

基于机器学习的军用软件过时淘汰评估方法研究

帅勇^{1,2}, 宋太亮³, 郑玉杰^{1,2}, 郑雪松^{1,2}, 唐浩^{1,2}

(1. 重庆赛宝工业技术研究院, 重庆 401332; 2. 重庆市电子信息产品可靠性工程技术研究中心, 重庆 401132; 3. 中国国防科技信息中心, 北京 100142)

摘要: 为了对军用软件进行科学系统的过时淘汰评估, 提出基于机器学习的软件过时淘汰评估模型; 首先使用机器学习预处理与缩放技术处理相关的特征数据, 然后基于主成分分析模型进行特征提取和降维, 消除特征数据中的噪音值并选择重要的军用软件过时淘汰特征数据, 使用由粒子群优化算法改进的支持向量机模型进行分类和评估建模, 并使用混淆矩阵的精度评估模型, 最后通过案例验证模型有效性、适用性和科学性。

关键词: 军用软件过时淘汰; 机器学习; 主成分分析; 支持向量机; 评估

Military Software Obsolescence Evaluation Based on Machine Learning

Shuai Yong^{1,2}, Song Tailiang³, Zheng Yujie^{1,2}, Zeng Xuesong^{1,2}, Tang Hao^{1,2}

(1. Chongqing Ceprei Industrial Technology Research Institute, Chongqing 401332, China;
2. Chongqing Engineering Research Center of Electronic Information Products Reliability, Chongqing 401332 China;
3. China Defense Science & Technology Information Center 100142, China)

Abstract: To evaluate military software obsolescence scientifically and systematically, the model of software obsolescence evaluation based on machine learning is pointed. Firstly, the machine learning preprocessing and scaling techniques are used to process the related feature data, then the principal component analysis model is used to extract feature and reduce dimension, eliminate the noise value in the feature data and select important military software obsolete feature data, use particle swarm optimization algorithm to optimize support vector machine parameters and build SVM classification and evaluation model, use confusion matrix accuracy to evaluate the machine model, finally the test results showed the model is effective, applicable, and scientific.

Keywords: military software obsolescence; machine learning; principle component analysis; support vector machine; evaluation

0 引言

随着信息技术的快速发展和高技术战争作战方式多样化程度的不断加剧, 软件在作战、训练和管理等军事活动中发挥着越来越重要的作用。软件过时淘汰特性^[1]是软件的重要特性之一, 是确定软件生命终止的重要标志, 在软件的使用与维护过程中, 发挥着与软件可靠性、安全性、可行性等特性同等重要的作用。开展软件过时淘汰研究, 对于提升软件的性能, 充分发挥软件的作用, 具有重要的军事价值和经济价值。然而, 当前军方还没有确定软件过时淘汰的相关方法和措施, 不利于有效控制软件的成本和生命周期。

论文针对软件过时淘汰的原理和相关数据的基本特性,

按照机器学习分析的基本思路建立军用软件过时淘汰评估模型。首先采集相关特征的数据, 使用 MinMaxScaler 算法对样本数据进行预处理, 然后利用主成分分析 (Principle Component Analysis, PCA) 模型进行特征提取和降维, 消除数据中的噪音值并选择重要的军用软件过时淘汰特征和数据, 使用改进的支持向量机 (Support Vector Machine, SVM) 模型进行分类和评估建模, 最后使用混淆矩阵对评估结果进行判断, 证明模型的有效性和科学性。软件过时淘汰评估机器学习建模总体思路如图 1 所示。

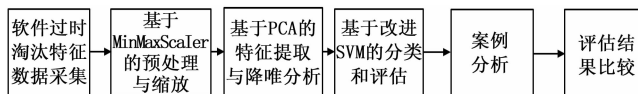


图 1 软件过时淘汰评估机器学习建模总体思路

收稿日期: 2018-11-09; 修回日期: 2018-12-07。

基金项目: 武器装备预先研究基金 (9140A19030314JB35275), 省部级科研项目 (2016YYF0203604, MGY1704040, MGY1804040), 重庆市技术创新与应用示范重点研发项目 (cstc2018jszx-cyzd0634)。

