

# 中南大学计算机学院

## 实训文献综述

课程名称	2019 年夏季实训
学 号	3901170505
学生姓名	孙佳程
专业班级	软件 1705
论文题目	基于 SVM 构建软件缺陷 预测模型的研究综述
批 阅 人	
论文成绩	

2019 年 7 月 6 日

# 基于 SVM 构建软件缺陷预测模型的研究综述

孙佳程

中南大学计算机学院, 湖南长沙 410083

**摘要:**支持向量机(Support Vector Machine)被广泛地运用于软件缺陷预测模型的构建中, 软件缺陷预测因其数据集的不同、度量元的选取不同, 模型之间会存在很大的差异。本文对使用 NASA MDP 数据, 基于 SVM 构建软件缺陷预测模型的研究进行总结分析, 主要包括: 数据集的噪声、降维、类不平衡问题的解决, SVM 的参数选取与寻优方法, 最后对 SVM 与其它机器学习方法进行简单比较并总结以 SVM 构建软件缺陷预测模型的特点。

**关键字:** 软件缺陷预测, 机器学习, 支持向量机

## A Survey of Research on Software Defect Prediction Model Based on SVM

Sun JiaCheng

School of Computer Science and Engineering Central South University, Changsha  
Hunan 410083

**Abstract:** Support Vector Machine is widely used in the construction of software defect prediction models. Because of the different data sets and the selection of metric elements, there are great differences between models. This paper summarizes the research on using NASA MDP data and based on SVM to build software defect prediction model, including: data set noise, dimensionality reduction, class imbalance problem solving, SVM parameter selection and optimization method, and finally SVM. Simple comparison with other machine learning methods and summarizing the characteristics of building software defect prediction models with SVM

**Keywords:** software defect prediction, machine learning, svm

缺陷是软件产品期望属性的偏离，是软件故障和软件失效的源头<sup>[1]</sup>。其存在于软件的任何生命周期，检测、修复缺陷的时间越晚，所需的代价也越大。然而，给予一个项目的时间、人力资源都是有限的，如何尽早、精确地定位缺陷是产业界的迫切需求。

软件缺陷预测<sup>[2,3]</sup>是目前一种可行的方案，通过挖掘、分析软件代码历史仓库，借助机器学习等方法构建缺陷预测模型是学界与工业界研究的主要方向。根据陈翔等人<sup>[4]</sup>对静态软件缺陷预测<sup>[2]</sup>的研究，静态软件缺陷预测有三个主要影响因素：软件度量元的设定、缺陷预测模型的构建方法和缺陷预测数据集处理相关问题。本文将分析机器学习中使用最广泛的方法——支持向量机(SVM)在构建缺陷预测模型中的应用。

为了对该问题进行系统地分析、比较和总结，首先在 IEEE、Elsevier 和 CNKI 等论文数据库中进行检索，检索时采用的主要英文关键字包括 software defect, software defect prediction, svm software defect. 最后经过筛选，得到 14 篇与使用 SVM 构建软件缺陷预测模型直接相关的论文。

本文第一部分介绍了综述论文内使用 SVM 构建缺陷预测模型的相关工作，第二部分就不同论文介绍的数据处理与改进方法做出总结，第三部分对 SVM 与其它机器学习构建缺陷预测模型做一个简单的比较，并总结 SVM 在构建软件缺陷预测模型方面的优势与劣势。

## 1 相关工作

### 1.1 支持向量机

SVM 是在 1995 年由 Cortes 和 Vapnik<sup>[5]</sup>提出的一种基于学习理论和结构风险最小化原则的统计机器学习算法，它的实质是一种监督学习的分类算法，在解决小样本、非线性及高维模式识别中表现出许多特有的优势。SVM 的基本思想是求解核函数和二次线性规划问题，可以通过核函数将线性不可分样本映射到高维空间，找到最优的分类平面作为决策曲面，所谓最优的分类平面就是求解一个平面  $g(x) = wx + b$ ，使得正例与反例之间的隔离边缘被最大化，即样本的几何间距最大，从而进行分类计算。SVM 常用的核函数有以下 3 种：

