Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 30 (2012) 174 – 182

Procedia Engineering

www.elsevier.com/locate/procedia

International Conference on Communication Technology and System Design 2011

Network Anomaly Detection by Cascading K-Means Clustering and C4.5 Decision Tree algorithm

Amuthan Prabakar Muniyandi[a], R. Rajeswari[b], R. Rajaram[c], a*

[a]*Department of IT, Thiagarajar College of Engineering, TN, INDIA*
[b]*Department of EEE, Government College of Engineering, TN, INDIA*
[c] *CSE/IT, Thiagarajar College of Engineering, TN, INDIA*

Abstract

Intrusions pose a serious securing risk in a network environment. Network intrusion detection system aims to identify attacks or malicious activity in a network with a high detection rate while maintaining a low false alarm rate. Anomaly detection systems (ADS) monitor the behaviour of a system and flag significant deviations from the normal activity as anomalies. In this paper, we propose an anomaly detection method using "K-Means + C4.5", a method to cascade k-Means clustering and the C4.5 decision tree methods for classifying anomalous and normal activities in a computer network. The k-Means clustering method is first used to partition the training instances into *k* clusters using Euclidean distance similarity. On each cluster, representing a density region of normal or anomaly instances, we build decision trees using C4.5 decision tree algorithm. The decision tree on each cluster refines the decision boundaries by learning the subgroups within the cluster. To obtain a final conclusion we exploit the results derived from the decision tree on each cluster.

Keywords: Anomaly detection; C4.5 decision tree; k - Means clustering;

1. Introduction

Network Intrusion Detection Systems (NIDS) have become a standard component in security infrastructures as they allow network administrators to detect policy violations. Anomaly-based network intrusion detection systems (A-NIDS) are currently a principal focus of research and development in the field of intrusion detection. Anomaly detection systems (ADS) monitor the behaviour of a system and flag significant deviations from the normal activity as anomalies. Recently, anomaly detection has been used for identifying attacks in computer networks, malicious activities in computer systems and misuses in

* Amuthan Prabakar.: +91-9900898748.
E-mail address: ap_sse@tce.edu.

doi:10.1016/j.proeng.2012.01.849

Web systems. A more recent class of ADS developed using machine learning techniques like artificial neural-networks, fuzzy classifiers, multivariate analysis, and others have become popular because of their high detection accuracies at low false positive rates. However, the ADS related studies cited above have drawbacks: The studies build anomaly detection methods with single machine learning techniques like artificial neural-networks, pattern matching, etc., while recent advances in machine learning show that selection, and cascading of multiple machine learning methods have a better performance yield over individual methods.

In this paper, we propose a supervised anomaly detection method, called "K-Means + C4.5" developed by cascading two machine learning algorithms: 1) the k-Means clustering and 2) the C4.5 decision tree. In the first stage, k-Means clustering is performed on training instances to obtain k disjoint clusters. Each k-Means cluster represents a region of similar instances, "similar" in terms of Euclidean distances between the instances and their cluster centroids. We choose k-Means clustering because: 1) it is a data-driven method with relatively few assumptions on the distributions of the underlying data and 2) the greedy search strategy of k-Means guarantees at least a local minimum of the criterion function, thereby accelerating the convergence of clusters on large data sets. In the second stage of K-Means+C4.5, the k-Means method is cascaded with the C4.5 by building decision trees using the instances in each k-Means cluster. Cascading the k-Means clustering method with C4.5 decision tree learning alleviates two problems in k-Means clustering: 1) the Forced Assignment problem and 2) the Class Dominance problem. The Forced Assignment problem arises when k parameter in k-Means is set to a value that is considerably less than the inherent number of natural groupings within the training data. The k-Means procedure initialized with a low k value underestimates the natural groups within the training data and, therefore, will not capture the overlapping groups within a cluster, forcing the instances from different groups to be a part of the same cluster. Such "forced assignments" in anomaly detection may increase the false positive rate or decrease the detection accuracy. The second problem, Class Dominance arises in a cluster when the training data have a large number of instances from one particular class and very few instances from the remaining classes. Such clusters, which are dominated by a single class, show weak association to the remaining classes. From the previous observations it is obvious that cascading the machine learning algorithms provide better results than individual machine learning implementations. Cascading the decisions from the k-Means and C4.5 methods involves two phases: 1) the Selection phase, and 2) the Classification phase. In the Selection phase, the closest cluster i.e., the nearest neighbour cluster to the test instance is selected. In the selected cluster the decision tree corresponding to that cluster is generated. In the Classification phase, the test instance is classified into normal or anomaly using the decision tree result and then it is included in the cluster with the classified label as normal or anomaly.

