

基于泛函网络的软件可靠性多模型综合预测方法

王二威¹ 吴祈宗²

(北京理工大学珠海学院 珠海 519088)¹ (北京理工大学管理与经济学院 北京 100081)²

摘 要 将泛函网络引入软件可靠性预测,利用其比神经网络更好的解释性及其他性能,提出了基于泛函网络的软件可靠性多模型综合预测方法。首先阐述了泛函网络的结构和学习过程,然后将多个单一模型的预测值作为泛函网络的输入,将实际值作为输出,建立泛函网络结构,给出了泛函网络的学习算法,制定了 3 种训练策略,并进行了实验分析。实验结果表明:在第三种训练策略下,基于泛函网络的软件可靠性多模型综合预测方法有较高的预测精度,其预测效果比单个模型和 Lyu 提出的线性综合模型都好。

关键词 软件可靠性,泛函网络,多模型综合

中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.10.035

Multi-model Synthesis Prediction of Software Reliability Based on Functional Networks

WANG Er-wei¹ WU Qi-zong²

(Beijing Institute of Technology, Zhuhai, Zhuhai 519088, China)¹

(School of Management and Economics, Beijing Institute of Technology, Beijing 100081, China)²

Abstract The functional networks were introduced into the prediction of software reliability, and based on its better explanatory and other attributes than neural network, a multi-model synthesis prediction method of software reliability based on functional networks was proposed. The estimated values of many single models were taken as the input, and the actual value was taken as the output, thus the structure of functional networks was established. The learning algorithm of functional networks was proposed and three training strategies were designed, all of which were conducted in tests accordingly. The test results show that in the third training strategy, the multi-model prediction method based on functional networks has better predictive accuracy, and is more effective than single-model and linear integrated models proposed by Lyu.

Keywords Software reliability, Functional networks, Multi-model Synthesis

1 引言

随着软件在代码量和复杂度两个方面的指数性增长,软件可靠性越发成为不可忽视的问题。但单个经典软件可靠性模型(如 JM 模型、GO 模型、NHPP 模型、LV 模型等)存在很大的局限性,对于一个软件可靠性的评估往往很难用一个模型来处理,而且很多软件可靠性模型都建立在概率分布假设的基础上,这就造成对一个软件有很好的适用性而对其他的软件的效果很差^[1,2]。20 世纪 90 年代以来,基于知识的方法被越来越多地应用到软件可靠性预测中^[3],如神经网络^[4-8]、遗传算法^[9]、模拟退火、支持向量机^[10]、粒子群^[11]等。然而这些方法算法复杂,计算性能较低,预测结果的精确性取决于结构的设计和模型的训练。Cai^[12]在大量对比实验的基础上指出:大多数情况下,采用神经网络方法很难得出满意的量化预测结果。香港中文大学的 Michael Lyu 提出了多个单一模型“综合”的思想^[1],并采用 4 种线性“综合”的方法验证了软件可靠性多模型综合预测比单个经典模型有更好的预测效果。

西班牙学者 Enrique Castillo 提出的泛函网络^[13,14]是神

经网络的推广,其由于很好的可解释性及优于神经网络的其他性能^[15],得到了广泛的应用^[15-18],如线性及非线性回归、非线性系统辨识、捕鱼预测优化、混沌时间序列预测及分类、电子商务推荐等。本文将泛函网络应用到软件可靠性预测中,提出了基于泛函网络的软件可靠性多模型综合预测方法,根据软件失效数据的客观特征,参考要综合的每个经典模型的特点,将多个经典模型的预测值通过泛函网络进行非线性综合,进而更准确地预测软件的可靠性。

2 泛函网络及其学习过程

2.1 泛函神经元模型

设泛函神经元 $f(\cdot)$ 是一个处理单元,对于输入向量 $x = (x_1, x_2, \dots, x_n)^T$, E. Castillo 提出的泛函神经元结构如图 1 所示,它与人工神经元类似,但缺少了连接线上的权重信息。

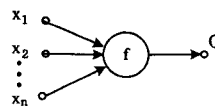


图 1 泛函神经元模型

到稿日期:2014-09-19 返修日期:2014-12-27 本文受广东省软科学基金项目(2013B070206004)资助。

王二威(1985—),男,硕士,讲师,主要研究方向为软件工程、信息管理, E-mail: wodezazhi@126.com; 吴祈宗(1947—),男,教授,博士生导师,主要研究方向为决策理论与方法。

