Improving Effectiveness of Intrusion Detection by Correlation Feature Selection

Hai Thanh Nguyen, Katrin Franke and Slobodan Petrović
NISlab, Department of Computer Science and Media Technology,
Gjøvik University College, P.O. box 191, N-2802 Gjøvik, Norway
Email: {hai.nguyen, katrin.franke, slobodan.petrovic}@hig.no

Abstract—The quality of the feature selection algorithm is one of the most important factors that affects the effectiveness of an intrusion detection system (IDS). Achieving reduction of the number of relevant traffic features without negative effect on classification accuracy is a goal that greatly improves the overall effectiveness of the IDS. Obtaining a good feature set automatically without involving expert knowledge is a complex task. In this paper, we propose an automatic feature selection procedure based on the filter method used in machine learning. In particular, we focus on Correlation Feature Selection (CFS). By transforming the CFS optimization problem into a polynomial mixed 0−1 fractional programming problem and by introducing additional variables in the problem transformed in such a way, we obtain a new mixed 0−1 linear programming problem with a number of constraints and variables that is linear in the number of full set features. The mixed 0−1 linear programming problem can then be solved by means of branch-and-bound algorithm. Our feature selection algorithm was compared experimentally with the best-first-CFS and the genetic-algorithm-CFS methods regarding the feature selection capabilities. The classification accuracy obtained after the feature selection by means of the C4.5 and the BayesNet machines over the KDD CUP’99 IDS benchmarking data set was also tested. Experiments show that our proposed method outperforms the best first and genetic algorithm search strategies by removing much more redundant features and still keeping the classification accuracies or even getting better performances.

Keywords—intrusion detection; correlation-based feature selection; polynomial mixed 0−1 fractional programming; mixed 0−1 integer linear programming.

I. INTRODUCTION

Intrusion detection systems (IDS) have become important security tools applied in many contemporary network environments. They gather and analyze information from various sources on hosts and networks in order to identify suspicious activities and generate alerts for an operator. The task of intrusion detection is often analyzed as a pattern recognition problem—an IDS has to tell normal from abnormal behaviour. It is also 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 [14], [15]). A model of this kind usually includes the representation algorithm (for representing incoming data in the space of selected features) and the 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 models presented in [14], 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 of interest 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 remains therefore a great research challenge.

In this paper, we propose an automatic feature selection procedure for intrusion detection purposes based on so-called filter method [5], [6] used in machine learning. The filter method directly considers statistical characteristics of the data set, such as correlation between a feature and a class or inter-correlation between features, without involving any learning algorithm. We focus on one of the most important filter methods, the Correlation Feature Selection (CFS) measure proposed by M. Hall [1].

The CFS measure considers correlation between a feature and a class and inter-correlation between features at the same time. This measure is used successfully in test theory [3] for predicting an external variable of interest. In feature selection, the CFS measure is combined with some search strategies, such as brute force, best first search or genetic algorithm, in order to find the most relevant subset of features. However, the brute force method can only be applied when the number of features is small. When the number of features is large, this method requires huge
computational resources. For example, with 50 features the brute force method needs to scan all $2^{50}$ possible subsets of features. That is impractical in general. With best first search or genetic algorithm, we can deal with high dimensional data sets, but these methods usually give us locally optimal solutions. It is desirable to get globally optimal subset of relevant features by means of the CFS measure with the hope of removing more redundant features and still keeping classification accuracies or even getting better performances.

The feature selection method that we propose in this paper finds the globally optimal subset of relevant features by means of the CFS measure. Firstly, we formulate the problem of feature selection by representing the CFS measure as an optimization problem. We then transform this optimization problem into a polynomial mixed $0-1$ fractional programming ($P01FP$) problem. We improve the Chang’s method [8], [9] in order to equivalently reduce this $P01FP$ to a mixed $0-1$ linear programming ($M01LP$) problem [9]. Finally, we propose to use the branch-and-bound algorithm to solve this $M01LP$, whose optimal solution is also the globally optimal subset of relevant features by means of CFS measure.