We perform experiments on the Network Anomaly Data, which is feature extracted from the 1999 MIT-DARPA network traffic, using an artificial neural network based nonlinear component analysis method. The data set contain representative anomalous and normal behavioural patterns from the domain of computer networks. Performance evaluation of the K-Means+C4.5 cascading approach is conducted using six measures:

1. Detection accuracy or true positive rate (TPR),
2. False positive rate (FPR),
3. Precision,
4. Total accuracy (or accuracy),
5. F-measure, and
6. Receiver operating characteristic (ROC) curves and areas under ROC curves (AUCs)

1.1. Contributions of the Proposed Scheme

The contributions of the proposed scheme are enumerated as follows:

- The paper presents a novel method to cascade the k-Means clustering and C4.5 decision tree learning methods for mitigating the Forced Assignment and Class Dominance problems of the k-Means method for classifying data originating from normal and anomalous behaviours in a computer network.
- The paper evaluates the performance of K-Means+C4.5 classifier, and compares it with the individual k-Means clustering and C4.5 decision tree methods using six performance measures.
- The paper presents a novel method for cascading two successful data partition methods for improving classification performance. From an anomaly detection perspective, the paper presents a high performance anomaly detection system.

The rest of the paper is organized as follows: In section 2, provides a brief related work. In Section 3, we briefly discuss the k-Means and C4.5 decision tree learning-based anomaly detection methods. In Section 4, we present the K-Means+C4.5 method for anomaly detection. In Section 5, we discuss the experimental datasets. In Section 6, we discuss the results. We conclude our work and give future directions in Section 7.

2. Related work

This section provides a detailed study on classification based anomaly detection methods and related application domains.

Classification based Anomaly Detection Techniques

Classification is used to learn a model (called as classifier) from a set of labelled data instances (called as training instances) and then, classify a test instance into one of the classes using the learnt model (testing) [5] classification based techniques can be divided into two phases 1) Training phase and 2) Testing phase. In the training phase, trains a classifier using the available labeled training data. In the testing phase, test instances are classified as normal or abnormal (anomaly) using classification algorithm. Classification anomaly detection methods are categorized into tow main classes, 1) Multi–class classification based anomaly detection techniques and 2) One–class classification based anomaly detection techniques. In the first technique, the training data contains labeled instances belonging to multiple normal classes [2] [26]. Such anomaly detection techniques learn a classifier to distinguish between each normal class against the rest of the classes. A test instance is considered anomalous if it's not classified as normal by any of the classifiers.

In the second technique, the training instances have only one class label. Such techniques learn a discriminative boundary around the normal instances using a one–class classification algorithm, for example, one–class SVMs [50], one–class Kernel Fisher Discriminates [23] [24]. Any test instances that does not fall with in the learnt boundary is declared as anomalous.

Bayesian network has been used for anomaly detection in the multi–class setting. A basic well known technique for a universal categorical data set using a Naïve Bayesian network estimate the posterior probability of observing a class label for a given test data instance. The class label with largest posterior is chosen as the predicted class for the given test instance. The likelihood of observing the test instance given a class, and the prior on the class probabilities, are estimates from the training data set. The basic technique can be generalized to multivariate categorical data set by aggregating the per–attribute posterior probabilities for each test instance and using the aggregated value to assign a class label to the test instance.

Several variants of the basic technique has been proposed for network intrusion detection [2] [4] [17], for novelty detection in video surveillance [4], for anomaly detection in text data and for disease out break detection.

Several methods have been proposed that capture the conditional dependencies between the different attributes using more complex Bayesian networks [3] [10].

Support Vector Machine (SVM) [28] has been applied to anomaly detection in the one class setting. Such techniques use one class learning techniques for SVM [22] and learn a region that contains the training data instances. Support Vector Machine (SVM) is used for supervised intrusion detection in Mukkamala and Janoski [19]. Kernels, such as Radial Basis Function (RBF) kernel, can be used to learn complex regions. For each test instance, the basic technique determines the test instance falls with in the learnt region. If it falls in learnt region, then it is declared as normal, otherwise it is declared as anomalous. Song et al. [25] uses Robust Support Vector Machine (RSVM) which is robust to the presence of anomalies in the training data. RSVM have been applied to system call intrusion detection [9].

