Intrusion Detection System using Modified C-Fuzzy Decision Tree Classifier

Krishnamoorthi Makkithaya†, N.V. Subba Reddy†, and U. Dinesh Acharya†

†Faculty of Computer Science and Engineering, Manipal Institute of Technology Manipal, Manipal University India

Summary
As the number of networked computers grows, intrusion detection becomes an essential component in keeping networks secure. Various approaches for intrusion detection are currently being in use with each one has its own merits and demerits. This paper presents the work to test and improve the performance of an intrusion detection system based on C-fuzzy decision tree, a new class of decision tree. The tree grows gradually by using fuzzy C-means clustering (FCM) algorithm to split the patterns in a selected node with the maximum heterogeneity into C corresponding children nodes. We investigated the usefulness of C-fuzzy decision tree for developing IDS with a data partition based on horizontal fragmentation. Emphirical results indicate the usefulness of our approach in developing the efficient IDS. This paper also used a modified fuzzy C-means algorithm with controllable membership ratio through an extended distance measure to include an additional higher order term. Effect of different membership ratio on the developed decision tree with each fragment is tested to select the best membership ratio. The results obtained have shown that our improved version performs better resulting in an effective intrusion detection system.

Key words:
Data mining, Intrusion detection, Fuzzy c- means clustering, Decision tree.

1. Introduction

The supervised learning methods significantly outperform the unsupervised ones if the test data contains no unknown attacks. In the presence of unknown attacks in the test data, the performance of all supervised methods drops significantly. The performance of unsupervised learning is not much affected by unknown attacks. This makes the unsupervised methods, which do not require a laborious labeling process, a clear forerunner for practical purposes if unknown attacks can be expected. However, the problem of test data being drawn from a different distribution cannot be solved within the purely supervised or unsupervised techniques. An emerging field of semi-supervised learning offers a promising direction of future research in developing IDS [1].

Generally the intrusion detection problem is viewed as a two-class classification problem. The goal is to classify patterns of the system behavior in two categories (normal and abnormal), using patterns of known attacks, which belong to the abnormal class, and patterns of the normal behavior. The normal and the abnormal behaviors in networked computers are hard to predict as the boundaries cannot be well defined. This prediction process may generate false alarms in many intrusion detection systems. However, with fuzzy logic, the false alarm rate in determining intrusive activities can be reduced [2].

Decision trees [3],[4] are the commonly used architectures of machine learning and classification systems. They come with a comprehensive list of various training and pruning schemes, a diversity of discretization (quantization) algorithms, and a series of detailed learning refinements [3], [4]. The objective of this study is to study and modify a new class of decision tree a semi supervised technique to develop an effective IDS. The underlying conjecture is that data can be perceived as a collection of information granules [5]. Thus, the tree becomes spanned over these granules, treated now as fundamental building blocks. In turn, information granules and information granulation are almost a synonym of clusters and clustering [6], [7]. Fuzzy clusters are used to capture the continuous nature of the classes so that there is no restriction of the use of these constructs to the discrete problems. Further, the fuzzy granulation helps to link the discretization problem with the formation of the tree in a direct and intimate manner. This paper presents the efforts that made, to improve the performance of IDS based on modify C- fuzzy decision tree.

For the construction of the modified *C*-fuzzy decision tree, as in [8], the tree is created initially by assigning all patterns to a root node and grows gradually by splitting the patterns in a selected node into *C* children nodes until certain stopping criteria is reached. The modifications considered are:

1. Data partition based on horizontal fragmentation.
2. Modified Fuzzy C-means clustering algorithm with controllable membership characteristics.

