A New Dependency and Correlation Analysis for Features

Guangzhi Qu, *Student Member*, *IEEE*, Salim Hariri, *Senior Member*, *IEEE*, and Mazin Yousif, *Senior Member*, *IEEE*

Abstract—The quality of the data being analyzed is a critical factor that affects the accuracy of data mining algorithms. There are two important aspects of the data quality, one is relevance and the other is data redundancy. The inclusion of irrelevant and redundant features in the data mining model results in poor predictions and high computational overhead. This paper presents an efficient method concerning both the relevance of the features and the pairwise features correlation in order to improve the prediction and accuracy of our data mining algorithm. We introduce a new feature correlation metric $Q_Y(X_i, X_j)$ and feature subset merit measure $e(S)$ to quantify the relevance and the correlation among features with respect to a desired data mining task (e.g., detection of an abnormal behavior in a network service due to network attacks). Our approach takes into consideration not only the dependency among the features, but also their dependency with respect to a given data mining task. Our analysis shows that the correlation relationship among features depends on the decision task and, thus, they display different behaviors as we change the decision task. We applied our data mining approach to network security and validated it using the DARPA KDD99 benchmark data set. Our results show that, using the new decision dependent correlation metric, we can efficiently detect rare network attacks such as User to Root (U2R) and Remote to Local (R2L) attacks. The best reported detection rates for U2R and R2L on the KDD99 data sets were 13.2 percent and 8.4 percent with 0.5 percent false alarm, respectively. For U2R attacks, our approach can achieve a 92.5 percent detection rate with a false alarm of 0.7587 percent. For R2L attacks, our approach can achieve a 92.47 percent detection rate with a false alarm of 8.35 percent.

Index Terms—Feature extraction, correlation measure.

1 Introduction and Background

Feature extraction in knowledge and date engineering is the process of identifying and removing as much of the irrelevant and redundant information as possible. Regardless of whether a machine learning algorithm attempts to select features itself or ignores the issue, feature extraction prior to learning can be beneficial. Reducing the dimensionality of the data reduces the size of the hypothesis space and allows algorithms to operate faster. In some cases, accuracy on future classification can be improved; in others, the result needs to be more compact and can be interpreted more easily.

Dash and Liu [6], Blum and Langley [4], and Hall and Holmes [8] presented a survey of the research on machine learning for feature extraction. In essence, many feature extraction methods model the task as a search problem, where each state in the search space specifies a distinct subset of the possible features. Dash and Liu categorize feature extraction into two major steps: generation procedure and evaluation function [6]. In the generation procedure, *complete*, *heuristic*, and *random* are different approaches for space searching. The searching space is exponential in the number of features. Hence, it is necessary to use a heuristic search procedure for an even medium number of features. Another important step in the feature selection is the evaluation function, which serves as the criterion in evaluating the relative merit of alternative feature subsets. Dash and Liu divide the evaluation function into five categories: *distance*, *information*, *dependency*, *consistency*, and *classifier error rate*. Hall and Holmes divide the feature extraction methods into two categories; one is based on the evaluation of individual feature, the other is based on evaluation of feature subsets. *Information gain attribute ranking* and *Relief/ReliefF* [10], [12] belong to the first category. *Correlation-based Feature Selection (CFS)*, *Consistency-based Feature Selection*, and *Wrapper Subset Selection* fall into the second category.

In this paper, we present a new approach that effectively removes irrelevant features from the ranked feature list based on the mutual information between each feature and the decision variable. We obtain the ranked lists of features by using a simple forward selection hill climbing search, starting with an empty set and evaluating each feature individually and forcing it to continue to the far side of the search space. Redundant features are removed through the pairwise decision dependent correlation analysis. The evaluation process of subset features is done in the abridged ranked lists of features after reducing irrelevant features.

The next section briefly reviews feature extraction algorithms. In Section 3, we present the new decision dependent correlation analysis measure and how to use it to quantify the dependency among features with respect to a particular decision task. Section 4 outlines the experimental methodology that was used to validate our approach.

- *G. Qu and S. Hariri are with the Electrical and Computer Engineering Department, The University of Arizona, PO Box 210104, 1230 E. Speedway Boulevard, Tucson, AZ 85721-0104. E-mail: {qug, hariri}@ece.arizona.edu.*
- *M. Yousif is with the Corporate Technology Lab, Intel Corporation, JF5-65, 2111 NE 25th Avenue, Hillsboro, OR 97124. E-mail: mazin.s.yousif@intel.com.*

Manuscript received 22 Nov. 2004; revised 23 Mar. 2005; accepted 30 Mar. 2005; published online 19 July 2005.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDESI-0478-1104.

Feature irrelevance and redundancy removal algorithm is also presented in this section. Section 5 presents the experimental results and compares our results with other methods. The last section summarizes our future research direction.

2 Feature Extraction Techniques

Feature extraction techniques can be categorized according to a number of criteria. One popular categorization consists of "filter" and "wrapper" to quantify the worth of features [5], [11]. Filters use general characteristics of the training data to evaluate attributes and operate independently of any learning algorithm. Wrappers, on the other hand, evaluate attributes by using accuracy estimates provided by the actual target learning algorithm. Due to the fact that the wrapper model is computationally expensive [13], the filter model is usually a good choice when the number of features becomes very large.

