基于改进 ACO 优化 BPNN 的软件缺陷预测模型

李克文，王秋宝+，于明晓

（中国石油大学 计算机与通信工程学院，山东 青岛 266580）

摘 要：针对用 BP 神经网络进行软件缺陷预测时出现的易陷入局部最优、学习速度缓慢等缺陷问题，提出一种基于信息素初始化和局部路径优化的蚁群优化算法优化 BP 神经网络的软件缺陷预测模型。对待预测的数据集进行基于互信息和自信息优化的主成分分析操作，降低数据的维数，提高运算效率；根据改进后的蚁群优化算法，计算最优的 BP 神经网络权值和阈值；使用 NASA 提供的软件缺陷数据集，利用提出的模型进行缺陷预测，基于十折交叉方法进行验证。通过与几种传统方法对比验证了所提方法具有更快的收敛速度和更高的预测准确度。

关键词：软件缺陷预测模型；BP 神经网络；蚁群优化算法；主成分分析；互信息

中图法分类号：TP311 **文献标识号**：A **文章编号**：1000-7024（2017）08-2137-05

doi：10.16208/j.issn1000-7024.2017.08.027

Optimizing software defect prediction model of BP neural network based on improved ACO algorithm

LI Ke-wen, WANG Qiu-bao+, YU Ming-xiao

(College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China)

Abstract: The traditional BP neural network algorithm has many drawbacks when it is used in software defects prediction. For example, it falls into local optimum easily and learns slowly. In view of these problems, a software defects prediction model based on the BP neural network improved by ant colony optimization algorithm (ACO) was proposed. Specifically, the ACO algorithm was based on pheromone initialization and local path optimization. The mutual information and self-information were integrated with the principal component analysis (PCA) method to reduce the dimensions of data set and increase the operation efficiency. The optimal BP neural network weights and thresholds were calculated using the optimized ant colony optimization algorithm. The performances of the proposed model were tested on NASA data sets according to the ten-fold cross method. Experimental results show that the proposed method achieves higher convergence rate and accuracy compared with conventional methods.

Key words: software defect prediction model; BP neural network; ant colony optimization; PCA; mutual information

0 引 言

目前贝叶斯网络[1]、隐马尔可夫模型[2]、支持向量机模型[3]、分类回归树模型[4]等已经被普遍应用于软件缺陷预测领域，但这些方法仍然存在问题，不能达到很好的效果。例如，当贝叶斯网络结构变得复杂时，其推理计算量会呈指数级增长，由于计算量过大而导致算法难以实现；隐马尔可夫模型适用于在比较小的数据集，对于复杂特征的数据集预测效果不理想；支持向量机参数设置主观性太强，核函数及其参数的选择都是根据经验来选取的；逻辑回归虽然进行二次判别可以明显提高计算效率和识别率，但是对于小样本数据预测效果不明显。

近年来，由于 BP 神经网络具有良好的分类能力、泛化能力和自我学习能力等原因，研究基于神经网络的软件缺陷预测逐渐受到关注，但人们在理论研究和实际应用中发现该类算法仍然存在一些局限，例如在权值和阈值的调整过程中，采用梯度下降方法，易找到局部最优解，不能保证获得全局最优以及学习速率缓慢等，这些问题严重影响

收稿日期：2016-08-08；修订日期：2016-09-06

基金项目：山东省自然科学基金项目（ZR2013FL034）

作者简介：李克文（1969-），男，山东青岛人，博士，教授，CCF 会员，研究方向为计算智能、软件工程、数据挖掘与机器学习；+通讯作者：王秋宝（1990-），男，山东青岛人，硕士研究生，研究方向为软件理论与服务计算、数据挖掘；于明晓（1994-），男，山东日照人，硕士研究生，研究方向为计算智能、数据挖掘。E-mail：wqb19901230@163.com

了网络的预测精度。Dorigo 等提出蚁群优化算法(ant colony optimization，ACO)[5]，它是一种模拟蚂蚁觅食的启发式搜索算法，具有鲁棒性、正反馈、全局优化以及易于其它算法相结合等优点，ACO 算法训练 BP 神经网络可有效克服 BP 神经网络的不足，并且可以提高 BP 神经网络的收敛速度及泛化能力。

为了提升软件缺陷预测的准确率，本文提出了一种基于改进 ACO 优化 BP 神经网络的软件预测模型。利用互信息优化后的主成分分析方法对待测数据预处理，并通过改进后的 ACO 优化 BP 神经网络，最终建立起一种软件缺陷预测模型。

