ISSN 2278 - 3091

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol.2 , No.5, Pages :47-53 (2013)
Special Issue of ICCECT 2013 - Held during September 20, 2013, Bangalore, India

Feature Selection Approach for Intrusion Detection System

Krishan Kumar[1], Gulshan Kumar[2], Yogesh Kumar[3]
[1]Associate Professor, SBS State Technical Campus, Ferozepur (Punjab)
[2]Assistant Professor, SBS State Technical Campus, Ferozepur (Punjab)
[3]Studnet, PIT(PTU Main Campus), Kapurthala (Punjab)

Abstract *:* At present, network security needs to be concerned to provide secure information channels due to increase in potential network attacks. Intrusion Detection System (IDS) is a valuable tool for the defense-in-depth of computer networks. However, building an efficient ID faces a number of challenges. One of the important challenges is dealing with data containing a high number of features. Current IDS examines all data features to detect intrusion or misuse patterns. Some of the features may be redundant or contribute little to the detection process; their usage can decrease the intrusion detection efficiency as well as taking more computational time for the effective response in real time environment. The purpose of this paper is to identify important input features in building IDS that is computationally efficient and effective. In this work we propose the feature selection method by ranking them using the various feature selection algorithms like InfoGain, GainRatio, OneR, RELIEF etc. Combining the features of the best algorithms whose performance is better by comparing the result with each other using J48 classifier. To evaluate the performance of the proposed technique several experiments are conducted on the KDDcup99 dataset for intrusion detection. The empirical results indicate that input features are important to detect the intrusions and reduces the dimensionality of the features, training time and increases overall accuracy.

Keywords: Intrusion Detection System, Feature Selection, OneR, Relief, J48.

INTRODUCTION

With the rapid development and popularity of Internet, the security of networks has been a focus in the current research [1]. In the recent past, several intruders can cause intrusions or attacks in many organizations to corrupt or theft the confidential data and create the serious problems for them. So, Intruder is a person who enters to another’s property without right or permission and an intrusion or attack can be defined as ”any set of actions that attempt to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of computer or network systems [2][3], while the Intrusion Detection is the process of monitoring the events occurring in a computer system or over a network. An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station. The Intrusion detection systems can be categorized into various classes depending upon different modules [4]. Based on data collection and storage, they can be classified into 2 categories: **Host based IDS**: Host based IDS collects the data from a host to be protected. They collect the data generally from system calls, operating system log files, NT events log file, CPU utilization, application log files, etc. **Network based IDS**: Network based IDS collects the data from the network directly in the form of packets. These IDS are operating system independent and easy to deploy to various systems. Based on data analysis and processing unit it can also be classified into 2 categories: **Misuse based IDS**: Misuse based IDS maintain a database of known attack signatures. The detection of attack involves comparison of data from the data collection unit and data stored in the database. If the match occurs then attack signal get generated. **Anomaly based IDS**: Anomaly based IDS reacts to anomalous behavior as defined by some history of monitoring systems, previous behavior or some previously defined profile of that system. The system matches the current profile with previous profile, if there is any significant deviation, then that activity is notified as an attack.

In complex domains, such as network Intrusion Detection System (IDS), a huge amount of activity data is collected from the network generating large log files and raw network traffic data, in which human inspection is impossible. Thus, these activity data must be compressed into highlevel events, called attributes. Over it, a set of attributes is obtained and monitored by the IDS in order to detect intrusion attempts. However, there are some attributes with false correlations, hiding the underlying process, and another that may be either irrelevant or redundant (its information is somehow included in other attributes). In this way, removing these attributes, or rather, selecting an optimal attributes set that adequately describes the network environment are essential in order to achieve fast and effective response against attack attempts, reduce the complexity and the computation time, and increase the precision of the IDS. In this way, we compare various feature selection algorithms like Infogain, GainRatio, SVM, OneR, Chi-square, Relief etc for selecting optimal attributes.