Das [5] combined both models into a hybrid one to improve the performance of a particular learning algorithm. In this paper, we focus on the filter model and present a novel feature extraction algorithm which can effectively remove both irrelevant and redundant information.

Evaluation of individual feature emphasizes the relevance of the feature to the final decision. There are two typical individual feature-based evaluation approaches. The first one is information-based feature ranking. In this approach, the mutual information between decision and feature is used to evaluate the importance of the feature with respect to the decision under consideration. This method is independent of the underlying distribution and especially efficient when the data sets have a sheer dimensionality. The second type of algorithms relies on the relevance evaluation of features such as *Relief* which is an instance-based feature ranking scheme introduced by Kira and Rendell [10], and *ReliefF*, which can handle multiple class data, is enhanced by Kononenko [12] from *Relief*. The rationale of *Relief* and *ReliefF* is that a useful feature should differentiate between instances from different classes and have the same value for instances from the same class. The *Relief* approach is based on randomly sampling a number (m) of instances from the training data set and then locating each feature's nearest neighbor from the same and opposite class. The values of the features of the nearest neighbors are compared to the sampled instance and used to update relevant scores for each feature.

Although feature extraction techniques that focus only on relevance can significantly reduce the number of features to be considered, it could not help remove the redundant information existing among multiple features. Hall [7] and Kohavi and John [11] show that redundant features, along with irrelevant features, severely affect the accuracy of the learning algorithms. The reason is that if we do not consider the dependency among features, the feature selection algorithm will select multiple highly correlated features. Our results show that the linear summation of the individual mutual information values with respect to a particular decision will not linearly decrease the uncertainty in the decision because of the dependency that exists between features.

Subset searching algorithms search through candidate feature subsets guided by a certain evaluation measure which captures the goodness of each subset. Some evaluation measures that have been effective in removing both irrelevance and redundancy include consistency measure [5], [2], [14] and correlation measure [7], [8]. The consistency method looks for the minimum combinations of features that could divide the training data into subsets containing a strong single class majority. This separation is hoped to be as consistent as the whole set of features. Correlation-based feature selection evaluates subsets of features rather than individual features. The ideal subsets should contain features that are highly correlated with the decision and have low level intercorrelation with each other.

3 Correlation Analysis Measure

It has been shown that dependency measure or correlation measures qualify the accuracy of decision to predict the value of one variable [6]. The main shortcomings of classical linear correlations are the assumption of linear correlation between the features and the requirement that all features contain numerical values [19]. To overcome these shortcomings, several information theory-based measures of association were introduced for the feature-class correlations and feature intercorrelations, such as the gain ratio [18] and information gain [1], the symmetrical uncertainty coefficient [17], and several others based on the minimum description length principle [12]. Good results were acquired through using the gain ratio for feature-class correlations and symmetrical uncertainty for feature intercorrelations [3], [7], [19], [20].

However, the symmetrical uncertainty measure is not accurate enough to quantify the dependency among features with respect to a given decision. A critical point was neglected that the correlation or redundancy between features is strongly related with the decision variable under consideration. In what follows, we will explain this property using a simple example.

For example, let us consider the case where the decision set consists of two decision variables, H and H^c, that denote two complementary decisions, with $P(H) = 0.6$. Suppose now that if H occurs, then both X and Y are likely to occur; that is, $P(X|H) = 0.8$, $P(Y|H) = 0.9$, and their independence requiring that $P(X \cap Y|H) = 0.8 \cdot 0.9 = 0.72$. On the other hand, if H^c occurs, then both X and Y are unlikely, say, $P(X|H^c) = 0.2, P(Y|H^c) = 0.1$. Again, independency requires that $P(X \cap Y|H^c) = 0.2 \cdot 0.1 = 0.02$. It is easy to check that the X and Y are dependent. Indeed,

$$\begin{aligned} P(X) &= P(X|H)P(H) + P(X|H^c)P(H^c) \\ &= 0.8 \cdot 0.6 + 0.2 \cdot 0.4 = 0.56, \end{aligned}$$

$$\begin{aligned} P(Y) &= P(Y|H)P(H) + P(Y|H^c)P(H^c) \\ &= 0.9 \cdot 0.6 + 0.1 \cdot 0.4 = 0.58. \end{aligned}$$

On the other hand,

$$\begin{aligned} P(X \cap Y) &= P(X \cap Y|H)P(H) + P(X \cap Y|H^c)P(H^c) \\ &= 0.72 \cdot 0.6 + 0.02 \cdot 0.4 = 0.44, \end{aligned}$$

which is not equal to $P(X)P(Y) = 0.56 \cdot 0.58 = 0.3248$.

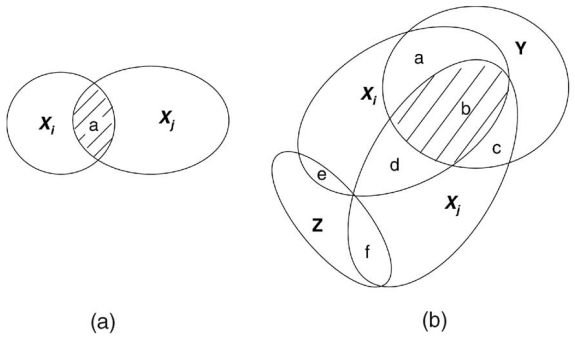


Fig. 1. Illustration of feature correlations with respect to multiple decisions in Venn Diagrams. (a) The decision independent correlation between two features X_i and X_j. (b) The decision dependent correlation for two features X_i and X_j with respect to two decision variables Y and Z.

