基于支持向量机的软件缺陷预测模型——王涛

摘要

软件缺陷预测在软件系统开发的各个阶段发挥着极为重要的作用。利用机器学习的相关方

法建立更好的预测模型已经被广泛研究。文章分析了支持向量机 SVM 作为二值分类模型应用到软

件缺陷预测中的实现方法，构造了基于 SVM 的可迭代增强的缺陷预测模型 SVM-DP。在 13 个基准数

据集上开展比较实验，定量地分析了应用各种核函数对 SVM-DP 模型性能的影响。实验结果显示，应

用线性内积核函数的 SVM-DP 具有最优的预测性能。同时，在与 J48 的比较实验中，最高超过 J48 预

测模型 20% 的性能进一步证明了 SVM-DP 模型应用于软件缺陷预测的有效性。

引言

软件测试是一项极其耗费成本的工作。通常认

为缺陷在软件系统中的分布是不均匀的，20% 的模

块包含超过 80% 的缺陷。因此， 软件缺陷预测的目

的，是在软件系统开发生命周期的各个阶段鉴别可

能存在缺陷的模块， 为测试人员提供缺陷分布等信

息，以此指导软件质量确保工作，这一流程已经成为

事实上的工业标准［1］ 。机器学习的方法将缺陷预

测看作二值分类问题。模型构造完成后， 对分析目

标( 可能是文件、模块、方法等) 进行分类预测， 从而

在一定条件下做出判决。B． Turhan 等人［2］ 提出了

一种多变量方法与贝叶斯理论结合的预测模型， 并

且强调了利用特征选取技术对静态代码属性进行筛

选的重要性。实验结果显示， 其预测模型有很好的

性能。T． Gyimothy 等人［3］ 分析并使用了逻辑回归、

线性回归、决策树及神经网络等四种方法构造预测

模型，并且试图找出最优的代码静态属性。在二值

分类方面支持向量机 SVM 一直被广泛采用［4］ ， SVM

是建立在统计学习理论的 VC 维理论和结构风险最

小原理基础上的， 根据有限的样本信息在模型的复

杂性和学习能力之间寻求平衡， 以期获得最好的泛

化能力［5］ 。本文分析了利用 SVM 构造缺陷预测模

型的理论方法， 构造了基于 SVM 的预测模型 SVMDP。分析了利用不同核函数对 SVM-DP 性能的影

响，并且与已知的其它方法进行了比较。

软件缺陷预测技术∗ 王青

摘要

软件缺陷预测技术从 20 世纪 70 年代发展至今,一直是软件工程领域最活跃的内容之一,在分析软件  
质量、平衡软件成本方面起着重要的作用.研究和讨论了软件缺陷预测技术的起源、发展和当前所面临的挑战,  
对主流的缺陷预测技术进行了分类讨论和比较,并对典型的软件缺陷的分布模型给出了案例研究.

引言

所谓缺陷(defect),目前为止,学术界、产业界有很多相关的术语和定义,比如故障、缺陷、 bug、错误、失误、  
失效、失败等.根据 ISO 9000 对缺陷的定义“未满足与预期或者规定用途有关的要求”,缺陷是软件中已经存在的一个部分,可以通过修改软件而消除.另一个重要的概念是失效(failure).当系统或者软件运行时,出现不正确的输出,则称为失效.严格地说,失效可能由软件缺陷引起,也可能由其他诸如人为因素、硬件故障等引起.如果我们每观察到一次失效,就可以发现一个或者多个软件缺陷,那么纠正这些缺陷就可以避免类似失效的重复出现.软件已经成为影响国民经济、军事、政治乃至社会生活的重要因素.高可靠和复杂的软件系统非常依赖于其采用的软件的可靠性.软件的缺陷是导致相关系统出错、失效、崩溃甚至机毁人亡的潜在根源.例如,1996 年6 月,欧洲“阿丽亚娜”号航天飞机因导航系统的计算机软件出现故障,致使航天飞机坠毁,造成了数亿美元的巨大损失;2005 年 3 月 31 日,欧空局的 SMART-1 月球探测器和 NASA 的“雨燕”太空望远镜的使用状态均因软件故障而受到了很大影响;2005 年 4 月,软件失灵、继而导航失误,导致耗资 1.1 亿美元的 NASA 自主交会任务DART 实验失败.