This paper proposes a new feature selection method for intrusion detection using the existing feature selection algorithms i.e. OneR and RELIEF. Compared the output of proposed method to each of the above algorithm using J48 classifier in WEKA tool. The effectiveness of the proposed method is evaluated by conducting several experiments on KDDCup99 network intrusion dataset. The results show that the proposed FS method increases the accuracy level and reduces the training time. The rest of this paper is organized as follows: Section 2 presents a background of the used methods, including Feature

Selection (FS), OneR, RELIEF and J48. Section 3 describes The KDDCup99 network intrusion dataset. Section 4 introduces the proposed Methodology for feature selection in IDS. Section 5 gives the implementation results and analysis. Finally, Section 6 concludes the result.

BACKGROUND

This section gives an overview of Feature Selection (FS), One Rule (OneR), RELIEF and J48 Classifier.

Feature Selection (FS)

Data mining is a multidisciplinary effort to extract theknowledge from data and feature reduction is an integral component of successful data mining. There are two main feature dimensionality reduction approaches are typically used: feature extraction and feature selection [5]. According to Jain et al., feature extraction is methods that create new features based on transformations or combinations of the original feature set. The term feature selection refers to methods that select the best subset of the original feature set. Feature selection is a process of selecting a subset of original features according to certain criteria, is an important and frequently used dimensionality reduction technique for data mining [6]. It reduces the number of features, removes irrelevant, redundant, or noisy data, and brings the immediate effects for applications: speeding up a data mining algorithm, and improving mining performance such as predictive accuracy and result comprehensibility.

To deal with these problems Feature selection algorithms can be classified into filters and wrappers [7]. Filter methods select subset of features as a preprocessing step, independent of the induction (learning) algorithm. Wrappers utilize the classifier (learning machine) performance to evaluate the goodness of feature subsets. Several different criteria have been used for evaluating the goodness of a feature [8] including distance measures, dependency measures, consistency measures, information measures, and classification error measures.

One Rule (OneR)

OneR, short for ”One Rule”, is a simple classification algorithm that generates a one-level decision tree. OneR is able to infer typically simple, yet accurate, classification rules from a set of instances. The OneR algorithm creates one rule for each attribute in the training data, and then selects the rule with the smallest error rate as its ’one rule’. To create a rule for an attribute, the most frequent class for each attribute value must be determined. The most frequent class is simply the class that appears most often for that attribute value. A rule is simply a set of attribute values bound to their majority class; one such binding for each attribute value of the attribute the rule is based on [9].

Algorithm 1: OneR Algorithm

1) For each predictor,
2) For each value of that predictor, make a rule as follows;
3) Count how often each value of the target (class) appears
4) Find the most frequent class
5) Make the rule assign that class to this value of the predictor
6) Calculate the total error of the rules of each predictor
7) Choose the predictor with the smallest total error.

RELIEF

RELIEF is a well-known feature-weighting (ranking) approach that first introduced by Kira and Rendell [10][11]. The basic idea is to measure the relevance of features in the neighborhoods around target samples. For each target sample, RELIEF finds the nearest sample in feature space of the same category, called the ”hit” sample, then measures the distance between the target and hit samples. It also finds the nearest sample of the other category, called the ”miss” sample, and then does the same work. RELIEF uses the difference between those measured distances as the weight of a target feature [12].

Algorithm 2: Relief Algorithm

Input: for each training instance a vector of attribute values and the class value

Output: the vector W of estimations of the qualities of attributes

1) Set all weights W [A] : = 0:0;
2) for i := 1 to m do begin
3) Randomly select an instance Ri;
4) Find nearest hit H and nearest miss M;
5) for A: = 1 to a do
6) W [A] :=W[A]diff (A; Ri; H)=m + diff (A; Ri; M)=m;
7) end;

J48 Classifier

J48 is an open source Java implementation of the C4.5 algorithm of the WEKA data mining tool. C4.5 is based on the ID3 algorithm developed by Ross Quinlan [13], with additional features to address problems that ID3 was unable to deal. In practice, the J48 is a Decision tree classifier algorithm. In this algorithm for classification of new item, it first needs to create a decision tree based on the attribute values of the available training data. It discriminates the various instances and identify the attribute for the same. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate that branch and assign it to the target value that we have obtained.

