# A Comparison of Feature-Selection Methods for Intrusion Detection

Hai Thanh Nguyen, Slobodan Petrović, and Katrin Franke

Norwegian Information Security Laboratory
Gjøvik University College, Norway
{hai.nguyen,slobodan.petrovic,katrin.franke}@hig.no

**Abstract.** Feature selection is an important pre-processing step in intrusion detection. Achieving reduction of the number of relevant traffic features without negative effect on classification accuracy is a goal that greatly improves overall effectiveness of an intrusion detection system. A major challenge is to choose appropriate feature-selection methods that can precisely determine the relevance of features to the intrusion detection task and the redundancy between features. Two new feature selection measures suitable for the intrusion detection task have been proposed recently [11,12]: the correlation-feature-selection (CFS) measure and the minimal-redundancy-maximal-relevance (mRMR) measure. In this paper, we validate these feature selection measures by comparing them with various previously known automatic feature-selection algorithms for intrusion detection. The feature-selection algorithms involved in this comparison are the previously known SVM-wrapper, Markov-blanket and Classification & Regression Trees (CART) algorithms as well as the recently proposed generic-feature-selection ($GeFS$) method with 2 instances applicable in intrusion detection: the correlation-feature-selection ($GeFS_{CFS}$) and the minimal-redundancy-maximal-relevance ($GeFS_{mRMR}$) measures. Experimental results obtained over the KDD CUP'99 data set show that the generic-feature-selection ($GeFS$) method for intrusion detection outperforms the existing approaches by removing more than 30% of redundant features from the original data set, while keeping or yielding an even better classification accuracy.

**Keywords:** intrusion detection; feature selection; polynomial mixed $0-1$ fractional programming; mixed $0-1$ integer linear programming.

## 1 Introduction

The problem of intrusion detection is often analyzed as a pattern recognition problem - an Intrusion Detection System (IDS) has to tell normal from abnormal behaviour of network traffic and/or command sequences on a host. In addition, it is of interest to further classify abnormal behaviour in order to undertake adequate counter-measures. An IDS can be modeled in various ways (see for example [9], [10]). A model of this kind usually includes a representation algorithm

I. Kotenko and V. Skormin (Eds.): MMM-ACNS 2010, LNCS 6258, pp. 242–255, 2010.

(for representing incoming data in the space of selected features) and a classification algorithm (for mapping the feature vector representation of the incoming data to elements of a certain set of values, e.g. normal or abnormal etc.) Some IDS, like the ones presented in [9], also include the feature selection algorithm, which determines the features to be used by the representation algorithm. Even if the feature-selection algorithm is not included in the model directly, it is always assumed that such an algorithm is run before the very intrusion detection process.

The quality of the feature selection algorithm is one of the most important factors that affect the effectiveness of an IDS. The goal of the algorithm is to determine the most relevant features of the incoming traffic, whose monitoring would ensure reliable detection of abnormal behaviour. Since the effectiveness of the classification algorithm heavily depends on the number of features, it is necessary to minimize the cardinality of the set of selected features, without dropping potential indicators of abnormal behaviour. Obviously, determining a good set of features is not an easy task. The most of the work in practice is still done manually and the feature selection algorithm depends too much on expert knowledge. Automatic feature selection for intrusion detection is therefore important. For automatic feature selection, the wrapper and the filter models from machine learning are frequently applied [18]. The wrapper model assesses the selected features by learning algorithm's performance. Therefore, the wrapper method requires a lot of time and computational resources to find the best feature subsets. The filter model considers statistical characteristics of a data set directly without involving any learning algorithm. Due to the computational efficiency, the filter method is usually used to select features from high-dimensional data sets, such as intrusion detection systems. The filter model encompasses two groups of methods: the feature ranking methods and the feature-subset-evaluating methods. The feature ranking methods assign weights to features individually based on their relevance to the target concept. The feature-subset-evaluating methods estimate feature subsets not only by their relevance, but also by the relationships between features that make certain features redundant. It is well known that the redundant features can reduce the performance of a pattern recognition system. Therefore, the feature-subset-evaluating methods are more suitable for selecting features for intrusion detection. A major challenge in the IDS feature selection process is to choose appropriate measures that can precisely determine the relevance of features to the intrusion detection task and the relationship between features of a given data set.