The above example demonstrates that the correlated two events could be independent with respect to decisions. On the other hand, it demonstrates that the symmetric uncertainty may provide false or incomplete information. Hence, to accurately quantify the dependency or correlation among features, we introduce the following new metric.

Definition 3.1. *Let X_i and X_j be two features. When there is no decision being considered with the features, we say the correlation between them is* decision independent correlation *(DIC). By using information theory, DIC is defined as the ratio between the mutual information and the uncertainty of the feature.*

$$DIC_{X_j}(X_i, X_j) = \frac{I(X_i; X_j)}{H(X_j)}, \tag{1}$$

$$DIC_{X_i}(X_i, X_j) = \frac{I(X_i; X_j)}{H(X_i)}. \tag{2}$$

Remark 3.1. $0 \le DIC(X_i, X_j) \le 1$ can be intuitively acquired. When $DIC(X_i, X_j) = 0$, features X_i and X_j are uncorrelated. Scenario $DIC(X_i, X_j) = 1$ implies full prediction between the features.

Remark 3.2. This correlation measure can also be used to find the correlation between one feature and one class.

Definition 3.2. *Let X_i and X_j be two features. When there is a decision (Y) associated with the features, we say the correlation between them is* decision dependent correlation *(DDC). Let (Ω, Γ, P) be an arbitrary probability space and let $X_i : (\Omega, \Gamma) \to (\Omega, \Gamma_i)$ for $i \in I_n = \{1, 2, \ldots, n\}$ be n random features. We define a correlation measure to quantify the information redundancy between X_i and X_j with respect to Y as follows:*

$$Q_Y(X_i, X_j) = \frac{I(Y; X_i) + I(Y; X_j) - I(Y; X_i, X_j)}{H(Y)}. \tag{3}$$

Remark 3.3. DDC is symmetric that $Q_Y(X_i, X_j) = Q_Y(X_j, X_i)$.

Proof. By noticing the symmetric property of the mutual information. □

We use the Venn diagrams in Fig. 1 to illustrate the idea of decision independent correlation (DIC) and decision dependent correlation (DDC). In Fig. 1a, the correlation between X_i and X_j could be quantified as the ratio between the shade area (Fig. 1a) and each individual area of X_i or X_j, which is generally the percentage of information with respect to its uncertainty acquired by knowing the other variable. This measure is a good reference for feature-class intercorrelation. In Fig. 1b, Y and Z are decision variables. X_i and X_j are features. By using Shannon's theoretic mutual information measure, we get $I(Y; X_i) = a + b$ and $I(Y; X_j) = b + c$. The mutual information between the decision Y and features X_i and X_j is $I(Y; X_i \cap X_j) = a + b + c$, which is obviously less than $I(Y;\ X_i) + I(Y; X_j)(a + 2b + c)$. Consequently, choosing both features X_i and X_j may include some redundancy $(b/H(Y)$ amount) which could be quantified by the DDC measure $Q_Y(X_i, X_j)$. If we did not use the new correlation measure, the symmetric uncertainty will zoom in the correlation between features. Another important fact not accounted for is that for decision Y, features X_i and X_j are highly correlated, but they are not correlated when we consider another decision variable Z, as shown in Fig. 1b. This fact cannot be captured by using symmetric uncertainty, which is constant and independent of the decision variable when given the two features. Our experimental results show that using the decision dependent correlation measure $Q_Y(X_i, X_j)$ in subset feature selection will significantly improve the accuracy of the decision variables.

Theorem 3.1. *Let (Ω, Γ, P), X_i, and $Q_Y(X_i, X_j)$ as above. Then, $Q_Y(X_i, X_j) \ge 0$ with equality if and only if X_i, X_j are uncorrelated with respect to decision Y.*

Proof. The uncertainty for the decision variable Y is always positive, which means $H(Y) > 0$. By definition, the mutual information between decision variable Y and features X_i and X_j is given by

$$\begin{aligned} I(Y; X_i, X_j) &= I(Y; X_i) + I(Y; X_j | X_i) \\ &= I(Y; X_j) + I(Y; X_i | X_j). \end{aligned} \tag{4}$$

Using conditional mutual information, we can write the following inequality:

$$I(Y; X_j | X_i) \le I(Y; X_j). \tag{5}$$

Now,

$$\begin{aligned} &I(Y; X_i) + I(Y; X_j) - I(Y; X_i, X_j) \\ &= \left(\frac{1}{2}\right)\big(2I(Y; X_i) + 2I(Y; X_j) - 2I(Y; X_i, X_j)\big). \end{aligned}$$

Applying (4) twice on $I(Y; X_i, X_j)$, we get

$$\begin{aligned} &= \left(\frac{1}{2}\right)(2I(Y; X_i) + 2I(Y; X_j) - I(Y; X_i) - I(Y; X_j) \\ &\quad - I(Y; X_i | X_j) - I(Y; X_j | X_i)) \\ &= \left(\frac{1}{2}\right)(I(Y; X_i) - I(Y; X_i | X_j) + I(Y; X_j) - I(Y; X_j | X_i)). \end{aligned} \tag{6}$$

From (5), we know (6) is nonnegative. So, we get $Q_Y(X_i, X_j) = \frac{I(Y;X_i) + I(Y;X_j) - I(Y;X_i,X_j)}{H(Y)} \ge 0$.