图1中, $O=f(x)$ 是泛函神经元的输出, 对该神经元的学习即是对泛函处理单元函数 $f(\cdot)$ 的学习。泛函神经元函数用基函数的线性组合来表示:

$$f(x) = \sum_{i=1}^n a_i \varphi_i(x) \quad (1)$$

其中, $\{\varphi_i(x) | i=1, 2, \dots, n\}$ 是任意给定的基函数簇, 是可以学习的, 可以根据特定的问题和数据来选择不同的函数簇, 比如三角函数基、多项式基、傅里叶基函数等; a_i 是参数, 也是可学习的, 可以用来无限逼近泛函神经元函数 $f(x)$ 一直达到期望的精度。

2.2 泛函网络的构成

设 $X=(x_1, x_2, \dots, x_p)$ 是一个节点集, 每一个节点 x_i 对应一个变量 X_i 。用元组 $U=(x, f, z)$ 来表示节点集和泛函神经元函数之间的关系, x 是输入节点集, z 是输出节点集, 则 $z=f(x)$, 其中 x 和 z 是 X 的非空子集。一个典型的泛函网络拓扑结构如图2所示。

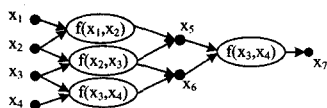


图2 一种典型的泛函网络拓扑结构

典型的泛函网络拓扑结构如图2所示。

(1) 几层存储单元: 一层输入单元, 用带有相应名字的实心圆表示, 如图2中 $\{x_1, x_2, x_3, x_4\}$ 。一层输出单元, 它是输出泛函网络的结果数据, 因此是最后一层单元, 输出单元也用带有相应名字的实心圆表示, 如图2中的 $\{x_7\}$ 。若干中间单元, 它用来存储中间层泛函神经元产生的信息, 中间单元也用带有相应名字的实心圆表示, 如图2中的 $\{x_5, x_6\}$, 也可以没有中间单元。

(2) 一层或者几层处理单元: 即是泛函神经元, 它处理一组来自前一层泛函神经元或者输入单元的输入值, 给下一层泛函神经元或者输出单元提供一组输入数据。图2中有两层处理单元, 即 $\{f_1, f_2, f_3\}$ 和 $\{f_4\}$ 。

(3) 定向连接线: 它将存储单元和处理单元连接起来, 箭头表示信息的流动方向。

所有这些元素一起组成了泛函网络的结构, 同时确定了网络的泛函能力。

2.3 泛函网络学习过程

有别于神经网络, 泛函网络是对网络结构和参数的学习。其学习步骤如下:

第一步, 定义问题, 分析问题, 确定网络的初始拓扑结构。

第二步, 网络结构学习, 简化网络结构。根据泛函方程, 看是否存在更简单的并与之等价的泛函网络。

第三步, 表达式的唯一性检查。对于一个给定的网络拓扑结构, 给出问题的初始条件使得对于任意的输入值, 网络都有相同的输出值。

第四步, 参数学习。即对式(1)中系数 a_i 的学习。

第五步, 模型确认。利用数据对网络进行测试, 对网络模型进行交叉确认, 如果测试结果满意, 则模型可以投入使用。

2.4 泛函网络与神经网络的区别

(1) 在拓扑结构的选择上, 泛函网络由于是问题驱动的, 可以根据实际问题从数据以及先验知识方面来确定, 还可以

参考泛函神经元函数的结合性、可交换性、恒定性等一些特性来进行网络结构的简化确定; 而神经网络只能依据数据和反复实验进行建模。

(2) 泛函网络的神经元函数和参数都通过数据进行学习; 而神经网络的神经元函数是事先给定的。

(3) 泛函网络的神经元函数可以是多目标函数, 而且同一层网络的神经元函数可以不同; 而神经网络的神经元函数是确定的单变量函数, 它仅是对所有输入信息的一个线性组合进行激活。

(4) 泛函网络对输入的数据没有限制, 每个神经元都可以是多输出的; 而神经网络的每个神经元都是单输出的, 对输入数据有要求, 如限制在 $[0, 1]$ 区间等。