然而软件技术发展至今,任何检验、验证手段都不可能发现并排除所有的缺陷,软件作为一种无形的产物,

虽然不会磨损用坏,却随时可能因为我们不易查知的原因出现故障甚至失效.

CeBASE(Center for Empirically-Based Software Engineering)[1]是美国国家科学基金支持的经验软件工程中心,拥有世界范围著名的专家和学术机构的合作组织.2002 年,CeBASE组织了几次著名的网上研讨,并在加拿大渥太华的第 8 届度量大会(METRICS 2002)上组办了名为“What we have learned about fighting defects”的研讨会.会后根据大家研讨的结果,整理出了软件缺陷所带来的影响及缺陷分布和检测问题:软件发布后发现和修复缺陷的成本大幅增加(例如,对于一个严重问题,其成本较之需求和设计阶段增加了 100 倍);软件项目返工工作量比例居高不下,但随着过程成熟度的提高而减少;软件工程中返工工作量与缺陷、软件失效与缺陷,以及缺陷在软件模块中的分布都是符合 2-8 原则的;同行评审、评审准备工作等有助于发现更多的缺陷;有经验的开发人员可以显著减少缺陷的引入率.上述讨论结果表明,缺陷对软件质量甚至对软件经济有重要影响;同时也说明缺陷分布问题的复杂性和差异性,以及现有的缺陷预测技术在解决实际问题上的不足等.事实上,从第一个软件诞生,就伴随出现软件缺陷的检测和预测技术.检测技术在于发现缺陷,而预测技术则在于预测还未发现的缺陷.20 世纪 70 年代,出现了利用统计学习技术,根据历史数据以及已经发现的缺陷等软件度量数据预测软件系统的缺陷数目及类型.缺陷预测技术的目的在于统计计算软件系统的缺陷数、没有发现但还可能存在的缺陷数,以决定系统是否可以交付使用.缺陷预测技术为软件质量的提高和保证起着非常重要的作用,同时,也促进了软件工程技术向前大大地发展了一步.

纵观软件缺陷预测技术的发展,从 20 世纪 70 年代起,软件缺陷预测技术大体上分为静态和动态两种缺陷

预测技术,如图 1 所示.静态预测技术,主要是指基于缺陷相关的度量数据,对缺陷的数量或者分布进行预测的技术;而动态技术则是基于缺陷或者失效产生的时间,对系统缺陷随时间的分布进行预测的技术

SOFTWARE DEFECT PREDICTION MODEL BASED ON LLE AND SVM

摘要

Software defect prediction strives to improve software security by helping testers locate the software defects accurately. The data redundancy caused by the overmuch attributes in defects data set will make the prediction accuracy decrease. A model based on locally linear embedding and support vector machine (LLE-SVM) is proposed to solve this problem in this paper. The SVM is used as the basic classifier in the model. And the LLE algorithm is used to solve data redundancy due to its ability of maintaining local geometry. The parameters in SVM are optimized by the method of ten-fold cross validation and grid  
search. The comparison between LLE-SVM model and SVM model was experimentally verified on the same NASA defect data set. The results indicate that the proposal LLE-SVM model performs better than SVM model, and it is available to avoid the accuracy decrease caused by the data redundancy.