Any feature selection algorithm selects relevant traffic features based on labelled data (Fig.1). In this research we used the KDD CUP’99 [11] data set for this purpose. The full feature set assigned to this data set consists of 41 features. For evaluating the performance of our feature selection approach, two available feature selection methods based on the CFS measure [16] were implemented. One was the best-first-CFS method using the best first search strategy to find the locally optimal subset of features by means of the CFS measure. The other one used the genetic algorithm. We called this feature selection approach as a genetic-algorithm-CFS method. To test the overall effectiveness of an IDS employing our feature selection algorithm, 10% of the overall (5 millions) KDD CUP’99 IDS benchmarking labelled data was also used to train and to test C4.5 [12] and BayesNet [13] machine learning algorithms with 5-folds cross-validation evaluation. Experiments show that an IDS implementing the best first and genetic algorithm search strategies in the feature selection process. Our method removes much more redundant features and the classification accuracies with the reduced feature set are kept at the same level or they become even better.

The paper is organized as follows. Section 2 describes the CFS measure in more detail. We show how to represent the problem of feature selection by means of the CFS measure as an optimization problem and then as a polynomial mixed $0-1$ fractional programming ($P01FP$) problem. The background regarding $P01FP$, $M01LP$ problems and Chang’s method are also introduced in this section. Section 3 describes our new approach. We present some experimental results in Section 4. The last section summarizes our findings.

II. BACKGROUND

A. 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”. This hypothesis gives rise to two concepts. One is the feature-classification ($r_{cf}$) correlation and another is the feature-feature ($r_{ff}$) correlation. There exist broadly two measures of the correlation between two random variables: the classical linear correlation and the correlation which is based on information theory (see Appendix A for correlation computation). The feature-classification correlation $r_{cf}$ indicates how much a feature $f_i$ is correlated to a target variable $C$, while the feature-feature correlation $r_{ff}$ is, as the very name says, the correlation between two features $f_i$, $f_j$. The following equation from [3] used in [1] gives the merit of a feature subset $S$ consisting of $k$ features:

$$\text{Merit}_S(k) = \frac{k r_{cf}}{\sqrt{k + k(k-1)r_{ff}}}.$$  \hspace{1cm} (1)

Here, $r_{cf}$ is the average feature-classification correlation, and $r_{ff}$ is the average feature-feature correlation, as given below:

$$r_{cf} = \frac{r_{cf_1} + r_{cf_2} + \ldots + r_{cf_k}}{k}$$

$$r_{ff} = \frac{r_{f_1 f_2} + r_{f_1 f_3} + \ldots + r_{f_k f_1}}{k(k-1)/2}$$

Therefore, we can rewrite (1) as follows:

$$\text{Merit}_S(k) = \frac{r_{cf_1} + r_{cf_2} + \ldots + r_{cf_k}}{\sqrt{k + 2(r_{f_1 f_2} + r_{f_1 f_3} + \ldots + r_{f_k f_1})}}$$  \hspace{1cm} (2)
In fact, the equation (1) is Pearson’s correlation coefficient, where all variables have been standardized. It shows that the correlation between the feature subset $S$ and the target variable $C$ is a function of the number $k$ of features in the subset $S$ and the magnitude of the inter-correlations among them, together with the magnitude of the correlations between the features and the target variable $C$. From the equation (1), the following conclusions can be drawn: The higher the correlations between the features and the target variable $C$, the higher the correlation between the feature subset $S$ and the target variable $C$; The lower the correlations between the features of the subset $S$, the higher the correlation between the feature subset $S$ and the target variable $C$.

The task of feature subset selection by means of the CFS measure: Suppose that there are $n$ full set features. We need to find the subset $S$ of $k$ features, which has the maximum value of $\text{Merit}_S(k)$ over all $2^n$ possible feature subsets:

$$\max_{S} \{ \text{Merit}_S(k), 1 \leq k \leq n \}. \quad (3)$$

When the number of features $n$ is small, we apply the brute force method to scan all these subsets. But when this number becomes large, the heuristic and random search strategies, such as the best first search or genetic algorithm, are usually chosen due to their computational efficiency. Consequently, the given results will always be approximate. It is desirable to get optimal subsets of features. In the sequel, we propose a new method to find these optimal subsets.