Rule based anomaly detection techniques learn rules that capture the normal behavior of a system. A basic multi–class rule based technique consists of two steps. First step is to learn rules from the training data using a rule learning algorithm, such as RIPPER, Decision Trees, etc. Second step is to find for each test instance, the rule that best captures the test instance. The inverse of the confidence associated with the best rule is the anomaly score of the test instance. Several minor variants of the basic–rule based technique have been proposed [6] [8] [11]. Association rule mining [49] has been used for one class anomaly detection by generating rules from the data in an unsupervised fashion. Association rules are generated from a categorical data set. To ensure that the rules correspond to strong patterns, a support threshold is used to prune out rules with low support [27]. Association rule mining based techniques have been used for network intrusion detection [2] [14] [15] [20], system call intrusion detection [12] [13] [21], Credit card fraud detection [3], and fraud detection in spacecraft house keeping data. Frequent item sets are generated in the intermediate step of association rule mining algorithms. He et al [7] propose an anomaly detection algorithm for categorical data sets in which the anomaly score of a test instance is equal to the number of frequent item sets it occurs in.

3. Anomaly detection with k-means clustering and c4.5 decision tree learning methods

In this section, we briefly discuss the k-Means clustering and the C4.5 decision tree classification methods for supervised anomaly detection.

3.1. Anomaly Detection with k-Means Clustering

The k-Means algorithm groups N data points into k disjoint clusters, where k is a predefined parameter. The steps in the k-Means clustering-based anomaly detection method are as follows:

Step 1: Select k random instances from the training data subset as the centroids of the clusters C1; C2; ...Ck.
Step 2: For each training instance X:
  a. Compute the Euclidean distance $D(C_i,X), i = 1...k$
  b. Find cluster C_q that is closest to X.
  c. Assign X to C_q. Update the centroid of C_q. (The centroid of a cluster is the arithmetic mean of the instances in the cluster.)
Step 3: Repeat Step 2 until the centroids of clusters C1; C2; ...Ck stabilize in terms of mean-squared-error criterion.
Step 4: For each test instance Z:
  a. Compute the Euclidean distance $D(C_i,Z), i = 1...k$. Find cluster C_r that is closest to Z.
  b. Classify Z as an anomaly or a normal instance using the Decision tree.

3.2. Anomaly Detection with C4.5 Decision Trees

Given a set S of cases, C4.5 first grows an initial tree using the divide-and-conquer algorithm as follows:

1. If all the cases in S belong to the same class or S is small, the tree is a leaf labeled with the most frequent class in S.
2. Otherwise, choose a test based on a single attribute with two or more outcomes. Make this test the root of the tree with one branch for each outcome of the test, partition S into corresponding subsets S1, S2,... according to the outcome for each case, and apply the same procedure recursively to each subset.

There are usually many tests that could be chosen in this last step. C4.5 uses two heuristic criteria to rank possible tests: information gain, which minimizes the total entropy of the subsets $\{S_i\}$ (but is heavily biased towards tests with numerous outcomes), and the default gain ratio that divides information gain by the information provided by the test outcomes

4. Proposed scheme

K-MEANS+C4.5 METHOD FOR ANOMALY DETECTION

In this section, we have proposed a network anomaly detection system by cascading K–Means and C4.5 decision tree algorithm. The proposed method is divided into two phases, 1) Training Phase and 2) Testing Phase.

1. Training Phase Z_i

We are provided with a training data set (X_i, Y_i), $i = 1,2,3,.....,N$ where X_i represents an n-dimensional continuous valued vector and $Y_i = \{0,1\}$ represents the corresponding class label with "0" for normal and "1" for anomaly. The proposed method has two steps: 1) training and 2) testing. During training, steps 1-3 of the k-Means-based anomaly detection method are first applied to partition the training space into k disjoint clusters $C_1, C_2, C_3, ..., C_K$. Then, C4.5 decision tree is trained with the instances in each k-Means cluster. The k-Means method ensures that each training instance is associated with only one cluster. However, if there are any subgroups or overlaps within a cluster, the C4.5 decision

tree trained on that cluster refines the decision boundaries by partitioning the instances with a set of if-then rules over the feature space.

2. Testing Phase

In the testing phase, we have two subdivided phases 1) Selection Phase and 2) Classification Phase. In selection phase, compute the Euclidean distance for every testing instance and find the closest cluster. Compute the decision tree for the closest cluster. In classification phase, apply the test instance Z_i over the C4.5 decision tree of the computed closest cluster and classify the test instance Z_i as normal or anomaly. The algorithm for the proposed method is given below.

K–Means+C4.5 Algorithm

Selection Phase

Input: Test instances Z_i, $i = 1,2,3,.....,N$.

Output: Closest cluster to the test instance Z_i.

Procedure **Selection**

Begin

Step 1: For each test instance Z_i

a. Compute the Euclidean distance $D(Z_i, r_j)$,j=1...k, and find the cluster closest to Z_i.

b. Compute the C4.5 Decision tree for the closest cluster.

