A Gaussian Distribution-based Lightweight Intrusion Detection Model

Yuchen Wang†, Shuxiang Xu††, Wei Liu† and Qiongfang Huang†

†College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China
††School of Computing and IS, University of Tasmania, Launceston, Australia

Summary
The important parts of building a lightweight intrusion detection model include selecting informative features and designing efficient classification process. In this paper, we propose a novel Gaussian distribution-based lightweight intrusion detection (GD-LID) model, which combines a Gaussian distribution filtering model with a particular machine learning algorithm. Initially, feature selection with information gain is performed to find out the features with the most discriminative information, and 2 features are selected for our model. Then, we build a Gaussian distribution describing normal data and carry out a threshold selection algorithm to establish our Gaussian distribution filtering model which distinguishes outliers, uncertain data and normal data. Finally, we incorporate 5 well-known machine learning algorithms respectively into our model to classify the uncertain data. Experimental results show that our GD-LID model has very similar accuracy rate compared with using the 5 machine learning algorithms directly, but it can filter 43.05% of total network traffic data with only 2 features.

Key words:
intrusion detection; lightweight; Gaussian distribution filtering model; feature selection

1. Introduction

Intrusion detection is the discovery of intrusion behavior in computer network systems. Its concept was first proposed by Anderson in 1980 [1]. He pointed out that the network traffic data and audit data contain valuable information, which can be exploited to detect abnormal behavior in computer networks. In order to verify intrusion detection methods for different communities around the world, the Knowledge Discovery and Data Mining (KDD) contest in 1999 established the KDD99 dataset [2] as a evaluation platform. This dataset was simulated by Defense Advanced Research Projects Agency (DARPA) in the laboratory network environment and contained well-labeled classes and sufficient pretreatment.

Although many practical intrusion detection devices have already been built, these devices are too sophisticated and expensive to be afforded for organizations or departments who lack capital and technology. Therefore, it is important to develop a lightweight intrusion detection system with low cost and high detection rate.

Recently, many researchers have proposed a bunch of lightweight algorithms for intrusion detection. Sindhu et al. [3] presented a lightweight method based on decision tree and wrapper. This method eliminates the redundant data and select 16 features and achieve better detection accuracy. Li et al. [4] developed a gradually feature removal method to get 19 features, and then used clustering, ant colony algorithm and support vector machine to build intrusion detection model. Amiri et al. [5] proposed a feature selection method combining linear correlation coefficient with nonlinear mutual information and used least squares support vector machines as classifier. The results showed that this method has a high accuracy for detecting remote to local (R2L) and user to remote (U2R). Bajaj et al. [6] chose features by using information gain, and compared the performance of 6 machine learning algorithms on NSL-KDD dataset. Part et al. [7] presented a hybrid feature selection based on correlation and established a lightweight intrusion detection model. This method can achieve a high detection rate, as well as reducing the training time and testing time significantly.

In this paper, we propose a novel Gaussian distribution-based lightweight intrusion detection (GD-LID) model, which combines a Gaussian distribution filtering model with a particular machine learning algorithm. The building process of our model can be summarized as follows. First, we select two features from KDD99 dataset based on information gain to fully exploit the discriminative information with the minimal number of features. Second, using the selected two features, we build a two-dimensional Gaussian distribution filtering model to distinguish outliers, uncertain data and normal data. Third, we adopt 5 kinds of machine learning algorithms to classify he uncertain data filtered by the previous step. Compared with applying machine learning algorithms directly to classify intrusion data, the proposed GD-LID model could be able to classify approximately half of total intrusion data with only 2 features, and has almost the same accuracy in comparison with these machine learning algorithms.

This paper is organized as follows. Section 2 briefly reviews related work. Section 3 presents the proposed GD-

Manuscript received September 5, 2015
Manuscript revised September 20, 2015

LID algorithm. Section 4 shows the experimental results. Section 5 summarizes this paper.