Firstly, we formulate the above mentioned task as an optimization problem. We use binary values of the variable $x_i$ in order to indicate the appearance ($x_i = 1$) or the absence ($x_i = 0$) of the feature $f_i$ in the optimal subset of features. Therefore, the problem of selecting features by means of the CFS measure can be described as an optimization problem as follows:

$$\max_{x=(x_1, ..., x_n)} F(x) = \frac{\sum_{i=1}^{n} r_{cf_i} x_i}{\sqrt{\sum_{i=1}^{n} x_i + \sum_{i \neq j} 2r_{f_i f_j} x_i x_j}}. \quad (4)$$

or in parameter form:

$$\max_{x=(x_1, ..., x_n)} F(x) = \frac{(\sum_{i=1}^{n} a_i x_i)^2}{\sum_{i=1}^{n} x_i + \sum_{i \neq j} b_{ij} x_i x_j}. \quad (5)$$

In the next subsection, we consider the optimization problem stated above as a polynomial mixed 0–1 fractional programming (P01FP) problem and show how to solve it.

B. Polynomial Mixed 0–1 Fractional Programming

A general polynomial mixed 0–1 fractional programming (P01FP) problem [8] is represented as follows:

$$\min_{i=1}^{m} \left( a_i + \sum_{j=1}^{n} a_{ij} \prod_{k \in J} x_k \right) \quad (6)$$

subject to the following constraints:

$$b_i + \sum_{j=1}^{n} b_{ij} \prod_{k \in J} x_k > 0, i = 1, ..., m,$$

$$c_p + \sum_{j=1}^{n} c_{pj} \prod_{k \in J} x_k \leq 0, p = 1, ..., m,$$

$$x_k \in \{0, 1\}, k \in J,$$

$$a_i, b_i, c_p, a_{ij}, b_{ij}, c_{pj} \in \mathbb{R}. \quad (8)$$

By replacing the denominators in (6) by positive variables $y_i(i = 1, ..., m)$, the P01FP then leads to the following equivalent polynomial mixed 0–1 programming problem:

$$\min_{i=1}^{m} \left( a_i y_i + \sum_{j=1}^{n} a_{ij} \prod_{k \in J} x_k y_j \right) \quad (7)$$

subject to the following constraints:

$$b_i y_i + \sum_{j=1}^{n} b_{ij} \prod_{k \in J} x_k y_j = 1; y_i > 0, i = 1, ..., m,$$

$$c_p + \sum_{j=1}^{n} c_{pj} \prod_{k \in J} x_k \leq 0, p = 1, ..., m,$$

$$x_k \in \{0, 1\}, k \in J,$$

$$a_i, b_i, c_p, a_{ij}, b_{ij}, c_{pj} \in \mathbb{R}. \quad (9)$$

In order to solve this problem, Chang [8] proposed a linearization technique to transfer the terms $\prod_{k \in J} x_k y_j$ into a set of mixed 0–1 linear inequalities. Based on this technique, the P01FP becomes then a mixed 0–1 linear programming (M01LP) which can be solved by means of the branch-and-bound method to obtain the global solution.

Proposition 1: A polynomial mixed 0–1 term $\prod_{k \in J} x_k y_j$ from (7) can be represented by the following program [9]:

$$\min z_i$$

subject to the following constraints:

$$z_i \geq M(\sum_{k \in J} x_k - |J|) + y_i, \quad (9)$$

$$z_i \geq 0,$$

where $M$ is a large positive value.

Proposition 2: A polynomial mixed 0–1 term $\prod_{k \in J} x_k y_j$ from (8) can be represented by a continuous
variable $v_i$, subject to the following linear inequalities [9]:

\[
\begin{align*}
    v_i &\geq M (\sum_{k \in J} x_k - |J|) + y_i, \\
    v_i &\leq M (|J| - \sum_{k \in J} x_k) + y_i, \\
    0 &\leq v_i \leq M x_i,
\end{align*}
\]

where $M$ is a large positive value.

In the following, we formulate the optimization problem of the CFS measure (5) as a polynomial mixed 0 – 1 fractional programming (P01FP) problem.

**Proposition 3:** The optimization problem of the CFS measure (5) can be considered as a polynomial mixed 0 – 1 fractional programming (P01FP) problem.

**Proof:** We change the sign of $F(x)$ in (5) to make a minimum problem and decompose the numerator of (5) as follows:

\[
\left(\sum_{i=1}^{n} a_i x_i\right)^2 = \sum_{i=1}^{n} a_i^2 x_i^2 + \sum_{i \neq j} 2a_i a_j x_i x_j.
\]

Therefore, (5) can be written as (6). ■

**Remark:** By applying the Chang’s method, we can transform this P01FP problem to the $M01LP$ problem. The number of variables and constraints will depend on the square of $n$, where $n$ is the number of features. The reason is the number of terms $\prod_{k \in J} x_k y$, which are replaced by the new variables in forms ($\sum_{i \neq j} 2a_i a_j x_i x_j y$) or ($\sum_{i \neq j} b_{ij} x_i x_j y$), is $n(n-1)/2$. The branch-and-bound algorithm can then be used in order 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 $M01LP$ has, the more complicated the branch-and-bound algorithm is.

