(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 108563556 A
(43)申请公布日 2018.09.21

(21)申请号 201810021357.6

(22)申请日 2018.01.10

(71)申请人 江苏工程职业技术学院
地址 226000 江苏省南通市青年中路87号

(72)发明人 曲豫宾 李芳 陈翔 谢萍丽

(51)Int.Cl.
G06F 11/36(2006.01)
G06N 3/00(2006.01)

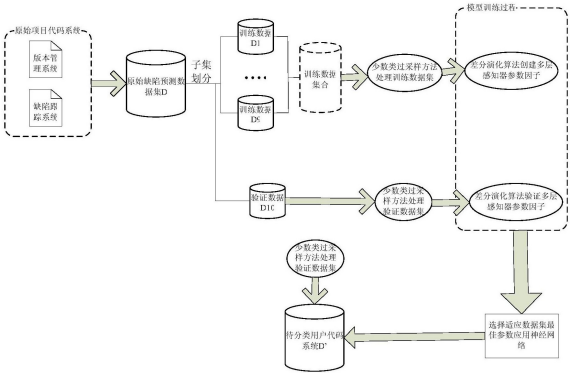
权利要求书2页 说明书6页 附图3页

(54)发明名称

基于差分演化算法的软件缺陷预测优化方法

(57)摘要

本发明公开了一种基于差分演化算法的软件缺陷预测优化方法，属于软件工程中的质量保证领域。包括如下步骤：(1)整理软件项目中模块，清洗代码中的注释等，建立软件缺陷数据代码集合；对给定缺陷集合进行整理，包括缺陷的度量设计以及缺陷数据标记等，生成软件缺陷数据集；借助差分演化算法，对缺陷预测数据集使用少数类过采样方法创建多数类和少数类比值为2:1的数据集，确定神经网络超参数的最优取值，使用训练的神经网络分类模型在测试集中测试，满足性能指标，则表示构建成功软件缺陷预测模型。本发明可以根据数据集的不同自动化确定分类模型构建中的相应参数因子，找到最适合当前数据集与分类模型的参数组合，提升软件缺陷预测模型的性能并减少模型构建中参数寻找的工作量。



CN 108563556 A

1.一种基于差分演化算法的软件缺陷预测优化方法,其特征在于:包括如下步骤:

步骤1)缺陷预测数据集的搜集:挖掘软件项目的版本管理系统和缺陷跟踪系统,从中抽取程序模块;随后对上述每个程序模块,通过分析缺陷跟踪系统内的缺陷报告信息进行标记;最后基于软件代码复杂度或软件开发过程分析,设计出与软件缺陷存在相关性的度量元,并借助这些度量元完成对每个程序模块的度量;通过对程序模块进行类型标记和软件度量,生成缺陷预测数据集D;

步骤2)借助差分演化算法,对缺陷预测数据集使用分层抽样及少数类过采样方法创建多数类和少数类比值的数据集,确定神经网络超参数的最优取值;

步骤3)基于缺陷预测数据集,使用少数类过采样方法,使用超参数取值方案,构建出缺陷预测模型。

2.根据权利要求1所述的一种基于差分演化算法的软件缺陷预测优化方法,其特征在于:步骤1)所述程序模块的粒度根据缺陷预测的目的设置为文件、包、类或函数。

3.根据权利要求1所述的一种基于差分演化算法的软件缺陷预测优化方法,其特征在于:步骤2)所述多数类和少数类比值为2:1。

4.根据权利要求1所述的一种基于差分演化算法的软件缺陷预测优化方法,其特征在于:步骤2)中分层抽样及少数类过采样方法,包括如下步骤:

2-1)将缺陷预测数据集按照分层抽样的方式划分,缺陷预测数据集里面多数类个数与少数类个数的比值为IR,缺陷预测数据集的个数为M,进行十折划分时候,每个子集的个数为M/10。对缺陷预测数据集按照少数类与多数类进行分组,在抽样时候,对于第i个子数据集,从多数类的样本中抽取M/10*(IR/(IR+1))个多数类,同时从少数类样本中抽取M/10*(1/(IR+1))个少数类,组合成为新的缺陷预测子集。重复以上过程九次。对于第10个子数据集,把所有抽样剩余的数据集作为第10个数据集。最终得到分层抽样的十个子数据集;