Since the relevance and the relationship are usually characterized in terms of correlation or mutual information [4,19], we focus on two feature selection measures for intrusion detection task: the correlation-feature-selection (CFS) measure [1] and the minimal-redundancy-maximal-relevance (mRMR) measure [2]. In [11,12], a new search method that ensures globally optimal feature sets by means of the CFS and the mRMR measures was proposed. It was shown that the proposed search method outperforms the heuristic search strategies by removing much more redundant features from the KDD CUP 1999 data set [7] and still keeping the classification accuracies or even getting better performances. In this paper, the feature selection measures proposed in [11,12] are validated by comparison with various previously known automatic feature-selection algorithms for intrusion detection. Thus, the feature-selection algorithms involved in the comparison are the previously known SVM-wrapper [13], Markov-blanket [14] and CART [14] and the new generic-feature-selection ($GeFS$) method with 2 instances applied in intrusion detection: the correlation-feature-selection ($GeFS_{CFS}$) [11] and the minimal-redundancy-maximal-relevance ($GeFS_{mRMR}$) [12] measures.

A theoretical basis for comparison of the methods proposed in [11,12] and the other methods is difficult to give. Such a basis would require the general solution of the problem of comparison of filter and wrapper methods, which is not known (sometimes, the filter methods perform better, but sometimes the wrapper methods perform better). Because of that, in this paper we present the results of practical comparison achieved on a particular data set. Then the generalization of the results of the comparison depends to a large extent on the quality and generality of the test data set. We believe that the data set used for this comparison with the modifications described below is general enough to claim that our comparison results can be generalized with high probability.

Any feature selection algorithm selects relevant traffic features based on labelled data (Fig.1). In this research, we used the KDD CUP'99 [7] data set for this purpose, since all the existing approaches involved in the comparison used the same data set for evaluation [13,14]. The full feature set assigned to this data set consists of 41 features. It is well known [15,16] that the KDD CUP'99 data set has several drawbacks regarding its suitability for representation of modern traffic. To avoid problems related to this data set, we split it into 4 parts according to the category of attack: DoS, Probe, U2R and R2L; we consider only two attack classes: DoS and Probe. This ensures more objective classification, since in such a way the influence of difference in cardinality of these subsets in the overall data set is reduced. We compare the feature-selection algorithms by the number of selected features as well as by the classification accuracy of machine learning algorithms chosen as classifiers for intrusion detection. Experimental results obtained over the KDD CUP'99 data set show that the $GeFS$ method outperforms the existing approaches by removing more than 30% of redundant features from the original data set, while keeping or yielding an even better classification accuracy. Even though the KDD CUP'99 data set does not reflect completely the characteristics of contemporary traffic, the results of our comparison indicate that the $GeFS$ method for selecting features would behave well on general intrusion detection data as well.

The paper is organized as follows. In Section 2, we give an overview of the feature-selection methods involved in the comparison. In Section 3, we present experimental setting as well as experimental results regarding the number of selected features and the classification accuracy obtained over the KDD Cup'99 data set. Section 4 summarizes our findings.

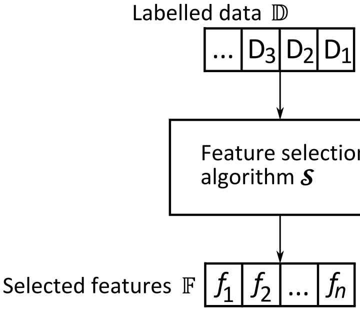
Labelled data $\mathbb{D}$
... $D_3$ $D_2$ $D_1$
Feature selection algorithm $S$
Selected features $\mathbb{F}$
$f_1$ $f_2$ ... $f_n$

**Fig. 1.** A feature selection algorithm

## 2 Feature-Selection Methods for Intrusion Detection

In this section, we first describe the previously known feature-selection methods used in intrusion detection. Then we give an overview of the recently proposed generic-feature-selection ($GeFS$) method together with 2 instances applied in intrusion detection: the correlation-feature-selection ($GeFS_{CFS}$) [11] and the minimal-redundancy-maximal-relevance ($GeFS_{mRMR}$) [12] measures.