a) 线性核函数：

$$K(x, x_i) = x^T x_i$$

b) 径向内积核函数：

$$K(x, x_i) = \exp(-\gamma \|x - x_i\|^2) \quad \gamma > 0$$

c) 多项式核函数：

$$K(x, x_i) = (\gamma x^T x_i + r)^p$$

其中径向基核函数因为有着较宽的收敛范围，被广泛地应用在 SVM 中。

## 1.2 缺陷数据集

### 1.2.1 NASA 数据集

80%的缺陷预测模型实验中都使用 NASA 公布的 NASA IV & V Facility Metrics Data Program( MDP) 数据集，如表 1 所示，MDP 包括 13 个不同的数据集，这些数据均来自 NASA 的实际软件项目，由最常见的开发语言编写。每个数据集包含来自不同软件项目的若干模块，规模从 125 /6k 到 17186 /315k(模块个数 /代码行)不等。MDP 数据集包含 LOC、McCabe、Halstead 以及其它等四类代码属性。所有类型的代码属性再加上是否包含缺陷的类型标记( defective) 一共 43 个度量元，MDP 数据集所提供的记录，一行代表一个模块，每个字段对应一个属性。依据 SVM，每一记录构成的向量即为一个样本。

表 1 NASA MDP 数据集基本信息

	CM1	JM1	KC1	KC3	KC4	MC1	MC2	MW1	PC1	PC2	PC3	PC4	PC5
开发语言	C	C	C++	Java	Perl	C++	C	C	C	C	C	C	C++
代码行数	20k	315k	43k	18k	25k	63k	6k	8k	40k	40k	26k	36k	164k
模块个数	505	10878	2107	458	125	9466	161	403	1107	5589	1563	1458	17186
包含缺陷 模块个数	48	2102	325	43	75	68	52	31	76	23	160	178	516

### 1.2.2 度量元的选择

在使用 NASA 数据集的实验中，度量元选取数据集给出的静态代码属性 LOC、McCabe、Halstead，在不使用 NASA 数据集的实验中，由 SVM 构建的缺陷预测模型，使用的度量元也是静态缺陷预测模型的静态代码属性，这一点使得各个实验之间存在着共性。

### 1.2.3 数据集的预处理

英国赫特福德大学计算机科学学院 Gray 等人指出了 NASA 的数据集中存在着大量的“重复数据”和“矛盾数据”，经过研究这些数据已经严重影响了软件缺陷预测的实验结果，而在使用 NASA 数据集进行的实验中，大多数没有指出这一点，所以需要进行的数据集预处理包含以下几点：

- 1) 删除“重复数据”和“矛盾数据”
- 2) 根据 error\_count 属性，设每个数据集中 error\_count  $\geq 1$  的模块为有缺陷模块，即添加 defective 类标记属性

3)可选地加入 error\_density 属性，计算方式为  $error\_density = 1000 \times (error\_count / loc\_total)$

4)在实际训练中需要将 error\_count 属性和 error\_density 属性从样本向量中移除

5)可选择地对数据进行归一化操作

## 1.4 模型评估方法

使用 SVM 构建的缺陷预测模型本质上是一个二分类问题，即对有缺陷模块和无缺陷模块的分类，故在所有的实验中，超过 80%使用了十折验证(10-fold)来进行实验，所有实验都使用了混淆矩阵来对模型性能进行评估

表 2 混淆矩阵

实际值	预测值	
	缺陷高倾向模块	缺陷低倾向模块
缺陷高倾向模块	正确的正例(TP)	错误的负例(FN)
缺陷低倾向模块	错误的正例(FP)	正确的负例(TN)