In the next section, we present an improvement of the Chang’s method to get an $M01LP$ with a linear number of variables and constraints in the number of full set variables. We also give a new search strategy to obtain the relevant subsets of features by means the CFS measure.

III. OPTIMIZATION OF THE CFS MEASURE

By introducing an additional positive variable, denoted by $y$, we now consider the following problem equivalent to (5):

\[
\min \{-F(x)\} = -\sum_{j=1}^{n} \left(\sum_{i=1}^{n} a_i a_j x_i x_j y\right).
\]

subject to the following constraints:

\[
\begin{align*}
    y &> 0, \\
    x &= (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n, \\
    \sum_{i=1}^{n} x_i y + \sum_{j=1}^{n} \left(\sum_{i=1, i \neq j}^{n} a_i a_j x_i x_j y\right) x_j y &= 1.
\end{align*}
\]

Here, all the terms $a_i a_j x_i x_j$ in the numerator and the terms $b_{ij} x_i x_j$ in the denominator of (5) have been grouped into the sum ($\sum_{j=1}^{n} \left(\sum_{i=1}^{n} a_i a_j x_i x_j y\right)$) and the sum ($\sum_{j=1}^{n} \left(\sum_{i=1, i \neq j}^{n} b_{ij} x_i x_j y\right)$), respectively. Each sum contains $n$ terms, which will be equivalently replaced by new variables with constraints following the two propositions given below:

**Proposition 4:** A polynomial mixed 0 – 1 term ($\sum_{i=1}^{n} a_i a_j x_i x_j y$) from (12) can be represented by the following program:

\[
\min z_j
\]

subject to the following constraints:

\[
\begin{align*}
    z_j &\geq M (x_j - 1) + \left(\sum_{i=1}^{n} a_i a_j x_i x_j y\right), \\
    z_j &\geq 0,
\end{align*}
\]

where $M$ is a large positive value.

**Proof.**

(a) If $x_j = 0$, then $z_j \geq M (0 - 1) + \left(\sum_{i=1}^{n} a_i a_j x_i x_j y\right) \leq 0$ will force min $z_j$ to be zero, because $z_j \geq 0$ and $M$ is a large positive value.

(b) If $x_j = 1$, then $z_j \geq M (1 - 1) + \left(\sum_{i=1}^{n} a_i a_j x_i x_j y\right) \geq 0$ will force min $z_j$ to be $\left(\sum_{i=1}^{n} a_i a_j x_i y\right)$, because $z_j \geq 0$.

Therefore, the above program on $z_j$ reduces to:

\[
\min z_j = \begin{cases} 
0, & \text{if } x_j = 0, \\
\left(\sum_{i=1}^{n} a_i a_j x_i y\right), & \text{if } x_j = 1.
\end{cases}
\]

which is the same as ($\sum_{i=1}^{n} a_i a_j x_i x_j y = \min z_j$). ■

**Proposition 5:** A polynomial mixed 0 – 1 term ($\sum_{i=1, i \neq j}^{n} b_{ij} x_i x_j y$) from (13) can be represented by a continuous variable $v_j$, subject to the following linear inequality constraints:
\[
\begin{cases}
    v_j \geq M(x_j - 1) + \left(\sum_{i=1, i \neq j}^{n} b_{ji}x_i\right)y, \\
    v_j \leq M(1 - x_j) + \left(\sum_{i=1, i \neq j}^{n} b_{ji}x_i\right)y, \\
    0 \leq v_j \leq Mx_j,
\end{cases}
\]

where \( M \) is a large positive value.

**Proof.**

(a) If \( x_j = 0 \), then (15) becomes

\[
\begin{cases}
    v_j \geq M(0 - 1) + \left(\sum_{i=1, i \neq j}^{n} b_{ji}x_i\right)y, \\
    v_j \leq M(1 - 0) + \left(\sum_{i=1, i \neq j}^{n} b_{ji}x_i\right)y, \\
    0 \leq v_j \leq 0,
\end{cases}
\]

\( v_j \) is forced to be zero, because \( M \) is a large positive value.