### 2.1 Existing Approaches

#### 2.1.1 SVM-Wrapper

Sung and Mukkamala [13] used the ranking methodology to select important features for intrusion detection: One input feature is deleted from the data at a time and the resultant data set is then used for the training and testing of the classifier Support Vector Machine (SVM) [17]. Then the SVMs performance is compared to that of the original SVM (based on all features) in terms of relevant performance criteria, such as overall accuracy of classification, training time and testing time. The deleted feature will be ranked as "important", "secondary" or "insignificant" according to the following rules:

- If accuracy decreases **and** training time increases **and** testing time decreases, **then** the feature is important.
- If accuracy decreases **and** training time increases **and** testing time increases, **then** the feature is important.
- If accuracy decreases **and** training time decreases **and** testing time increases, **then** the feature is important.
- If accuracy is not changed **and** training time increases **and** testing time increases, **then** the feature is important.
- If accuracy is not changed **and** training time decreases **and** testing time increases, **then** the feature is secondary.
- If accuracy is not changed **and** training time increases **and** testing time decreases, **then** the feature is secondary

– If accuracy is not changed **and** training time decreases **and** testing time decreases, **then** the feature is insignificant.
– If accuracy increases **and** training time increases **and** testing time decreases, **then** the feature is secondary.
– If accuracy increases **and** training time decreases **and** testing time increases, **then** the feature is secondary.
– If accuracy increases **and** training time decreases **and** testing time decreases, **then** the feature is insignificant

In [13] the experiment was conducted on a part of KDD CUP'99 data set [7]. This data set contains normal traffic and four main attack classes: Denial-of-Service (DoS) attacks, Probe attacks, User-to-Root (U2R) attacks and Remote-to-Local (R2L) attacks. Some important features were selected and the obtained data set after removing irrelevant features was classified by SVM [17]. The results are given in Table 1.

**Table 1.** Performance of SVM using selected features (SF) [13]

| **Classes** | **Number-of-SF** | **Accuracy** |
|---|---|---|
| Normal | 25 | 99.59% |
| DoS | 19 | 99.22% |
| Probe | 7 | 99.38% |
| U2R | 8 | 99.87% |
| R2L | 6 | 99.78% |

### 2.1.2 Markov-Blanket

Markov blanket MB($T$) of the output variable $T$ is defined as the set of input variables such that all other variables are probabilistically independent of $T$. Knowledge of MB($T$) is sufficient for perfectly estimating the distribution of $T$ and thus for classifying $T$. Markov blanket has been applied for feature selection in many domains [4]. In 2004, Chebrolu et. al. [14] proposed to use Markov blanket for selecting important features for intrusion detection. In order to do that, they constructed a Bayesian Network (BN) from the original data set. A Bayesian network $B = (N, A, Q)$ is a Directed Acyclic Graph (DAG) $(N, A)$ where each node $n \in N$ represents a domain variable (e.g. a data set attribute or variable), and each arc $a \in A$ between nodes represents a probabilistic dependency among the variables. A BN can be used to compute the conditional probability of one node, given values assigned to the other nodes. From the constructed BN, the Markov blanket of a feature $T$ is the union of $T$'s parents, $T$'s children and eventually other parents of $T$'s children. An example of a Bayesian Network is given in Fig.2. The gray-filled nodes constitute the MB($T$):

For conducting the experiment, Chebrolu et. al. [14] randomly chose 11,982 instances from the overall (5 millions of instances) KDD CUP'99 data set [7]. 17 features were selected and the Bayesian Network [17] was used for classifying

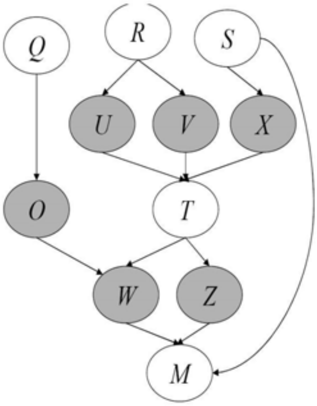

**Fig. 2.** An example of Markov blanket

**Table 2.** Performance of Bayesian Network using selected features (SF) [14]

| Classes | Number-of-SF | Accuracy |
|---|---|---|
| Normal | 17 | 99.64% |
| DoS | 17 | 98.16% |
| Probe | 17 | 98.57% |
| U2R | 17 | 60.00% |
| R2L | 17 | 98.93% |

the obtained data set after removing irrelevant features. The results are given in Table 2.

