a clustering method and select 1 feature in each group. And we could construct 4 groups and select 2 features per group as well. Moreover, we could select different numbers of features in different groups. Where hierarchical clustering method is used to create groups in this work, we chose the selecting 1 feature from each group strategy. This strategy is simple and easy to implement. And another reason is there might be only one feature in one group by using agglomerative hierarchical clustering method.

3.2 Implemented Algorithm

In this section, we will show the algorithm put forward by this paper. The detailed algorithm is shown as follows.

Input: A training dataset T=D(F,C), number of clusters n.
Output: Selected features S.
(1) Initialize parameters: F←'initial set of all features', C ←'class labels', S=∅.
(2) Calculate the mutual information of every pair of features f_i and f_j in F, denote as $I(f_i; f_j)$.
(3) Create hierarchical cluster tree by using agglomerative hierarchical clustering method base on $I(f_i; f_j)$.
(4) Construct clusters from a hierarchical cluster tree by given n.
(5) For each cluster, calculate mutual information between each feature and class label in C, and then find the maximum value M_c.
(6) Select feature f_s which has the M_c in each group, and put f_s into S, S←'f_s'.

First of all, the algorithm set initialization parameters and F is a set of all the features in the training dataset. And C denotes class labels and C represents class labels. Then, the algorithm calculates the mutual information of every pair of features in F and composes a matrix based on them. After that, it creates a hierarchical cluster tree based on the matrix by using an agglomerative hierarchical clustering method. Moreover, it constructs clusters from a hierarchical cluster tree by given n. And n clusters mean n groups containing candidate features. Furthermore, in each cluster, it calculates mutual information between each feature and class label in C, and then finds the maximum value M_c. Finally, it selects feature f_s which has the M_c in each group, and put f_s into S.

4 Experimental Results

4.1 KDD99 Dataset

KDD99 is the most widely used data set for the evaluation of anomaly detection methods. This data set is built based on 7 weeks of TCP connections in network traffic, and there are about 5 million connection records in the training dataset and around 2 million connection records. Each connection is labeled by either normal or attack. The attack type is divided into four categories of 39 types of attacks [21]. Only 22 types of attacks are in the training dataset and the other 17 unknown types are in the test dataset. It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training data which makes the task more realistic. The KDD dataset consists of three components, which are detailed in Table 1.

The "10% KDD" dataset is employed for the purpose of training. The KDD training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features, with exactly one specific attack type or normal type. This dataset contains 22 attack types and is a more concise version of the "whole KDD" dataset. It contains more connections of attacks than normal connections and the attack types are not represented equally. Denial of service attacks account for the majority of the dataset [22].

Table 1. Basic characteristics of the KDD 99 intrusion detection datasets

Dataset	*Normal*	*DoS*	*U2R*	*R2L*	*Probe*
"10%KDD"	97278	391458	52	1126	4107
"Corrected KDD"	60593	229853	70	16347	4166
"Whole KDD"	972780	3883370	52	1126	41102

On the other hand, the "Corrected KDD" dataset (test dataset) provides a dataset with different statistical distributions than either "10% KDD" or "Whole KDD" and contains 14 additional attacks. The list of class labels and their corresponding categories for "10% KDD" are detailed in [23].

4.2 Measures of Performance Evaluation

The implemented method in this paper is conducted by six measures: True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, F-Measure. The six measures could be calculated by True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN), as follows.

True positive rate (TPR): TP/(TP+FN), also known as detection rate (DR) or sensitivity or recall. False positive rate (FPR): FP/(TN+FP) also known as the false alarm rate. Precision (P): TP/(TP+FP) is defined as the proportion of the true positives against all the positive results. Total Accuracy (TA): (TP+TN)/(TP+TN+FP+FN) is the proportion of true results (both true positives and true negatives) in the population.

Recall (R): TP/(TP+FN) is defined as percentage of positive labeled instances that were predicted as positive. F-measure: 2PR/(P+R) is the harmonic mean of precision and recall.

We use the training dataset to construct the decision tree model and then reevaluate on the test dataset and get TP, FP, TN, FN. After that, we calculate precision, total accuracy and F-measure for the test dataset.