1 研究基础

1.1 主成分分析方法

主成分分析算法(principal component analysis，PCA)[6]由 Parson 提出的，是统计学中一种简化数据集的技术。它的主要思想是通过在样本中各变量分析的基础上，通过线性变换的方式，将多个原始变量转换为几个互不相关的综合变量(即主成分)，同时保持原始变量中的绝大部分信息。

PCA 的主要步骤如下：

(1) 假设用 n 个随机向量来描述研究对象，计算它的协方差矩阵，记为：$\sum$。

(2) 求出矩阵 $\sum$ 的特征值，并按照从大到小的顺序排列($\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$)，特征值的大小直接影响了各主成分的影响力。

(3) 通过步骤(2)得到的特征值计算前 m 个主成分累计贡献率为 $\sum_{i=1}^{m}\lambda_i / \sum_{i=1}^{n}\lambda_i$，一般取累计贡献率达到 85%～95%的特征值 $\lambda_1,\lambda_2,\lambda_3,\cdots,\lambda_m$ 所对应的第 1，第 2，…，第 $m(m<n)$ 个主成分。

1.2 互信息和自信息

随着信息科学对现代社会生活各方面影响的不断增加，人们对信息论的认识和价值估计也不断加深[7]。在信息理论中为了更好地描述诸属性之间的联系，引入了互信息的概念。

互信息反映了两个随机变量 X 和 Y 之间的关联关系，表示两个变量之间公有的信息量。若两个变量各自的边缘概率分布和联合概率分布分别为 $p(x)$，$p(y)$ 和 $p(x,y)$，则它们之间的互信息 $I(X;Y)$ 定义为

$$I(X;Y)=\sum_{x,y}p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{1}$$

特别地，当 $Y=X$ 时，表示自信息；由式(1)得到的 $I(X;Y)$ 值越大，表示它们相互关联程度越高，所包含的相同信息也越多。

1.3 蚁群优化算法

蚁群优化算法是由 Dorigo 等学者提出，是模拟自然界中真实蚁群的觅食行为发展成的一种智能算法。作为一种全局优化的方法，蚁群算法具有正反馈性、并行计算、自组织性以及易与其它算法结合等优点，使其在旅行商问题和组合优化等领域得到了广泛应用。

蚁群优化算法的执行过程如下：

(1) 蚁群数量、信息素浓度等信息初始化。

(2) 根据式(2)和式(3)进行下一个城市的转移

$$p_{ij}^{k}(t+1)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}(\eta_{ij})^{\beta}}{\sum\limits_{k\notin tabu_k}[\tau_{ik}(t)]^{\alpha}(\eta_{ik})^{\beta}}, j\notin tabu_k\\ 0,\text{其它}\end{cases} \tag{2}$$

$$j=\begin{cases}\arg\max\limits_{j\in J_k(i)}\{[\tau_{ij}(t)]^{\alpha}(\eta_{ij})^{\beta}\}, q\leqslant q_0\\ S,\text{其它}\end{cases} \tag{3}$$

其中，q 是在[0，1]区间的随机数，q_0 的大小决定了利用先验知识与探索新路径之间的相对重要性；$\tau_{ij}(t)$ 是描述 t 时刻城市 i、j 之间的信息素；η_{ij} 是描述启发式信息；α、β 是描述信息素和启发式因子相对重要程度的控制参数；S 是一随机变量，根据式(3)得到选择城市 j 的概率，$tabu_k$ 是蚂蚁 k 已经走过的城市集合(即禁忌表)。

(3) 根据式(4)和式(5)对它们所经过的路径进行局部信息素更新

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\Delta\tau_{ij}^{k} \tag{4}$$

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{l_{jb}}, (i,j)\in T^k\\ 0,\text{其它}\end{cases} \tag{5}$$

其中，l_{jb} 是第 k 只蚂蚁经过的路径长度，Q 为常数，T^k 为蚂蚁 k 经过的路径集合。

(4) 若所有蚂蚁都完成本次城市遍历，则执行步骤(5)，否则转向步骤(2)。

(5) 根据式(6)和式(7)对本次循环最优的蚂蚁所经过的路径进行全局信息素更新

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\Delta\tau_{ij}^{best} \tag{6}$$

$$\Delta\tau_{ij}^{best}=\begin{cases}\dfrac{1}{l_{best}}, (i,j)\in T^k\\ 0,\text{其它}\end{cases} \tag{7}$$