Manuscript received November 5, 2008
Manuscript revised November 20, 2008

2. Over all Architecture of Cluster based Decision Tree

The architecture of the cluster–based decision tree develops around fuzzy clusters that are treated as generic building blocks of the tree. The training data set X is clustered into C clusters so that the data points (patterns) that are similar are put together. These clusters are completely characterized by their prototypes (centroids). Tree growing start with them positioned at C top nodes of the tree structure. The way of building the clusters implies a specific way in which elements of X allocated to each cluster. In other words, each cluster comes with a subset of X, namely X1, X2 … Xc. The process of growing tree is guided by a certain heterogeneity criterion that quantifies the diversity of the data (with respect to the output variable y) falling under the given cluster (node) denoted by V!, V2,… ,Vc respectively (Fig. 1) The node with the highest value of the criterion is chosen and treated it as a candidate for further refinement. The process is repeated by selecting the most heterogeneous node out of all final nodes. The growth of the tree is carried out by expanding the nodes and building their consecutive levels that capture more details of the structure. It is noticeable that the node expansion leads to the increase in either the depth or width (breadth) of the tree. The pattern of the growth is very much implied by the characteristics of the data as well as influenced by the number of the clusters.

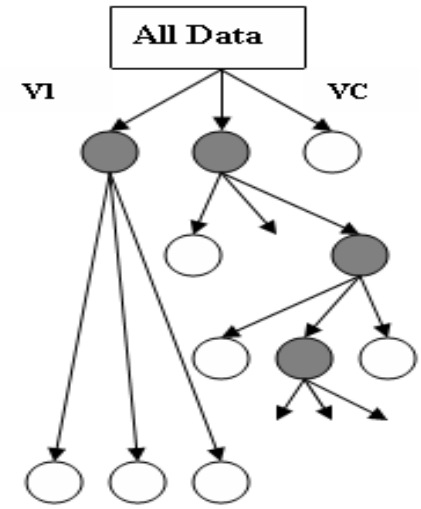

Fig. 1.Growing a decision tree by expanding nodes (which are viewed as clusters located at its nodes). Shadowed nodes are those with maximal diversity criterion and thus being subject to the split operation.

The architecture of C-fuzzy decision tree for intrusion detection is similar to that in [8]. Once the tree has been constructed, we can use the tree to classify an unknown pattern or predict the corresponding output of the pattern. We traverse the tree by starting from the root node and then selecting a path according to the similarity calculations to move down from the current level to the next level until a leaf node is reached. Finally, the local linear model of the reached leaf node is used to calculate the output of the unknown pattern.

Fuzzy clustering is a core functional part of the overall tree. It builds the clusters and provides its full description. We confine ourselves to the standard fuzzy C-means (FCM), which is an omnipresent technique of information granulation. The description of this algorithm is well documented in the literature [6]. The FCM algorithm is an example of an objective-oriented fuzzy clustering where the clusters are built through a minimization of some objective function.

2.1Heterogeneity Calculation

The purpose of the heterogeneity calculation for each leaf node in the current tree is to find out the node with highest diversity of the included patterns for splitting. Therefore, we use the same heterogeneity measure as that in [8]. The variability of the data in the output space at a node (Vi) is taken as a spread around the representation of the same node. In the next step the node with highest value of V say Vjmax selected for tree expansion.

3. Data Partition

One of the key problems that arise in a great variety of fields, including pattern recognition and machine learning, is the so-called feature selection. Feature selection not only obtains better accuracy of the predictor but also reduces training and inference time. Thus, selecting relevant features for the construction of classifiers has received a great deal of attention. As computational cost depends upon the sample size, number of features, it is required some approaches to reduce sample size and number of features to make these IDS models work efficiently [11]. Domain knowledge can be systematically incorporated into the sample size reduction [12].

The accuracy by which the c-fuzzy decision tree classifies depends heavily on the performance of the FCM. The aim of FCM here is to cluster the network traffic data to find useful pattern to detect intrusion. Sometimes this may be considered as partitioning or division of data into groups using simple techniques to achieve the goal of clustering. Data partitioning in the preprocessing phase group the most similar data together so that it may help to understand the structure of data and relative importance of each attribute for the selection or classification process. In this section we explain our technique borrowed from distributed database system to partition the network traffic data for intrusion detection.. We are proposing a simple domain knowledge based fragmentation for data partition.

We analyzed network traffic data and observed following details which are very useful for data partition.