综上, 由于泛函网络的问题驱动特征, 可以不考虑输入数据形式的要求, 根据软件可靠性综合预测问题本身进行网络结构的构建。泛函网络对输入的综合正好符合软件可靠性多模型综合预测的思想, 泛函网络的输入正是各经典模型的预测值, 输出正好是多模型综合预测的输出, 中间层表征了经典模型的融合, 泛函神经元为经典模型预测性能的整合(线性或非线性)提供了可能。

3 基于泛函网络的软件可靠性多模型预测

3.1 问题定义

根据软件失效数据的客观特征, 参考要综合的每个经典模型的特点, 将多个经典模型的预测值通过泛函网络进行综合, 即将多个经典模型的预测值作为泛函网络的输入, 通过泛函神经元函数及网络结构对输入进行综合, 期望输出更精确的预测值。

软件失效数据一般有两类^[19]: 一类是相邻失效间隔时间, 一类是某一时间段内的失效次数, 一般使用前者能够获得更精准的预测。第一类软件失效数据又可以用两种形式表示: 一种是记录相邻两次失效的间隔时间, 一种是记录失效发生时软件运行的累计时间, 两者可以相互转换。泛函网络对输入的数据没有格式上的要求, 因此可以使用两种形式的软件失效数据。

3.2 一般步骤

第一步, 经典模型的选择。参考Lyu提出的经典模型的选择原则^[1, 20], 即首先选择有很好预测效果而且应用广泛的经典模型, 其次选择预测的偏好可以相互抵消的模型, 这些模型的预测偏好综合起来以后既不悲观也不乐观。

第二步, 网络结构的确定。根据问题定义进行分析, 采用多输入单输出可分离的泛函网络模型, 输入的节点数就是所选择经典模型的个数, 其结构如图3所示。

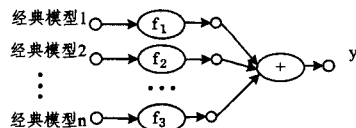


图3 对于软件可靠性多模型综合预测问题的泛函网络结构

图3中的泛函神经元函数 f_i 用基函数的线性组合来表示, 选择基函数簇时要考虑相对应经典模型的特点。设

$$f_i = \sum_{s=1}^m a_{is} \phi_s \quad (2)$$

其中, ϕ_s 是基函数; a_s 是基函数的系数, 即网络参数; m 是每个基函数簇中基函数的个数。

则泛函网络输出可表达为:

$$y=\sum_{i=1}^n f_i=\sum_{i=1}^n \sum_{s=1}^m a_s \phi_s \tag{3}$$

根据所选择经典模型的特点确定泛函神经元的基函数簇,进而确定泛函神经元函数。由泛函网络的结构和泛函神经元函数可以看出,该方法是多个经典模型的非线性综合。

第三步,表达式唯一性验证。由于采用的是成熟的可分离的泛函网络结构,因此满足表达式唯一性。

第四步,参数学习。将多个经典模型预测值作为输入,将实际值作为输出,采用参数学习算法对网络进行训练,进而确定网络参数。

第五步,模型预测。用训练好的网络模型对新数据进行预测,检验模型的有效性。

4 试验分析(一)

4.1 试验数据

由于在实际的软件项目背景下搜集失效数据很困难,为了不影响软件可靠性模型的研究和方法的验证,这里选择 Musa 公开发表的 DACS 数据集中的 3 组数据 (SYS1、SYS2、SYS3,见表 1)^[21] 进行试验,这 3 组数据是在实际软件工程中获取的软件可靠性数据。

表 1 3 组数据的采样点

失效数据集	采样点	数据形式
SYS1	136	失效时的累计时间
SYS2	87	失效时的累计时间
SYS3	207	失效时的累计时间

4.2 经典模型选择

为了进行对比分析,这里根据 Lyu 提出的模型选择原则,同样选择 GO、MO、LV 3 个经典模型,它们都有很好的预测效果并在工程中有广泛的应用。其中,GO 是指数型 NHPP 模型,预测结果比较乐观;MO 是对数型 NHPP 模型,预测结果有悲观也有乐观;LV 是贝叶斯模型,预测结果比较乐观。