End /*End Procedure*/

Classification Phase

Input: Test instance Z_i.

Output: Classified test instance Z_i as normal or anomaly

Procedure **Classification**

Begin

Step 1: Apply the test instance Z_i over the C4.5 decision tree of the computed closest cluster.

Step 2: Classify the test instance Z_i as normal or anomaly and include it in the cluster.

Step 3: Update the centre of the cluster.

End /*End Procedure*/

5. Experiment Results and Discussions

In this section, we present a performance analysis for the proposed algorithm with the related classification algorithms. We make use of the well known KDD Cup 1999 data (KDD99) [30] for network anomaly to make relevant experiments. First, we make an experiment for the proposed cascading algorithm of K–Means and C4.5 with relevant classification algorithms commonly used in Network Anomaly Detection System, including K–Means [4], SVM algorithms [17], ID3 decision tree [16] [18], Naïve Bayes algorithm, TCM–KNN (Transductive Confidence Machines for K-Nearest Neighbors) [13]

and K–Nearest Neighbor (K–NN) algorithm [1]. Second, we make comparison in order to validate the performance analysis of proposed algorithm.

5.1. Data Sets

In this section, we discuss the experimental data set: KDD99 data set [30] for network anomaly detection. The KDD99 data set [30] will contain one type of normal data and 24 different types of attacks that are categorized into four types such DoS (Denial of Service), Probes, U2R (User to Root), and R2L (Remote to Local). The packet information in the original TCP dump files were summarized into connections. The KDD99 data set contains 41 features for each instance. We have considered 15000 training instances from the KDD99 data set. The training data set will contains 60% of normal data and 40% abnormal data (Anomaly data) approximately. For testing, we considered 2500 testing instances randomly selected from the KDD99 data set.

5.2. Experiment Evaluation

The experiment was conducted a testing phase by usins the KDD99 data set [30] (41 features) as it is with out applying any feature selection methods (data set will contain 41 features). The experiments were performed in a windows machine having configuration Intel® Pentium® Core™ 2 Duo CPU E4500 @ 2.20GHz, 2.19GHz, 0.99GB of RAM, and the operating system is Microsoft Windows XP professional (SP2). We have used an open source machine learning framework Weka (Weka 3.5) [29]. Weka is a collection of machine learning algorithm for data mining applications. We have use this tool for performance comparison of our algorithm with the related other classification algorithms.

Table 1 illustrates the performance of k-Means, ID3, Naïve Bayes, K–NN, SVM, TCM–KNN and the proposed cascading algorithm of K–Means and C4.5 algorithms averaged over 5 trials for KDD99 data set for 41 features.

Classifier Algorithms	Performance Measures in %				
	TPR	FPR	Precision	Accuracy	F–Measure
K–Means	96.3	5.7	90.3	89.4	84.2
ID3	97.1	4.3	93.1	93.0	91.7
Naïve Bayes	97.1	4.2	92.5	93.2	91.5
K–NN	97.5	4.5	93.1	93.0	91.7
SVM	98.7	2.7	90.7	95.5	92.3
TCM–KNN	99.7	0	94.7	95.7	93.5
K–Means + C4.5	99.6	0.1	95.6	95.8	94.0

Table 1: Performance evaluation without applying Feature Selection algorithm for KDD99 data set [30]

6. Conclusion

Most of the Network Anomaly detection systems are designed based on availability of data instances. Many anomaly detection techniques have been specifically developed fir certain application domains, while others are more generic. In this paper, we present a cascaded algorithm using K–Means and C4.5

algorithms for Supervised Anomaly Detection. The proposed algorithm is used for detect the anomalies presented in the supervised data set. We use KDD99 data set for conducting the experiments. Performance analysis is measured by using five measures, 1) detection accuracy (or) True Positive Rate (TTR), 2)False Positive Rate (FPR), 3) Precision, 4) Total Accuracy (TA), and 5) F–Measures (FM). The proposed algorithm gives impressive detection accuracy in the experiment results.

References

[1] Aksoy, S., "K–Nearest Neighbor Classifier and Distance Functions," Technical Report, Department of Computer Engineering, Bilkent University (February 2008)

[2] Brause, R., Langsdorf, T., and Hepp, M., "Neural data mining for credit card fraud detection," In *Proceedings of IEEE International Conference on Tools with Artificial Intelligence*, 1999, pp. 103–106

[3] Das, K. and Schneider, J., "Detecting anomalous records in categorical datasets," In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* 2007. ACM Press.