The equality of $I(Y; X_i) = I(Y; X_i | X_j)$ implies that X_i, X_j are uncorrelated with respect to the decision variable Y. □

Algorithm 1. Feature Extract Algorithm (*FEA*)

1. Calculates the mutual information between the feature X_i and the decision Y, $I(Y; X_i)$.

2. Generating relevant features set R by comparing the mutual information $I(Y; X_i)$

   if $I(Y; X_i) \geq \delta_1$ then $R \leftarrow R + \{X_i\}$.

3. Creates working set W by copying R.

4. Creates goal set G = *null*.

5. While $e(G) < \delta_2$ do

6. if W = *null* then break.

7. choose $X_k \in W$ that subjects to

8. (a) $I(Y; X_k) \geq I(Y; X_l) \quad \forall l \neq k, X_l \in W$

9. (b) $Q_Y(X_k, X_n) \leq Q_Y(X_m, X_n) \quad \forall m \neq k, X_m \in W, \forall n, X_n \in G$

10. remove X_k from the working set $W \leftarrow W - \{X_k\}$ and put X_k into the target set

    $G \leftarrow G + \{X_k\}$

11. End loop

Fig. 2. Feature Extraction Algorithm.

Remark 3.4. When features X_i and X_j are fully correlated and they contribute 100 percent information in determine decision Y, the decision dependent correlation will be equal to 1 ($Q_Y(X_i, X_j) = 1$), which means that the features X_i and X_j are completely correlated with respect to decision Y.

Definition 3.3. *Let* (Ω, Γ, P), $Q_Y(X_i, X_j)$, I_n *as above. Let* S *denote a features subset with index set* $I_m = \{o_1, o_2, \ldots, o_m\}$ *and* $I_m \subseteq I_n$. *We define the new subset evaluation measure* $e(S)$ *in (7).*

$$e(S) = \frac{\sum_{\forall j \in I_m} I(Y; X_j)}{H(Y)} - \sum_{\forall i,j \; i \neq j \; i,j \in I_m} Q_Y(X_i, X_j). \quad (7)$$

This evaluation heuristic intuitively specifies a subset in which the mutual information of individual features with regard to the decision functions as an award for the merit of this subset, while the decision dependent correlation (DDC) between features is regarded as the penalty. So, the bigger the value of this metric $e(S)$, the better the feature subset in making a decision.

Remark 3.5. If the dependency among three or more features can be ignored, then $e(S) \leq 1$. Otherwise, the existing dependency among three or more features makes the $e(S) > 1$ in some cases.

4 Experimental Methodology

The research presented in this paper is part of a large effort to develop an autonomic control and management environment (AUTONOMIA) that provides self-configuring, self-optimizing, self-healing, and self-protecting services. The research presented here focuses on developing an online analysis to support the development of an AUTONOMIA self-protection engine. Our work aims at using the DDC metrics to identify the minimal sets of features that must be monitored and analyzed online in order to detect abnormal behaviors due to network attacks and, consequently, minimize and eliminate their impacts on the network operations and services.

4.1 Feature Extraction Algorithm (*FEA*)

The algorithm is based on the decision dependent correlation (DDC) measure discussed in the previous section. The goal of the feature selection algorithm is to select the minimum set of features that are strongly related to the desired decision variable and have the least redundancy among them.

The algorithm shown in Fig. 2 consists of two functional modules. The first one focuses on removing irrelevance. We use a user defined threshold δ_1 to determine which feature is relevant to the final decision (lines 1 and 2). In this part of the algorithm, irrelevant features are removed from the original feature set. The second part focuses on eliminating redundancy from the features to be selected (line 3). We

Algorithm 2 Learning Classifier ()

INPUT:

Dataset $\vec{D}$, FeatureSet $\vec{X}$

{* preprocessing the training data $\vec{D}$ according to feature set $\vec{X}$ which is generated through the algorithm 1 (*FEA*) *}

1. $\vec{D}^* \leftarrow feature\_discretization(\vec{D}, \vec{X})$;

{* initialize the coefficients for each feature in the classifier *}

2. $\vec{W} \leftarrow initialize\_weights(\vec{X})$;

3. $H \leftarrow \vec{W} \times \vec{X}^t$;

{* apply the classifier on each class of data, namely, dos, u2r, r2l, probe *}

4. $\vec{S} \leftarrow training\_data(\vec{D}^*, H)$;

5. **WHILE NOT** $(\vec{S} - |\vec{D}| * \vec{P} > 0)$

{* update the weights for the classifier *}

6. $\vec{W}^{new} \leftarrow update\_classiifer(\vec{W})$;

7. $H^{new} \leftarrow \vec{W}^{new} \times \vec{X}^t$;

{* training the new classifier *}

8. **FOR** each record $\vec{X}_i$ in $\vec{D}$ and each class C_j

9. **IF** $(H^{new}(\vec{X}_i) > T_j) \wedge (\vec{X}_i \in C_j)$

10. $\vec{S}_{C_j}.count = \vec{S}_{C_j}.count + 1$;

11. End **IF**

12. End **FOR**

13. End **WHILE**

Fig. 3. Learning the classifiers for multiple features.

quantify a final state criterion as the distance of subset evaluation metric $e(S)$ from the user defined threshold δ_2 (line 5). For each pass, the feature X_k is chosen which satisfies two conditions simultaneously. The first one is that feature X_k should be the most relevant one compared with the rest of features in the working set (line 8). The second one is that feature X_k should have the least correlation with all the features in goal set G when compared with the other features in the working set W (line 9). We use the DARPA KDD benchmark data set in validating our approach.