2-2)将训练数据D1到D9合并为一个训练集,在该训练集中使用少数类过采样算法来解决缺陷预测数据类不平衡问题,具体方法步骤如下:

2-2-1)从缺陷预测数据集D中选择有缺陷的少数类集合,设定为S_{min},$S_{min}\in D$,在S_{min}集合中选择一个样本X_i;

2-2-2)根据欧式距离计算出来与样本X_i距离最近的六个样本;

2-2-3)计算随机值ζ,ζ取值范围为(0,1),在离X_i最近的六个样本中随机取出一个X_{ik},k取值范围为(1,6),以此计算生成的新样本X_{new},$X_{new}=X_i+\zeta*(X_{ik}-X_i)$;

2-2-4)重复以上步骤中的2-2-1),2-2-2),2-2-3),直到缺陷预测数据集中的多数类与少数类直接的比值为2:1终止循环。

5.根据权利要求1所述的一种基于差分演化算法的软件缺陷预测优化方法,其特征在于:步骤2)中差分演化算法包括如下步骤:

3-1)随机初始化种群:以神经网络为缺陷预测模型的分类器,需要随机初始化学习速率learningRate,候选取值为0.1到0.4,隐藏层的层数hiddenLayers,候选取值范围为3到10,迭代次数epoch,候选取值为100到300,选择代理的规模NP>=4;根据确定的NP数目初始化得到一组染色体,初始化交叉因子CR,设定CR等于0.5,设定缩放因子F等于2;

3-2)计算种群中每个染色体的适应值:使用十折交叉验证法计算当前训练数据集上的相应因子的AUC值,该值作为该染色体的适应值;

3-3)染色体变异:从代理候选集合里面随机选择一个染色体X,并随机选择其他三个染色体,分别为A,B,C,这三个染色体与染色体X都互不相同;

3-4)染色体交叉:设定染色体的维度为N,随机选择一个整数n作为当前染色体交叉的因子,n的取值范围为0到N-1,如果n等于0,或者均匀分布随机变量小于交叉因子CR,那么就计算分量:

$y_0=a_0+F*(b_0-c_0)$

如果n等于1,或者均匀分布随机变量小于交叉因子CR,那么就计算分量:

$y_1=a_1+F*(b_1-c_1)$

如果n等于2,或者均匀分布随机变量小于交叉因子CR,那么就计算分量:

$y_2=a_2+F*(b_2-c_2)$

3-5)染色体选择:当前变异交叉后产生的染色体为(y_0,y_1,y_2),依据该染色体在神经网络中使用十折交叉验证法计算当前训练数据集上的相应因子的AUC值,该值作为该染色体的函数值,将该染色体与X染色体的函数值比较,如果新产生染色体的AUC值大于X染色体的AUC值,则删除染色体种群的中的X染色体,添加当前染色体到种群中去;

3-6)重复执行步骤3-3,3-4)及3-5),直到满足算法的终止条件为止,算法的终止条件是或者迭代次数达到100次,或者是最大的AUC值大于0.9,在一次循环过程中,步骤3-3),3-4)及3-5)重复执行染色体的规模NP次,完成对所有染色体的随机变异。

6.根据权利要求5所述的一种基于差分演化算法的软件缺陷预测优化方法,其特征在于:步骤3-2)的AUC值具体包括如下步骤:

步骤3-2-1)将数据集借助分层抽样方法划分为10份,i=1,对缺陷预测数据集D进行切分,按照层次抽样的方式进行分割,划分为10个数据集,随机选择其中9个数据集合并作为训练数据集,剩余的1个数据集作为验证数据集,分别对训练数据集和验证数据集使用少数类过采样算法进行预处理,获得待处理数据集;

步骤3-2-2)将第i份设置为测试集,将剩余数据设置为训练集;

步骤3-2-3)在训练集上使用少数类过采样方法,使用染色体对应的超参数取值训练模型,并在测试集上得到测试结果,记为AUC_i;