实际正例个数  $P = TP + FN$ ，实际负例个数  $N = FP + TN$ ，实例总数  $C = P + N$ 。

模型评价指标的定义如下：

准确度(accuracy)表示正确分类的测试实例的个数占测试实例总数的比例,计算公式如下：

$$accuracy = \frac{TP + TN}{C}$$

查准率(precision)表示正确分类的正例个数占分类为正例的实例个数的比例,计算公式如下：

$$precision = \frac{TP}{TP + FP}$$

查全率(recall)表示正确分类的正例个数占实际正例个数的比例,计算公式如下：

$$recall = \frac{TP}{P}$$

F1(F-measure)表示查全率与查准率的调和平均 ,计算公式如下：

$$F - measure = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

## 2 研究综述

### 2.1 数据噪音的处理

在挖掘软件历史存档时，在对程序模块进行类型标记和软件度量时均可能产生噪音，这些噪音的存在会影响到缺陷预测模型的构建，并会对已有实证研究结论的有效性产生严重影响。论文实验中采用的多是 NASA 的数据集，前文已经提到过 NASA 的数据集存在“重复数据”和“矛盾数据”，而论文中对此的不同处理将对实验结果的比较造成影响。

### 2.2 数据维度灾难的处理

在程序模块度量时，若考虑了大量与代码或开发过程相关的度量元，会使得一些数据集存在维数灾难问题(the curse of dimensionality)。因此，需要提出有效方法来重点识别出数据集中的冗余特征(在软件缺陷预测问题中，特征指度量元)和无关特征。其中，冗余特征大量或完全重复了其他单个或多个特征中含有的信息，而无关特征则对采用的数据挖掘算法不能提供任何帮助。研究结果表明：无关特征和冗余特征的存在会提高缺陷预测模型的构建时间，并会降低模型的预测性能。

在使用 SVM 构建软件缺陷预测模型的实验中，使用特征子集选择进行数据降维是被采用最多的方法。其中主要被使用到的方式有：主成分分析方法、LASSO 方法、降噪编码器、NPE 算法四种方式。

#### 2.2.1 主成分分析方法

主成分分析方法 (PCA)<sup>[6]</sup>，主成分分析将给定的一组相关变量通过线性变换转换为另一组不相关的变量，这些新的变量按照方差递减的顺序排列，构成相应的主成分。在姜慧研等人的《基于 ACO-SVM 的软件缺陷预测模型的研究》<sup>[7]</sup>使用的就是 PCA 数据降维方式。

#### 2.2.2 LASSO 方法

Tibshirani 于 1996 年提出了一种有偏估计的方法，被用来处理高维数据的特征选择问题，该方法称为最小绝对值压缩与选择方法(LASSO)<sup>[8]</sup>LASSO 可以通过惩罚函数将影响较小的观察变量的回归系数收缩到 0，这样做虽然牺牲了一定的估计偏差，但能降低预测的方差从而提高预测的精确性。吴晓萍等人<sup>[9]</sup>在 LASSO-SVM 构建缺陷预测模型的实验中，使用的是由 Efron 等人<sup>[10]</sup>于 2004 年提出的改进算法 LARS。LARS 算法是每次在回归残差的基础上选择新的变量，该过程是一个残差不断减小的过程，具有很高的计算效率。回归残差综合

了类标签变量与已选变量的信息，将该算法引入 LASSO 问题中，能高效地求得最优解，即所要求的变量选择序列。LARS 算法的基本思想是：首先选择一个与响应变量相关性最大的观察变量，然后沿这个方向走一定的长度，直到出现第二个观察变量。这两个观察变量与残差的相关性相同，就沿着这两个变量等角度的方向继续走。以此类推，选择出需要的观察变量。

### 2.2.3 降噪编码器

甘露等人<sup>[11]</sup>指出了使用 PCA 与 LDA<sup>[12]</sup>的方式进行数据降维和特征提取都只能通过线性转换进行特征提取，无法对特征进行非线性转换，从而具有一定的局限性。如果将输入属性记为 $(x_1, x_2, \dots, x_n)$ ，预测结果用  $y$  表示，那么 PCA 和 LDA 只能得到  $y = a_1x_1 + a_2x_2 + \dots + a_nx_n$  的线性关系，而如果某个属性对于是否为缺陷是非线性的关系时，如代码行数和是否为缺陷存在  $y = ax_{1k}$  时，PCA 和 LDA 并不能得到，而且降噪自动编码器<sup>[13,14]</sup>(Denoising Autoencoder, DA)可以压缩特征并提取特征间非线性的依赖关系。