4.3 泛函网络结构的确定

以 GO、MO、LV 3 个经典模型的预测失效数分别作为 3 个输入,以实际失效数作为输出,确定 3 输入单输出的可分离泛函网络结构。

在为泛函神经元函数选择基函数时,一般需要根据问题的描述来确定合适的基函数簇。考虑到选取的 3 个经典模型的特点,对于第一层泛函神经元函数 $f_i(i=1,2,3)$,取其基函数簇为 $\phi_s=\{1,e^x,e^{-x}\},s=1,2,3$ 。而对于 f_4 ,为了保证其可逆性,取其基函数簇为 $\phi_4=\{1,x,x^2\}$ 。

4.4 参数学习算法

采用 Castillo 提出的 Minimax 学习算法^[15,22],即最小化

误差。该算法具有较快的学习速度。

由 4.3 节确定的基函数可得:

$$f_i(x)=\sum_{s=1}^3 a_s \phi_s, i=1,2,3,4 \tag{4}$$

其中, a_s 是参数。

则误差

$$e_j=f_1(x_{1j})+f_2(x_{2j})+f_3(x_{3j})-f(x_{4j})$$

其中 j 为训练样本数。

为了保证 f_4 的可逆性,对应的基函数簇中的每一个函数都是可逆的,而且其系数 a_{4i} 是非负的,并有 $\sum a_{4i}=1$ 。

根据 Minimax 学习算法,建立如下线性规划模型:

$$\begin{aligned} \text{Min } \epsilon & \\ \left\{ \begin{aligned} & -\sum_{i=1}^3 a_{1i} \phi_{1i}(x_{1j}) - \sum_{i=1}^3 a_{2i} \phi_{2i}(x_{2j}) - \sum_{i=1}^3 a_{3i} \phi_{3i}(x_{3j}) + \\ & \sum_{i=1}^3 a_{4i} \phi_{4i}(x_{4j}) + \epsilon \geq 0, \forall j \\ & \sum_{i=1}^3 a_{1i} \phi_{1i}(x_{1j}) + \sum_{i=1}^3 a_{2i} \phi_{2i}(x_{2j}) + \sum_{i=1}^3 a_{3i} \phi_{3i}(x_{3j}) - \\ & \sum_{i=1}^3 a_{4i} \phi_{4i}(x_{4j}) + \epsilon \geq 0, \forall j \\ & \sum_{i=1}^3 a_{4i} = 1 \\ & a_{4i} \geq 0, \forall i \end{aligned} \right. \tag{5} \end{aligned}$$

根据训练样本,求解此线性规划问题,可得到各参数值。

4.5 网络训练策略

本试验采用两种网络训练策略:第一种策略,选取各条数据 90% 的样本点进行网络训练,对剩余的样本点进行预测,以分析长期预测效果;第二种策略,选取各条数据的前 $n-2$ 个样本点进行网络训练,对最后的两个样本点进行预测,以分析短期预测效果。

4.6 模型评价标准

对软件可靠性模型预测性能进行评价时,最主要是看模型的有效性,它可以用定量的方法来测量。本文采用的评价标准如下。

(1)平均误差 (Average Error, AE)^[23]:在整个测试阶段模型预测的误差均值,只能针对同一数据进行比较。平均误差越小,模型的预测效果越好。

(2)平均偏差 (Average Bias Error, AB)^[23]:在整个测试阶段模型预测总的偏差趋势,既可以为正值,也可以是负值,正值表示预测结果比较悲观,负值表示预测结果比较乐观。平均偏差越接近于零,模型的预测效果越好。

(3)均方根误差 (Root Mean Square Error, RMSE)^[1]:在整个测试阶段模型预测值与实际值的距离,均方根误差值越小,模型的拟合度越高,预测性能越好。

4.7 结果分析

经典模型和多模型综合预测方法的预测性能见表 2—表 4。表中 FN-1、FN-2 分别表示第一种、第二种训练策略下的泛函网络预测模型。

表 2 各模型对数据 SYS1 预测能力评价标准的比较

评价标准	第一种训练策略				第二种训练策略			
	MO	LV	GO	FN-1	MO	LV	GO	FN-2
AB	-0.01621	0.02201	0.00681	0.03797	-0.004558	0.01897	-0.00115	-0.00374
AE	0.01621	0.02308	0.00765	0.03797	0.00458	0.02097	0.00115	0.00374
RMSE	2.49372	3.67871	1.28377	5.75508	0.67882	6.16636	0.02828	0.66567