步骤3-2-4)将i取值加1,跳转到步骤2-2-2,当i取值大于10的时候,跳转到步骤2-2-5);

步骤3-2-5)将10次预测结果取均值,$AUC=(AUC_1+AUC_2+AUC_3+AUC_4+AUC_5+AUC_6+AUC_7+AUC_8+AUC_9+AUC_{10})/10$,作为该染色体对应的适应值。

基于差分演化算法的软件缺陷预测优化方法

技术领域

[0001] 本发明属于软件质量保障领域，具体涉及一种基于差分演化算法的软件缺陷预测优化方法。

背景技术

[0002] 软件组织的有限的软件质量保证资源更多的关注软件模块中的bug，比如源代码更有可能出现缺陷。因此，缺陷检测使用统计方法或者是机器学习的方法来识别源代码中可能存在的错误。这些机器学习中的分类器需要有提前设置的超参数，这些超参数没有启发式的规则可用，很多分类器使用默认的超参数。比如说随机森林中的决策树的数目就需要提前做配置，这个超参数就无法通过数据建模方式获得。目前可用的方法就是在超参数空间中寻找与当前数据集最匹配的参数，来建立分类器的模型。

[0003] 超参数空间搜索方法存在如下问题：(1)超参数空间巨大，要完成整个超参数空间的搜索几乎是不可能完成的任务；(2)超参数空间搜索无法做到自动化处理。超参数空间的搜索与模型建立有直接关联，没有成熟的方法提出如何来直接在超参数搜索空间中找到参数以后直接与现有的模型训练结合起来。(3)超参数空间在搜索过程中往往会遇到缺陷预测数据集中的类不平衡问题。如何将两者有效的结合来一起解决也没有成熟的方案可以借鉴。

[0004] 目前已经有使用网格搜索(grid search)方法搜索超参数的空间，该方法对待搜索的参数空间进行组合，使用模型计算该组合，设定停止标准，找到最优参数解以后则停止搜索。基于差分演化算法的超参数空间搜索是一种遗传算法的变异搜索方式，这个方法能够借助遗传学中的多样性，在不同的染色体序列直接进行差分变异，快速找到最优解。在其他领域经过实践，该方法可以用于寻找最优解。

[0005] 综上所述，为有效解决自动化的超参数搜索空间搜索问题和类不平衡问题，有必要设计出一种有效的基于差分演化算法的软件缺陷预测优化方法。本发明由此而生。

发明内容

[0006] 发明目的：本发明的目的是为了解决现有技术中的不足，提供一种基于差分演化算法的软件缺陷预测优化方法。

[0007] 技术方案：本发明所述的一种基于差分演化算法的软件缺陷预测优化方法，包括如下步骤：

[0008] 步骤1)缺陷预测数据集的搜集：挖掘软件项目的版本管理系统和缺陷跟踪系统，从中抽取程序模块；随后对上述每个程序模块，通过分析缺陷跟踪系统内的缺陷报告信息进行标记；最后基于软件代码复杂度或软件开发过程分析，设计出与软件缺陷存在相关性的度量元，并借助这些度量元完成对每个程序模块的度量；通过对程序模块进行类型标记和软件度量，生成缺陷预测数据集D；

[0009] 步骤2)借助差分演化算法，对缺陷预测数据集使用分层抽样及少数类过采样方法

创建多数类和少数类比值的数据集，确定神经网络超参数的最优取值；

[0010] 步骤3)基于缺陷预测数据集，使用少数类过采样方法，使用超参数取值方案，构建出缺陷预测模型。

[0011] 进一步的，步骤1)所述程序模块的粒度根据缺陷预测的目的设置为文件、包、类或函数。

[0012] 进一步的，步骤2)所述多数类和少数类比值为2:1。

[0013] 进一步的，步骤2)中分层抽样及少数类过采样方法，包括如下步骤：：

[0014] 2-1)将缺陷预测数据集按照分层抽样的方式划分，缺陷预测数据集里面多数类个数与少数类个数的比值为IR，缺陷预测数据集的个数为M，进行十折划分时候，每个子集的个数为M/10。对缺陷预测数据集按照少数类与多数类进行分组，在抽样时候，对于第i个子数据集，从多数类的样本中抽取M/10*(IR/(IR+1))个多数类，同时从少数类样本中抽取M/10*(1/(IR+1))个少数类，组合成为新的缺陷预测子集。重复以上过程九次。对于第10个子数据集，把所有抽样剩余的数据集作为第10个数据集。最终得到分层抽样的十个子数据集；