A fundamental understanding of TCP/IP protocol is essential for using network based intrusion detection. Attackers often take the advantage of obscure aspects of specific protocols to execute their attacks. Decoding at the protocol level allows signatures to be created that stipulate allowable and unallowable aspects of specific protocols. This type of signature allows for much broader rules, while still keeping a low false alert threshold. Attackers do not fully adhere to the protocol rules. TCP (Transmission Control Protocol) is the primary transport control protocol used on the Internet. It is basically a connection oriented service where as UDP (User datagram protocol) and ICMP (internet control message protocol) are connectionless service. UDP communications can be spoofed much easier than can TCP communications because UDP is a connectionless protocol and does not use sequence numbers. While UDP attacks are less common, they can also be more difficult to detect and decipher. UDP communications are session less. There are no flags to use as tell-tale indicators of malicious activity as with TCP communications. Unfortunately several severe attacks occur via UDP. This means that each transport layer protocol requires separate strategy and feature set to detect intrusion using network traffic data. Therefore, we believe fragmentation of the network traffic data based on different transport layer protocol will result in a better data set, resulting in effective classifiers.

3.1Fragmentation

The decomposition of global relation into fragments can be performed by applying two different types of fragmentation: horizontal fragmentation and vertical fragmentation. In all types of fragmentation, a fragment can be defined by an expression in a relational language, which takes global relation as operand and produces fragments as result. There are, however, some rules which must be followed when defining fragments. They are completeness, reconstruction condition, and disjoint condition.

Horizontal fragmentation consists of partitioning the tuples of a global relation into subsets, where each subset can contain data which have common properties. It can be defined by expressing each fragment as a selection operation on the global relation.

Horizontal fragments are defined using selection operation on global relation .Our proposed horizontal fragmentation is as follows:

We assumed the**Training-data** as the global relation and transport layer protocol feature is selected for horizontal fragmentation.

The horizontal fragmentation is defined as follows:

Training-data1= **SL** protocol=“Tcp”
Training-data

Training-data2= **SL** protocol = “Udp”
Training-data

Training-data3= **SL** protocol = “Icmp”
Training-data

Where **SL** is the selection operation on the relation **Training-data**.

The above fragmentation satisfies the completeness condition as “Tcp”, “Udp”, “Icmp” are the only possible values of the protocol attribute in the global relation Training-data. The reconstruction condition is easily verified, because it is always possible to reconstruct the Training-data through following operation

Training-data = Training-data1 **UN** Training-data2 **UN** Training-data3.

Where **UN** is the union operation. The disjoint condition is clearly verified.

Our proposed fragmentation makes data set for modified C-fuzzy decision tee and modified tree is as shown in the Fig 2. This divides data set into three processes, where same program run on each subset to generate c-fuzzy decision tree. Each process results in independent decision tree and these trees can be used separately for training and testing.

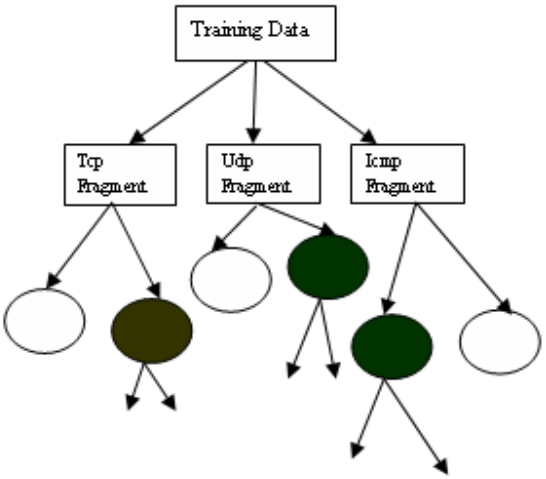

Fig. 2. Horizontal fragmentation based modified c-fuzzy decision tree

4.Fuzzy Clustering Method with Controllable Membership Characteristics

As mentioned earlier, we use a modified fuzzy *C*-means clustering algorithm with controllable membership characteristics to cluster the patterns in the selected node. As we know, fuzzy *C*-means is a well-known objective-oriented fuzzy clustering algorithm where

clusters are generated through a minimization of some objective function. The objective function is of the following form:

$$J(U,V)= \sum_{i=1}^{c} \sum_{k=1}^{n} u^{m}_{ki} \quad (1)$$

where C and m (usually $m = 2$) are user-defined parameters and represent the number of clusters and fuzzification factor, respectively, Nl ($Nl \leq N$) denotes the number of patterns in the leaf node l to be split, and μik represents the degree of the kth pattern belonging to the ith cluster. Therefore, we have a C by Nl matrix $\mathbf{U} = [\mu ik]$, $i = 1, 2. . . C$, $k = 1, 2, . . ., Nl$, which is named partition matrix and satisfies the following three conditions:

1) $\mu_{ik} \in [0, 1]$

2) $\sum_{i=1}^{C} \mu_{ik} = 1$

3) $\sum_{k=1}^{n} \mu_{ik} \in [1, N]$

In convectional fuzzy c-means algorithm, the distance measure is defined by $\| x_k - v_i \|^2$

This is extended to include an additional higher-order term, as defined in [11]

$$d_{ki} = a_1 \| x_k - v_i \|^2 + a_2 \| x_k - v_i \|^4 \quad (2)$$

where x_k is the kth input datum, vi is the ith cluster center, and a1 and a2 are two parameters specified by the user. With this new definition of distance measure, the fuzzy clustering problem can be reformulated as a new optimization problem

$$\min_{U,V} J(U,V) \overset{\Delta}{=} \sum_{i=1}^{c} \sum_{k=1}^{n} u^{m}_{ki} (a_1 \| x_k - v_i \|^2 + a_2 \| x_k - v_i \|^4) \quad (3)$$

$$\text{subject to} \quad \sum_{i=1}^{C} u_{ik} = 1, \forall k \quad (4)$$

where U is the fuzzy partition matrix, V is the collection of cluster centers, n is the number of data samples, c is the number of clusters, m is the weighting exponent, and u_{ki} is the membership value of x_k with respect to cluster i, for $k = 1, \cdots, n$ and $i = 1, \cdots c$

The *alternating algorithm* of the conventional fuzzy c-means method can also be applied to the new fuzzy clustering method with different updating functions for the cluster centers and fuzzy partition matrix. We summarize the solution algorithm as follows:

1) Initialize cluster centers. Usually this is performed by random assignment.
2) Update the fuzzy partition matrix.
3) Update the cluster centers.
4) Check for convergence. If not yet converged, go to Step 2 and proceed.

4.11 Controllable membership function

The accuracy by which the C-fuzzy decision tree classifies depends heavily on the performance of the FCM. With a new distance measure which adds a higher-term to the original one membership curves can be controlled resulting in more flexible clustering method. Memberships are affected by the ratio a_2/a_1. With an increasing a_2/a_1 ratio, the clustering gradually changes from soft partition toward harder partitions. Cluster centers change slightly with varying combinations of a_1 and a_2 [11]. In principle, a higher membership value is observed around the same neighborhood of a cluster centre. The unique property of this fuzzy clustering scheme is that it is capable of controlling the membership curve through adjusting the values of a_1 and a_2.The optimal choice of the ratio is linked with dataset being investigated and thus varies according to the application

5. Experimental Consideration

In this section the results of different experiments conducted with modified C-decision tree are presented.

5.1Data set

The KDD Cup 1999 Intrusion detection contest data [12] (KDD cup 99 Intrusion detection data set) is used in our experiments. This data was prepared by the 1998 DARPA Intrusion Detection Evaluation program by MIT Lincoln Laboratory. Lincoln labs acquired nine weeks of raw TCP dump data. The raw data was processed into connection records, which consist of about 5 million connection records. The data set contains 24 attack types. These attacks fall into four main categories: DOS, Probe, u2r, and r2l. The data set has 41 attributes for each connection record plus one class label. The data set for our experiments contained 37016 records which were randomly generated from the MIT data set. Random generation of data include the number of data from each class proportional to its size. This data set is again divided into training data with 29612 records and testing data with 7404 records. The data is partitioned into the two classes of ‘‘Normal’’ and ‘‘Attack’’ patterns where Attack is the collection of four classes (Probe, DOS, U2R, and R2L) of attacks. The objective is to separate normal and attack pattern to make classifier into two class classifier [40]. Only the 34 numeric features are used in our experiments.