其中，l_{best} 表示到目前为止找出的全局最优路径的长度。

(6) 重复进行步骤(2)～步骤(5)，直到循环的次数达到最大迭代次数停止算法，输出结果。

2 软件缺陷预测模型

2.1 基于互信息和自信息的主成分分析方法

通过数据集协方差矩阵计算而来的主成分特征向量描述的仅仅是线性相关关系，但是各属性之间总是会有某种非线性依赖关系的存在，所以为了避免忽视非线性依赖关系的问题，提出一种基于互信息和自信息的主成分分析方法。

由于传统主成分分析方法中属性之间的非线性关系无法度量的问题，而互信息和自信息却是信息论中衡量属性相关性的重要指标[8]，这种相关性度量不局限于线性关系，而且还兼顾了变量之间的非线性关系。所以我们用属性的自信息和互信息矩阵替换协方差矩阵来计算主成分，即

$$\sum_{i,j} = I(\mathrm{X}_i, \mathrm{X}_j) \tag{8}$$

其中，当 $i=j$ 时，表示自信息；$\sum_{i,j}$ 表示信息矩阵第 i 行第 j 列的信息，$I(X_i, X_j)$ 是属性 X_i 和属性 X_j 两个属性的互信息值。

2.2　基于改进蚁群优化算法对模型参数的优化

2.2.1　基于信息素初始化和局部路径优化的蚁群优化算法

蚁群算法虽然有许多有优点，但同时也有不少缺点，例如：算法的搜索效率低下，收敛速度缓慢，在算法的运行过程中极易出现早熟现象，极易陷入局部最优解[9]。为此，本文提出一种基于信息素初始化和局部路径优化的蚁群优化算法。

在传统蚁群算法的初始化阶段，人们通常是将每一条路径上的信息素浓度初始化为相同的常数值，这就造成了在初始阶段蚂蚁只能依靠启发式信息的指导进行搜索，进而导致大部分蚂蚁都倾向于选择最短的路径，造成了局部收敛过快，易陷入局部最优解的问题。因此，我们提出“差异初始化信息素浓度”，基本思想是将蚂蚁随机分布在各个城市，计算每只蚂蚁到达其余 $n-1$ 个城市的距离，并按照由小到大的顺序排列，取前 k 个城市作为搜索范围，并且按照式（9）在相对较长路径上初始化较高浓度的信息素而在相对较短路径上初始化较低浓度的信息素的策略，这样在初始阶段既兼顾了信息素和启发式信息两种因素同时也可以提高蚂蚁搜索下一个城市的随机性

$$\tau_{ij(0)} = \begin{cases} \alpha + \Delta\tau_{ij}, j \in \Phi(i) \\ \alpha, \text{否则} \end{cases} \tag{9}$$

$$\Delta\tau_{ij} = \begin{cases} \dfrac{d_{ij}}{\max(d_{ij})}, j \in \Phi(i) \\ 0, \text{否则} \end{cases} \tag{10}$$

其中，α 是一个信息素基数，$\Delta\tau_{ij}$ 是根据路径长短额外添加的信息素浓度，$\Phi(i)$ 是选取的 k 个搜索范围，d_{ij} 是城市 i 到城市 j 的距离。

此外，为加快收敛速度和提高解的质量，提出结合遗传算法中的变异操作进行局部二次搜索，当每只蚂蚁遍历完所有城市以后，然后把所有蚂蚁遍历的路径按照递增排序排列，并且截取前面的一半蚂蚁路径作为优化范围，根据式（11）计算出该范围内种群应该产生变异的个体个数 k，随机选取 k 个蚂蚁路径进行变异，然后从变异点进行局部二次搜索，如果变异后经二次搜索得到的路径长度小于未变异时的路径长度，则更新本次遍历路径表，否则不更新

$$k = \frac{d_{\max} - d_{\min}}{d_{\max}} \cdot m \tag{11}$$

其中，$d_{\max}$ 是城市 i 到城市 j 的最大值，$d_{\min}$ 是城市 i 到城市 j 的最小值，m 是选取变异个体数量，在这里是种群数量的一半。