### 2.1.3 CART

The Classification and Regression Trees (CART) approach [17] is based on binary recursive partitioning. The process is binary because parent nodes are always split into exactly two child nodes and recursive because it is repeated by treating each child node as a parent. The key elements of CART methodology are a set of splitting rules in a tree; deciding when the tree is complete and assigning a class to each terminal node. Feature selection for intrusion detection is based on the contribution of the input variables to the construction of the decision tree from the original data set. The importance of features is determined by the role of each input variable either as a main splitter or as a surrogate. Surrogate splitters are considered as back-up rules that closely mimic the action of primary splitting rules. For example, in the given model, the algorithm splits data according to the variable *protocol_type* and if a value for *protocol_type* is not available then the algorithm might use the *service* feature as a good surrogate. Feature importance, for a particular feature is the sum across all nodes in the tree of the improvement scores that the predictor has when it acts as a primary or surrogate splitter. For

**Table 3.** Performance of CART using selected features (SF) [14]

| Classes | Number-of-SF | Accuracy |
|---|---|---|
| Normal | 12 | 100% |
| DoS | 12 | 85.34% |
| Probe | 12 | 97.71% |
| U2R | 12 | 64.00% |
| R2L | 12 | 95.56% |

example, for the node $i$, if the feature appears as the primary splitter then its importance could be given as $i_{importance}$. But if the feature appears as the $n^{th}$ surrogate instead of the primary variable, then the importance becomes $i_{importance} = (p^n) \times i_{improvement}$ in which $p$ is the *surrogate improvement weight* which is a user controlled parameter set between 0 and 1.

Chebrolu et. al. [14] conducted the experiment on the data set, which contains randomly chosen 11,982 instances from the overall (5 millions of instances) KDD CUP'99 data set [7]. 12 features were selected and the CART [17] was used for classifying the obtained data set after removing irrelevant features. The results are given in Table 3.

## 2.2 A New Generic-Feature-Selection Measure

In this subsection, we give an overview of the generic-feature-selection ($GeFS$) method together with 2 instances applied in intrusion detection: the ($GeFS_{CFS}$) and the ($GeFS_{mRMR}$) measures.

### 2.2.1 Definitions

**Definition 1:** A generic-feature-selection measure used in the so-called filter model is a function $GeFS(x)$, which has the following form [12]:

$$GeFS(x) = \frac{a_0 + \sum_{i=1}^{n} A_i(x)x_i}{b_0 + \sum_{i=1}^{n} B_i(x)x_i}, x = (x_1, \dots, x_n) \in \{0,1\}^n \tag{1}$$

In this definition, binary values of the variable $x_i$ indicate the appearance ($x_i = 1$) or the absence ($x_i = 0$) of the feature $f_i$; $a_0$, $b_0$ are constants; $A_i(x)$, $B_i(x)$ are linear functions of variables $x_1, \dots, x_n$.

**Definition 2:** The feature selection problem is to find $x \in \{0,1\}^n$ that maximizes the function $GeFS(x)$ [12]:

$$\max_{x \in \{0,1\}^n} GeFS(x) = \frac{a_0 + \sum_{i=1}^{n} A_i(x)x_i}{b_0 + \sum_{i=1}^{n} B_i(x)x_i} \tag{2}$$

There are several feature selection measures, which can be represented by the form (1), such as the correlation-feature-selection (CFS) measure [1], the minimal-redundancy-maximal-relevance (mRMR) measure [2], Mahalanobis distance, etc.

A major challenge in the IDS feature-selection process is to choose appropriate measures that can precisely determine the relevance of features to the intrusion detection task and the redundancy between features. Since the relevance and the redundancy are usually characterized in terms of correlation or mutual information [4], the following measures for application in intrusion detection were considered in [11,12]: the correlation-feature-selection (CFS) measure [1] and the minimal-redundancy-maximal-relevance (mRMR) measure [2].

### 2.2.2 Correlation Feature Selection Measure

The Correlation Feature Selection (CFS) measure evaluates subsets of features on the basis of the following hypothesis: *"Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other"* [1]. The following equation gives the merit of a feature subset $S$ consisting of $k$ features:

$$Merit_S(k) = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

Here, $\overline{r_{cf}}$ is the average value of all feature-classification correlations, and $\overline{r_{ff}}$ is the average value of all feature-feature correlations. The CFS criterion is defined as follows:

$$\max_{S_k} \left[\frac{r_{cf_1} + r_{cf_2} + ... + r_{cf_k}}{\sqrt{k + 2(r_{f_1 f_2} + .. + r_{f_i f_j} + .. + r_{f_k f_1})}}\right] \tag{3}$$