[0015] 2-2)将训练数据D1到D9合并为一个训练集，在该训练集中使用少数类过采样算法来解决缺陷预测数据类不平衡问题，具体方法步骤如下：

[0016] 2-2-1)从缺陷预测数据集D中选择有缺陷的少数类集合，设定为S_{min}，$S_{min}\in D$，在S_{min}集合中选择一个样本X_i；

[0017] 2-2-2)根据欧式距离计算出来与样本X_i距离最近的六个样本；

[0018] 2-2-3)计算随机值ζ，ζ取值范围为(0,1)，在离X_i最近的六个样本中随机取出一个X_{ik}，k取值范围为(1,6)，以此计算生成的新样本X_{new}，$X_{new}=X_i+\zeta*(X_{ik}-X_i)$；

[0019] 2-2-4)重复以上步骤中的2-2-1)，2-2-2)，2-2-3)，直到缺陷预测数据集中的多数类与少数类直接的比值为2:1终止循环。

[0020] 进一步的，步骤2)中差分演化算法包括如下步骤：

[0021] 3-1)随机初始化种群：以神经网络为缺陷预测模型的分类器，需要随机初始化学习速率learningRate，候选取值为0.1到0.4，隐藏层的层数hiddenLayers，候选取值范围为3到10，迭代次数epoch，候选取值为100到300，选择代理的规模NP>＝4；根据确定的NP数目初始化得到一组染色体，初始化交叉因子CR，设定CR等于0.5，设定缩放因子F等于2；

[0022] 3-2)计算种群中每个染色体的适应值：使用十折交叉验证法计算当前训练数据集上的相应因子的AUC值，该值作为该染色体的适应值；

[0023] 3-3)染色体变异：从代理候选集合里面随机选择一个染色体X，并随机选择其他三个染色体，分别为A，B，C，这三个染色体与染色体X都互不相同；

[0024] 3-4)染色体交叉：设定染色体的维度为N，随机选择一个整数n作为当前染色体交叉的因子，n的取值范围为0到N-1，如果n等于0，或者均匀分布随机变量小于交叉因子CR，那么就计算分量：

[0025] $y_0=a_0+F*(b_0-c_0)$

[0026] 如果n等于1，或者均匀分布随机变量小于交叉因子CR，那么就计算分量：

[0027] $y_1=a_1+F*(b_1-c_1)$

[0028] 如果n等于2，或者均匀分布随机变量小于交叉因子CR，那么就计算分量：

[0029] $y_2=a_2+F*(b_2-c_2)$

[0030] 3-5)染色体选择:当前变异交叉后产生的染色体为(y_0,y_1,y_2),依据该染色体在神经网络中使用十折交叉验证法计算当前训练数据集上的相应因子的AUC值,该值作为该染色体的函数值,将该染色体与X染色体的函数值比较,如果新产生染色体的AUC值大于X染色体的AUC值,则删除染色体种群的中的X染色体,添加当前染色体到种群中去;

[0031] 3-6)重复执行步骤3-3,3-4)及3-5),直到满足算法的终止条件为止,算法的终止条件是或者迭代次数达到100次,或者是最大的AUC值大于0.9,在一次循环过程中,步骤3-3),3-4)及3-5)重复执行染色体的规模NP次,完成对所有染色体的随机变异。

[0032] 进一步的,步骤3-2)的AUC值具体包括如下步骤:

[0033] 步骤3-2-1)将数据集借助分层抽样方法划分为10份,i＝1,对缺陷预测数据集D进行切分,按照层次抽样的方式进行分割,划分为10个数据集,随机选择其中9个数据集合并作为训练数据集,剩余的1个数据集作为验证数据集,分别对训练数据集和验证数据集使用少数类过采样算法进行预处理,获得待处理数据集;

[0034] 步骤3-2-2)将第i份设置为测试集,将剩余数据设置为训练集;

[0035] 步骤3-2-3)在训练集上使用少数类过采样方法,使用染色体对应的超参数取值训练模型,并在测试集上得到测试结果,记为AUC_i;

[0036] 步骤3-2-4)将i取值加1,跳转到步骤2-2-2,当i取值大于10的时候,跳转到步骤2-2-5);

[0037] 步骤3-2-5)将10次预测结果取均值,$AUC＝(AUC_1+AUC_2+AUC_3+AUC_4+AUC_5+AUC_6+AUC_7+AUC_8+AUC_9+AUC_{10})/10$,作为该染色体对应的适应值。

[0038] 有益效果:本发明可以根据数据集的不同自动化确定分类模型构建中的相应参数因子,找到最适合当前数据集与分类模型的参数组合,提升软件缺陷预测模型的性能并减少模型构建中参数寻找的工作量。

附图说明

[0039] 图1是本发明的总体流程图;

[0040] 图2是十折交叉验证计算染色体的适应值AUC的过程图;

[0041] 图3为AUC具体计算过程图。

具体实施方式

[0042] 为了更详尽的表述上述发明的技术路线,以下本发明人列举出具体的实施例来说明技术效果;需要强调的是,这些实施例是用于说明本发明而不限于限制本发明的范围。

[0043] 实施例1

[0044] 本实施例的基于差分演化算法的软件缺陷预测优化方法的总体流程图如图1所示,包含如下步骤:

[0045] (1)挖掘软件项目的业务软件系统和缺陷跟踪系统,从中抽取程序模块;所述程序模块的粒度根据缺陷预测的目的设置为文件、包、类或函数;随后对上述每个程序模块,通过分析缺陷跟踪系统内的缺陷报告信息进行标记;最后基于软件代码复杂度或软件开发过程分析,设计出与软件缺陷存在相关性的度量元,并借助这些度量元完成对每个程序模块的度量;通过对程序模块进行类型标记和软件度量,生成缺陷预测数据集D。

[0046] 若将数据集存储为Weka软件支持的格式，则来自某一实际项目的缺陷预测数据集的具体内容如下所示(其中//后面是注释)。

[0047] //下面列出了数据集考虑的度量元名称以及类型，最后一个是模块的类型(若取值为Y表示是有缺陷模块，否则取值为N表示是无缺陷模块)：

[0048] @relation EQ//数据集的名称

[0049] @attribute ck_oo_numberOfPrivateMethods numeric

[0050] @attribute LDHH_lcom numeric

[0051] fMethodsInherited numeric

[0052] @attribute numberOfBugsFoundUntil:numeric

[0053] @attribute LDHH_fanOut numeric

[0054] ......(省略)

[0055] @attribute LDHH_numberOfMethodsInherited numeric

[0056] @attribute LDHH_rfc numeric

[0057] @attribute ck_oo_numberOfMethodsInherited numeric

[0058] @attribute ck_oo_numberOfPublicMethods numeric

[0059] @attribute LDHH_cbo numeric

[0060] @attribute WCHU_numberOfLinesOfCode numeric

[0061] @attribute CvsExpEntropy numeric

[0062] @attribute LDHH_numberOfMethods numeric

[0063] @attribute Defective{Y,N}

[0064] @data

[0065] 3,0.002547,0.002555,4,0,3.04,0.393707,0.003049,1.01,0.004091,10.1322,0,1.01,2.05,0,0,2,33,0,3.04,0.308139,0,0,3.1,5,125,0,11,1,2,0,0,55,0,36,0.004387,0.004425,0.004354,2.04,2.4,7,3.22,11,0,11,2,2.04,0,3.05,0.043005,1.01,4,0.005627,0,0.004406,8,8,0.008431,3.5,0.103594,0.003611,Y