2.2.2　基于改进 ACO 算法对模型参数的优化

本文利用改进后的 ACO 算法对预测模型中 BP 神经网络权值和阈值的优化。即利用改进后的 ACO 算法遍历出最优路径的方式来确定 BP 神经网络的最优权值和阈值。在整个参数优化的过程中，我们不采用蚂蚁遍历的路径长度作为标准来全局更新路径上的信息素，而是通过蚂蚁遍历完成后得到的参数（阈值和权值）应用于 BP 神经网络预测模型，在预测结果中找到具有最小预测准确率误差的蚂蚁来全局更新信息素浓度。为了利用改进后的 ACO 算法来优化 BP 神经网络的参数，我们假设 BP 神经网络里一共有 m 个待优化的参数，这其中包含隐含层和输出层所有的阈值与权值。首先，将这些参数进行有序排列，记为：$p_1, p_2, p_3, \cdots, p_m$，将任一神经网络参数 $p_i (1 \leqslant i \leqslant m)$ 设置为可能取值范围（一般情况下都是 $[-1, 1]$）内的 N 个随机非零值）形成集合 I_{pi}。然后定义蚂蚁数量 S，每只蚂蚁从集合 I_{pi} 出发。根据集合中每个元素的信息素和式（2）、式（3）从每个集合中选择一个元素，蚂蚁在所有集合中完成元素的选择后，就认为此时蚂蚁到达了食物源，完成了一次寻找食物的过程。这一过程反复执行，当全部蚂蚁收敛到同一路径时或达到最大迭代次数，找到最优路径便表示找到网络参数的最优解。

改进 ACO 算法优化 BP 神经网络的软件缺陷预测模型的参数优化执行过程如下：

（1）将 BP 神经网络权值和阈值按照前述方法设置，并且设定网络节点数、训练误差 ε_0 等神经网络参数信息；

（2）设蚂蚁个数 S，当前迭代次数为 NC，最大迭代次数为 $NC\_\max$、最优路径长度为 l_{best}、城市 i 与城市 j 距离为 d_{ij}、当前路径禁忌表为 $tabu$，所有蚂蚁遍历完的路径表 T，依据式（9）和式（10）得到初始信息素浓度等；

（3）设置蚂蚁初始位置。初始化蚂蚁的禁忌表 $tabu_k (k=1,2,3,\cdots,s)$，将 S 只蚂蚁随机放置于 n 个城市结点上，把相应的这 S 个值添加到禁忌表中；

（4）根据式（2）和式（3）决定其下一个要访问的城市；

（5）根据式（4）和式（5）对蚂蚁经过的路径进行信息素更新。如果蚂蚁遍历完成所有城市，则转到步骤（6）；否则转到步骤（4）；

（6）局部路径优化：

结合遗传算法的变异操作进行二次搜索局部优化蚁群算法构造的解。当每只蚂蚁遍历完所有城市以后，把所有

蚂蚁遍历的路径按照由短到长升序排序，然后截取前面的一半路径作为优化范围，根据式（11）计算出该范围内种群中将会发生变异的个体数量 k，随机选取 k 只蚂蚁路径进行变异操作，再从变异点开始按照步骤（4）进行二次搜索；

（7）将所有蚂蚁每次迭代遍历得到的权值和阈值训练 BP 神经网络，找到具有最低误差值的最优蚂蚁，根据式（6）和式（7）全局更新最优蚂蚁遍历路径上的信息素，如果预测误差达到预先设定值或者迭代次数超过最大循环次数，则转到步骤（8），否则转到步骤（3）；

（8）得到最优的网络权值和阈值后，建立最终优化的软件预测模型，并利用测试数据集对软件模块进行预测。

3 仿真实验及分析

3.1 实验数据

实验数据是美国国家航空航天局（NASA）提供的软件缺陷数据集（metrics data program，MDP）[10]。MDP 数据集有 13 个不同的子数据集，每个数据集都提供了许多条记录，一条记录代表一个模块，每条记录的各个数据对应不同属性。所以针对测试模型来说，数据集中的每一条记录即为一个测试样本。我们选取其中的 4 个数据集 JM1，CM1，KC1，PC1 做缺陷预测模型的实验数据，见表 1。

表 1 实验数据集

数据集	模块数	缺陷模块	缺陷率/%
JM1	10 885	2106	19.35
CM1	505	49	9.83
KC1	2109	326	15.45
PC1	1109	77	6.94