Suppose that there are $n$ full-set features. Binary values of the variable $x_i$ are used to indicate the appearance ($x_i = 1$) or the absence ($x_i = 0$) of the feature $f_i$ in the globally optimal feature set [11]. Therefore, the problem (3) can be rewritten as an optimization problem as follows:

$$\max_{x \in \{0,1\}^n} \left[\frac{(\sum_{i=1}^{n} a_i x_i)^2}{\sum_{i=1}^{n} x_i + \sum_{i \neq j} 2b_{ij} x_i x_j}\right] \tag{4}$$

It is obvious that the CFS measure is an instance of the GeFS measure. In [12], this measure was denoted by $GeFS_{CFS}$.

### 2.2.3 The mRMR Feature Selection Measure

In 2005, Peng et. al. [2] proposed a feature-selection method, which is based on mutual information. In this method, the relevance of features and the redundancy between features are considered simultaneously. In terms of mutual information, the relevance of a feature set $S$ for the class $c$ is defined by the mean value of all mutual information values between the individual feature $f_i$ and the class $c$ as follows:

$$D(S, c) = \frac{1}{|S|} \sum_{f_i \in S} I(f_i; c)$$

The redundancy between features in the set $S$ is the mean value of all mutual information values between the feature $f_i$ and the feature $f_j$:

$$R(S) = \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i; f_j)$$

The mRMR criterion is a combination of two measures given above and is defined as follows:

$$\max_S \Big[\frac{1}{|S|} \sum_{f_i \in S} I(f_i; c) - \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i; f_j)\Big] \tag{5}$$

By using binary values of the variable $x_i$ as in the case of the CFS measure to indicate the appearance or the absence of the feature $f_i$ and by denoting the mutual information values $I(f_i; c)$ and $I(f_i; f_j)$ by constants $c_i$ and $a_{ij}$, respectively, the problem (5) can be described as an optimization problem as follows:

$$\max_{x \in \{0,1\}^n} \Big[\frac{\sum_{i=1}^n c_i x_i}{\sum_{i=1}^n x_i} - \frac{\sum_{i,j=1}^n a_{ij} x_i x_j}{(\sum_{i=1}^n x_i)^2}\Big] \tag{6}$$

It is also obvious that the mRMR measure is an instance of the GeFS measure. In [12], this measure was denoted by $GeFS_{mRMR}$.

Both the $GeFS_{CFS}$ and the $GeFS_{mRMR}$ feature-selection problems are solved by means of the technique that involves the Polynomial Mixed 0-1 Fractional Programming ($PM01FP$). The details are given below.

### 2.2.4 Polynomial Mixed 0-1 Fractional Programming

A general polynomial mixed $0-1$ fractional programming ($PM01FP$) problem [5] is represented as follows:

$$\min \sum_{i=1}^m \Big(\frac{a_i + \sum_{j=1}^n a_{ij} \prod_{k \in J} x_k}{b_i + \sum_{j=1}^n b_{ij} \prod_{k \in J} x_k}\Big) \tag{7}$$

$$\textit{such that} \begin{cases} b_i + \sum_{j=1}^n b_{ij} \prod_{k \in J} x_k > 0, i = 1, .., m, \\ c_p + \sum_{j=1}^n c_{pj} \prod_{k \in J} x_k \leq 0, p = 1, .., m, \\ x_k \in \{0, 1\}, k \in J, \\ a_i, b_i, c_p, a_{ij}, b_{ij}, c_{pj} \in \Re. \end{cases}$$

By replacing the denominators in (7) by positive variables $y_i (i = 1, .., m)$, the $PM01FP$ then leads to the following equivalent polynomial mixed $0-1$ programming problem:

$$\min \sum_{i=1}^m \Big(a_i y_i + \sum_{j=1}^n a_{ij} \prod_{k \in J} x_k y_i\Big) \tag{8}$$

$$\textit{such that} \begin{cases} b_i y_i + \sum_{j=1}^n b_{ij} \prod_{k \in J} x_k y_i = 1; y_i > 0, \\ c_p + \sum_{j=1}^n c_{pj} \prod_{k \in J} x_k \leq 0, p = 1, .., m, \\ x_k \in \{0, 1\}, k \in J, \\ a_i, b_i, c_p, a_{ij}, b_{ij}, c_{pj} \in \Re. \end{cases} \tag{9}$$

In order to solve this problem, Chang [5] proposed a linearization technique to transfer the terms $\prod_{k\in J} x_k y_i$ into a set of mixed $0-1$ linear inequalities. Based on this technique, the $PM01FP$ becomes then a mixed $0-1$ linear programming ($M01LP$), which can be solved by means of the branch-and-bound method to obtain the globally optimal solution.