As we can see from Fig. 2, the main computational part of the algorithm involves computing the mutual information values for $Q_Y(X_i, X_j)$ and $e(\cdot)$, which has linear complexity $O(N)$ in term of the number of instances (N) in the training data set. The complexity of the algorithm that deals with determining the relevant features is of order $O(M)$, where M denotes the number of features, that is, the algorithm has linear complexity to determine the feature set from the relevant ones (assuming all features are selected first as relevant ones). The best complexity of the algorithm occurs when only one feature is selected and all of the other

TABLE 1
Ranked Features for DoS and Probe Attacks

DOS		Probe	
Feature	I(Y;X)/H(Y)	Feature	I(Y;X)/H(Y)
count	0.89973	src_bytes	0.617323
service	0.823221	service	0.508163
dst_bytes	0.711719	dst_host_diff_srv_rate	0.45012
logged_in	0.545972	dst_bytes	0.425978
dst_host_same_src_port_rate	0.531105	rerror_rate	0.343698
srv_count	0.475077	count	0.341383
protocol_type	0.43273	flag	0.329652
dst_host_count	0.428534	dst_host_srv_diff_host_rate	0.313887
src_bytes	0.403728	same_srv_rate	0.313486

TABLE 2
Ranked Features for U2R and R2L Attacks

U2R		R2L	
Feature	I(Y;X)/H(Y)	Feature	I(Y;X)/H(Y)
service	0.481635	service	0.559618
root_shell	0.37445	dst_host_srv_count	0.328744
dst_host_srv_count	0.281805	dst_host_same_src_port_rate	0.205641
duration	0.26578	dst_host_srv_diff_host_rate	0.183676
num_file_creations	0.255163	is_guest_login	0.159472
dst_host_count	0.177618	srv_count	0.149472
dst_host_same_src_port_rate	0.134272	dst_bytes	0.136806
srv_count	0.113392	dst_host_count	0.131907
dst_host_srv_diff_host_rate	0.091564	count	0.131043
src_bytes	0.086327	src_bytes	0.088246

features are removed and the worst-case complexity of order $O(M^2)$ when all features are selected. In general cases, when $k(1 < k < M)$ features are selected, the number of evaluations performed by FEA will typically be much less than the worst-case scenario. In summary, our method approximates relevance and redundancy among features by selecting a minimal set of features that meets the user specified threshold δ_2.

4.2 Learning Classification Algorithm

Given a feature set and a training data set $\{\vec{D}\}$, machine learning approaches could be used to learn a classification function. There are many optimization methods that have been proposed in the literature [15], [16]. In our system, we develop a learning algorithm based on genetic algorithms [9] to train the classification functions as shown in Fig. 3.

The feature discretization (line 1) is adopted to add continuous features to the discrete features list so they can both be considered with respect to a decision variable. The discretization of the features will be processed according to a coding strategy. For the selected discrete feature, the nominal values will be mapped into values 1 to M, which is the total number of nominal values. This mapping is done according to the frequency of the nominal values, i.e., the most frequent nominal value will map to 1 and the least frequent nominal value will be mapped into M. In this way, the weights adjustment will be done in a finer granularity. For continuous features, we decompose the continuous range of values into several intervals, where each interval has a finite set of values. In validating our approach, we use the DARPA KDD99 benchmark data set where the decision variable aims at detecting the occurrence of network attacks and their types. The data set consists of four types of network attacks such as Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probe attacks.

The classifier $(\vec{H})$ used in our algorithm has a linear function of weighted features, that is, $f(\vec{X}) = \sum_{\forall i} w_i \cdot x_i$, where x_i is the discretized value of selected ith feature and w_i is the weight assigned to this feature. The classifier typically divided the multidimensional feature space into different subspaces where each contains the majority of one category of data. During the initialization, the weights are generated randomly (lines 2 and 3). In the training process, the weights are adjusted such that the accuracy of the detection rate satisfies certain requirement (lines 4-12).

Stopping criterion (line 5) is a critical factor because it determines the accuracy of the detection rate of the algorithm. Strict stopping criteria will increase the computational overhead of the algorithm. Let $\vec{P}$ denote the detection satisfaction vector for all types of attacks. This parameter can be determined by the users based on domain knowledge and the desired detection rate for each type of network attacks. We set the stopping criteria for DoS, U2R, R2L, and Probe as $\vec{P} = \{95\%, 80\%, 80\%, 95\%\}$, respectively. That means, for U2R and R2L attacks, the desired detection rates are 80 percent, while the desired detection rate for the other types of attacks is equal to 95 percent.