作者简介: 帅勇 (1983-), 男, 重庆璧山人, 博士, 高级研发工程师, 主要从事装备保障信息化、大数据与人工智能技术方向的研究。

1 军用软件过时淘汰技术相关研究

美军早在 2014 就开始了对软件过时淘汰技术的研究, 其报告《Software obsolescence in defence》^[2]中指出, 尽管已经对软件过时淘汰进行了各种分析和研究, 并开发了用于降低软件和组件成本的工具, 但没有开发用于评估软件过时淘汰成本的框架。即使软件功能完整, 但如果无法使用, 软件也会被过时淘汰。如果不主动检查和控制, 将会

造成巨大的成本负担。

当前对于软件过时淘汰的研究主要分为以下四个方面：

一是对过时淘汰机理分析研究。如文献 [3] 将软件过时淘汰的原因归纳为平台过时、技术过时和介质过时，文献 [4] 从正反两方面论述了软件过时淘汰的标志和未达到过时淘汰的标志，文献 [5-6] 将软件过时淘汰的原因解释为技术过时、功能过时和后勤保障过时，并分析了软件过时淘汰的重要性。

二是过时淘汰分析与评估模型建立。如文献 [7] 通过设计分析、技术开发分析和成本分析对是否需要软件过时淘汰进行评估；文献 [6] 提出使用软件生命周期的瀑布图，并将螺旋迭代和增量生命周期映射到 CADMID 生命周期图中来评估软件过时淘汰；文献 [1] 采用模糊贝叶斯网路算法评估软件过时淘汰的影响，并辅助管理者进行决策。

三是过时淘汰对策分析。如文献 [8] 开发了过时淘汰地图软件来确定软件过时的范围和驱动因素，用以监控和管理软件过时淘汰，减少过时淘汰并避免业务中断；文献 [9] 开发了一个软件过时淘汰测试模块接收器，用于判断何种状态下应该进行软件过时淘汰；文献 [10] 指出应对软件过时淘汰的策略是确定旧版本产品的第三方供应商支持、找到备用商业应用程序以执行该功能、考虑为有限数量的特定软件应用程序提供升级策略；文献 [2] 以关键的成本动因为基础制定一个判断软件过时淘汰的框架，以辅助管理者制定正确的软件过时淘汰决策。

四是过时淘汰理论与技术的行业应用。如文献 [11] 中，美国航空航天局采用低成本可扩展平台和自适应维护策略来解决航空航天软件的过时淘汰问题；文献 [12] 为了减轻石油和天然气行业软件过时的风险，使用一种新框架来评估软件类型，并识别和量化软件过时淘汰的影响；文献 [13] 针对航空航天企业在软件过时淘汰上面临的问题和挑战，提出一种航空航天工业软件过时淘汰触发图，用以增强航空航天公司管理软件应用程序的综合能力，避免业务中断并优化维护成本；文献 [14] 针对卫星地面控制软件中常见的软件过时及缺乏理想的功能和灵活性等问题，开发了基于卫星地面控制软件对象处理方法轮模型和处理软件过时淘汰指导方针系统，利用数据库驱动的应用程序以及对象过程方向和模块化的概念，辅助卫星运营商灵活地创建新功能，并使软件开发人员能够更快地发现错误并更有效地修复错误。

以上对于软件过时淘汰的研究大部分来源于军方开展的工作，虽然解决了当前存在的部分问题，但是仍然过度依赖于专家知识和机理分析，并限定在特定的行业中，其科学性、适用性和可行性较差。本文从数据角度出发，基于多个机器学习算法构建综合模型，对软件过时淘汰相关特征数据进行处理，并且从中提取出其背后的因素，从而发现评估军用软件过时淘汰的方法，更具有科学性和适用性。