### 2.2.4 NPE 算法

NPE 算法认为样本空间中任意样本点都可由其领域内的  $K$  个样本点线性表示，当数据集嵌入到更低维度时，将非线性数据特征转化为多个线性特征，保留了数据的全局几何结构信息。<sup>[4,15]</sup>王玉红等人在研究中指出<sup>[16]</sup>改进的 NPE 算法，奇异矩阵的特征计算问题，将特征向量的求解转化为计算两矩阵的特征分解，在简化广义特征分解计算问题的同时，保留了有用的原始判别条件，求得了更加稳定的特征向量具有以下优良特性：克服了高维、小采样条件下奇异矩阵的广义特征分解问题，在不增加算法复杂度、不损失有用信息的情况下，可以有效地对高维数据进行特征分解，得到的特征向量具有完备性和稳健性。

## 2.3 类不平衡的处理

类不平衡问题是软件缺陷预测训练数据集中普遍存在的问题，其产生的根源在于缺陷在软件模块中的分布大致符合二八原则，即，80%的缺陷集中分布于 20%的程序模块内。因此在软件缺陷预测中，无缺陷程序模块(即多数类)的数量要远超过有缺陷程序模块(即少数类)的数量，因此会造成模型对少数类的预测精度(例如查准率和查全率)较低。同时，在缺陷预测模型的构建过程中需要考虑不同预测错误类型的开销，其中，错误类型 I 是将 FP 模块错误地预测为 NFP 模块，该错误类型将造成测试人员不会对 FP 模块进行必要的代码审查和测试，并最终将含有缺陷的软件产品推向市场；错误类型 II 是将 NFP 模块错误地预测为 FP 模块，

该错误类型会造成测试人员对 NFP 模块进行大量的代码审查和测试，因此造成测试资源的浪费。显然，第 1 类错误会给企业带来更大的损失。

类不平衡学习<sup>[16]</sup>是当前机器学习中的一个热点研究领域，可以有效缓解数据集中的类不平衡问题。已有方法可以大致分为两类。一类是从调整数据集中的类分布角度出发，经典方法包括随机过采样(random oversampling)法、随机欠采样(random undersampling)法和 SMOTE(synthetic minority oversampling technique)法。另一类从修改学习算法角度出发。即：通过修改算法的训练过程，使得学习算法可以针对少数类取得更好的预测精度，典型方法包括代价敏感学习 fcost. sensitive learning)方法。代价敏感学习方法的目标并不是简单地使得预测错误数最少，而是使得错误分类开销总和最小，因此更符合实际需求。

在基于 SVM 构建缺陷预测模型的研究中，更多的是代价敏感学习，任胜兵等人<sup>[17]</sup>在代价敏感 SVM 的基础上构建了 CSS-SVM(ClusterCostSensitive-SVM)软件缺陷预测模型。该模型首先通过聚类找到每类样本的中心点，根据样本到其中心点的距离定义每个样本的类别置信度，给每个样本不同的误分代价系数，通过给每个样本分配不同的误分代价来建立 SVM 模型，提升分类性能。

除此之外，程启元人<sup>[18]</sup>提出了使用模糊支持向量机来构建软件缺陷预测模型。SVM 的缺点在于并没有对特异点(outliner)进行处理。因此 Chun-FuLin 和 Sheng-DeWang 提出了模糊支持向量机算法<sup>[19]</sup>。在模糊支持向量机中，给每个数据点增加模糊隶属度(fuzzy member ship value)，新的数据点的形式为  $(y_i, x_i, s_i), \dots, (y_l, x_l, s_l)$ ， $i = 1, 2, \dots, l$ ， $x_i$  和  $y_i$  的定义与之前相同， $s_i$  是模糊隶属度，且  $\sigma \leq s_i \leq 1$ ， $\sigma$  大于 0 且足够小。在软件缺陷预测中，模糊隶属度需要考虑除了特异点的影响外，还要考虑类不平衡的影响，通过定义参数  $r^+$  和  $r^-$ ，且  $r^+ > r^-$ 。正类的模糊隶属度属于区间  $[0, r^+]$ ，设置  $[0, r^-]$  与  $[0, r^+]$  的模糊隶属度同时考虑了特异点问题和类不平衡问题。