5 Results on Large Data Sets

We analyzed the benchmark KDD99 data set [21] used in the Third International Knowledge Discovery and Data Mining Tools Competition to validate our approach. Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical US Air Force LAN. A connection is a sequence of TCP packets starting and ending at some well-defined times, between which data flows to and from a source IP address to a target IP address. Each connection is labeled as either normal or as an attack, with exactly one specific attack type. It is important to note that the testing data is not from the same probability distribution as the training data. There are 494,021 records in the training data set and the number of records in the testing data set is about five million. The data set contains a total of 22 different attack types. There are 41 features for each connection record that are divided into discrete sets and continuous sets.

We have implemented a genetic algorithm to discretize the continuous variable values into intervals to maximize the mutual information between the continuous features and the decision variable.

Features are ranked in descending order according to their relevance to the final decision. When set $\delta_{dos,1} = 0.4$, $\delta_{probe,1} = 0.31$, $\delta_{u2r,1} = 0.09$, $\delta_{r2l,1} = 0.08$, features are chosen as shown in Tables 1 and 2. Without feature-feature correlation analysis, we were able to get good detection rates for the *dos* and *probe* attacks, as shown in Table 3. However, the results

TABLE 3
Results Comparison of Different Approaches

Class	Our Approach using Continuous and Discrete Features	Our Approach using Discrete Features only	Winner Entry using C5.0	CTree
Normal	98.45%	98.34%	99.5%	92.78%
Dos	99.93%	99.33%	97.1%	98.91%
U2R	75.34%	63.64%	13.2%	88.13%
R2L	41.34%	5.86%	8.4%	7.41%
PROBE	99.91%	93.95%	83.3%	50.35%

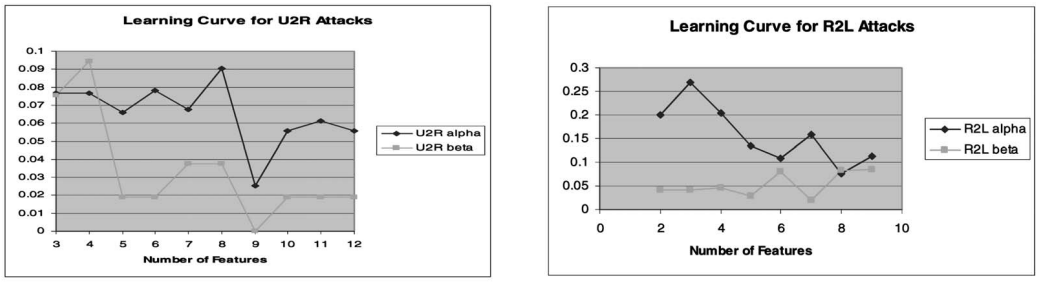


Fig. 4. Learning curves for U2R and R2L attacks with sequential selection.

are not good for U2R and R2L attacks. Similarly, other algorithms failed to achieve good detection rates for these attacks.

Results by sequentially choosing features in detecting U2R and R2L attacks are shown in Fig. 4. We use *alpha* and *beta* to denote false alarm and false negative, respectively. For U2R attacks, sequentially choosing the first nine features can gain a good detection rate (99.9 percent) with a false alarm of (2.4 percent). In other cases, the false alarm is around 6 percent and the detection rate is around 98 percent. For R2L attacks, the best case wasfound when using the first eight features, which gave a detection rate of 91.66 percent with 7.609 percent false alarm. Using the first seven features, the detection rate is around 98.05 percent and 15.93 percent false alarm. If the number of features is less than five, the false alarm will be above 15 percent.

We calculated the decision dependent correlation (DDC) among the features and obtained two correlation matrices for U2R and R2L attacks, as shown in Tables 4 and 5. $X_i, i = 1..10$ is used to denote the ith features in Table 1 and Table 2, respectively, e.g., X_6 in Table 4 is feature *dst_host_count*, while, in Table 5, it stands for feature *srv_count*. From Remark 3.3, we know the correlation matrix is symmetric and, because of that, we only show the DDC in the upper triangle part of the matrix.

By applying the feature extraction algorithm shown in Fig. 2, the features chosen for U2R and R2L attacks with respect to different thresholds level are shown in Tables 6 and 7. In Table 6, when set $\delta_2 = 0.99$, features *service* (x_1), *dst_host_srv_count* (x_3), *num_file_creations* (x_5), *dst_host_count* (x_6), and *dst_host_same_src_port_rate* (x_7) were selected from the autonomic feature extraction algorithm. Training based on these features, the learning algorithm gets the classifier for U2R attacks

$$f(x_1, x_3, x_5, x_6, x_7) = (-508)x_1 + (499)x_3 + (-908)x_5 + 480x_6 + (-90)x_7.$$

TABLE 4
Correlation Matrix for U2R Attacks

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
X1	0	0.2253	0.1345	0.1724	0.1119	0	0	0.0734	0.0203	0
X2		0	0.0824	0.1163	0.1030	0.0360	0.0051	0.0248	0.0227	0
X3			0	0.0371	0.0079	0	0	0.0918	0.0580	0
X4				0	0.0457	0	0	0.0187	0.0063	0
X5					0	0.0003	0	0.0184	0	0
X6						0	0.0706	0.0353	0	0
X7							0	0.0130	0	0
X8								0	0.0185	0
X9									0	0
X10										0