2 机器学习建模

2.1 预处理与缩放

由于 PCA 和 SVM 模型对于样本数据非常敏感，因此，对于采集的软件过时淘汰相关特征数据进行预处理和缩放，可以使特征数据更加适用于 PCA 和 SVM 算法，提高算法的精度。

对于定量的数据，scikit-learn 提供了四的数据预处理的方式，如表 1 所示。

表 1 数据预处理方法概述

预处理方法名称	方法函数	处理思路
标准尺度法	StandardScaler	确保每个特征的均值为 0，方差为 1，使所有特征都位于同一量级
健壮缩放法	RobustScaler	确保每个特征的中位数为 0，四分位数为 1，使所有特征都位于同一量级
最小最大缩放法	MinMaxScaler	使所有特征都位于 [0,1] 之间
正则化缩放法	Normalizer	对每个数据点进行缩放，使得特征向量的欧式长度为 1

考虑到 SVM 和 PCA 算法的特性，论文采用 MinMaxScaler 函数对数据预处理与缩放，例如对维护费效值特征缩放后与缩放前数据比较如图 2、图 3 所示。对于定性数据，按照其定性表述的值，由人工确定在 [0, 1] 之间取值。如对于特征“功能适应性”，其评估值可能是包括“优、良、中、差”四个等级，则定义其取值为 0.9、0.7、0.5、0。

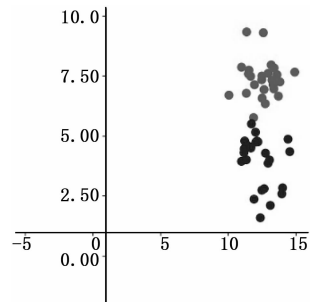


图 2 缩放前数据

2.2 基于 PCA 的特征提取与降维分析

主成分分析是一种旋转数据集的方法，旋转后的特征在统计上不相关，旋转后根据新特征对解释数据的重要性来选择子集。^[15] 使用 PCA 保留一部分主成分来使用并进行降维，可以去掉数据中的噪音，并找到影响军用软件过时淘汰的关键参数。主成分分析的基本思路如图 4 至图 7 所示。