引言

With the high-speed progress of computer science and  
software technology, the process of informationization has  
been rapidly developed. Software systems are playing an  
increasingly important role in the socio-economic and  
political life. Software defects are increasing with software function variety and software structure complexity.  
Software defects may cause software failure, which do  
serious harm to software security [1]. So it is necessary to  
study software defect prediction technology in this background. About 80% of the defects come from 20% of the  
modules, and more than half of the modules are defects  
free [2]. It has very important significance that software  
testers find out those modules which contain defects and  
this is the goal of defect prediction. Software defect prediction technology can be classified into two kinds. They  
are static defect prediction technology and dynamic defect  
prediction technology. The static technology is based on  
the metric data of defects, which is used to predict the  
quantity or distribution of defects; and the dynamic technology is based on the generate time of defects or failures,  
which is used to predict the defects over time [1].  
There are many outstanding techniques appeared in the  
dynamic defect prediction technology. The software defect  
prediction model based on grey prediction theory (GPT)  
was mentioned in [3], the idea of the method was: the  
number of software defects was a grey value changed  
within a certain range, and the random process was a time  
related gray process changed in a certain range. The  
method using GPT predicted software defects which  
might appear in the iteration according to the historical  
statistics of software defect number [3]. An improved  
model of software defect prediction based on Rayleigh  
model was mentioned in [4]. The model corrected the unreasonable hypothesis of Rayleigh model considering the effect of removing failure on defect prediction  
result [4].  
There are a lot of static defect prediction technologies  
such as support vector machine (SVM) [5], neural network (NN) [6], and Bayesian network (BN) [7]. The advanced software defect prediction model based on SVM  
was mentioned in [8]. The defect prediction was regarded  
as a binary classification problem, and SVM model was  
used to predict the current version of the project. New  
defect information found was added to the history data  
and then new incremental data sets would be produced. A  
kind of improved prediction model would be constructed  
based on the incremental data sets in the new project life  
cycle [8]. The technology involved in this paper is static  
prediction techniques.  
Software defect prediction is a kind of technology which  
can reveal the possibility whether a software system contains defects by analyzing the metric data of software.  
More and more attributes are being introduced to metric  
software with the developing of technology. Facing the  
growing number of attributes, one of the problems which  
must be solved in software defect prediction area is how  
to deal with the data redundancy. This problem may lead  
to higher cost and lower prediction accuracy. For this reason a software defect prediction model based on locally  
linear embedding and support vector machine is proposed  
in this paper. Locally linear embedding algorithm is used  
to reduce dimensionalities and maintain the local geometry of the metric data set, and support vector machine is  
proposed to classify the data set of software defects.

基于 DA-SVM 的软件缺陷预测模型 甘 露

摘要

特征提取是软件缺陷预测技术研究中的重要环节，而现有的特征提取方法无法准确获得特征之间的非线性依赖关  
系，因而无法提高软件缺陷预测的准确性。针对该问题， 本文构建基于降噪编码器和支持向量机的软件缺陷预测模型  
( Denoising Autoencoder Support Vector Machine，DA-SVM) 。首先利用降噪编码器进行特征提取， 然后将提取的特征作为  
支持向量机的输入向量，最后再进行软件缺陷预测。实验结果表明， DA-SVM 提高了软件缺陷预测的准确度， 同时降低  
了历史数据中的噪声，增强了软件预测模型的鲁棒性。

引言

近 30 年来， 软件缺陷预测［1］ 一直是软件测试领

域中重要的研究课题。在软件开发周期中，利用软件

缺陷预测技术预测当前开发模块的缺陷情况，从而决

定是否可以进入下一阶段的开发。软件缺陷预测技

术不仅可以调节软件的开发过程，在测试阶段还可以

将有限的资源集中在缺陷较多的模块， 合理利用资

源，提高软件的安全性能，因此，软件缺陷预测技术的

研究具有深远意义。

软件属性对于构建软件缺陷预测模型起着重要

的作用。软件缺陷模型的性能和有效性取决于软件

模块的特征属性，利用这些特征属性可以快速有效地

对软件模块进行预测。用来描述软件的属性有很多，

由于某些属性不包含重要的信息或存在不相关的冗

余信息，从而会影响软件缺陷预测模型的准确性。如

果属性选择不当，构建的缺陷预测模型的预测性能则

会降低。因此， 为了提高缺陷预测模型的效率和性

能，采用合适的特征选择［2-3］ 算法对于构建软件缺陷

预测模型尤为重要。

基于 LASSO-SVM 的软件缺陷预测模型研究吴晓萍

针对当前大多数软件缺陷预测模型预测准确率较差的问题，提出了结合最小绝对值压缩和选择方法与  
支持向量机算法的软件缺陷预测模型。首先利用最小绝对值压缩与选择方法的特征选择能力降低了原始数据  
集的维度，去除了与软件缺陷预测不相关的数据集; 然后利用交叉验证算法的参数寻优能力找到支持向量机的  
最优相关参数; 最后运用支持向量机的非线性运算能力完成了软件缺陷预测。仿真实验结果表明，所提出的缺  
陷预测模型与传统的缺陷预测模型相比具有较高的预测准确率，且预测速度更快。