TABLE 5
Correlation Matrix for R2L Attacks

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
X1	0	0.247	0	0.0069	0.1574	0.0777	0.1284	0	0.0592	0.0482
X2		0	0	0.0845	0.0357	0.0920	0.0822	0	0.0861	0
X3			0	0.0359	0	0.0663	0.0275	0	0.0541	0
X4				0	0.0086	0.0827	0.0830	0.0407	0.0718	0
X5					0	0.0425	0	0	0.0363	0.0263
X6						0	0.0520	0.0196	0.1194	0
X7							0	0.0088	0.049	0
X8								0	0.0326	0
X9									0	0
X10										0

TABLE 6
Different Feature Subsets and Their Prediction for U2R Attacks

$\delta_{u2r,2}$	Subset (S)	$e(S)$	False alarm	Detection Rate
0.99	{ x1, x3, x5, x6, x7}	100.4% †	0.007587	92.55%
0.9	{ x1, x3, x5, x6}	94.05%	0.01431	96.23%
0.85	{ x3, x4, x5, x6}	88.93%	0.01531	91.47%
0.8	{ x1, x3, x6}	80.55%	0.019583	94.34%
0.7	{ x1, x3, x5}	77.03%	0.067961	90.06%

† *Refer to Remark 3.5.*

TABLE 7
Different Feature Subsets and Their Prediction for R2L Attacks

$\delta_{r2l,2}$	Subset (S)	$e(S)$	False alarm	Detection Rate
0.99	{ x1, x3, x4, x8}	99.73%	0.092581	91.13%
0.9	{ x1, x3, x4}	90.61%	0.09476	92.37%
0.85	{ x1, x3, x4}	90.61%	0.09476	92.37%
0.7	{ x1, x3}	76.53%	0.083524	92.46%

Applying this classifier on the testing data set resulted in a detection rate of 92.5 percent with a 0.7587 percent false alarm. We also note that if we set $\delta_2 = 0.9$, the classifier based on feature set $\{x_1, x_3, x_5, x_6\}$ can lead to a detection rate of 96.2 percent with a 1.43 percent false alarm. These results are significantly better than those obtained using the sequential feature selection approach.

For R2L attacks detection, the feature extraction algorithm yields a feature subset that consists of service (x_1), *dst_host_same_src_port_rate* (x_3), *dst_host_srv_diff_host_rate* (x_4), and *dst_host_count* (x_8). Using these features in detecting R2L attacks, we get a 91.13 percent detection rate with a 9.258 percent false alarm. When training on features *service* (x_1) and *dst_host_same_src_port_rate* (x_3), we obtained a detection rate of 92.47 percent with 8.35 percent false alarm. The results are comparable to the optimal sequential selection of eight features. However, the small number of features will result in a much faster learning process and it will reduce the overhead in collecting data when used in a real network environment.

6 Conclusions and Future Work

In this paper, an efficient algorithm for feature extraction is proposed to remove the irrelevance and redundancy information during the data preparation period. Our validation results show that the new decision dependent correlation measure $Q_Y(X_i, X_j)$ and the subset evaluation heuristic metric $e(S)$ can be used to select the near optimal feature subset. Based on these features, the learning algorithm can be a better classifier when compared with the sequential selection strategy. Our results show a significant improvement in the detection rates for the most difficult to detect attacks (e.g., U2R and R2L). For U2R attacks, our approach can achieve a 92.5 percent detection rate with false alarm of 0.7587 percent. For R2L attacks, our approach can achieve a 92.47 percent detection rate with false alarm of 8.35 percent. Nevertheless, R2L attacks detection results suggest some special recommendations. Due to the probable existence of interdependency among three or more features, the learning algorithm may pick up some inaccuracy in its classification.

We are currently investigating techniques to integrate our approach with online monitoring, filtering, and a self-protection system that can be used in a real network environment.

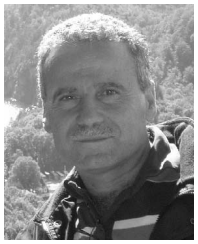
Acknowledgments

This work is supported in part by grants from Intel Corporation ISTG R&D Council, US National Science Foundation/NGS Contract 0305427, and US National Science Foundation/SEI(EAR) Contract 0431079.