We conducted a five fold experiment with random sub sampling to derive a classifier with training data and then to estimate accuracy of the classifier with test data.
Experiments are conducted with training and testing data to evaluate the performance of C-fuzzy decision tree. A simple output node representation (centroids) is considered, to use the decision tree in the classification mode. Sensitivity, specificity, and accuracy are calculated separately for both training and test data as shown below.

$$\text{sensitivity} = \frac{t\_pos}{pos} \quad \text{……..(5)}$$

$$\text{specificity} = \frac{t\_neg}{neg} \quad \text{……(6)}$$

$$\text{accuracy} = \text{sensitivity}\ \frac{pos}{(pos+neg)} + \text{specificity}\ \frac{neg}{(pos+neg)} \quad \text{…..(7)}$$

Where **t_pos** is the number of true positives, **t_neg** is the number of true negatives, **pos** is the number of positives, and **neg** is the number of negative samples.

5.2 Experiment 1

This experiment is conducted to see the effect of fragmentation on the performance of the classifier. The sensitivity, specificity and accuracy are evaluated separately with training and test data set for the each tree (fixed size depth wise only) and tabulated in Table 1 and Table 2. As per the results, it is very clear that our fragmentation based C-fuzzy decision tree performs better compared to that of original tree. Specificity of ICMP fragmentation is very less, this is due to non uniform distribution of normal and intruder class in the ICMP fragment. But overall accuracy is still improved because of better sensitivity.

Table 1. Performance of C-fuzzy decision tree with and without training data

Performance measures	With out Fragmentation	With Fragmentation		
		TCP	UDP	ICMP
Sensitivity (%)	83.55	93.36	92.15	98.90
Specificity (%)	95.52	96.63	99.59	31.24
Accuracy (%)	89.27	94.46	95.78	98.13

Table 2 Performance of C-fuzzy decision tree with and without fragmentation (Test data)

Performance measures	With out Fragmentation	With Fragmentation		
		TCP	UDP	ICMP
Sensitivity (%)	84.22	92.56	93.15	97.45
Specificity (%)	93.52	95.33	98.39	31.24
Accuracy (%)	88.97	93.96	95.97	97.23

5.3 Experiment 2

After finding the effectiveness of fragmentation in developing C-fuzzy decision tree based IDS, experiments are conducted to analyze the effect of modified FCM with controllable membership characteristics. The combined diversity critrion, accuracy, sensitivity, and specificity are evaluated for the each C-decision tree with different membership ratio. The results are as shown in the Table 3, Table 4, and Table 5. From the results we observed following things:

1. A membership ratio 0 gives higher accuracy, sensitivity and low diversity criterion for the TCP fragment.

2. UDP fragment decision tree performs better with membership ratio 0 when compared with other ratios.

3. ICMP fragment performance is very good with membership ratio equal to 10.

Table 3 Performance of TCP fragment C-Fuzzy decision tree for different membership ratio

Data set size	Tree Size	Membership ratio	Sensitivity (%)	Specificity (%)	Accuracy (%)	Combined diversity crterion
27189	6	**0**	**93.36**	**96.63**	**94.89**	**671.2958**
		1	91.81	95.14	93.50	804.5718
		10	92.25	94.40	93.30	872.8119
		100	91.89	94.09	92.67	915.9116

Table 4 Performance of UDP fragment C-Fuzzy decision tree for different membership ratio

Data set size	Tree Size	Membership ratio	Sensitivity (%)	Specificity (%)	Accuracy (%)	Combined diversity crterion
1281	5	**0**	**92.15**	**99.59**	**95.78**	**30.4429**
		1	91.71	98.77	95.13	30.0968
		10	91.66	99.26	95.38	32.1842
		100	88.63	99.36	94.02	42.5120

Table 5 Performance of ICMP fragment C-Fuzzy decision tree for different membership ratio

Data set size	Tree Size	Membership ratio	Sensitivity (%)	Specificity (%)	Accuracy (%)	Combined diversity crterion
1142	5	0	98.30	31.24	98.13	5.8901
		1	99.49	37.09	99.15	6.0446
		10	**99.88**	**54.35**	**99.18**	**3.3128**
		100	98.60	45.56	98.04	5.4371