引言

目前统计方法、机器学习和混合统计与机器学习三种方法  
已经被广泛地应用于软件缺陷预测领域。在这三种方法中，混  
合统计与机器学习方法已经被证明可以更好地实现软件缺陷  
预测［1］ 。  
在软件缺陷预测领域， 由于软件模块的复杂性、软件复杂  
性度量和易出错模块类别的对应关系很难确定，研究人员提出  
了采用大量的原始复杂性度量属性数据来进行软件缺陷预测。  
然而不加选择地引入所有的复杂性度量属性来进行缺陷预测，  
往往不能取得良好的效果［2］ 。这是因为过多地引入复杂性度  
量属性会导致“维数灾难”和“组合爆炸”［3］ ， 使得计算量空前  
增大，估计和预测精度也会下降; 此外， 在一些情况下， 获得某  
些软件复杂性度量属性数据代价昂贵，如果这些数据本身对预  
测结果的影响微乎其微，那么势必会造成缺陷数据的收集和模  
型应用的费用不必要地增大。  
针对上述问题，混合统计与机器学习方法的核心是先利用  
统计方法 的 思 想 对 要 进 行 缺 陷 预 测 的 数 据 集 进 行 特 征选  
择［4，5］ 处理，达到降低数据集维数的目的; 之后用机器学习方  
法对处理后的数据集进行缺陷预测，获得所需要的预测精度。  
本文基于上述思想， 提出了一种基于 LASSO-SVM 的软件  
缺陷预测方法。其主要工作如下: 分析了 LASSO 方法的核心  
思想？？？，以及如何使用它的特征选择能力来得到精简的复杂性度  
量属性子集; 分析了 SVM 机器学习算法的核心思想， 并且使用  
交叉验证算法对 SVM 的相关参数进行优化， 获得 SVM 优化模  
型; 将使用 LASSO 方法得到的属性精简子集输入到基于 SVM  
机器学习算法的软件缺陷预测模型中，并与其他传统软件缺陷  
预测方法进行对比，验证了该方法的有效性和预测准确率。

基于 NPE-SVM 的软件缺陷预测模型王玉红

针对软件缺陷预测中数据集的类不均衡、高维、小采样以及非线性降维问题，提出基于领域保持嵌入支持向量  
机的软件缺陷预测模型。模型采用 NPE 算法对数据集进行降维处理，通过将 NPE 算法中奇异的广义特征计算转化为两  
个特征分解问题，得到了更准确的稳健解，有效规避了属性约减后导致的预测精度下降问题。选用支持向量机作为基础  
分类器，仿真实验结果表明，与其他方法相比，预测模型的查全率及 F-measure 值指标显著提高了2%～4%

软件缺陷预测，即识别软件系统中含缺陷的高风  
险模块，合理分配软件测试资源［1］ 。软件缺陷预测已  
成为软件系统安全中的一项重要的技术，然而，实际应  
用中的软件缺陷测试集都是高维的， 难以识别。且研  
究学者 Boehm 指出软件缺陷分布符合 2 － 8 原则， 即  
80%的缺陷包含在20%的软件模块中， 表明软件缺陷  
数据集也存在严重的类不均衡问题。因此， 降低软件  
缺陷测试集维度、获取软件缺陷特征在软件缺陷预测  
中显得尤为重要。  
目前，软件缺陷预测领域的研究人员采用最多的  
就是机器学习的方法， 如聚类分析、人工神经网络、线  
性判别分析等，但都存在一些问题: 聚类分析可重复性  
差且成本较高; 人工神经网络所需样本较多; 线性判别  
分析预测精度比较低［2－4］ 。针对上述问题， 采用领域  
保持嵌入( neighborhood preserving embedding， NPE) 算  
法，通过转化 NPE 算法中奇异的广义特征计算问题改  
进 NPE 算法，发现低维空间中数据的真实结构。由于  
支持向量机( support vector machine， SVM) 在处理小样  
本、非线性及高维模式识别问题中具有独特的优势，因  
此选用 SVM 进行分类预测。解决软件缺陷数据集的  
类不均衡及高维、小采样问题。