**Proposition 1:** A polynomial mixed $0 - 1$ term $\prod_{k\in J} x_k y_i$ from (8) can be represented by the following program [5], where $M$ is a large positive value:

$$\min z_i$$

$$\textit{such that} \begin{cases} z_i \geq 0, \\ z_i \geq M(\sum_{k\in J} x_k - |J|) + y_i \end{cases} \tag{10}$$

**Proposition 2:** A polynomial mixed $0-1$ term $\prod_{k\in J} x_k y_i$ from (9) can be represented by a continuous variable $v_i$, subject to the following linear inequalities [5], where $M$ is a large positive value:

$$\begin{cases} v_i \geq M(\sum_{k\in J} x_k - |J|) + y_i, \\ v_i \leq M(|J| - \sum_{k\in J} x_k) + y_i, \\ 0 \leq v_i \leq M x_i, \end{cases} \tag{11}$$

The feature selection problem (2) is formulated as a polynomial mixed $0 - 1$ fractional programming ($PM01FP$) problem as follows:

**Proposition 3:** The feature selection problem (2) is a polynomial mixed $0 - 1$ fractional programming ($PM01FP$) problem.

**Remark:** By applying Chang's method [5], this $PM01FP$ problem can be transformed into an $M01LP$ problem. The number of variables and constraints is quadratic in the number $n$ of full set features. This is because the number of terms $x_i x_j$ in (2), which are replaced by the new variables, is $n(n + 1)/2$. The branch-and-bound algorithm can then be used to solve this $M01LP$ problem. But the efficiency of the method depends strongly on the number of variables and constraints. The larger the number of variables and constraints an $M01LP$ problem has, the more complicated the branch-and-bound algorithm is.

In [11,12], an improvement of the Chang's method was proposed in order to get an $M01LP$ problem in which the number of variables and constraints is linear in the number $n$ of full set features. Details of the improvement are given below:

### 2.2.5 Optimization of the GeFS Measure

By introducing an additional positive variable, denoted by $y$, the following problem equivalent to (2) is considered:

$$\min_{x\in\{0,1\}^n} (-GeFS(x)) = -a_0 y - \sum_{i=1}^{n} A_i(x) x_i y \tag{12}$$

$$\textit{such that} \begin{cases} y > 0, \\ b_0 y + \sum_{i=1}^{n} B_i(x) x_i y = 1 \end{cases} \tag{13}$$

This problem is transformed into a mixed 0-1 linearning programming problem as follows:

**Proposition 4:** A term $A_i(x)x_iy$ from (12) can be represented by the following program, where $M$ is a large positive value [12]:

$$\min z_i$$

$$\textit{such that} \begin{cases} z_i \geq 0, \\ z_i \geq M(x_i - 1) + A_i(x)y, \end{cases} \tag{14}$$

**Proposition 5:** A term $B_i(x)x_iy$ from (13) can be represented by a continuous variable $v_i$, subject to the following linear inequality constraints, where $M$ is a large positive value [12]:

$$\begin{cases} v_i \geq M(x_i - 1) + B_i(x)y, \\ v_i \leq M(1 - x_i) + A_i(x)y, \\ 0 \leq v_i \leq Mx_i \end{cases} \tag{15}$$

Each term $x_iy$ in (14), (15) is substituted by new variable $t_i$ satisfying constraints from Proposition 2. Then the total number of variables for the $M01LP$ problem will be $4n + 1$, as they are $x_i$, $y$, $t_i$, $z_i$ and $v_i(i = \overline{1, n})$. Therefore, the number of constraints on these variables will also be a linear function of $n$. As we mentioned above, with Chang's method [5] the number of variables and constraints depends on the square of $n$. Thus the method [11,12] actually improves Chang's method by reducing the complexity of the branch and bound algorithm.