(b) If \( x_j = 1 \), then (15) becomes

\[
\begin{cases}
    v_j \geq M(1 - 1) + \left(\sum_{i=1, i \neq j}^{n} b_{ji}x_i\right)y, \\
    v_j \leq M(1 - 1) + \left(\sum_{i=1, i \neq j}^{n} b_{ji}x_i\right)y, \\
    0 \leq v_j \leq M,
\end{cases}
\]

\( v_j \) is forced to be \( \left(\sum_{i=1, i \neq j}^{n} b_{ji}x_i\right)y \), because \( M \) is a large positive value.

Therefore, the constraints on \( v_j \) reduce to:

\[
v_j = \begin{cases}
    0, & \text{if } x_j = 0, \\
    \left(\sum_{i=1, i \neq j}^{n} b_{ji}x_i\right)y, & \text{if } x_j = 1.
\end{cases}
\]

which is the same as \( \left(\sum_{i=1, i \neq j}^{n} b_{ji}x_i\right)x_jy = v_j \). □

We substitute each term \( x_iy \) in (13) by new variables \( t_i \) satisfying constraints from Proposition 2. Then the total number of variables for the M01LP problem will be \( 4n + 1 \), as they are \( x_i, y, t_i, z_j \) and \( v_j(i, j = 1, \ldots, n) \). Therefore, the number of constraints on these variables will also be a linear function of \( n \). As we mentioned above, with Chang’s method [8] the number of variables and constraints depends on the square of \( n \), thus our new method actually improves his method by reducing the complexity of branch and bound algorithm.

We now present a new search strategy for obtaining subsets of relevant features by means of the CFS measure.

The new search method for subsets of relevant features by means of the CFS measure:

- **Step 1:** Calculate all feature-feature \( (r_{f_i f_j}) \) and feature-classification \( (r_{f_i y}) \) correlations from the training data set.
- **Step 2:** Construct the optimization problem (4) from the correlations calculated above. In this step, we can use expert knowledge by assigning the value 1 to the variable \( x_i \) if the feature \( f_i \) is relevant and the value 0 otherwise.
- **Step 3:** Transform the optimization problem of CFS to a mixed 0–1 linear programming (M01LP) problem, which is to be solved by the branch-and-bound algorithm. A non-zero integer value of \( x_i \) from the optimal solution indicates the relevance of the feature \( f_i \) regarding the CFS measure.

## IV. Experiment

### A. Experimental Setting

For evaluating the performance of our new CFS-based approach, two available feature selection methods based on the CFS measure [16] are implemented. One is the best-first-CFS method, which uses the best first search strategy to find the locally optimal subset. The other one uses the genetic algorithm for search. Note that the best first search and genetic algorithm may not guarantee to find the globally optimal solution. However, we can overcome this issue with our new method. We did not choose the exhaustive search method since it is not feasible for feature selection from data sets with a large number of features. Even for this experiment we have no access to required computing resource. We applied machine learning algorithms for evaluating the classification accuracy on selected features, since there is no standard IDS.

We performed our experiment using 10% of the overall (5 millions) KDD CUP’99 IDS benchmarking labelled data [11]. This data set contains normal traffic and four main attack classes: (i) Denial of Service (DoS) attacks, (ii) Probe attacks, (iii) User to Root (U2R) attacks and (iv) Remote to Local (R2L) attacks. The number of instances for the four attack classes and normal class is quite different, e.g the relation of the number of U2R to the number of DoS is \( 1.3 * 10^{-4} \). Details of the number of class instances are given in Table I.

### Table I

<table>
<thead>
<tr>
<th>Classes</th>
<th>Number-of-instances</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDD99-normal</td>
<td>97,278</td>
<td>18.30%</td>
</tr>
<tr>
<td>KDD99-DoS</td>
<td>391,458</td>
<td>73.74%</td>
</tr>
<tr>
<td>KDD99-Probe</td>
<td>41,113</td>
<td>7.74%</td>
</tr>
<tr>
<td>KDD99-U2R</td>
<td>52</td>
<td>0.01%</td>
</tr>
<tr>
<td>KDD99-R2L</td>
<td>1,126</td>
<td>0.21%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>531,027</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>
In more detail we testified the performance of our newly proposed CFS-based feature selection method as follows:

1) Feature selection is performed on the basis of the whole data set: (1a) Each attack class and the normal class are processed individually, so that a five-class problem can be formulated for feature extraction and classification with one single classifier. (1b) All attack classes are fused so that a two-class problem can be formulated, meaning the feature selection and classification for normal and abnormal traffic is performed. It might be well possible that the attack-recognition results are not satisfactory for all of the classes, since the number of class instances are unevenly distributed, in particular classes U2R and R2L are under-represented. The feature selection algorithm and the classifier, which is used for evaluation of the detection accuracy on selected features, might concentrate only on the most frequent class data and neglect the others. As consequence, we might miss relevant characteristics of the less represented classes.