NETWORK INTRUSION DATASET: KDDCUP99

To evaluate IDS schemes, NSL-KDD dataset [14] benchmarks are used as, for instance, the intrusion dataset available in Knowledge Discovery and Data Mining Competition KDDCup99 [15] for both training and testing. This dataset is still used by researchers because it has the capability to compare different intrusion detection techniques on a common dataset base. In the KDD99 database, any network connection (or instance) is comprised of 41 attributes and each instance is labelled either as normal or as an attackspecified type. In KDD99 database, there are 494,021 instances in which 97,278 are considered normal and 396,744 are labelled as attacked by 22 different types that can be classified into 4 main categories as follows:

Denials-of Service (DoS) attacks have the goal of limiting or denying services provided to the user, computer or network. A common tactic is to severely overload the targeted system. (E.g. apache, smurf, Neptune, Ping of death, back, mailbomb, udpstorm, SYNflood, etc.).

Probing or Surveillance attacks have the goal of gaining knowledge of the existence or configuration of a computer system or network. Port Scans or sweeping of a given IP address range typically fall in this category. (e.g. saint, portsweep,mscan, nmap, etc.).

User-to-Root (U2R) attacks have the goal of gaining root or super-user access to a particular computer or system on which the attacker previously had user level access. These are attempts by a non-privileged user to gain administrative privileges (e.g. Perl, xterm, etc.).

Remote-to-Local (R2L) attack is an attack in which a user sends packets to a machine over the internet, which the user does not have access to in order to expose the machine vulnerabilities and exploit privileges which a local user would have on the computer (e.g. xclock, dictionary, guestpassword, phf, sendmail, xsnoop, etc.).

It is clear that the total number of connection records to be used for training and testing of the classifiers is very large. Moreover the number of connection records related to U2R and R2L is very less as compared to other attack classes. So, in order to reduce non-uniformity in the dataset, we randomly selected maximum of 44,000 connection records of each attack type for the purpose of training and testing the classifiers in an unbiased manner. In order to test the classifiers, we randomly selected 37,791 connection records as a training data set. Table 1 shows the detail of 6763 connection records in the test dataset. KDD dataset contains symbolic as well as continuous features. The dataset is pre-processed before it is used for training and testing the classifiers.

Table 1: Details of Connection Records in used dataset

Label	# Training Dataset	# Testing Dataset
Normal	12533	1609
Probe	11656	1607
DoS	12555	1628
R2L	52	200
U2R	995	1719
Total Records	37791	6753

The pre-processing of NSL-KDD dataset involves following steps [16]:

1) Convert Symbolic features to a numeric value.
2) The attack type feature is mapped to one of attack classes namely Normal, Probe, DoS, U2R and R2L.

Normalize the feature values, since the data have significantly varying resolution and ranges. The feature values are scaled to the range [0, 1], using the following equation.

$$Normalizedvalue_i = normalize(\ln(\mathrm{val}_i + 1))$$

$$normalize(\mathrm{x}_i) = \frac{x_i - \ln(Min_i + 1)}{\ln(Max_i + 1) - \ln(Min_i + 1)}$$

Where val_i is the initial value of feature i and Min_i and Max_i are the minimum and maximum values of feature i, respectively. For some feature reduction methods which process only discrete values of the features, the mapped & normalized dataset is further discretized to obtain discrete values for continuous features using WEKA.

PROPOSED FEATURE SELECTION METHOD

Proposed feature selection method efficiently reduced the dimensionality of the NSL-KDD dataset from 41 features to 12 features, which reduces 70.73% of the feature dimension space. We obtained the ranking of all 41 attributes using the various existing single attribute evaluator algorithms InfoGain, GainRatio, SVM, Chi-square, OneR, Relief etc. According to the set of rules [17] that are applied to the ranked attributes and get the subset of the relevant features of each algorithm. Comparing the each subset with other and the original dataset.