基于代价敏感支持向量机的软件缺陷预测研究 任胜兵

软件缺陷预测是典型的非平衡学习问题。基于 ＣＳ－ＳＶＭ 和聚类算法改进代价敏感支持向量  
机（ＳＶＭ）算法，提出了 ＣＣＳ－ＳＶＭ 软件缺陷预测模型。在 ＣＣＳ－ＳＶＭ 预测模型中，将ＳＶＭ 与类别误分代  
价结合起来，以非平衡数据评价指标作为目标函数，优化错分代价因子，提升少数类样本的识别率。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，给每个样本分配不  
同的误分代价系数，并把样本的置信度引入到代价敏感ＳＶＭ 优化问题中，提高算法鲁棒性，提升ＳＶＭ 分  
类性能。此外，为了提高模型的泛化能力，使用遗传算法优化特征选择和模型参数。通过美国航空航天局  
ＮＡＳＡ ＭＤＰ数据集实验表明，本文方法的Ｇ－ｍｅａｎ 和Ｆ－ｍｅａｓｕｒｅ 模型评价值有明显的提升。

引言

软件缺陷是计算机软件或程序中存在的某种  
破坏正常运行能力的问题、错误，或者隐藏的功能  
缺陷。缺陷的存在会导致软件产品在某种程度上  
不能满足用户的需要，被认为是影响软件质量最主  
要的原因。在程序发布后修正软件开发阶段的所  
有缺陷，维护成本可能非常高。因此，在程序发布  
之前尽早检测出所有可能的错误非常重要［１，２］。  
目前，有大量研究根据程序静态属性用机器学  
习来建立软件缺陷预测模型。为了得到高性能预  
测模型，主要研究工作集中在如何选择有效的机器  
学习算法。Ｍｅｎｚｉｅｓ等［３］用朴素贝叶斯 ＮＢ（Ｎａｉｖｅ  
Ｂａｙｅｓｉａｎ）构造分类器预测软件缺陷，并利用信息  
增益 对 特 征 选 择 进 行 研 究。Ｓｈｕａｉ等［４］提 出 了  
ＧＡ－ＣＳＳＶＭ（ＧｅｎｅｔｉｃＡｌｇｏｒｉｔｈｍ ＣｏｓｔＳｅｎｓｅｓｉｔｉｖｅ  
ＳｕｐｐｏｒｔＶｅｃｔｏｒＭａｃｈｉｎｅ）预测模型，使用遗传算法  
调整 ＣＳ－ＳＶＭ（Ｃｏｓｔ－ＳｅｎｓｉｔｉｖｅＳＶＭ）的惩罚参数，  
实验表明该方法基于 ＡＵＣ 评价 函 数 取 得 了 比 较  
好的结果。Ｓｅｌｉｙａ等［５］提出了“ＲＢＢａｇ”预测模型，  
用朴素贝叶斯和 Ｃ４．５结合数据抽样构建ｂａｇｇｉｎｇ  
预测模型，实验结果表明该方法比不建立 ｂａｇｇｉｎｇ  
和数据抽样的模型性能更好。然而，这些研究在处  
理数据不平衡问题方面都存在一些问题。软件缺  
陷数据是典型的非平衡数据，因为大部分缺陷都集  
中在少量的模块。比如 ＮＡＳＡ 的 ＰＣ１ 项目包含  
７０５个模块，其中６１个模块有缺陷，占比所有模块  
的８．６５％［６］。所以，虽然他们的实验结果表明这  
些算法能够得到不错的预测性能，但由于他们的实  
验主要是基于准确率来评价模型的，这在软件缺陷  
预测问 题 上 是 不 合 理 的。 假 设 有 缺 陷 模 块 占 比  
１０％，把所有的有缺陷模块都预测为无缺陷模块，  
准确率仍然是９０％！  