2) As the attack classes distribute so differently, we preferred to process these attack classes separately. With the specific application of IDS we can also formulate four different two-class problems. Four classifiers shall be derived using specific features for each classifier in order to detect (identify) a particular attack. The rationale for this approach is that we predict the most accurate classification if each of the four intrusion detectors (classifiers) is fine-tuned according to significant features. This approach might also be very effective, since the four light-weight classifiers can be operated in parallel.

For understanding the effect, as mentioned in 1), we conducted a small experiment. The aim was to show that the classifier highly neglected U2R attack instances. In order to do that, we mixed all attack classes to get only one data set and considered five-class (normal, DoS, Probe, U2R and R2L) problem. The C4.5 machine learning algorithm was used as a classifier. We applied 5-folds cross-validation for evaluating the detection accuracy of the C4.5. The result of the experiment is given in Table II. It can be seen from Table II that the C4.5 highly misclassified U2R attack instances with 34.6% error.

In order to perform the experiment 2), we added normal traffic into each attack class to get four data sets: KDD99-normal&DoS, KDD99-normal&Probe, KDD99-normal&U2R and KDD99-normal&R2L. With each data set, we ran three feature selection algorithms: our new CFS-based method, the best-first CFS-based and the genetic algorithm CFS-based methods. The numbers of selected features and their identifications are given in Tables IV and V, respectively. We then applied the C4.5 and the BayesNet machine learning algorithm on each original full set as well as each newly obtained data set that includes only those selected features from feature selection algorithms. By applying 5-folds cross-validation evaluation on each data set, the classification accuracies are reported in Table V.

Our new CFS-based method was compared with the best-first-CFS and genetic-algorithm-CFS methods regarding the number of selected features and regarding the classification accuracies of 5-folds cross-validation of BayesNet and C4.5 learning algorithms. Weka tool [2] was used for obtaining the results. In order to solve the M01LP problem, we used TOMLAB tool [10]. All the obtained results are listed in Tables III, IV and V.

B. Experimental Results

Table III shows the number of features selected by our approach and those selected by using the best-first and GA search strategies. The identification of selected features is given in Table IV (for feature names, see Appendix B). Table V summarizes the classification accuracies of the BayesNet and the C4.5 performed on four data sets (see above).

It can be observed from Table III that our CFS-based approach selects the smallest number of relevant features in comparison with the full and the feature sets selected by the best-first and GA search strategies. Especially in some cases, our new method compresses the full set of features extremely. For example, only one feature was selected out of 41 features of the KDD99-normal&U2R data set.

In the Table V, it can be observed that with our approach the average classification accuracies are slightly different from the ones obtained by using the best-first search or the genetic algorithm. The absolute difference between them does not overcome 0.69%. In the case of the C4.5 classifier, we got better performance. Even though the gain of classification accuracy is not very high compared to other methods, the overall gain of the feature selection classification procedure lies in significantly improved
Table III

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Full-set</th>
<th>Our-method</th>
<th>Best-first</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDD99-normal&amp;Dos</td>
<td>41</td>
<td>3</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>KDD99-normal&amp;Probe</td>
<td>41</td>
<td>6</td>
<td>7</td>
<td>17</td>
</tr>
<tr>
<td>KDD99-normal&amp;U2R</td>
<td>41</td>
<td>1</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>KDD99-normal&amp;R2L</td>
<td>41</td>
<td>2</td>
<td>5</td>
<td>8</td>
</tr>
</tbody>
</table>

Table IV

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Identifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDD99-normal&amp;Dos</td>
<td>5, 6, 12</td>
</tr>
<tr>
<td>KDD99-normal&amp;Probe</td>
<td>5, 6, 12, 29, 37, 41</td>
</tr>
<tr>
<td>KDD99-normal&amp;U2R</td>
<td>14</td>
</tr>
<tr>
<td>KDD99-normal&amp;R2L</td>
<td>10, 22</td>
</tr>
</tbody>
</table>