5.4 Experiment 3

So far fixed size tree is used which has grown only depth wise. For the complete analysis purpose tree has grown fully (both depth wise and breadth wise) using training data set. The combined diversity criterion and number of points assigned to a node are used as the stopping criteria to stop the tree growth. The performance of the each tree developed is tabulated in Table 6 and Table 7

Table 6 Performance of complete C-Fuzzy decision tree with and without fragmentation (Training data)

Performance measures	C-Fuzzy Decision Tree			
	Without Fragmentation	With Fragmentation		
		TCP	UDP	ICMP
Data set	29612	27189	1281	1142
Tree size	13	9	7	5
Number of leaves	14	10	8	6
Membership Ratio	1	0	0	10
Sensitivity (%)	94.34	97.28	97.67	99.82
Specificity (%)	97.02	98.96	98.91	82.35
Accuracy (%)	95.39	97.81	97.88	99.55

Table 7 Performance of complete C-Fuzzy decision tree with and without fragmentation (Test data)

Performance measures	C-Fuzzy Decision Tree			
	Without Fragmentation	With Fragmentation		
		TCP	UDP	ICMP
Data set	7404	6798	321	285
Sensitivity (%)	94.84	97.89	98.32	99.45
Specificity (%)	96.82	99.01	99.89	80.56
Accuracy (%)	95.63	98.98	98.93	99.32

From the results it is observed that the decision tree size and number of leaf nodes comparatively less with fragmentation thus reducing the complexity of the tree. The classification accuracy also significantly improved due to the fragmentation thus resulting in effective Intrusion Detection System.

6. Conclusion

This research work presents a modified C-fuzzy decision tree model to detect intrusion with a focus on improving the performance selecting relevant data set and using a modified FCM. It is evident from the results that the proposed data partition and modified FCM enhance the performance of the intrusion detection system. The experimental results with KDD cup 99 data emphasizes the importance of modified decision tree in developing improved IDS.

References

[1] Pavel Laskov, Patrick D¨ussel, Christin Sch¨afer and Konrad Rieck Fraunhofer- "Learning intrusion detection: supervised or unsupervised?, *FIRST.IDA, Kekul´estr. 7,12489 Berlin, Germany*

[2] J. Gomez and D. Dasgupta, "Evolving fuzzy classifiers for intrusion detection," *Proceedings of the 2002 IEEE Workshop on the Information Assurance,* West Point, NY, USA, June 2001.

[3] J. R. Quinlann, "Induction of decision trees," *Mach. Learn. 1*, pp. 81–106, 1986.

[4] W. P. Alexander and S. Grimshaw, "Tree regression," *J. Computational Graphical Statistics*, vol. 5, pp. 156–175, 1996.

[5] W. Pedrycz and Z. A. Sosnowski,"Designing decision trees with the use of fuzzy granulation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 2, pp. 151 159, Mar. 2000.

[6] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Functions", *New York: Plenum,* 1981.

[7] A. K. Jain *et al.*, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[8] Witold Pedrycz, Fellow, IEEE, and Zenon A. Sosnowski, "C–Fuzzy Decision Trees", *IEEE Transactions on Systems, Man, and Cybernetics —Part C: Applications and Reviews,* VOL. 35, NO. 4, November 2005.

[9] Wenke Lee and Salvatore J. Stolfo, Philip K. Chan. "Real Time Data Mining- based Intrusion Detection." http://www.ncsu.edu/faculty/lee/project/id.html/

[10] Wenke Lee and Salvatore J. Stolfo, "Combining Knowledge Discovery and Knowledge Engineering to build IDSs". http://www.ncsu.edu/faculty/lee/project/id.html

[11] Dian-Rong Yang, Leu-Shing Lan* , and Shih-Hung Liao "A New Fuzzy Clustering Method with Controllable Membership Characteristics", *IEEE International Conference on Fuzzy Systems ,* Vancouver, BC, Canada July 16-21, 2006.

[12] KDD-Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html