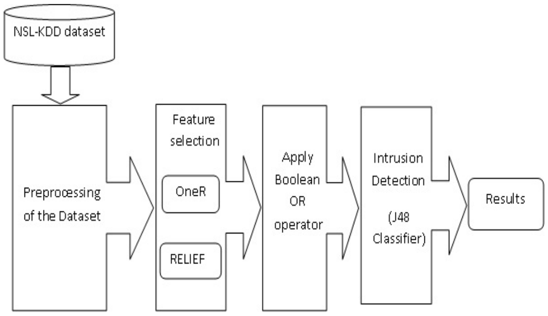
NSL-KDD dataset
Preprocessing of the Dataset
Feature selection
OneR
RELIEF
Apply Boolean OR operator
Intrusion Detection (J48 Classifier)
Results

Fig 1: The structure of the proposed feature selection method for network intrusion detection.

Feature Selection Phase

Table 2 Shows the comparison of the various existing single attribute evaluator algorithms InfoGain, GainRatio, SVM, Chi-square, OneR, Relief etc. Comparison shows that the accuracy level of OneR & Relief algorithms with 10 attributes each is far better than other algorithms, means that both algorithms can ranked most relevant attributes.

Combining Technique

Apply Boolean operators OR, AND & EXOR on attributes of best algorithms i.e OneR and RELIEF. After performing various experiments on the reduced set, results show that with the use of the Boolean logical OR operator (Disjunction) between the relevant feature subset of OneR & Relief algorithms got the reduced feature subset of the 12 features out of 41 and reduces the complexity and the computation time, and increase the precision of the IDS.

J48 Classifier

The dataset which has been reduced by disjunction Boolean operator is passed to the J48 classifier to be classified. The algorithm of J48 [18] classifier is described in Algorithm 3.

Algorithm 3: J48 Classifier Algorithm

1) It builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy.
2) At each node of the tree, J48 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other.
3) Its criterion is the normalized information gain that results from choosing an attribute for splitting the data.
4) The attribute with the highest normalized information gain is chosen to make the decision.
5) For each attribute, the gain is calculated and the highest gain is used in the decision node.

IMPLEMENTATION RESULTS AND ANALYSIS

In order to evaluate the proposed attribute selectionschemes, it was used the WEKA toolkit (Waikato Environment for Knowledge Analysis) [19] on NSL-KDD dataset, where 37,789 records are randomly taken for training and testing purposes. All experiments have been performed using the Intel Core i3 2.40 GHz processor with 4GB of RAM. The experiments have been implemented using Java language environment with a ten-fold cross-validation.

Performance Metric

To evaluate our system, besides the classical accuracy measure, the three standard metrics of detection rate (DR) / True Positive Rate (TPR), False positive rate (FPR) & F-measure developed for network intrusions [20], have been used. Table 3 shows these standard metrics.

Table 3: Confusion Matrix

Class	Predicted Normal	Predicted Attack
Actual Normal	TN	FP
Actual Attack	FN	TP

A confusion matrix that summarizes the number of instances predicted correctly or incorrectly by a classification model.

Table 2: Comparison of Various Feature Selection Algorithms.

FS Algorithms	KDDCup99	Chi-square		Relief		Infogain		Gainratio		SVM		OneR	
No. of Attributes	41	10	15	10	15	10	15	10	15	10	15	10	15
Accuracy with J48	61.39	65.39	64.91	66.51	64.88	63.05	65.07	63.46	61.94	62.14	60.22	65.56	65.25
Training time (sec.)	7.67	2.24	4.21	2.62	5.06	2.17	3.12	2.82	3.74	3.28	4.74	2.93	3.25

False positive (FP): Or false alarm, Corresponds to the number of detected attacks but it is in fact normal.

False negative (FN): Corresponds to the number of detected normal instances but it is actually attacked, in other words these attacks are the target of intrusion detection systems.

True positive (TP): Corresponds to the number of detected attacks and it is in fact attack.

True negative (TN): Corresponds to the number of detected normal instances and it is actually normal.

The accuracy of an intrusion detection system is measured regarding to Detection Rate (DR) and False Positive Rate (FPR).

$$DR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

$$F - measure = \frac{2 * TP}{(2 * TP) + FP + FN}$$

These metrics are important because they measure the percentage of intrusions the system is able to detect and how many misclassifications it makes. To visualize the tradeoff between the false positive and the detection rates, the ROC (Receiving Operating Characteristic) curves are also depicted. Furthermore, to compare classifiers it is common to compute the area under the ROC curve, denoted as AUC. The higher the area, the better is the average performance of the classifier.