2. Related work

2.1 Gaussian distribution

In probability theory, Gaussian distribution is commonly used to represent random variables whose distributions are unknown. Its definition is written as:

$$p\left(x,\mu,\sigma\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\left(x-\mu\right)^2}{2\sigma^2}\right) \quad (1)$$

Given a training set $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$, with each sample containing n dimension. The estimated parameters (mean and variance) of Gaussian distribution are calculated as follows:

Mean:

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \quad (2)$$

Variance:

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)} - \mu_j\right)^2 \quad (3)$$

The Gaussian distribution filtering model [8] assumes that every feature of a sample is generated from a Gaussian distribution. A sample with n dimension has n corresponding Gaussian distributions, which control the density of all features of the sample. In particular, the density at the mean point is maximal, and the variance reflects the variation degree of the data.

The density of a sample is defined as the product of each dimension:

$$p\left(x\right) = \prod_{j=1}^{n} p\left(x_j;\mu_j,\sigma_j^2\right) = \prod_{j=1}^{n}\frac{1}{\sqrt{2\pi}\sigma_j}\exp\left(-\frac{\left(x_j-\mu_j\right)^2}{2\sigma_j^2}\right) \quad (4)$$

This value can be used to represent the probability of occurrence of a sample. By defining a threshold ε. The sample density with a probability less than ε is identified as anomaly. This process can be written as:

$$if\ \ p\left(x\right)\begin{cases} \le \varepsilon & anomaly \\ > \varepsilon & normal \end{cases} \quad (5)$$

2.2 Machine learning algorithms

Machine learning is kind of technique which can mimic the learning process of humans to automatically acquire new knowledge without being programmed directly to solve a specific problem. In 1998, Tom Mitchell [9] created a formal definition for machine learning which can be described as: for a computer program, given a task T and a evaluation method P, if the P, when doing T, is improved by offering experience E, it can be concluded that the program has learned from E.

Based on the expected output of algorithms and the types of input data, machine learning techniques are generally divided into 3 categories: supervised learning [10], unsupervised learning [11] and reinforcement learning [12]. We mainly adopt the supervised learning algorithms to classify data. The input data used by supervised learning algorithms includes a feature vector and a desired output value. The algorithms then analyze the data and generate hypotheses to map new input data to different categories. This means that the learning algorithm need to have strong generalization capability to predict unknown data. There are many kinds of supervised learning algorithms. In this paper, we apply 5 well-known algorithms, which are Decision Tree [13], Naive Bayes [14], Radial Basis Function Network (RBF) [15], Logistic Regression [16] and Support Vector Machine (SVM) [17].

2.3 Feature selection based on information gain

Information gain [18] is a commonly used method in feature selection [19]. Given a set of labeled training data $S = \{(x^{(1)}, y^{(1)}),(x^{(2)}, y^{(2)}),...,(x^{(m)}, y^{(m)})\}$, where the number of samples is m, every sample has n features, and the number of samples with class i is si, the expected amount of information to distinguish a sample is computed by the following formula:

$$I\left(s_1,s_2,...,s_n\right) = -\sum_{i=1}^{n}\frac{s_i}{m}\log_2\left(\frac{s_i}{m}\right) \quad (6)$$

Given a feature F of a sample with a range of $\{f_1, f_2,... f_v\}$, this feature divides training set S into V subsets $\{S_1,S_2,...S_v\}$ where Sj indicates the subsets with feature fi. Let sij denotes the number of class i in subset Sj. The entropy of feature F could be calculated as:

$$I\left(s_1,s_2,...,s_n\right) = -\sum_{i=1}^{n}\frac{s_i}{m}\log_2\left(\frac{s_i}{m}\right) \quad (7)$$

The information gain of the feature F can be calculated by:

$$Gain\left(F\right) = I\left(s_1,s_2,...,s_n\right) - E\left(F\right) \quad (8)$$

Based on above formula, Gain(fi) could be calculated for each feature. Next, the feature selection could be performed by utilizing cross validation method. The pseudo code is described as follows:

Table 1. Feature selection based on information gain

Algorithm: Feature selection based on information gain
Input: { f_1 ,…, f_d} /* features sorted in descending order in terms of Gain(f_i) */ S_{train} /* Training set */ S_{cv} /* Cross validation set */ M /* Classification method */
Output: F_{best} /* Feature set with minimum error */
set $F_0 = \varnothing$ For $i = 1,...n$ set $F_i = F_{i-1} \cup \{f_i\}$ $h_i = train(M, F_i, S_{train})$ $\hat{\varepsilon}_i = test(S_{cv}, h_i)$ set $F_{best} = \arg_{F_i} \min \hat{\varepsilon}_i$

3. Gaussian distribution-based lightweight intrusion detection

3.1 Feature selection

In this paper, we adopt KDD99 dataset, which has 41 features in total, including basic connection features, connection content features, 2 seconds traffic statistics features, and statistical characteristics of recent 100 connections.