# 3 Experimental Results

## 3.1 Experimental Setting

For comparison of the generic-feature-selection ($GeFS$) measure for intrusion detection [11,12] with the previously known ones [13,14], we implemented the $GeFS_{CFS}$ and the $GeFS_{mRMR}$ algorithms. The goal was to find globally optimal feature subsets by means of these two measures. Since different intrusion detection systems used different feature-selection methods and different classifiers with the aim of achieving the best classification results, we compared general performance of intrusion detection systems in terms of numbers of selected features and the classification accuracies of the machine learning algorithms giving the best classification results. For our experiment, we used the decision tree algorithm C4.5 [8] as classifier for the full-set data as well as for the data sets obtained by removing irrelevant features by means of the $GeFS_{CFS}$ and $GeFS_{mRMR}$ measures.

We performed our experiment using 10% of the overall (5 millions of instances) KDD Cup'99 data set [7], since all the existing approaches involved in the comparison used the same data set for evaluation [13,14]. This data set contains normal traffic (Normal) and four attack classes: Denial-of-Service (DoS),

**Table 4.** The partition of KDD CUP'99 data set used in the experiment

| Classes | Number-of-instances | Percentage |
|---|---:|---:|
| Normal | 97.278 | 18.35% |
| DoS | 391.458 | 73.88% |
| Probe | 41.113 | 7.77% |
| Total | 529.849 | 100% |

Probe, User-to-Root (U2R) and Remote-to-Local (R2L) attacks. As the two attack classes U2R and R2L have been criticized [15,16], we did not consider them for our experiment. Details of numbers of class instances are given in Table 4.

As the attack classes distribute so differently, the feature selection algorithm might concentrate only on the most frequent class data and neglect the others. Therefore, we chose to process these attack classes separately. In order to do that, we added normal traffic into each attack class to get two data sets: Normal&DoS and Normal&Probe. With each data set, we ran two feature-selection algorithms: the $GeFS_{CFS}$ and the $GeFS_{mRMR}$. The number of selected features is given in Fig.3. We then applied the C4.5 machine learning algorithm on each original full-set as well as each newly obtained data set that includes only those selected features from the feature-selection algorithms. We applied 5-fold cross-validation on each data set. The classification accuracies are given in Fig.4.

The $GeFS_{CFS}$ and the $GeFS_{mRMR}$ feature-selection methods were compared with the existing ones (the SVM-wrapper, the Markov-Blanket and the CART) regarding the number of selected features and regarding the classification accuracies of machine learning algorithms chosen as classifiers for intrusion detection process. Weka tool [3] that implements the machine learning algorithms (C4.5, SVM and BayesNet) was used for obtaining the results. In order to solve the $M01LP$ problem, we used TOMLAB tool [6]. All the obtained results are shown in Fig.3 and Fig.4.

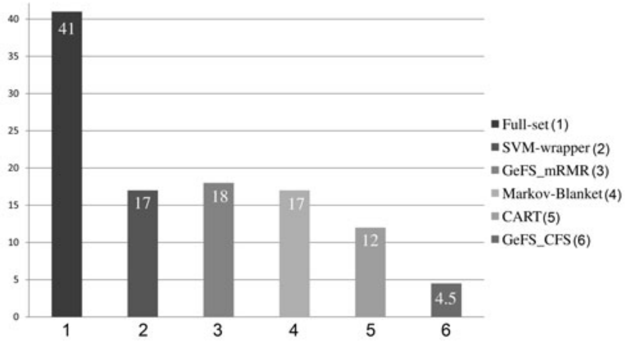

**Fig. 3.** Number of selected features (on average)

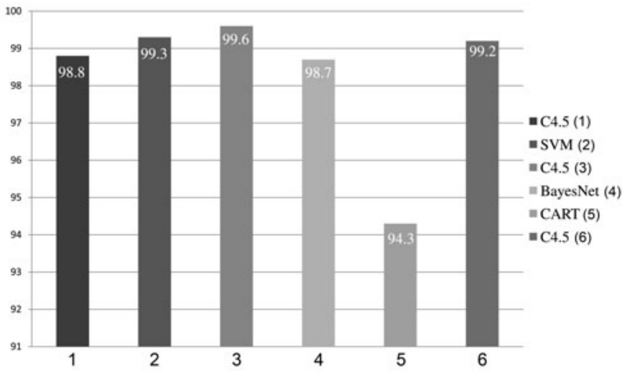

**Fig. 4.** Classification accuracies (on average)

### 3.2 Experimental Results

Fig.3 shows the average number of features selected by the *GeFS* feature-selection method and those selected by existing approaches. Fig.4 summarizes the average classification accuracies of chosen machine learning algorithms as classifiers for intrusion detection process. It can be observed from Fig.3 that the $GeFS_{CFS}$ feature-selection method selects the smallest number of relevant features. Fig.4 shows that with the approach from [11,12] the average classification accuracies are approximately the same or even better than those achieved by applying other methods.