3.2 实验结果与分析

为了验证模型的预测能力，本实验平台采用 MATLAB R2012a，并采用十折交叉验证方法[11]进行实验。将本文提到的方法通过与传统的预测方法：朴素贝叶斯算法（Naive Bayes）、逻辑回归算法（logic）、BP 神经网络（BPNN）、ACO 优化的 BP 神经网络（ACO-BPNN）等比较，并结合之前研究的相关实验数据[12]得到在 4 个数据集上计算的准确度、查准率、查全率、F1 值[13]的比较结果如图 1～图 4 所示。

由图 1 至图 4 可知，基于朴素贝叶斯方法和逻辑回归方法建立的模型预测结果相对不理想，这是由于朴素贝叶斯方法假设属性之间不存在依赖关系，而这种假设本身就存在一定的误差性；基于逻辑回归方法的模型容易受到样本容量的限制，只有样本容量比较大的时候才可能产生较理想的预测效果，并且预测时间较长。基于 BP 神经网络的模型相对而言效果好一些，但是受到阈值和权值的影响，其分类准确率和泛化能力较差。基于蚁群优化算法优化的

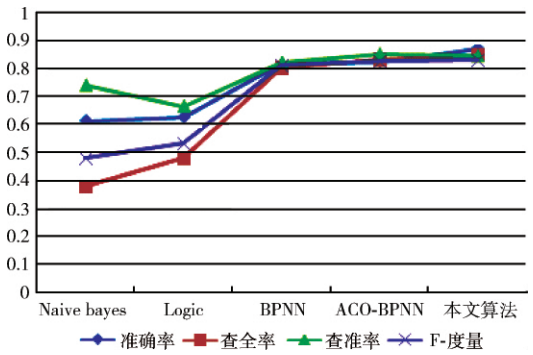

图 1 JM1 数据集预测结果

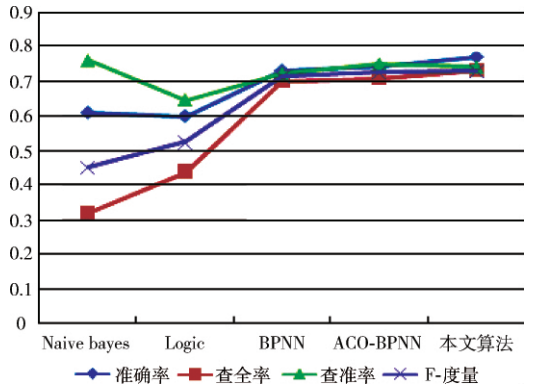

图 2 CM1 数据集预测结果

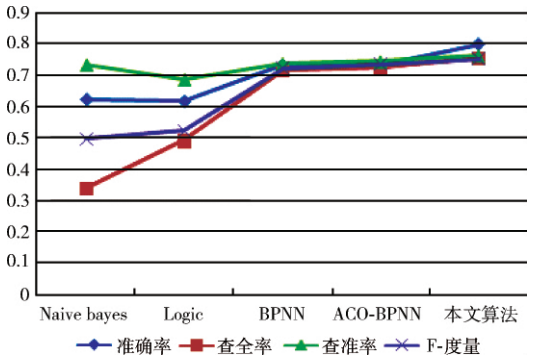

图 3 KC1 数据集预测结果

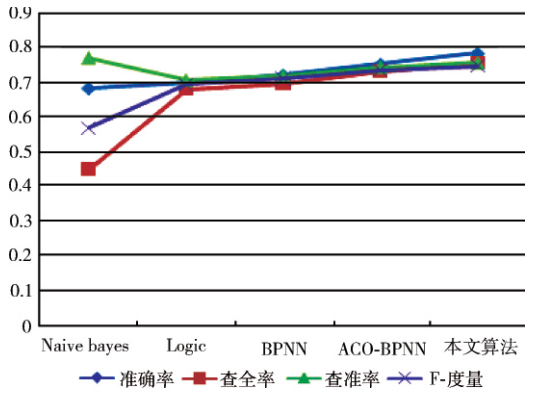

图 4 PC1 数据集预测结果

BP 神经网络模型较传统的 BP 神经网络在准确率上得到一定的改善，但是受到蚁群算法自身易陷入局部最小值的影响，得到的权值和阈值并不是很理想且预测能力不稳定。本文提出改进的蚁群优化算法，对收敛速度和解空间都做了提升，使改进后的蚁群算法鲁棒性更强且也加强了寻优能力，最终，通过利用改进后的主成分分析法实现特征降维，并利用改进后的 ACO 算法优化 BP 神经网络参数，使得优化后的预测模型在准确率、查准率等各项评价指标中均优于 Naive Bayes、Logic、BPNN、ACO-BPNN 模型。