Table V

<table>
<thead>
<tr>
<th>Data Set</th>
<th>C4.5</th>
<th>BayesNet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Full-Set</td>
<td>Our-method</td>
</tr>
<tr>
<td>KDD99-normal&amp;Dos</td>
<td>97.80</td>
<td>98.89</td>
</tr>
<tr>
<td>KDD99-normal&amp;Probe</td>
<td>99.98</td>
<td>99.70</td>
</tr>
<tr>
<td>KDD99-normal&amp;U2R</td>
<td>99.97</td>
<td>99.96</td>
</tr>
<tr>
<td>KDD99-normal&amp;R2L</td>
<td>98.70</td>
<td>99.11</td>
</tr>
<tr>
<td>Average</td>
<td>99.11</td>
<td>99.41</td>
</tr>
</tbody>
</table>

V. Conclusions

We have proposed a new search method to get the globally optimal subset of relevant features by means of the correlation feature selection (CFS) measure. Actually we transformed the CFS optimization problem into polynomial mixed 0 − 1 fractional programming (P01FP) problem. From this P01FP problem, we then applied our improved Chang’s method to get mixed 0 − 1 linear programming (M01LP) problem with linear dependence of the number of constraints and variables on the number of features in the full set. We used branch-and-bound algorithm in order to solve that M01LP. Experimental results showed that our approach outperforms the best-first-CFS and genetic-algorithm-CFS methods by removing much more redundant features and still keeping the classification accuracies or even getting better performances.

Appendix A.

Correlation Computation

1) For continuous class problems: For a pair of random variables \((X, Y)\), the linear correlation coefficient \(\rho(X, Y)\) is given by the formula:

\[
\rho(X, Y) = \frac{E((X - \mu_X)(Y - \mu_Y))}{\sigma_X \sigma_Y}
\]

where \(\mu_X, \mu_Y\) are expected values of variables \(X\) and \(Y\), respectively; \(\sigma_X, \sigma_Y\) are standard deviations; \(E\) is the expected value operator.

2) For discrete class problems: For a pair of random variables \((X, Y)\), the correlation is defined as symmetrical uncertainty \(SU(X, Y)\) coefficient by the formula:

\[
SU(X, Y) = 2\left(\frac{H(X) - H(X|Y)}{H(X) + H(Y)}\right)
\]
where $H(X)$, $H(Y)$ are the entropy of variables $X$ and $Y$, respectively; $H(X|Y)$ is the conditional entropy.

**Appendix B.**

**Names and Identifications (ID) of Selected Features**

Here we want to keep the order of features from the original KDD CUP’99 data set.

<table>
<thead>
<tr>
<th>ID</th>
<th>Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>duration</td>
</tr>
<tr>
<td>3</td>
<td>service</td>
</tr>
<tr>
<td>5</td>
<td>src_bytes</td>
</tr>
<tr>
<td>6</td>
<td>dst_bytes</td>
</tr>
<tr>
<td>9</td>
<td>urgent</td>
</tr>
<tr>
<td>10</td>
<td>hot</td>
</tr>
<tr>
<td>11</td>
<td>num_failed_logins</td>
</tr>
<tr>
<td>12</td>
<td>logged_in</td>
</tr>
<tr>
<td>14</td>
<td>root_shell</td>
</tr>
<tr>
<td>16</td>
<td>num_root</td>
</tr>
<tr>
<td>17</td>
<td>num_file_creations</td>
</tr>
<tr>
<td>22</td>
<td>num_comp_login</td>
</tr>
<tr>
<td>26</td>
<td>srv_serror_rate</td>
</tr>
<tr>
<td>27</td>
<td>srv_rerror_rate</td>
</tr>
<tr>
<td>29</td>
<td>same_srv_rate</td>
</tr>
<tr>
<td>31</td>
<td>srv_diff_host_rate</td>
</tr>
<tr>
<td>32</td>
<td>dst_host_count</td>
</tr>
<tr>
<td>33</td>
<td>dst_host_srv_count</td>
</tr>
<tr>
<td>37</td>
<td>dst_host_srrv_diff_host_rate</td>
</tr>
<tr>
<td>39</td>
<td>dst_host_srv_serror_rate</td>
</tr>
<tr>
<td>41</td>
<td>dst_host_srv_rerror_rate</td>
</tr>
</tbody>
</table>

**References**