Results and Analysis

1) Experiment 1: Using the J48 Classifier

Table 4 shows the accuracy measurements achieved by the 41 full dimension features of the NSL-KDD dataset. While, Table 5 gives the accurate measurements for the proposed feature selection method for network intrusion detection system with reduced 12 dimension features with the use of the J48 classifier.

Table 4 & 5 depict the performance of the J48 by using the original 41-variable data set and the 12-variable reduced data set. The False positive rate for each class decreases from 0.121 to 0.104 and the area under the curve increases from 0.772 to .855 when the 12-variable data set is used. It is clear that the classification accuracy achieved by reduced feature set is improved than using 41-variable dataset with J48 standalone classifier.

Table 4: J48 Accuracy Measurements (41 Feature Dimensionality)

Class Name	TPR	FPR	F-Measure	ROC
Normal	0.965	0.322	0.644	0.84
Probe	0.787	0.16	0.684	0.844
DoS	0.741	0.023	0.818	0.861
U2R	0.135	0	0.238	0.813
R2L	0.059	0.002	0.111	0.553
Weighted Avg.	0.614	0.121	0.548	0.772

To facilitate the comparison, we also use InfoGain (IG) a well known filter based methods of feature selection. Table 6 shows the classification accuracy of combining IG feature selection algorithm with the J48 classifier.

Table 5: Proposed Technique Accuracy Measurements (12 Feature Dimensionality)

Class Name	**TPR**	**FPR**	**F-Measure**	**ROC**
Normal	0.964	0.358	0.62	0.876
Probe	0.775	0.03	0.828	0.897
DoS	0.831	0.038	0.851	0.896
U2R	0.202	0	0.239	0.505
R2L	0.212	0.001	0.342	0.799
Weighted Avg.	0.668	0.104	0.637	0.855

Table 6: J48 Accuracy Measurements with InfoGain (20 Feature Dimensionality)

Class Name	**TPR**	**FPR**	**F-Measure**	**ROC**
Normal	0.963	0.37	0.612	0.824
Probe	0.671	0.018	0.777	0.844
DoS	0.833	0.035	0.857	0.89
U2R	0.015	0.022	0.017	0.482
R2L	0.106	0.053	0.169	0.7
Weighted Avg.	0.617	0.115	0.58	0.803

Validation of Experiment 1

To validate the proposed method, the testing accuracy, feature numbers and timing speed of the proposed system is compared with different feature selection methods. Table 7 shows the comparison results of the proposed system with 41-variable, 20-variable of IG,15-variable of GR and 10-variable of Relief using J48 classifier for intrusion detection. Table 7 illustrates that, the proposed method for network intrusion detection systems gives the best accuracy performance (66.807%). Also the proposed method for network intrusion detection systems reduced the feature space to 12, which leads to enhance the timing speed to 3.07 sec which is very important for real time network applications.

Table 7 : Testing accuracy, No. of Features and Timing comparison

System	**Accuracy Level**	**Attribute No.**	**Training Time**
Original Features	61.3929	41	7.67 sec
Infogain	61.659	20	5.99 sec
Gainratio	61.94	15	3.74 sec
Relief	59.515	10	2.9 sec
Proposed Technique	66.8047	12	3.07 sec

CONCLUSION

Focus on research we proposed the feature selection approach in Intrusion detection, earlier most of the existing IDS's use all 41 features in the network to evaluate and look for intrusive patterns and some of these features are redundant and irrelevant. The drawback of that approach is time-consuming in detection process and degrading the performance of IDS. To solve this problem we proposed a method for feature selection by combining different feature selection algorithms for intrusion detection. We used the feature selection method using union of the two best algorithms i.e. OneR & Relief.. The goal of this work is to reduce the dimensionality of the data while retaining as much as possible of the variation present in the original dataset. Tests and comparison are done on KDDcup99 dataset. The test data contains 4 kinds of different attacks in addition to normal system call. Our experimental results showed that the proposed model gives better and robust representation of data as it was able to reduces 70.73% of the feature dimension space and approximately 55-60%reduction in training time ,and classification accuracy increased from 61.39% to 66.80% in detecting attacks. Meantime it significantly reduce a number of computer resources, both memory and CPU time, required to detect an attack. This shows that our proposed feature selection method is applicable to select the most relevant features in intrusion detection