表3 各模型对数据 SYS2 预测能力评价标准的比较

评价标准	第一种训练策略				第二种训练策略			
	MO	LV	GO	FN-1	MO	LV	GO	FN-2
AB	0.00759	0.01153	0.01118	-0.03268	0.00623	0.00888	0.00382	-0.00279
AE	0.00968	0.01249	0.01224	0.03268	0.00623	0.00888	0.00382	0.00279
RMSE	0.99503	1.23922	1.27922	3.14437	0.76692	0.92424	0.46966	0.24156

表4 各模型对数据 SYS3 预测能力评价标准的比较

评价标准	第一种训练策略				第二种训练策略			
	MO	LV	GO	FN-1	MO	LV	GO	FN-2
AB	-0.01935	0.04848	-0.00787	0.01354	-0.00268	0.06121	-0.00158	-0.00384
AE	0.02098	0.04848	0.01212	0.01401	0.00268	0.06121	0.00158	0.00384
RMSE	5.07611	10.1904	2.73168	3.31165	0.78065	12.6526	0.46103	0.79319

从表2—表4可以得出:

(1)在两种训练策略下 AB 和 AE 的值都很理想,说明泛函网络预测模型有较好的预测性能;两种训练策略下 AB 的值均为负,说明泛函网络预测模型的预测结果比较乐观;第二种训练策略下泛函网络预测模型的预测性能远远优于第一种。

(2)第一种训练策略下,泛函网络预测模型的预测性能不如单个的经典模型。

(3)第二种训练策略下,泛函网络预测模型的预测性能一般都比单个经典模型的预测性能强,泛函网络预测模型对短期预测效果更好。

(4)第一种训练策略下 RMSE 的值过大,造成这种结果的原因就是网络没能很好地描述整个软件发生失效的时间过程,使得网络对局部的数据预测得很准确,对其他部分的数据预测的准确度稍差一些。

5 试验分析(二)

从试验分析(一)的预测数据中可以发现,泛函网络预测模型对训练样本点之后的2至5个预测样本点有很好的预测效果,但对以后的预测样本点则预测效果稍差一些。为了进一步验证模型的短期预测性能,对泛函网络预测模型的训练策略进行一些改进:首先仍然选取90%的样本点作为训练样本点,对训练样本点之后的两个预测样本点进行预测;然后将这两个样本点纳入到训练样本中,再对训练样本点之后的两个预测样本点进行预测;依此类推,直到所有预测样本点预测完成。同时与Lyu提出的4种经典模型线性综合方法(ELC、MLC、ULC、DLC)^[1]进行对比试验。

使用SYS1和SYS2数据进行试验分析,各模型的预测性能如表5和表6所列,表中FN-3指第三种训练策略下泛函网络预测模型。

表5 几种综合模型的比较(SYS1)

评价标准	ELC	MLC	ULC	DLC	FN-3
AB	0.00421	-0.01621	-0.00600	0.00156	-0.00492
AE	0.00966	0.01621	0.00892	0.00724	0.00652
RMSE	1.40127	2.49373	1.53180	1.07565	0.95858

表6 几种综合模型的比较(SYS2)

评价标准	ELC	MLC	ULC	DLC	FN-3
AB	0.01010	—	—	—	-0.01023
AE	0.01140	—	—	—	0.01023
RMSE	1.15983	—	—	—	0.99435

对比表2—表4得出:在第三种训练策略下泛函网络预

测模型的预测结果是比较乐观的;其预测性能比单个经典模型的预测性能更优。

从表5和表6可以得出:对于SYS1,第三种训练策略下泛函网络预测模型在AB标准下比ELC和DLC稍差,而在第三种训练策略下泛函网络预测模型的其余标准是最好的。对于SYS2,运用Lyu的方法只能得出ELC模型数据,其预测效果也比第三种训练策略下泛函网络预测模型的预测效果差。

由以上试验分析可以得出以下结论:第三种训练策略下泛函网络预测模型的预测结果偏向乐观;第三种训练策略下泛函网络预测模型有较高的预测精度,其预测性能优于Lyu提出的线性综合模型。