## 2.4 SVM 的参数优化选择

### 2.4.1 遗传算法

遗传算法(Genetic Algorithm, GA)<sup>[20]</sup>是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。

其主要特点是直接对结构对象进行操作，不存在求导和函数连续性的限定；具有内在的隐并行性和更好的全局寻优能力；采用概率化的寻优方法，不需要确定的规则就能自动获取和指导优化的搜索空间，自适应地调整搜索方向。



遗传算法以一种群体中的所有个体为对象，并利用随机化技术指导对一个被编码的参数空间进行高效搜索。其中，选择、交叉和变异构成了遗传算法的遗传操作；参数编码、初始群体的设定、适应度函数的设计、遗传操作设计、控制参数设定五个要素组成了遗传算法的核心内容。在 SVM 构建缺陷预测模型的实验中利用遗传算法对模型参数的  $C$  与  $\alpha$  占了超过 50%，其中任胜兵等人<sup>[17]</sup>在代价敏感学习中同时对  $C^+$  (正类的惩罚参数)、 $C^-$  (负类的惩罚参数)、 $\sigma$  (RBF 的核函数参数)和特征的选择状态( $f$ )进行优化。

#### 2.4.2 粒子群算法

粒子群算法(PSO)<sup>[21, 22]</sup>是模拟鸟群捕食行为提出的一种全局搜索算法。PSO 算法先初始化一组随机种群，又叫粒子群，每个粒子按初始化得到的速度穿过解空间，其速度是该粒子和种群其他粒子的历史行为(速度、位置和适应度)的函数，在每一次迭代中都会发生变化。其中每个粒子的位置属性对应一个潜在的解，通过计算当前粒子的适应度来评价当前解的优劣，如果未能符合解的要求则进行粒子速度和位置的更新，并进入下一个迭代过程直到找到符合要求的最优解或者到达最大迭代次数。朱朝阳等人<sup>[23]</sup>与张飞<sup>[24]</sup>在以 SVM 构建缺陷预测模型的实验中使用了这一方法来寻找 SVM 的  $C$  与  $\alpha$  的参数最优值。

#### 2.4.3 蚁群优化算法

姜慧研等人<sup>[25]</sup>使用了蚁群优化算法(ACO)的寻优能力提出了基于 ACO-SVM 的软件缺陷预测模型。基本蚁群算法引进了正反馈和并行机制，较好地解决了旅行商 (Traveling Salesman Problem, TSP)问题，但是收敛速度较慢。1997 年 Dorigo 等人<sup>[26]</sup>在基本蚁群算法的基础上提出了蚁群系统，证明了该算法优于模拟退火、进化计算等仿生算法。ACO 使用模型的准确率(Accuracy)来取代路径长度来更新信息素，蚂蚁在  $10 \times 10$  的图上走完一条路径后，根据小数点的位置可以计算图上的有效位与有效值，根据最优路径就能计算出离散化了的模型的最优参数  $C$  与  $\alpha$ 。

### 3 SVM 使用总结

#### 3.1 SVM 的优劣势

SVM 是平衡数据下非常优秀的二分类器，在解决小样本、非线性及高维模式识别中表现出许多特有的优势。

但是 SVM 本身对参数  $C$  与  $\alpha$  非常敏感，而在软件缺陷预测模型构建中，参数的选取没有一定的理论指导，所以就依赖一定的参数寻优算法对 SVM 做出优化。

王涛等人<sup>[27]</sup>提出了 SVM-DP 模型，这一模型的特点在于结合软件缺陷预测挖掘历史数据仓库的这一特点将每一次模型预测正确的结构最为历史仓库加入到数据集中，实现可迭代增强的预测模型。这也指出了 SVM 本身的灵活性。