4.3 Experiment Evaluation

The experiments were conducted by using KDD 99 dataset and performed on a Windows machine having configuration and Intel (R) Core (TM) i5-2400 CPU@ 3.10GHz, 3.10 GHz, 4GB of RAM, the operating system is Microsoft Windows 7 Professional. We have used an open source machine learning framework Weka 3.5.0. This tool is used to do classification for performance comparison of our method with other feature selection methods. We used C4.5 as the classification algorithm for all the feature selection methods.

Table 2 shows comparison results by different feature selection methods using 13 selected features. The first algorithm C4.5 used 41 features to do the classification. DMIFS is dynamic mutual information feature selection method proposed by Huawen Liu [9]. FGMI is feature grouping based on mutual information method implemented previously by the authors of this paper. This method is construct groups base on mutual information among features. AHC is agglomerative hierarchical clustering algorithm implemented by this paper. We can see from the comparison that AHC algorithm produces better performance on F-measure and achieves good performance on other measures.

Table 2. Comparison results by different algorithms using 13 selected features

Algorithm	*TP Rate*	*FP Rate*	*Precision*	*Recall*	*F-Measure*	*Class*
C4.5 (41)	**0.994**	0.09	0.728	**0.994**	0.841	Normal
	0.91	**0.006**	**0.999**	0.91	0.952	Anomaly
DMIFS	0.993	0.086	0.736	0.993	0.846	Normal
	0.914	0.007	0.998	0.914	0.954	Anomaly
FGMI	**0.994**	0.085	0.739	0.994	0.848	Normal
	0.915	**0.006**	0.998	0.915	0.955	Anomaly
AHC	0.993	**0.077**	**0.757**	0.993	**0.859**	Normal
	0.923	0.007	0.998	**0.923**	**0.959**	Anomaly

Table 3 describes comparison results by different feature selection methods using 10 selected features. C4.5, DMIFS, FGMI and AHC have the meaning as table 2. MMIFS is modified mutual information feature selection method raised by Jingping in 2014 [24]. And we can see from the comparison that AHC could get better performance nearly in all measures.

Table 3. Comparison results by different algorithms using 10 selected features

Algorithm	*TP Rate*	*FP Rate*	*Precision*	*Recall*	*F-Measure*	*Class*
C4.5 (41)	**0.994**	0.09	0.728	**0.994**	0.841	Normal
	0.91	**0.006**	**0.999**	0.91	0.952	Anomaly
DMIFS	0.993	0.086	0.736	0.993	0.846	normal
	0.914	0.007	0.998	0.914	0.954	anomaly
MMIFS	0.99	0.084	0.741	0.99	0.848	normal
	0.916	0.01	0.997	0.916	0.955	anomaly
FGMI	0.994	0.082	0.746	0.994	0.852	Normal
	0.918	0.006	0.998	0.918	0.957	Anomaly
AHC	**0.994**	**0.08**	**0.751**	**0.994**	**0.856**	Normal
	0.92	**0.006**	0.998	**0.92**	**0.958**	Anomaly

The purpose of comparison in table 2 and table 3 is compare AHC and other algorithms by the same number of selected features. Figure 1 illustrates the precision comparison of AHC by different number of features.

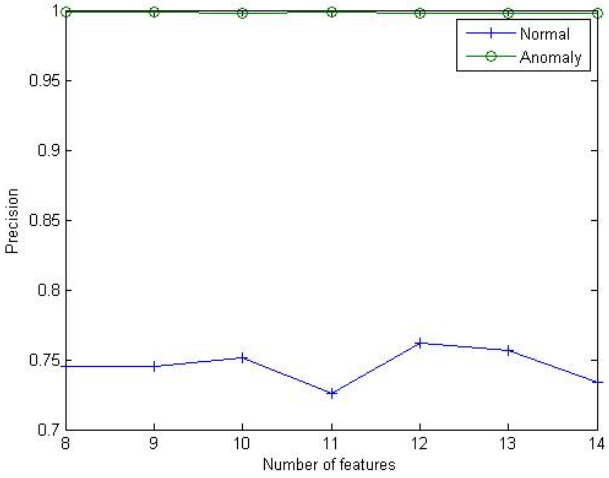

Fig. 1. Precision comparison of AHC by different number of selected features

Figure2 and figure 3 show the F-measure and total accuracy comparison of AHC by different number of features respectively.