Ｗａｎｇ等［７］比较了已 有的 软 件 缺 陷 预 测 方 法  
和非平衡数据分类方法，包含 欠 采 样 和 过 采 样 技  
术、阈值 移 动、集 成 学 习 等 算 法，结 果 表 明 Ａｄａ－  
Ｂｏｏｓｔ．ＮＣ 优 于 朴 素 贝 叶 斯和随 机 森林 ＲＦ（Ｒａｎ－  
ｄｏｍＦｏｒｅｓｔ）等方法。最近，支持向量机在软件缺  
陷预测领域和非平衡学习上都表现出了不错的预  
测性能［８－１１］。Ｅｌｉｓｈ 等［１２］比较了 ＳＶＭ 和八种传  
统的统 计 和 学 习 算 法，比 如 决 策 树 ＤＴ（Ｄｅｃｉｓｉｏｎ  
Ｔｒｅｅ）、朴素贝叶斯、ＡｄａＢｏｏｓｔ等，实 验 结 果 显 示  
ＳＶＭ 在多数数据集上优于其它算法。通过调研分  
析发现，ＳＶＭ 在软件缺陷预测上表现出了不错的  
预测效果，而且代价敏感是处理非平衡数据常用的  
方法。在本文中，我们结合 ＳＶＭ 和代价敏感来建  
立软件缺陷预测模型。  
ＳＶＭ 是平衡数据下非常优 秀 的 二 分 类 器，但  
标准的ＳＶＭ 并不能很好地处理非平衡数据［１３，１４］。  
为了使ＳＶＭ 能处理非平衡问题，常见的做法主要  
有两种：一是数据预处理［１５，１６］，虽然这种方法被广  
泛用作简单的解决不平衡问题的解决方案，但大多  
数数据预处理方法由于随机抽样而具有共同的限  
制。预处理后可能得到完全不同于原来的数据集，  
从而导致严重偏差的数据分布。另一种方法是算  
法层面的改进，代价敏感学习是最常见的解决方  
法［４，１７，１８］，已有的研究工作都是通过调整对负类和  
正类的惩罚因子，使分类面向少数类偏移。但是，  
因为ＳＶＭ 训练过程中对所有的训练样本是平等  
对待的，这会使ＳＶＭ 分类器对噪音和孤立点数据  
样本极为敏感，进而导致了过拟合的情况发生，已  
有的研究方法并不能很好地解决这个问题。我们  
在代 价 敏 感 ＳＶＭ 的 基 础 上 构 建 了 ＣＣＳ－ＳＶＭ  
（ＣｌｕｓｔｅｒＣｏｓｔＣｅｎｓｉｔｉｖｅ－ＳＶＭ）软 件 缺 陷 预 测 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。然后用遗传算法优化模型参数和特征选择。  
本文的实验是基于 ＮＡＳＡ ＭＤＰ（ＭｅｔｒｉｃｓＤａ－  
ｔａＰｒｏｇｒａｍ）［１９］数据集，这是软件缺陷预测研究使  
用最广泛的数据集［３－５，７，１２，２０］。本文将 ＣＣＳ－ＳＶＭ  
与 ＡｄａＢｏｏｓｔ、ＡｄａＢｏｏｓｔ．ＮＣ、朴素贝叶斯（ＮＢ）、随  
机森林（ＲＦ）、ｋ 近邻（ｋ－ＮＮ）、决策树（ＤＴ）进行了  
比较，评价指标包括 Ｆ－ｍｅａｓｕｒｅ 值和Ｇ－ｍｅａｎ 值。  
实验结果表明，本文提出的算法在Ｆ－ｍｅａｓｕｒｅ 值和  
Ｇ－ｍｅａｎ 值 上 都 有 明 显 的 提 升，在 Ｇ－ｍｅａｎ 值上比  
其余算 法 中 的 最 优 算 法 平 均 高 出 ９ 个 点，在 Ｆ－  
ｍｅａｓｕｒｅ 值上平均高出５个点。