### 3.2 与其他机器学习的比较

目前机器学习用于构建软件缺陷模型的研究数不胜数，主要包括马尔可夫模型[28]、分类回归树模型[29]、人工神经网络模型[30]、线性判别分析模型[31]、时间序列分析模型、分类树模型[32]。然而它们都或多或少的存在一些问题。马尔可夫模型需要对软件内部错误及失效过程的特性做出很多假设；分类回归树模型的泛化能力差；人工神经网络模型的网络结构选择尚无统一完整的理论指导。

在其它的比较实验中，可以看到传统的 SVM 与 LR 回归、K-NN 临近算法、随机森林、朴素贝叶斯等机器学习算法在构建软件缺陷预测模型上，对某些数据集各类算法差异很小，而在一些数据集上则差异很大。究其原因，不在于构建模型的算法本身，而是在度量元的选取与数据集预处理上。然后在这一点上 SVM 有着很好的适应性。

## 4 总结与展望

本次文献综述聚焦在基于 SVM 构建软件缺陷预测模型，并使用 NASA 数据集进行实验的论文上，而软件缺陷预测模型本身这一论题是非常广泛的，不仅使用不同的机器学习算法构建模型种类繁多，如有监督的学习方法、无监督的学习方法、半监督的学习方法、迁移学习等等，还存在基于缓存构建模型这一思路；不仅存在静态缺陷预测模型，还有考量时间与缺陷关系的动态缺陷预测模型；在度量元的选取上除了基于代码的静态属性的，还有基于软件开发过程的度量、基于开发人员的度量、基于程序模块间依赖分析的度量、基于项目团队组织架构的度量<sup>[4]</sup>，不同的度量元选取将对模型的建构产生巨大的影响；度量元的选取关系到数据集本身，数据集除了存在噪声、维度灾难、类不平衡问题外，还有可能存在不具备历史仓库的跨项目软件缺陷预测问题。上述阐述表明了本次综述只是展现软件缺陷预测领域的冰山一角，但也希望梳理出一个简略的研究脉络，如陈翔等人<sup>[4]</sup>提出的该领域的研究还存在数据集共享困难、研究对缺陷预测模型的应用关注得较少等问题。推动软件缺陷预测的发展任重而道远。

## 参考文献

- [1] 聂林波, 刘孟仁. 软件缺陷分类的研究. 计算机应用研究. 1001- 3695 (2004) 06- 0084- 03
- [2] Wang Q, Wu SJ, Li MS. Software defect prediction. Ruan Jian Xue Bao / Journal of Software, 2008, 19(7): 1565—1580 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/1565.htm>
- [3] Hall T, Beecham S, Bowes D, Gray D, Counsell S. A systematic literature review on fault prediction performance in software engineering. IEEE Trails. on Software Engineering, 2012, 38(6): 1276—1304 [doi: 10.1109/TSE.2011.103]
- [4] 陈翔, 顾庆, 刘望舒, 刘树龙, 倪超. 静态软件缺陷预测方法研究. 软件学报, 2016, 27(1): 1—25. <http://www.jos.org.cn/1000-9825/923.htm>
- [5] Cortes C, Vapnik V. Support-vector networks [J]. Machine Learning, 1995, 20(3): 273-297.
- [6] 姜慧研, 宗茂, 刘相莹. 基于 ACO-SVM 的软件缺陷预测模型. 计算机学报. 10.3724/SP.J.1016.2011.01148
- [7] KAYNAR D, LYNCH N, SEGALA R, et al. The theory of timed I/O automata [M]. [S.l.]: Morgan & Claypool Publishers, 2005.
- [8] 吴晓萍, 赵学靖, 乔辉, 刘东梅, 王志. 基于 LASSO-SVM 的软件缺陷预测模型研究. 计算机应用研究. doi: 10.3969/j.issn.1001-3695.2013.09.047
- [9] GUREVICH Y. Sequential abstract state machines capture sequential algorithms [J]. ACM Trans on Computational Logic, 2000, 1(1): 77-111.
- [10] Pearson K. Principal components analysis [J]. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1901, 6(2): 559.
- [11] 甘露, 臧冽, 李航. 基于 DA-SVM 的软件缺陷预测模型. 计算机与现代化 10.3969/j.issn.1006-2475.2017.02.007
- [12] Fisher R A. The use of multiple measurements in taxonomic problems [J]. Annals of Eugenics, 1936, 7(2): 179-188.
- [13] Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders [C] // Proceedings of the 25th ACM International Conference on Machine Learning. 2008: 1096-1103.
- [14] Wang Hao, Shi Xingjian, Yeung Dit-yan. Relational stacked denoising autoencoder for tag recommendation [C] // Proceedings of the 29th AAAI Conference on Artificial Intelligence. 2015: 3052-