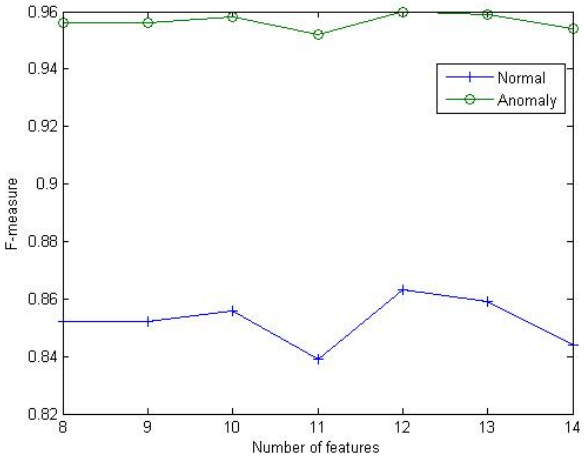

Fig. 2. F-measure comparison of AHC by different number of selected features

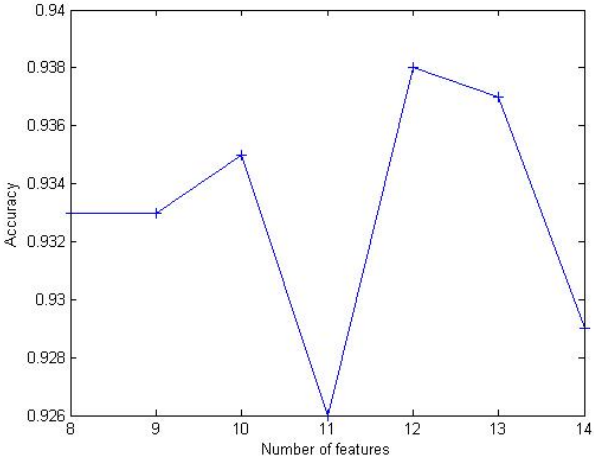

Fig. 3. Total accuracy comparison of AHC by different number of selected features

From the comparison of figure 1 to figure 3, we could see better performance could be achieved when selecting 12 features by AHC. And table 4 shows detailed comparison of AHC by different number of selected features.

We can see from table 4 that AHC algorithm could get best performance by selecting 12 features. And for F-measure, both normal and anomaly could achieve highest value when using 12 selected features.

Table 4. Comparison results of AHC by different number of selected features

NO. of Features	*TP Rate*	*FP Rate*	*Precision*	*Recall*	*F-Measure*	*Class*
8	**0.995**	0.082	0.745	**0.995**	0.852	normal
	0.918	**0.005**	**0.999**	0.918	0.956	anomaly
9	**0.995**	0.082	0.745	**0.995**	0.852	normal
	0.918	**0.005**	**0.999**	0.918	0.956	anomaly
10	0.994	0.08	0.751	0.994	0.856	normal
	0.92	0.006	0.998	0.92	0.958	anomaly
11	**0.995**	0.091	0.726	**0.995**	0.839	normal
	0.909	**0.005**	**0.999**	0.909	0.952	anomaly
12	0.994	**0.075**	**0.762**	0.994	**0.863**	normal
	0.925	0.006	0.998	**0.925**	**0.96**	anomaly
13	0.993	0.077	0.757	0.993	0.859	normal
	0.923	0.007	0.998	0.923	0.959	anomaly
14	0.994	0.087	0.734	0.994	0.844	normal
	0.913	0.006	0.998	0.913	0.954	anomaly

5 Conclusion

This paper has presented a feature grouping method based on agglomerative hierarchical clustering method. It described how to compose the group by hierarchical tree, how to get the number of groups and how to select features in each group. First of all, the mutual information between each pair of two features is calculated to be used to construct the hierarchical tree. Moreover, the proposed algorithm creates groups by a given number. Finally, the mutual information between a feature and class labels is used to select one feature in one group. Experiment results on KDD 99 dataset indicate that the proposed approach generally outperforms DMIFS, MMIFS, and FGMI algorithm. Furthermore, the comparison by different number of features shows that 12 features could get best performance indicator.

Whilst promising, the presented work opens avenues for further investigation. For instance, the mutual information between features and class labels can be used to design new algorithm. And other clustering or classification algorithms can be applied to compose groups. Moreover, more than one feature could be selected in a certain group. In future work, the proposed algorithm will be tested on other datasets and look for more effective measures or methods than mutual information theory.