基于粒子群优化 SVM 的面向对象软件缺陷预测模型朱朝阳

针对电力信息系统软件安全问题，分析了软件缺陷预测方法在面向对象软件开发过程中的重要性， 并提  
出了一种与面向对象软件特征相应的基于粒子群优化的支持向量机软件预测模型。模型主要包括三部分: 首先是对  
原数据进行归一化和特征选择的预处理模块; 然后是以预测准确度作为适应度评价的动态惯性权重粒子群优化支持  
向量机( SVM) 参数的模块; 最后则是利用第二个模块中的最优参数进行降维数据预测的 SVM 分类模块。实验结果表  
明，该模型在四个数据集合上的准确率高于对比模型 8． 2% ～ 12． 2% ， 在精准度、查全率和 F 值上平均高出 9． 9% ，  
5． 6% 和 7． 7% ，说明了该模型的有效性。

电力信息系统软件在长期发展过程中， 主要是以面向对  
象设计技术来开发的。从本质上讲， 面向对象系统设计是寻  
求电力软件结构和电力软件功能模型解决方案的过程。因而  
为了功能的全面性、可靠性，面向对象电力系统软件逐步从功  
能单一向全面发展，软件本身结构和模型更加复杂，对象规模  
也更加庞大，导致整个电力信息系统的软件所面临的安全问  
题也愈加严重，软件缺陷预测能够在软件开发早期预测待测  
的模块、对象是否有出错倾向从而提出相应的解决方案，包括  
人力资金等资源的分配， 开发进度的控制和安排。因此软件  
开发过程中尽早地发现软件缺陷、漏洞并尽快解决，才能为国  
民生产、市场正常运行提供必要的保障，也是使得电网公司降  
低日后测试代价和周期、提高软件质量的重要途径［1］ 。  
软件缺陷预测技术可分为静态和动态两种缺陷预测技  
术。动态缺陷预测技术关注软件整个生命周期或者测试阶段  
缺陷分布及数量随时间的变化规律， 并据此预测软件未来缺  
陷分布; 而更加常用的静态预测技术则更关注软件关于缺陷  
的度量指标，结合各个不同的缺陷相关属性的度量进行缺陷  
预测，在软件分析和设计阶段及开发的早期阶段均可提供相  
应的预测功能。现有的静态缺陷预测技术基本上都是基于不  
同的机器学习算法提出的， 例如决策树、随机森林、朴素贝叶  
斯、BP 神经网络以及人工免疫系统等分类算法， 根据某种描  
述软件特性的软件度量获取模块属性， 然后对其进行分类  
( 有缺陷或无缺陷) ［2］ 。这些方法均拥有一定程度的缺陷预  
测能力，但又或多或少地隐含一些问题， 例如决策树过度拟  
合，忽略特征属性之间相关性的问题; 朴素贝叶斯需要已知先  
验概率、对属性独立性要求较高; 神经网络容易陷入局部最优  
或者拟合程度不够的问题，同贝叶斯模型一样，需要根据专家  
经验来获取与缺陷相关的因子，计算效率低; 支持向量机具有  
良好的学习和扩展能力， 但其最优参数的设置没有统一高效  
的方法。而且在应对面向对象软件时各类算法都不可避免地  
需要应对非常多的类和对象特征属性来度量软件， 导致“维  
数灾难”，检测时间过长，预测模型实用性降低。

软件度量方法作为获取软件架构和模块属性的一类标准  
化方法，实质是按照相应的度量规则，定量地分析软件实体的  
属性、描述软件及其特征， 为软件质量评估、基于度量的软件  
缺陷预测等工作提供必需的数据来源。传统的结构化软件度  
量主要包括 McCabe 结 构 复 杂 性 度 量、LOC 语句行度量和  
Halstead 软件科学度量等。复杂性软件度量针对的是软件模  
块内属性，而随着面向对象软件技术的发展，模块间的交互更  
加显著，因此复杂度度量标准也应当考虑到模块间的属性。  
以前的结构化度量在面向对象分析和面向对象设计过程中无  
法提取出关于数据抽象、封装、继承、多态、信息隐藏、内聚和  
耦合等面向对象软件特有的属性［3 － 4］ ， 依照传统的度量标准  
得到的软件的特征属性不足以充分表示面向对象软件的内在  
特性，因而，需要针对面向对象的软件提出特定的度量模型。  
目前最具代表性的面向对象软件的度量模型为 MOOD 模型  
和 Chidamber 和 Kemerer 提出的 CK 度量［5］ ， 在 CK 度量中包  
含 6 个具有严格度量理论基础的用于描绘面向对象软件设计  
规模和复杂度的度量属性，这些度量指标包括: 类的加权方法  
数 WMC、继承树的深度 DIT、类的孩子数目 NOC、对象之间的  
耦合 CBO、类的响应集合 RFC 和类方法内聚缺乏度 LCOM。  
这六个指标包含了面向对象软件的大部分特征， 而且在各个  
数据集中，又对其属性集进行了相应的扩充以应对软件复杂  
程度的提高。本文中采用较为灵活的 CK 度量进行软件缺陷  
检测来验证我们所提模型的有效性［3］ 。  
为了优化软件缺陷预测方法的性能， 本文提出了一种基  
于粒子群优化 SVM( Support Vector Machine) 的面向对象软件  
缺陷预测模型，即先通过 Relief 算法进行特征选择， 再利用结  
合了 SVM 较强的泛化能力和随粒子聚集程度动态更新惯性  
权重的 PSO( Particle Swarm Optimization) 算法高效的寻优能  
力的模型进行缺陷预测，获取最优的预测结果。