By applying information gain on the 41 features of KDD99, we get the feature list sorted in descending order. Specifically, the sequence numbers of the list are 5, 23, 3, 6, 36, 12, 24, 2, 32, 37, 33, 35, 31, 34, 29, 30, 39, 38, 26, 4, 25, 1, 41, 40, 10, 28, 16, 19, 13, 17, 8, 22, 18, 7, 20, 27, 21, 11, 9, 15, 14. The corresponding figure is shown below:

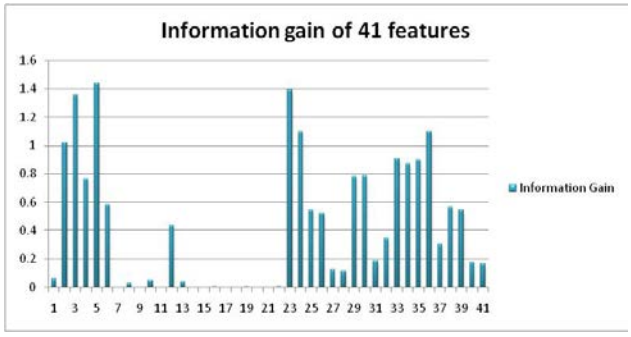

Figure 1. Information gain of 41 features

Fig. 3 and Fig. 4 show the F1Score and error rate by modeling the first 1 features to the first 15 features from the sorted list with logistic regression:

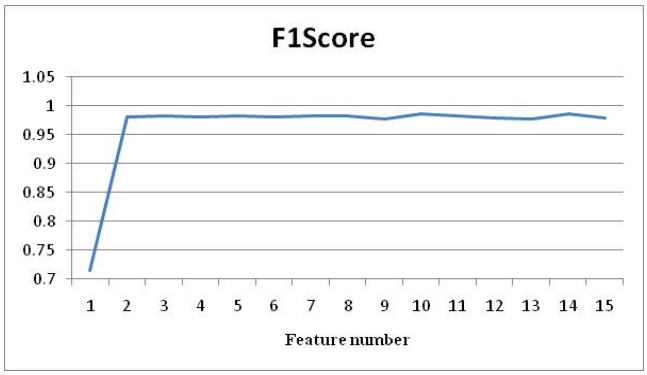

Figure 2. F1Score with different feature numbers

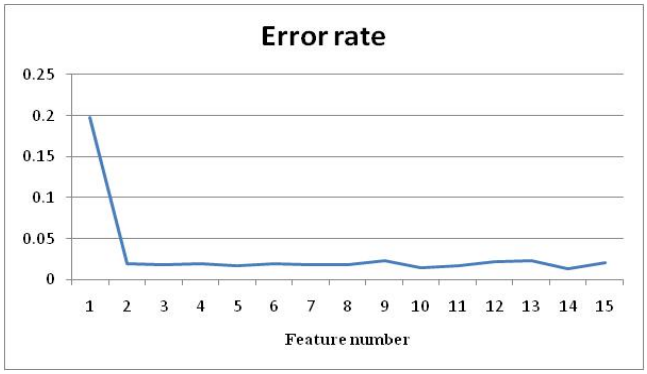

Figure 3. Error rate with different feature numbers

As can be seen from the figures, by utilizing the two largest features with respect to information gain, which are 5th feature src_bytes and 23rd feature count, the F1Score of logistic regression algorithm has increased to a fairly high level, and error rate has decreased to a very low number. This means that the two features contain a lot of category information of samples. Therefore, we choose to use these 2 features to build Gaussian distribution filtering model.