References

1. Kim, H.J., Kim, H.-S., Kang, S.: A memory-dfficient bit-split parallel string matching using pattern dividing for intrusion detection systems. IEEE Transactions on Parallel and Distributed Systems 22(11), 1904–1911 (2011)

2. García-Teodoroa, P., Díaz-Verdejoa, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security 28, 18–28 (2009)
3. Horng, S.-J., Su, M.-Y., Chen, Y.-H., Kao, T.-W., Chen, R.-J., Lai, J.-L., Perkasa, C.D.: A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Systems with Applications 38, 306–313 (2011)
4. Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A.: Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset. Expert Systems with Applications 38, 5947–5957 (2011)
5. Sobh, T.S.: Anomaly Detection Based on Hybrid Artificial Immune Principles. Information Management & Computer Security 21(14), 1–25 (2013)
6. Mehdi, M., Zair, S., Anou, A., Bensebti, M.: A Bayesian Networks in Intrusion Detection Systems. Journal of Computer Science 3(5), 259–265 (2007)
7. Shan, S., Karthik, V.: An approach for automatic selection of relevance features in intrusion detection systems. In: Proc. of the 2011 International Conference on Security and Management, pp. 215–219 (2011)
8. Battiti, R.: Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on Neural Networks, 537–550 (1994)
9. Liu, H., Suna, J., Liu, L., Zhang, H.: Feature selection with dynamic mutual information. Pattern Recognition 42, 1330–1339 (2009)
10. Vinh, L.T., Lee, S., Park, Y.-T., d'Auriol, B.J.: A novel feature selection method based on normalized mutual information. International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies 37(1), 100–120 (2012)
11. Muniyandia, A.P., Rajeswarib, R., Rajaramc, R.: Network Anomaly Detection by Cascading K-Means Clustering and C4.5 Decision Tree algorithm. In: International Conference on Communication Technology and System Design, pp. 174–182 (2012)
12. Chebrolu, S., Abraham, A., Thomas, J.P.: Feature deduction and ensemble design of intrusion detection systems. Journal of Computers & Security 24(4), 295–307 (2005)
13. Mukkamala, S., Sung, A.H.: Feature ranking and selection for intrusion detection systems using support vector machines. In: International Conference on Information and Knowledge Engineering (ICIKE), pp. 503–509 (2002)
14. Lin, S.-W., Ying, K.-C., Lee, C.-Y., Lee, Z.-J.: An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Applied Soft Computing 12, 3285–3290 (2012)
15. Amiri, F., Yousefi, M.R., Lucas, C., Shakery, A., Yazdani, N.: Mutual information-based feature selection for intrusion detection systems. Journal of Network and Computer Applications 34, 1184–1199 (2011)
16. Oh, S.-J., Kim, J.-Y.: A hierarchical clustering algorithm for categorical sequence data. Information Processing Letters 91, 135–140 (2004)
17. Cilibrasi, R.L., Vitanyi, P.M.B.: A fast quartet tree heuristic for hierarchical clustering. Pattern Recognition 44, 662–677 (2011)
18. Kojadinovic, I.: Agglomerative hierarchical clustering of continuous variables based on mutual information. Computational Statistics & Data Analysis 46, 269–294 (2004)
19. Özdamar, L., Demir, O.: A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. Transportation Research Part E 48, 591–602 (2012)
20. Liu, X., Lang, B., Xu, Y., Cheng, B.: Feature grouping and local soft match for mobile visual search. Pattern Recognition Letters 33, 239–246 (2012)

21. Kayacik, H.G., Zincir-Heywood, A.N., Heywood, M.I.: Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets. In: Proceedings of the Third annual Conference on Privacy, Security and Trust (2005)
22. Cho, J., Lee, C., Cho, S., Song, J.H., Lim, J., Moonam, J.: A statistical model for network data analysis: KDD CUP 99' data evaluation and its comparing with MIT Lincoln Laboratory network data. Simulation Modelling Practice and Theory 18, 431–435 (2010)
23. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A Detailed Analysis of the KDD CUP 99 Data Set. In: Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications (2009)
24. Song, J., Zhu, Z., Scully, P., Price, C.: Modified Mutual Information-based Feature Selection for Intrusion Detection Systems in Decision Tree Learning. Journal of computers 9(7), 1542–1546 (2014)