REFERENCES

[1] C. F. L. Lima, F. M. de Assis, and C. P. de Souza, “An empirical investigation of attribute selection techniques based on shannon, r´enyi and tsallis entropies for network intrusion detection,” *American Journal of Intelligent Systems*, vol. 2, no. 5, pp. 111–117, 2012.

[2] K. Scarfone and P. Mell, “Guide to intrusion detection and prevention systems (idps),” *NIST Special Publication*, vol. 800, no. 2007, p. 94, 2007.

[3] J. McHugh, A. Christie, and J. Allen, “Defending yourself: The role of intrusion detection systems,” *Software, IEEE*, vol. 17, no. 5, pp. 42–51, 2000.

[4] G. Kumar, K. Kumar, and M. Sachdeva, “The use of artificial intelligence based techniques for intrusion detection: a review,” *Artificial Intelligence Review*, vol. 34, no. 4, pp. 369–387, 2010.

[5] K. Jain, R. P. W. Duin, and J. Mao, “Statistical pattern recognition: A review,” Pattern Analysis and Machine Intelligence, *IEEE Transactions on*, vol. 22, no. 1, pp. 4–37, 2000.

[6] P. A. Est´evez, M. Tesmer, C. A. Perez, and J. M. Zurada, “Normalized mutual information feature selection,” Neural Networks, *IEEE Transactions on*, vol. 20, no. 2, pp. 189–201, 2009.

[7] G. H. John, R. Kohavi, K. Pfleger et al., “Irrelevant features and the subset selection problem.” in *ICML*, vol. 94, 1994, pp. 121–129.

[8] M. Dash and H. Liu, “Feature selection for classification,” *Intelligent data analysis*, vol. 1, no. 3, pp. 131–156, 1997.

[9] G. Buddhinath and D. Derry, “A simple enhancement to one rule classification,” *Technique Report at*, 2006.

[10] K. Kira and L. A. Rendell, “The feature selection problem: Traditional methods and a new algorithm,” in *AAAI*, 1992, pp. 129–134.

[11] K. Kira and L. A. Rendell, “A practical approach to feature selection,” in *Proceedings of the ninth international workshop on Machine learning*. Morgan Kaufmann Publishers Inc., 1992, pp. 249–256.

[12] M. Robnik-ˇ Sikonja and I. Kononenko, “Theoretical and empirical analysis of relieff and rrelieff,” *Machine learning*, vol. 53, no. 1-2, pp. 23–69, 2003.

[13] J. R. Quinlan, *C4. 5: programs for machine learning*. Morgan kaufmann, 1993, vol. 1.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, “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, 2009.

[15] KDD, “Kddcup99 intrusion dataset.” [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/

[16] G. K. Ahuja, K. K. Saluja, and M. Sachdeva, “An empirical comparative analysis of feature reduction methods for intrusion detection,” *International Journal of Information and Telecommunication Technology (ISSN: 0976-5972)*, vol. 1, no. 1, 2010.

[17] A. H. Sung and S. Mukkamala, “Identifying important features for intrusion detection using support vector machines and neural networks,” in *Applications and the Internet, 2003. Proceedings. 2003 Symposium on. IEEE*, 2003, pp. 209–216.

ISSN 2278 - 3091

[18] M. Revathi and T. Ramesh, “Network intrusion detection system using reduced dimensionality,” *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 2, no. 1, pp. 61–67, 2011.

[19] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[20] G. Folino, C. Pizzuti, and G. Spezzano, “Gp ensemble for distributed intrusion detection systems,” in *Pattern Recognition and Data Mining*. Springer, 2005, pp. 54–62.