结束语 本文将泛函网络应用到软件可靠性的预测中,提出了一种基于泛函网络的软件可靠性多模型综合预测方法。有别于Lyu的线性综合模型,该方法是对多模型的非线性综合。然后给出了模型的学习算法,制定了3种训练策略,并分别进行了试验分析。试验结果表明在第三种训练策略下,基于泛函网络的软件可靠性多模型综合预测方法具有很好的预测效果;并且将其与Lyu提出的4种线性综合模型进行了试验比较,结果表明基于泛函网络的软件可靠性多模型综合预测方法有更优的预测性能。

参考文献

- [1] Lyu M R, Nikora A. Applying reliability models more effectively [J]. IEEE Software, 1992, 9(4): 43-52
- [2] Phame H. Reliability Handbook [S]. New York: Springer Verlag, 2002
- [3] 耿技, 聂鹏, 秦志光. 软件可靠性模型现状与研究[J]. 电子科技大学学报, 2013, 42(4): 565-570
Geng Ji, Nie Peng, Qin Zhi-guang. Status and Research of Software Reliability Models[J]. Journal of University of Electronic Science and Technology of China, 2013, 42(4): 565-570
- [4] Karunanithi N, Whitley D, Malaiya Y K. Using neural networks in reliability prediction[J]. IEEE Software, 1992, 9(4): 53-59
- [5] 乔辉, 周雁舟, 邵楠, 等. 基于AGA-LVQ神经网络的软件可靠性预测模型研究[J]. 计算机科学, 2013, 40(1): 179-182
Qiao Hui, Zhou Yan-zhou, Shao Nan, et al. Research of Software Reliability Prediction Model Based on AGA-LVQ[J]. Computer Science, 2013, 40(1): 179-182
- [6] Su Yu-shen, Huang Chin-yu. Neural-network-based approaches for software reliability estimation using dynamic weighted com-