2.3 基于改进 SVM 的分类和评估

由于 SVM 在结构、性能、学习速度、泛化能力、数据处理和维度免疫等方面具有显著的优势，且可以推广到更

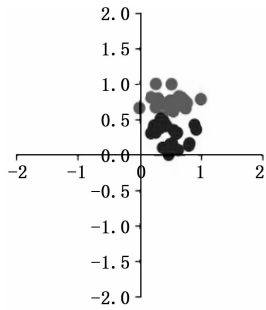


图 3 缩放后数据

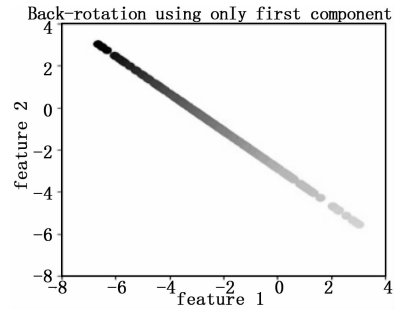


图 7 还原后的主成分

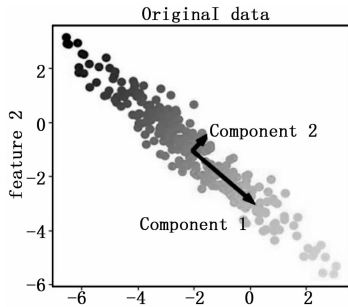


图 4 原始数据的主成分

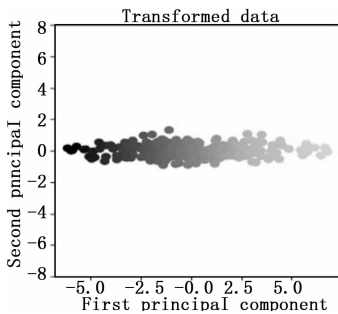


图 5 数据转换后的主成分

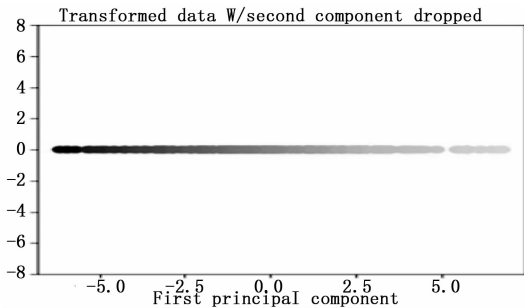


图 6 特征缩减后的主成分

复杂的无法被输入空间的超平面定义的模型的扩展, SVM 模型已广泛应用于分类和回归。^[16]考虑到军用软件过时淘汰评估的结论是淘汰或不淘汰, 因此可以将此类问题归纳为一个二分类问题进行处理, 从而建立软件过时淘汰的评估模型。

SVM 算法建模步骤如下^[18]:

第一步: 构建 SVM 估计函数。

假设有一样本集为 $\{(y_1, x_1), (y_2, x_2), \dots, (y_l, x_l)\}$, 且 y

$\in R$ 。支持向量机线性方程表示如下, 其估计函数为:

$$f(x) = W^T \varphi(x) + b \quad (1)$$

第二步: 最优问题的转化。

基于线性可分定理, 利用 ϵ 不敏感损失函数将估计函数转化为优化问题:

$$\min_{w, b, \epsilon, \epsilon^*} \frac{1}{2} W^T W + C \sum_{i=1}^l \epsilon_i + C \sum_{i=1}^l \epsilon_i^* \quad (2)$$

$$\begin{cases} W^T \varphi(x_i) + b - y_i \leq \epsilon + \xi_i \\ y_i - W^T \varphi(x_i) - b \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, i = 1, \dots, l \end{cases} \quad (3)$$

上式的约束条件为:

其中 $W^T W$ 代表与模型相关的复杂度指标, W^T 代表 W 的维数, C 为惩罚函数, ϵ 为不敏感顺势函数, ξ_i 和 ξ_i^* 为松弛变量, ξ_i 为上限值, ξ_i^* 为下限值表示样本偏离 ϵ 不敏感区域的程度, 惩罚函数、不敏感系数、松弛变量及核函数相关参数均对 SVM 的性能有着重要的影响。各参数的最优组合通过改进的粒子群优化算法获得, 详细步骤在 5.3.3 节中进行论述。

第三步: 采用对偶理论将上式转化为二次规划求解, 则约束表达式的对偶式为:

$$\min \frac{1}{2} (a - a^*)^T Q (a - a^*) + \epsilon \sum_{i=1}^l (a_i + a_i^*) + \sum_{i=1}^l y_i (a_i - a_i^*) \quad (4)$$

其约束条件为:

$$\sum_{i=1}^l (-a_i + a_i^*) = 0, 0 \leq a_i, a_i^* \leq C, i = 1, \dots, l \quad (5)$$

第四步: 通过二次规划算法计算 SVM 模型。SVM 模型为:

$$f(x) = \sum_{i=1}^l (-a_i + a_i^*) K(x_i, x) + b \quad (6)$$

式中, $K(x_i, x)$ 称为核函数, 需要满足 Mercer 条件, 且需要满足:

$$Q_{ij} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \quad (7)$$

论文采用径向基函数 (Radial Basis Function, RBF) 作为 SVM 的核函数, 其计算方法见 5-8。

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\delta^2}\right), \delta \in +R \quad (8)$$

SVM 训练的优化主要针对核函数、惩罚系数 C 和不敏

感损失系数 γ 。C 和 γ 可通过粒子群优化算法 (particle swarm optimization, PSO) 进行优化, 相关算法参见文献 [19]。通过对训练集的优化, 得到 SVM 的最佳参数组合, 将该参数组合应用到支持向量机分类模型中, 计算出基于现有参数集的军用软件过时淘汰结果, 并与实际的过时淘汰结果进行比较。