3.2 Construction of lightweight intrusion detection model

The main idea of this paper is as follows: First, we need to establish the Gaussian distribution which describes the possibility of normal data with the features of src_bytes and count. Next, we perform our threshold selection algorithm to find out the thresholds for classifying data. Then, for each sample to be detected, we compute the product of the 2 probability density, and then compared it with 2 thresholds ε anomaly and ε normal. If the product is less than ε anomaly, the occurrence probability of the detected sample is small so that it is classified to be abnormal. By contrast, if the product is bigger than ε normal, it means that the detected sample is likely to be

normal. If the product number is between ε anomaly and ε normal, the category of the sample cannot be determined by the Gaussian distribution filtering model and needs to be fed into machine learning algorithms for final classification. The above process can be presented as:

$$if \quad p(x)\begin{cases} \le \varepsilon_{anomaly} & anomaly \\ > \varepsilon_{anomaly} \wedge \le \varepsilon_{normal} & undetermined \\ > \varepsilon_{normal} & normal \end{cases} \tag{9}$$

In summary, the proposed GD-LID model is constructed in following 5 steps:

1) According to the last section, the feature src_bytes and count are selected to build the Gaussian distribution filtering model. A sample can be represented by a vector x, where x1 is src_bytes, x2 is count, and x3 is the sample's category:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_{src\_bytes} \\ x_{count} \\ x_{class} \end{bmatrix} \tag{10}$$

2) Using the data with normal label, we can compute mean μ and variance σ to get the Gaussian distribution model which describes normal network behavior. In other words, this distribution gives information about the occurrence tendency of normal data.

3) According to Eq. 4, using the established Gaussian distribution in last step, we can calculate the density for all samples in cross validation set and sort them in ascending order. Every density we get represents the occurrence possibility of a particular sample.

4) The threshold selection algorithm for Gaussian distribution filtering model is summarized in table 2. Specifically, to find the threshold ε anomaly and ε normal, we conduct an iterative process with each iteration using the k smallest samples with respect to density from the ascending list. These k samples are predicted as anomaly and the rest of samples are expected as normal data, thus getting the precision rate and recall rate. The number k ranges from 1 to the size of cross validation set. We choose the threshold ε anomaly when the precision rate is bigger than 0.997 and choose ε normal when the recall rate is bigger than 0.997.

5) We also train 5 machine learning algorithms on whole training set, which can be used to classify the data that cannot be determined by previous Gaussian distribution filtering model. A machine learning algorithm and Gaussian distribution filtering model together comprise our GD-LID model, which can filter data efficiently and also get a reasonable overall result.

Table 2. Threshold selection algorithm

Algorithm: threshold selection algorithm
Input: { x(1),…, x(m)} /* cross validation samples in ascending order in terms of density */ m /* cross validation set size */
Output: ε anomaly , ε normal /* threshold for classifying normal an abnormal data */ $TP = 0$ $FP = 0$ $FN = attackNumber$ $TN = normalNumber$ For $i = 1,...n$ $if\ x_3^{(i)} == attack$ $TP++$ $FN--$ $else$ $FP++$ $TN--$ $Presion_i = TP/(TP+FP)$ $Recall_i = TP/(TP+FN)$ $\varepsilon_{anomaly} = \max P(x^{(i)} \mid Presion_i > 0.997)$ $\varepsilon_{normal} = \min P(x^{(i)} \mid Recall_i > 0.997)$

3.1 Model diagram

The whole constructing and detecting process of our proposed GD-LID is demonstrated in Fig. 5, including the process of training Gaussian distribution filtering model, training 5 types of machine learning algorithms and classification.

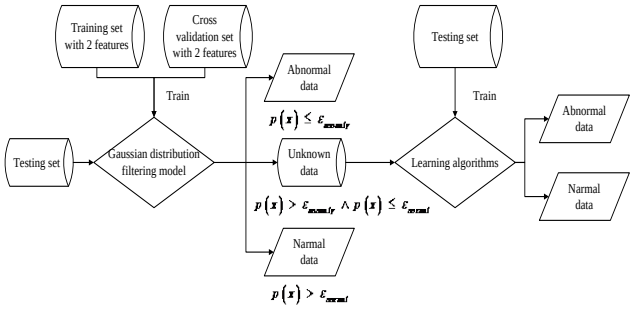

Figure 4. Diagram of GD-LID lightweight intrusion detection process

4. Experimental results

In this section, we present experimental results on KDD99 dataset to illustrate the effectiveness of our method. To establish the Gaussian distribution filtering model and hypotheses of machine learning algorithms, we use the following experimental data which are randomly picked from the whole dataset:

Table 3. Experimental data for GD-LID model

	Training set for Gaussian distribution filtering model	Cross validation set	Training set for machine learning algorithms	Testing set
Samples size	5847	12267	123505	12266

where the 5847 samples are all normal data used for building Gaussian distribution describing normal data and cross validation set is used for selecting the thresholds. Training set for machine learning algorithms and testing set are also presented.