## 4 Conclusions

In this paper, we compared, regarding the number of selected features and the classification accuracy, some previously known feature selection methods applicable for intrusion detection purposes with the feature selection methods for intrusion detection proposed in [11,12]. The previously known feature-selection algorithms involved in this comparison were the SVM-wrapper, Markov-blanket and CART algorithms. The feature selection algorithms proposed in [11,12] included in this comparison are instances of a generic-feature-selection (*GeFS*) method for intrusion detection: the correlation-feature-selection ($GeFS_{CFS}$) and the minimal-redundancy-maximal-relevance ($GeFS_{mRMR}$). Experimental results obtained over the KDD CUP'99 data set show that the *GeFS* method outperforms the previously known approaches by removing more than 30% of redundant features from the original data set, while keeping or yielding an even better classification accuracy. In spite of all the known limitations of the KDD CUP'99 data set used for comparison and the difficulties in establishing a more general theoretical basis for the comparison, there is a high probability that comparison results similar to ours could be obtained on other data sets as well.

## References

1. Hall, M.: Correlation Based Feature Selection for Machine Learning. In: Doctoral dissertation. Department of Computer Science, University of Waikato (1999)
2. Peng, H., Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1226–1238 (2005)
3. Weka, the Data Mining Software in Java, `http://www.cs.waikato.ac.nz/ml/weka/`
4. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature Extraction: Foundations and Applications. Studies in Fuzziness and Soft Computing. Springer, Heidelberg (2006)
5. Chang, C.T.: On the Polynomial Mixed 0-1 Fractional Programming Problems. European Journal of Operational Research 131, 224–227 (2001)
6. TOMLAB, The Optimization Environment in MATLAB, `http://tomopt.com/`
7. KDD Cup 1999 Data Set (1999),
   `http://www.sigkdd.org/kddcup/index.php?section=1999&method=data`
8. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
9. Gu, G., Fogla, P., Dagon, D., Lee, W., Skoric, B.: Towards an Information-Theoretic Framework for Analyzing Intrusion Detection Systems. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) ESORICS 2006. LNCS, vol. 4189, pp. 527–546. Springer, Heidelberg (2006)
10. Crescenzo, G.D., Ghosh, A., Talpade, R.: Towards a Theory of Intrusion Detection. In: Capitani, S., Syverson, P., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 267–286. Springer, Heidelberg (2005)
11. Nguyen, H., Franke, K., Petrović, S.: Improving Effectiveness of Intrusion Detection by Correlation Feature Selection. In: International Conference on Availability, Reliability and Security (ARES), pp. 17–24. IEEE Press, New York (2010)
12. Nguyen, H., Franke, K., Petrović, S.: Optimizing a Class of Feature Selection Measures. In: NIPS 2009 Workshop on Discrete Optimization in Machine Learning: Submodularity, Sparsity & Polyhedra (DISCML), Vancouver, Canada (2009)
13. Sung, A.H., Mukkamala, S.: Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks. In: International Symposium on Applications and the Internet (SAINT), pp. 209–217. IEEE Press, Los Alamitos (2003)
14. Chebrolu, S., Abraham, A., Thomas, J.: Feature Deduction and Ensemble Design of Intrusion Detection Systems. Computers & Security 4, 295–307 (2005)
15. McHugh, J.: Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. ACM TISSEC 3, 262–294 (2000)
16. Sabhnani, M., Serpen, G.: Why Machine Learning Algorithms Fail in Misuse Detection on KDD Intrusion Detection Data Set. Intelligent Data Analysis 8, 403–415 (2004)
17. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley& Sons, New York (2001)
18. Chen, Y., Li, Y., Cheng, X.Q., Guo, L.: Survey and Taxonomy of Feature Selection Algorithms in Intrusion Detection System. In: Lipmaa, H., Yung, M., Lin, D. (eds.) Inscrypt 2006. LNCS, vol. 4318, pp. 153–167. Springer, Heidelberg (2006)
19. Liu, H., Motoda, H.: Computational Methods of Feature Selection. Chapman & Hall/CRC, Boca Raton (2008)