2.4 基于混淆矩阵模型评估与验证

混淆矩阵是用 $N \times N$ 的矩阵形式来评估机器学习精度的一种方式。^[15]对于二分类问题, 混淆矩阵是一种最全面的表示方法, 使用混淆矩阵输出一个 2×2 的数据, 其中行对应于真实的类别, 列对应于预测的类别。混淆矩阵主对角线上的元素对应于正确的分类, 而其他元素则表示一个类别中有多少样本被错误地划分到其他类别中。观察混淆矩阵, 可以发现不同模型的问题, 如较差的模型总是预测同一个类别, 或者其真正例和真反例很少, 而较好的模型其真正例和真反例更多, 而假正例和假反例数量很少。

混淆矩阵的评估方法包括精度、准确率、召回率和 f -分数。^[15]论文采用精度来评估混淆矩阵, 其计算公式如下:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

其中: TP 表示被判定为负样本, 事实上也是负样本的样本数; TN 表示被判定为正样本的样本数, 事实上也是正样本; FP 表示被判定为正样本, 但事实上是负样本的样本数; FN 表示被判定为负样本, 但事实上是正样本的样本数。

3 案例分析

论文使用某装甲机械化部队装备管理方面使用的 34 个软件系统在 2010~2015 年间的过时淘汰相关特征数据及评估数据进行案例分析, 并验证模型的有效性, 建模环境为 python3.5, 机器学习库为 scikit-learn, 深度学习库为 TensorFlow。

3.1 案例运算

基于论文的建模思路, 首先采集软件过时淘汰数据集。考虑到军用软件过时淘汰的特性, 为了减少专家决策对样本数据的影响, 论文的样本特征采集使用基于文本挖掘^[17]的特征获取方法, 采集公开发表的相关论文中涉及到软件过时淘汰的 20 个高频特征参数, 包括: 维护费效值、硬件匹配度、满负荷运转率、软件平台兼容性、功能适应度、客户满意度、环境适应度、任务完成度、风险管理能力、平均响应时间、最长响应时间、满负荷每小时处理请求数目、功能实现程度、平均维护时间、平均维护金额、维护服务是否提供、关键容错率、自修复率、致命性故障间隔时间、平均无故障时间。依据上述特征参数, 获取 34 个样本软件中上述特征对应的数据并进行评估建模。

步骤 1: 对于采集的样本特征数据, 使用 MinMaxScaler 算法归一化处理定量数据和定性数据, 使得所有样本值都位于 $|0 \sim 1|$ 之间。为了保证训练集、验证集和测试集的独立性和可信性, 使用 sklearn 的 train_test_split 函数

对数据集进行两次划, 第一次将样本数据集划分训练集和测试集, 第二次将训练集划分为用于构建模型的训练数据集和用于选择模型参数的验证数据集。

步骤 2: 使用 PCA 进行建模, 选择与军用软件过时淘汰关联性强的特征参数, 并以前两个特征为例, 生成对于训练数据的热图和特征间的关联结果, 如图 8 所示。从图中可以发现, 第一特征的值较大, 距离 1 更近, 这意味着其他测量值接近于 1 的可能性也较大, 所有特征之间存在普遍的相关性; 第二个主成分的值较小, 距离 0 更近, 说明该特征与其他特征的相关性不如特征 1 明显。这种情况下, 优先选择第一个主成分作为下一步 SVM 建模的特征参数。基于此规则进行降维, 选择特征相关性最强的 8 个特征参数进行下一步的评估建模, 相关参数为维护费效值、环境适应度、平均响应时间、风险管理能力、满负荷每小时处理请求数目、自修复率、致命性故障间隔时间、平均无故障时间。