References

[1] A. Al-Ani and M. Deriche, "Feature Selection Using a Mutual Information Based Measure," *Proc. 16th Int'l Conf. Pattern Recognition,* vol. 4, pp. 82-85, 2002.
[2] H. Almuallim and T.G. Dietterich, "Learning with Many Irrelevant Features," *Proc. Ninth Nat'l Conf. Artificial Intelligence,* pp. 547-552, 1991.
[3] R. Battiti, "Using Mutual Information for Selecting Features in Supervised Neural Net Learning," *IEEE Trans. Neural Networks,* vol. 5, pp. 537-550, 1994.
[4] A. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning," *Artificial Intelligence,* vol. 97, nos. 1-2, pp. 245-271, 1997.
[5] S. Das, "Filters, Wrappers and a Boosting-Based Hybrid for Feature Selection," *Proc. 18th Int'l Conf. Machine Learning,* pp. 74-81, 2001.
[6] M. Dash and H. Liu, "Feature Selection for Classification," *Intelligent Data Analysis: An Int'l J.,* vol. 1, pp. 131-156, 1997.
[7] M.A. Hall, "Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning," *Proc. 17th Int'l Conf. Machine Learning,* pp. 359-366, 2000.
[8] M.A. Hall and G. Holmes, "Benchmarking Attribute Selection Techniques for Discrete Class Data Mining," *IEEE Trans. Knowledge and Data Eng.,* 2002.
[9] K.A. De Jong, "An Analysis of Behavior of a Class of Genetic Adaptive Systems," PhD Dissertation, Dept. of Computer and Comm. Sciences, Univ. of Michigan, 1975.
[10] K. Kira and L.A. Rendell, "A Practical Approach to Feature Selection," *Proc. Ninth Int'l Workshop Machine Intelligence,* 1992.
[11] R. Kohavi and G. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence,* pp. 273-324, 1997.
[12] I. Kononenko, "Estimating Attributes: Analysis and Extensions of Relief," *Proc. Seventh European Conf. Machine Learning,* pp. 171-182, 1994.
[13] P. Langley, "Selection of Relevant Features in Machine Learning," *Proc. AAAI Fall Symp. Relevance,* 1994.
[14] H. Liu and R. Setiono, "A Probabilistic Approach to Feature Selection: A Filter Solution," *Proc. 13th Int'l Conf. Machine Learning,* pp. 319-327, 1996.
[15] J.A. Miller, W.D. Potter, R.V. Grandham, and C.N. Lapena, "An Evaluation of Local Improvement Operators for Genetic Algorithms," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 23, pp. 1340-1351, Sept./Oct. 1993.
[16] Y. Peng and J.A. Reggia, "A Connectionist Model for Diagnostic Problem Solving," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 19, pp. 285-298, Mar./Apr. 1989.
[17] W.H. Press, B.P. Flannery, S.A. Teukolski, and W.T. Vetterling, *Numerical Recipes in C.* Cambridge Univ. Press, http://www.library.cornell.edu/nr/bookcpdf.html, 2005.
[18] J.R. Quinlan, *C4.5: Programs for Machine Learning.* San Mateo, Calif.: Morgan Kaufmann, 1993.
[19] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," *Proc. 20th Int'l Conf. Machine Learning (ICML-2003),* 2003.
[20] L. Yu and H. Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy," *J. Machine Learning Research,* vol. 5, pp. 1205-1224, Oct. 2004.
[21] http://kdd.ics.uci.edu/databases/kddcup99/task.html, 2005.

Guangzhi Qu is a PhD candidate in the Electrical and Computer Engineering Department at The University of Arizona. His research interests lie in the area of networking and, particularly, in using information theory, data mining and statistical methods in network measurement, monitoring, performance analysis, reliability, and service assurance. He is also interested in the analytics of wireless and sensor networks. He is a student member of the IEEE.



Salim Hariri received the PhD degree in computer engineering from University of Southern California in 1986 and the MSc degree from The Ohio State University in 1982. He is a professor in the Department of Electrical and Computer Engineering at The University of Arizona. He is the director of the Center for Advanced TeleSysMatics (CAT): Next Generation Network Centric Systems. Dr. Hariri is the editor-in-chief for the *Cluster Computing Journal* (Springer, http://www.springerlink.com/link.asp?id=101766). He is the founder of the IEEE International Symposium on High Performance Distributed Computing (HPDC) and the cofounder of the IEEE International Conference on Autonomic Computing. His current research focuses on autonomic computing, high-performance distributed computing, design and analysis of high speed networks, benchmarking and evaluating parallel and distributed systems, developing software design tools for high performance computing and communication systems, and network-centric applications. He is coauthor/editor of three books on parallel and distributed computing: *Tools and Environments for Parallel and Distributed Computing* (Wiley, 2004), *Virtual Computing: Concept, Design and Evaluation* (Kluwer, 2001), and *Active Middleware Services* (Kluwer, 2000). He is a senior member of the IEEE.

Mazin Yousif received the masters and PhD degrees from Pennsylvania State University in 1987 and 1992, respectively. He is a principle engineer and manager in the Corporate Technology Group of Intel Corporation in Hillsboro, Oregon. He currently leads a team that focuses on platform provisioning and virtualization to enable platform autonomics and Capacity on Demand (CoD). Prior to that, he was in the Enterprise Product Group focusing on InfiniBand and datacenter I/O interconnects. During his involvement with the InfiniBand Architecture, he chaired the InfiniBand Trade Association (IBTA) Management Working Group (MgtWG). From 1993 to 1995, he was an assistant professor in the Computer Science Department at Louisiana Tech University. He worked for IBM's xSeries Server Division in Research Triangle Park (RTP), North Carolina, from 1995-2000. His research interests include computer architecture, clustered architectures, workload characterization, networking, and performance evaluation. He has published more than 50 articles in his areas of research. He chaired the program committee of several conferences and workshops, was on the program committees of many others, and led several panels. He is on the advisory board of the *Journal of Pervasive Computing and Communications* (JPCC), and is an editor of *Cluster Computing*, *The Journal of Networks, Software Tools and Applications*. He is a senior member of the IEEE.