4 结束语

本文针对软件缺陷预测问题提出了一种预测模型：基于改进 ACO 算法优化 BP 神经网络的软件缺陷预测模型。针对原始软件缺陷数据集中存在着许多对最终分析结果影响微小的属性集，我们采用基于互信息和自信息的主成分分方法做了数据预处理，仅保留了一小部分对分析结果影响显著的属性子集，然后根据改进后的蚁群优化算法来计算最优的 BP 神经网络权值和阈值，用优化好的 BP 神经网络算法对预处理的缺陷数据集进行软件缺陷预测，并通过与其它软件缺陷预测模型对比，验证了该方法的有效性和优越性。但是该模型在 BP 神经网络参数寻优过程阶段需要较长的时间，一定程度上影响了模型的执行效率，下一步的研究方向是如何进一步提高模型的执行效率和预测准确率。

参考文献：

[1] ZHANG Jing. Based on the software defect prevention research and application of Bayesian network [D]. Hebei: Heibei University of Technology, 2014 (in Chinese). [张婧. 基于贝叶斯网络的软件缺陷预防研究与应用 [D]. 河北：河北工业大学，2014.]

[2] Joshua Landon. A Markov modulated Poisson model for software reliability [J]. European Journal of Operational Research, 2013, 229 (2): 404-410.

[3] WANG Tao, LI Weihua, LIU Zun, et al. A software DP (defects prediction) model based on SVM (support vector machine) [J]. Journal of Northwestern Polytechnical University, 2011, 29 (6): 864-870 (in Chinese). [王涛，李伟华，刘尊，等. 基于支持向量机的软件缺陷预测模型 [J]. 西北工业大学学报，2011，29 (6)，864-870.]

[4] Asry Faidhul Ashaari Pinem, Erwin Budi Setiawan. Implementation of classification and regression tree (CART) and fuzzy logic algorithm for intrusion detection [C] //3rd International Conference on Information and Communication Technology, 2015: 266-271.

[5] Speranskii. Ant colony optimization algorithms for digital device diagnostics [J]. Antomatic Control and Computer Sciences, 2015, 49 (2): 82-87.

[6] Mohamed Morchid, Richard Dufour. Feature selection using principal component analysis for massive retweet detection [J]. Pattern Recognition Letters, 2014, 49: 33-39.

[7] WANG Weiling, LIU Peiyu, CHU Jianchong. An improved feature selection algorithm based on conditional mutual information [J]. Journal of Computer Applications, 2007, 27 (2): 433-435 (in Chinese). [王卫玲，刘培玉，初建崇. 一种改进的基于条件互信息的特征选择算法 [J]. 计算机应用，2007，27 (2)：433-435.]

[8] Amiri F, Rezaei M. Mutual information based feature selection for intrusion detection system [J]. Journal of Network and Computer Applications, 2011, 34 (4): 1184-1199.

[9] XIA Yamei, CHENG Bo, CHEN Junliang. A denotation and application of preference ontology for service composition [J]. Chinese Journal of Computers, 2012, 35 (2): 270-280 (in Chinese). [夏亚梅，程渤，陈俊亮. 基于改进蚁群算法的服务组合优化 [J]. 计算机学报，2012，35 (2)：270-280.]

[10] JIANG Huiyan, ZONG Mao, LIU Xiangying. Research of software defect prediction model based on ACO-SVM [J]. Chinese Journal of Computers, 2011, 34 (6): 1148-1153 (in Chinese). [姜慧研，宗茂，刘相莹. 基于 ACO-SVM 的软件缺陷预测模型的研究 [J]. 计算机学报，2011，34 (6)：1148-1153.]

[11] Purushotham Swarnalatha, BK Tripathy. Evaluation of classifier models using stratified tenfold cross validation techniques [M]. Global Trends in Information Systems and Software Applications. Springer, 2012 (11): 680-690.

[12] LI Kewen, CHEN Chenxi. Software defect prediction using fuzzy integral fusion based on GA-FM [J]. Wuhan University Journal of Natural Sciences, 2014, 19 (5): 405-408.

[13] Han Jiawei, Micheling Kamber. Data mining concepts and techniques [M]. 3rd ed. Beijing: China Machine Press, 2012: 236-240.