3058.

[15] 张强. 局部线性嵌入算法的改进及其在人脸识别中的应用 [D] . 重庆: 重庆理工大学, 2017.

[16] 王玉红, 范菁, 雷敏, 孙汇中. 基于 NPE-SVM 的软件缺陷预测模型 10. 16836/j. cnki. jcuit. 2018. 03. 011

[17] 任胜兵, 廖湘荡. 基于代价敏感支持向量机的软件缺陷预测研究. 计算机工程与科学. 10.3969/j. issn. 1007-130X. 2018. 10. 010

[18] 程元启, 姚淑珍, 谭火彬, 李丹丹. 基于模糊支持向量机的软件缺陷预测技术. 计算机工程与设 10. 16208/j. issn1000-7024. 2018. 09. 010

[19] Mitchell M. An introduction to genetic algorithms [M]. Cambridge: MIT Press, 1998

[20] Lin CF, Wang SD. Fuzzy support vector machines [J]. IEEE Transactions on Neural Networks, 2002, 13 (2): 464-471.

[21] 郭平. 软件可靠性工程中的计算智能方法[M]. 北京: 科学出版社, 2012.

[22] KENNEDY J. Particle swarm optimization [M] // Encyclopedia of Machine Learning. Berlin: Springer, 2011: 760 — 766

[23] 朱朝阳, 陈相舟, 王志宏, 张信明. 基于粒子群优化 SVM 的面向对象软件缺陷预测模型. 计算机应用. ISSN 1001-9081

[24] ZHANG Fei. Software defect prediction based on improved PSO-ISVM algorithm. Computer Engineering and Applications, 2016, 52(11): 17-21.

[25] 姜慧研, 宗茂, 刘相莹. 基于 ACO-SVM 的软件缺陷预测模型的研究. 计算机学报. 10.3724/SP. J. 1016 . 2011. 01148

[26] M. Dorigo, L.M. Gambardella, Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem, IEEE Transactions on Evolutionary Computation, 1997, 1(1), 53-66.

[27] 王涛, 李伟华, 刘尊, 史豪斌. 基于支持向量机的软件缺陷预测模型.

[28] Lyu Michael R. Handbook of Software Reliability Engineering .New York :IEEE Computer Society Press and McGraw Hill Book Company , 1996

[29] Khoshgoftar Taghi M, Seliya Naeen .Tree-based software quality estimation models for fault prediction// Proceedings of the 8th International Symposium on Software Metrics. Washington , DC , USA , 2002 :123-128

[30] Stich Timothy Janes, Spoerre Julie K , Velasco Tomas .The application of artificial neural networks to monitoring and control of an induction hardening process. Journal of Industrial Technology , 2000 , 16(1):1-11

[31] Ohlsson Niclas, Alberg Hans .Predicting fault-prone software modules in telephone switches. IEEE

Transactions on Software Engineering , 1996 , 22(12):886-894

[32] Khoshgoftaar TaghiM, Seliya Naeem .Software quantity classification modeling using the SP RINT decision tree algorithm// Proceeding s of the 14th IEEE International Conference on Tools with Artificial Intelligence .Washington , DC ,US A , 2002 :365-367