[4] Diehl, C. and Hampshire, J., "Real-time object classi¯cation and novelty detection for collaborative video surveillance", In *Proceedings of IEEE International Joint Conference on Neural Networks*, 2002 IEEE, Honolulu, HI

[5] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification (2nd Edition)*. Wiley–Interscience, 2000

[6] Fan, W., Miller, M., Stolfo, S. J., Lee, W., and Chan, P. K., "Using artificial anomalies to detect unknown and known network intrusions", In *Proceedings of the 2001 IEEE International Conference on Data Mining*. IEEE Computer Society, pp. 123–130

[7] He, Z., Xu, X., Huang, J. Z., and Deng, S., "A frequent pattern discovery method for outlier detection", 2004, pp. 726–732

[8] Helmer, G., Wong, J., Honavar, V., and Miller, L., "Intelligent agents for intrusion detection", In *Proceedings of IEEE Information Technology Conference*, 1998. pp. 121–124

[9] Hu, W., Liao, Y., and Vemuri, V. R., "Robust anomaly detection using support vector machines", In *Proceedings of the International Conference on Machine Learning*, 2003 Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 282–289

[10] Janakiram, D., Reddy, V., and Kumar, A., "Outlier detection in wireless sensor networks using bayesian belief networks", In *First International Conference on Communication System Software and Middleware ,* 2006, pp.1–6

[11] Lee, W., Stolfo, S., and Chan, P., "Learning patterns from Unix process execution traces for intrusion detection", In *Proceedings of the AAAI 97 workshop on AI methods in Fraud and risk management*,1997.

[12] Lee, W. and Stolfo, S., "Data mining approaches for intrusion detection", In *Proceedings of the 7th USENIX Security Symposium*, 1998, San Antonio, TX.

[13] Lee, W., Stolfo, S. J., and Mok, K. W., "Adaptive intrusion detection: A data mining approach," *Artificial Intelligence Review*, 2000 *14,* 6, pp. 533–567

[14] Mahoney, M. V. and Chan, P. K., "Learning nonstationary models of normal network traffic for detecting novel attacks," In *Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, ACM Press, pp. 376–385.

[15] Mahoney, M. V. and Chan, P. K., "Learning rules for anomaly detection of hostile network traffic"' In *Proceedings of the 3rd IEEE International Conference on Data Mining*, 2003, IEEE Computer Society, pp. 601.

[16] Mahoney, M. V., Chan, P. K., and Arshad, M. H., "A machine learning approach to anomaly detection", Tech. Rep. CS–2003–06, Department of Computer Science, Florida Institute of Technology Melbourne FL 32901. March 2003

[17] Mingming, N. Y., "Probabilistic networks with undirected links for anomaly detection", In *Proceedings of IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop* 2000, pp. 175–179.

[18] T. Mitchell, Machine Learning. McGraw-Hill, 1997

[19] Mukkamala S, Janoski GH, "Intrusion detection: support vector machines and neural networks", In Proceedings of the *IEEE international joint conference on neural networks*, Honolulu, USA; 2002, pp. 1702–07

[20] Otey, M., Parthasarathy, S., Ghoting, A., Li, G., Narravula, S., and Panda, D., "Towards nic-based intrusion detection," In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, ACM Press, New York, NY, USA, pp. 723–728

[21] Qin, M. and Hwang, K., "Frequent episode rules for internet anomaly detection," In *Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications*, 2004, IEEE Computer Society

[22] Ratsch, G., Mika, S., Scholkopf, B., and Muller, K.R., "Constructing boosting algorithms from svms: An application to one-class classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence 24,* 9, 2002, pp. 1184–1199.

[23] Roth, V., "Outlier detection with one-class kernel fisher discriminants", In *NIPS* 2004.

[24] Roth, V., "Kernel fisher discriminants for outlier detection", *Neural Computation 18,* 4, 2006, pp. 942–960.

[25] Song, Q., Hu, W., and Xie, W., "Robust support vector machine with bullet hole image classification", *IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and* Reviews,2002, *32,* 4.

[26] Stefano, C., Sansone, C., and Vento, M., "To reject or not to reject: that is the question–an answer in case of neural classifiers", *IEEE Transactions on Systems, Management and Cybernetics 30,* 2000, 1, pp. 84–94.
[27] Tan, P.N., Steinbach, M., and Kumar, V., "Introduction to Data Mining", *Addison-Wesley*, 2005.
[28] Vapnik, V. N. , "The nature of statistical learning theory", *Springer-Verlag New York, Inc.,* 1995, New York, NY, USA
[29] "Weka–Data Mining Machine Learning Software", http://www.cs.waikato.ac.nz/ml/weka/.
[30] KDD Cup 1999 Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.