According to the Eq. 2 and Eq. 3, we can get the parameters of Gaussian distribution describing normal data by using the normal samples only:

Table 4. Parameters of Gaussian describing normal data

	src_bytes	count
mean	2.349685096	1.198818532
variance	0.534154664	0.194765851

Next, we calculate the density of every sample in cross validation set and sort them in ascending order. After this, we predict the k smallest samples as abnormal data with respect to density and then get the precision rate and recall rate, where the range of k is from 1 to the size of cross validation set (12267). The curves of precision and recall rate depending on k is shown below:

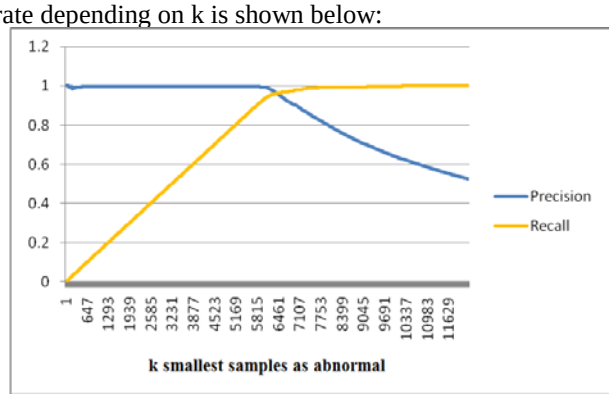

Figure 5. The curves of precision and recall rate with different k

The result illustrates that the samples with less density are more possible to be attack data, and vice versa. We then select the threshold $\varepsilon_{anomaly}$ when the precision rate is bigger than 0.997 and choose ε_{normal} when the recall rate is bigger than 0.997. This can ensure our thresholds can filter the input data with a significantly high accuracy. The exact number of two thresholds are shown in the following table:

Table 5. Thresholds of Gaussian distribution filtering model

	$\varepsilon_{anomaly}$	ε_{normal}
density	0.00000000353	1.191596412

We then do the testing on testing set, and get below results:

Table 6. Test results on testing set

	Detected as normal	Detected as attack	Not sure
Sample number	2029 (19 false detection)	3252 (9 false detection)	6985
Proportion	43.05%		56.95%

It is clear from the above table that it is only with 2 features that the Gaussian distribution filtering model manages to classify almost half of total data (43.05%) and achieve a performance of 99.7% of precision and recall rate. For the rest 6985 uncertain data, we feed them into the trained hypotheses of the 5 machine learning algorithms respectively to obtain 5 final classification results.

Besides testing on our GD-LID model, we also take the testing set as the input of the 5 machine learning algorithms directly for comparison. Fig. 6, in terms of 5 kinds of learning algorithms, compares the accuracy rates achieved between using one particular machine learning algorithm directly and using the GD-LID model incorporated with that algorithm. The exact variation rate is presented in table 7. We can see from the table that our GD-LID model is only marginally less than using J48, Logistic, SVM and RBF directly, with the declines reaching 0.230%, 0.083%, 0.018% and 0.001% respectively. In addition, GD-LID can improve the performance of Naive Bayes by 0.626%. Despite the slight decreases on 4 algorithms, our model is able to utilize only 2 features to filter 43.05% data, which is nearly half of total number of data.

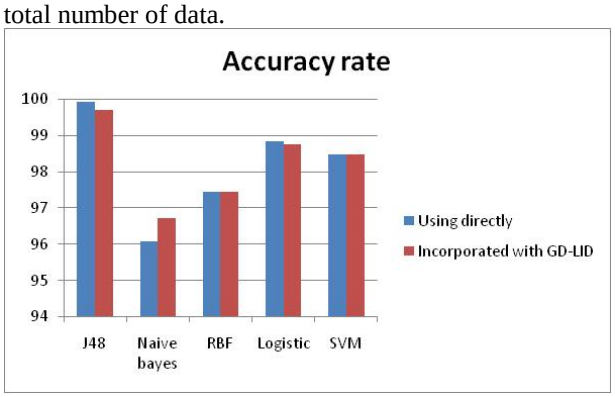

Figure 6. Comparisons between using machine learning algorithms and their GD-LID version