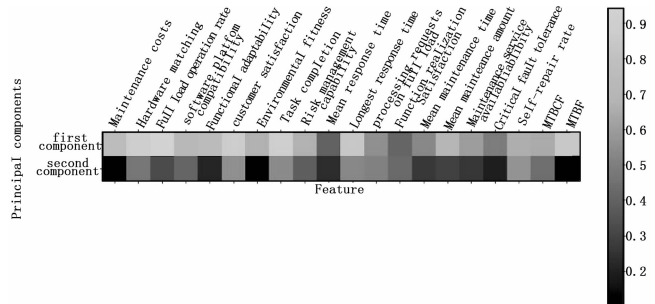


图 8 基于 PCA 的特征系数可视化热图

步骤 3: 使用 SVM 进行分类与评估, 建立软件过时淘汰评估模型。对于评估结果, 以 0.5 为阈值。基于上述八个特征对软件的过时淘汰特性进行评估, 当某个软件的评估值低于 0.5 时, 认定其需要进行过时淘汰; 当其评估值大于等于 0.5 时, 认定其不需要进行过时淘汰。

利用 SVM 建立评估模型, 首先使用 PSO 算法^[14]对 SVM 的 C 和 γ 进行优化, 并选择典型的 C 和 γ 进行热图比较, 可以发现, 当 $C=100$, $\gamma=0.01$ 时, 训练精度最高, 为 0.97。SVM 中 C 和 γ 训练值比较如图 9 所示。

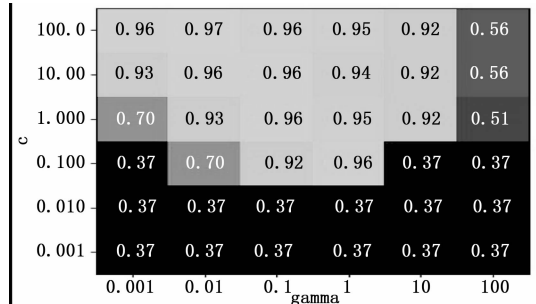


图 9 SVM 中 C 和 γ 训练值比较

SVM 模型使用最优的 C 和 γ 值对测试集进行运算, 得到模型的评估值与真实值的比较结果, 并使用

KNN、朴素贝叶斯和逻辑回归法进行运算并比较评估结果。使用混淆矩阵比较各算法混淆矩阵的训练精度和测试精度, 结果如表 2 所示。

表 2 各模型训练精度与测试精度比较

模型	训练精度	测试精度	排名
KNN	0.78	0.77	④
朴素贝叶斯	0.80	0.79	③
逻辑回归	0.85	0.82	②
本文方法	0.96	0.95	①

3.2 结果分析

通过案例分析可以发现, 论文使用的机器学习综合模型不仅可以在训练过程中达到最优的效果, 对于测试集的评估精度同样有效, 说明论文的机器学习模型具有较好的适应性; 同时, 由于建模过程中没有采用专家学习等主观性较强的方法, 保证了结论不受人为影响, 体现了模型的科学性和客观性。

4 结论

机器学习是从数据中提取知识, 是统计学、人工智能和计算机学科的交叉研究领域。^[15]针对机器学习算法的高效性和科学性, 论文利用机器学习预处理与缩放、PCA、SVM 等算法构建软件过时淘汰评估模型, 并通过案例进行了分析建模, 结果显示机器学习模型具有更高的测试精度, 更强的客观性和科学性。

参考文献:

- [1] 帅 勇, 宋太亮. 基于模糊贝叶斯网络的软件过时淘汰影响研究 [J]. 计算机测量与控制, 2014, 35 (12): 4207-4212.
- [2] Rajagopala S, Erkoyuncu J A, Roy R. Software obsolescence in defence [A]. 3rd CIRP Global Web Conference [C]. 2014, 22 (10): 76-80.
- [3] Kouroush Jenab, Kouroush Noori, Philip D. Weinsier, Sam Khoury. Reliability Paper: A dynamic model for hardware/software obsolescence [J]. International Journal of Quality & Reliability Management, 2014, 31 (5): 1-13.
- [4] Erich Schmid, George Kosugi, Jorge Ibsen, Morgan Griffith. Don't get taken by surprise: planning for software obsolescence management at the ALMA Observatory [J]. Proceedings of the SPIE, 2016, 9913 (46): 1-11.
- [5] Bjoern Bartels, Ulrich Ermel, Michael Pecht, Peter Sandborn. Strategies to the Prediction, Mitigation and Management of Product Obsolescence [M]. John Wiley & Sons Inc, 2012: 143-155.
- [6] Rajagopal S, Erkoyuncu J A, Roy R. Impact of software obsolescence in defence manufacturing sectors [J]. 3rd CIRP Global Web Conference, 2015, 28: 197-201.
- [7] Darling, Green. Decision software as applied to obsolescence [A]. AIAA's 3rd Annual Aviation Technology, Integration,

- and Operations (ATIO) [C]. 2003, 11: 1-10.
- [8] Raul Gonzlez Munuz, Essam Shehab, Martin Weinitzke, Chris Fowler, Paul Baguley. Software Obsolescence Drivers in Aerospace: An Industry Analysis [J]. World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2017, 11 (9): 961-966.
- [9] Cathleen R. Kennedy. Software replacement of test module adapters for depot instrumentation due to obsolescence [J]. Aerospace and Electronic Systems Magazine, 2004, 19 (7): 25-30.
- [10] Leigh Merola. The COTS Software Obsolescence Threat [J]. Proceedings of the Fifth International Conference on Commercial-off-the-Shelf (COTS) -Based Software Systems (IC-CBSS 2006), 2006, 6: 127-132.
- [11] Brian J. Giovannoni, Carole Boyles. Hidden Costs of Unsupported Software, Obsolescence and Non-Standards; The Importance and Value of a Multi-Mission Software Program [A]. SpaceOps Conferences [C]. 2016, 5: 1-8.
- [12] John Ahmet Erkoyuncu, Stanley Ononiwu, Rajkumar Roy. Mitigating the Risk of Software Obsolescence in the Oil and Gas Sector [A]. 3rd International Conference on Through-life Engineering Services [C]. 2014, 22: 81-86.
- [13] Raúl González Muñoz, Essam Shehab, Martin Weinitzke, Rachel Bencec, Chris Fowler, Stephen Tothill, Paul Baguley. Key Challenges in Software Application Complexity and Obsolescence Management within Aerospace Industry [Z]. CIRPe 2015 Understanding the life cycle implications of manufacturing, 2015, 37: 24-29.
- [14] Dov Dori, Somwang Thipphayathethana. Model-based guidelines for user-centric satellite control software development [J]. International Journal of Satellite Communications and Networking, 2015, 34 (7): 295-319.
- [15] Andreas C. Muller, Sarah Guido. Introduction to Machine Learning with Python [M]. O'Reilly Media, 2016, 5.
- [16] 帅 勇, 宋太亮, 王建平. 考虑全过程优化的支持向量机预测方法 [J]. 系统工程与电子技术, 2017, 39 (4): 931-940.
- [17] Shuai Yong, Song Tailiang, Wang Jianping, Zhan Wenbin. Hybrid Reliability Parameter Selection Method Based on Text Mining, Frequent Pattern Growth algorithm and Fuzzy Bayesian Network [J]. Journal of Shanghai Jiao Tong University (Science), 2018, 23 (3): 423-428.
- [18] Guzzetti F, Peruccacci S, Rossi M, et al. The Rainfall Intensity-duration Control of Shallow Landslides and Debris Flows: an Update [J]. Landslides, 2008, 5 (1): 3-17.
- [19] Shuai Yong, Song Tailiang, Wang Jianping. An integrated parallel forecasting model based on fuzzy time series and SVM [J]. Journal of Systems Engineering and Electronics, 2017, 28 (7): 766-775.