- bination models[J]. Journal of Systems and Software, 2007, 80(4): 606-615
- [7] 刘遒, 郭立红, 肖辉, 等. 基于参数动态调整的动态模糊神经网络的软件可靠性增长模型[J]. 计算机科学, 2013, 40(2): 186-190
Liu Luo, Guo Li-hong, Xiao Hui, et al. Software Reliability Growth Model Based on Dynamic Fuzzy Neural Network with Parameters Dynamic Adjustment [J]. Computer Science, 2013, 40(2): 179-182
- [8] 张柯, 张德平, 汪帅. 软件可靠性混沌神经网络模型[J]. 计算机科学, 2014, 41(4): 172-177
Zhang Ke, Zhang De-ping, Wang Shuai. Chaotic Neural Network Model for Software Reliability[J]. Computer Science, 2014, 41(4): 172-177
- [9] Yang Bo, Li Xiang, Xie Min. A generic data-driven software reliability model with mining technique[J]. Reliability Engineering and System Saft, 2010, 95: 671-678
- [10] 崔正斌, 汤光明, 乐峰. 遗传优化支持向量机的软件可靠性预测模型[J]. 计算机工程与应用, 2009, 45(36): 71-74
Cui Zheng-bin, Tang Guang-ming, Yue Feng. Software reliability prediction model based on support vector machine optimized by genetic algorithm[J]. Computer Engineering and Applications, 2009, 45(36): 71-74
- [11] 侯雪梅, 高飞, 宋瑞丽, 等. 基于量子粒子群的软件模糊可靠性分配模型[J]. 信息工程大学学报, 2013, 14(1): 124-128
Hou Xue-mei, Gao Fei, Song Rui-li, et al. Software Fuzzy Reliability Allocation Based on Quantum Particle Swarm Algorithm [J]. Journal of Information Engineering University, 2013, 14(1): 124-128
- [12] Cai Kai-yuan, Cai Lin, Wang Wei-dong, et al. On the neural network approach in software reliability modeling[J]. The Journal of Systems and Software, 2001, 58(1): 47-62
- [13] Castillo E. Functional Networks[J]. Neural Processing Letters, 1998, 7(3): 151-159
- [14] Castillo E, Cobo A, Gomez-Nesterkin R, et al. A general framework for functional networks[J]. Networks, 2000, 35(1): 70-82
- [15] Castillo E, Gutiérrez J M. Nonlinear time series modeling and prediction using functional networks. Extracting Information M asked by Chaos[J]. Physics Letter Apply, 1998, 244(1): 71-84
- [16] Iglesias A, Arcay B, Ctos J M, et al. A Comparison between Functional Networks and Artificial Neural Networks for the Prediction of Fishing Catches[J]. Neural computer&applied, 2004, 13: 24-31
- [17] 崔强, 武春友, 匡海波. BP-DEMATEL 在空港竞争力影响因素识别中的应用[J]. 系统工程理论与实践, 2013, 33(6): 1471-1478
Cui Qiang, Wu Chun-you, Kuang Hai-bo. Influencing factors research of airports competitiveness based BP-DEMATEL model [J]. Systems Engineering-Theory & Practice, 2013, 33(6): 1471-1478
- [18] 崔春生. 基于泛函网络的组合推荐算法[J]. 系统工程理论与实践, 2014, 34(4): 1034-1042
Cui Chun-sheng. Hybrid recommendation based on functional network[J]. Systems Engineering-Theory & Practice, 2014, 34(4): 1034-1042
- [19] 徐仁佐, 谢旻, 郑人杰. 软件可靠性模型及应用[M]. 北京: 清华大学出版社, 1994: 125-129
Xu Ren-zuo, Xie Min, Zheng Ren-jie. Software Reliability Model and Application [M]. Beijing: Tsinghua University press, 1994: 125-129
- [20] Lyu M R, Nikora A. CASRE-A computer-aided software reliability estimation tool[C]// Computer-Aided Software Engineering Proceedings. 1992: 264-275
- [21] Vladicescu F P. Performance evaluation of computers courses technical, University of Denmark Informatics and Mathematical Modeling[OL]. 2006 [2014-3-11]. <http://www.imm.dtu.dk/popentiu/pec/pec.html>
- [22] Castillo E, Gutiérrez J M, Cobo A. A minimax method for learning functional networks[J]. Neur Proc Lett, 2000, 1(1): 39-49
- [23] Malaiya Y K, Karunanithi N. Predictability measures for software reliability models[C]// Fourteenth Annual International Computer Software and Applications Conference, 1990 (COMP-SAC 90). 1990, 1: 7-12
- [24] 王涛, 杨娟, 陈代国. 基于贝叶斯网络的构件软件体系建模及可靠性研究[J]. 四川兵工学报, 2009, 30(5): 4-7, 13

(上接第 146 页)

- [14] Wang C, Wang Q, Ren K, et al. Privacy-preserving public auditing for data storage security in cloud computing[C]//Proceeding of IEEE INFOCOM. 2010: 1-9
- [15] The format of the digital certificate[EB/OL]. 2009-5-21 [2010-9-29]. <http://baike.baidu.com/view/356572.htm>
- [16] Woo J. Market Basket Analysis algorithms with MapReduce [J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2013, 3(6): 445-452
- [17] Qin J, Zhang Y P, Zong P. Research on Data Destruction Mechanism with Security Level in HDFS[J]. Advanced Materials Research, 2014, 834: 1795-1798
- [18] Shen H Y, Li Z, Li J. A DHT-Aided Chunk-Driven Overlay for Scalable and Efficient Peer-to-Peer Live Streaming [J]. IEEE Transactions on Parallel and Distributed Systems, 2013, 11(11): 2125-2137
- [19] Zou G L, Ma J J. The Data Self-Destruction Technology Research Used in Marine Environmental Monitoring Buoy System [J]. Applied Mechanics and Materials, 2013, 336: 2195-2198
- [20] Croft J, Signorile R. Secure distribution of confidential information via self-destructing data[C]//Proceedings of the 8th World Scientific and Engineering Academy and Society (WSEAS) International Conference on Data Networks, Communications, Computers Baltimore, USA, 2009
- [21] 郑光, 苏锦海, 孙万忠. 闪存数据应急销毁算法的研究与设计 [J]. 计算机应用与软件, 2013, 30(9): 305-308
Zheng Guang, Su Jin-hai, Sun Wan-zhong. Research and design of flash memory data emergency disposal algorithm[J]. Computing Applications and Software, 2013, 30(9): 305-308