Table 7. Performance variation by applying GD-LID

Algorithms	Incorporated with GD-LID
J48	- 0.230%
Naive bayes	+ 0.626%
RBF	- 0.001%
Logistic	- 0.083%
SVM	- 0.018%

4. CONCLUSION

In this paper, we propose a GD-LID model for detecting network malicious data. By using information gain, feature src_bytes and count are selected from KDD99 dataset for the filtering process. Then, by building the Gaussian distribution describing normal data and performing the threshold selection algorithm, the Gaussian distribution filtering model is established to filter the network traffic data. The experimental results demonstrate that GD-LID model has very similar accuracy rate compared with using 5 kinds of machine learning algorithms directly, but it only uses 2 features to filter 43.05% data, which speeds up the detection rate and reduces computational consumption, making it possible to be applied to places with limited software and hardware resources such as small hydropower stations.

References

[1] J.E Anderson. Compmer security threat monitoring and surveillance,Technical Report,James E Anderson Company, Fort Washington, Pennsylvania, April 1980.

[2] Pfahringer B. Winning the KDD99 classification cup: bagged boosting[J]. ACM SIGKDD Explorations Newsletter, 2000, 1(2): 65-66.

[3] Sivatha Sindhu S S, Geetha S, Kannan A. Decision tree based light weight intrusion detection using a wrapper approach[J]. Expert Systems with applications, 2012, 39(1): 129-141.

[4] Li Y, Xia J, Zhang S, et al. An efficient intrusion detection system based on support vector machines and gradually feature removal method[J]. Expert Systems with Applications, 2012, 39(1): 424-430.

[5] Amiri F, Rezaei Yousefi M M, Lucas C, et al. Mutual information-based feature selection for intrusion detection systems[J]. Journal of Network and Computer Applications, 2011, 34(4): 1184-1199.

[6] Bajaj K, Arora A. Dimension Reduction in Intrusion Detection Features Using Discriminative Machine Learning Approach[J]. International Journal of Computer Science Issues (IJCSI), 2013, 10(4).

[7] Park J S, Shazzad K M, Kim D S. Toward modeling lightweight intrusion detection system through correlation-based hybrid feature selection[C]//Information Security and Cryptology. Springer Berlin Heidelberg, 2005: 279-289.

[8] Leon E, Nasraoui O, Gomez J. Anomaly detection based on unsupervised niche clustering with application to network intrusion detection[C]//Evolutionary Computation, 2004. CEC2004. Congress on. IEEE, 2004, 1: 502-508.

[9] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training[C]//Proceedings of the eleventh annual conference on Computational learning theory. ACM, 1998: 92-100.

[10] Jordan M I, Rumelhart D E. Forward models: Supervised learning with a distal teacher[J]. Cognitive science, 1992, 16(3): 307-354.

[11] Hofmann T. Unsupervised learning by probabilistic latent semantic analysis[J]. Machine learning, 2001, 42(1-2): 177-196.

[12] Sutton R S. Introduction: The challenge of reinforcement learning[M]//Reinforcement Learning. Springer US, 1992: 1-3.

[13] Mašetic Z, Subasi A. Detection of congestive heart failures using C4. 5 Decision Tree[J]. SouthEast Europe Journal of Soft Computing, 2013, 2(2).

[14] Salah K, Kahtani A. Performance evaluation comparison of Snort NIDS under Linux and Windows Server[J]. Journal of Network and Computer Applications, 2010, 33(1): 6-15.

[15] Qasem S N, Shamsuddin S M. Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis[J]. Applied Soft Computing, 2011, 11(1): 1427-1438.

[16] Hosmer Jr D W, Lemeshow S. Applied logistic regression[M]. John Wiley & Sons, 2004.

[17] Mohabatkar H, Mohammad Beigi M, Esmaeili A. Prediction of GABAA receptor proteins using the concept of Chou's pseudo-amino acid composition and support vector machine[J]. Journal of Theoretical Biology, 2011, 281(1): 18-23.

[18] Hall M A. Correlation-based feature selection for machine learning[D]. The University of Waikato, 1999.

[19] Guyon I, Elisseeff A. An introduction to variable and feature selection[J]. The Journal of Machine Learning Research, 2003, 3: 1157-1182.