[0066] 37,0.008643,0.004756,71,0,14.37,2.09375,0.001481,2.02,0.015332,38.2812,0,13.18,14.23,1,0,2,300,0.010354,11.3,1.12823,0,0,34.22,53,1217,0,48,1,1,0,0,1128,0,605,0.041329,0.015698,0.061578,16.3,23.21,54,39.75,67,4,38,11,3.03,0,16.31,0.149026,1.01,78,0.018761,0.001486,0.060301,7,7,0.021602,43.12,0.328692,0.009906,Y

[0067] 3,0.001479,0.009143,5,1.01,3.08,0.484675,0,6.17,0.001953,9.49506,0.00026,2.03,2.05,0,6,0,48,0.001951,2.04,0.287726,2.06,0,4.38,16,214,0,8,1,11,0,0,28,0,79,0.00466,0.002373,0.002221,3.12,2.25,26,3.64,7,0,4,1,0,0,8.28,0.043268,1.01,5,0.003117,0.001486,0.002325,7,2,0.011859,4.68,0.125841,0.001655,N

[0068] 1,0.00135,0,1,1.01,1.01,0.03194,0.000876,0,0,7.55094,0.000402,1.01,2.02,1,0,0,10,0.000577,0,0.204387,0,0,2.04,1,31,1,8,2,1,0,0,28,0,12,0.001433,0.000615,0.001485,1.01,2.13,2,2.06,1,0,0,0,1.01,0,1.01,0.02254,3.05,1,0.000566,0.003017,0.001492,22,7,0.000652,2.13,0.055912,0.001572,Y

[0069] 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,7,0,0,0,0,0,N

[0070] 8,0.000667,0,7,0,2.04,0.748927,0,0,0.002712,7.74631,0,1.06,1.06,0,0,2,49,0,2.04,0.255544,0,0,5.32,16,198,29,13,2,0,0,0,78,4.06,118,0.008936,0.00262,0.00876,2.06,1.93,16,4.51,6,0,6,3,0,0,2.06,0.039137,1.01,7,0.008318,0,0.004431,24,2,0.008008,6.08,0.117569,0.000473,Y

[0071] 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,2,26,0,2,1,0,0,0,1,0,18,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,1,0,0,0,9,2,0,0,0,0,N.....(省略)。

[0072] 2)借助差分演化算法,对缺陷预测数据集使用分层抽样及少数类过采样方法创建多数类和少数类比值为2:1的数据集,确定神经网络超参数的最优取值。

[0073] 3)基于缺陷预测数据集,使用少数类过采样方法,使用超参数取值方案,构建出缺陷预测模型。

[0074] 步骤2)中分层抽样及少数类过采样方法,包括如下步骤::

[0075] 2-1)将缺陷预测数据集按照分层抽样的方式划分,缺陷预测数据集里面多数类个数与少数类个数的比值为IR,缺陷预测数据集的个数为M,进行十折划分时候,每个子集的个数为M/10。对缺陷预测数据集按照少数类与多数类进行分组,在抽样时候,对于第i个子数据集,从多数类的样本中抽取M/10*(IR/(IR+1))个多数类,同时从少数类样本中抽取M/10*(1/(IR+1))个少数类,组合成为新的缺陷预测子集。重复以上过程九次。对于第10个子数据集,把所有抽样剩余的数据集作为第10个数据集。最终得到分层抽样的十个子数据集;

[0076] 2-2)将训练数据D1到D9合并为一个训练集,在该训练集中使用少数类过采样算法来解决缺陷预测数据类不平衡问题,具体方法步骤如下:

[0077] 2-2-1)从缺陷预测数据集D中选择有缺陷的少数类集合,设定为S_{min},$S_{min}\in D$,在S_{min}集合中选择一个样本X_i;

[0078] 2-2-2)根据欧式距离计算出来与样本X_i距离最近的六个样本;

[0079] 2-2-3)计算随机值ζ,ζ取值范围为(0,1),在离X_i最近的六个样本中随机取出一个X_{ik},k取值范围为(1,6),以此计算生成的新样本X_{new},$X_{new}=X_i+\zeta*(X_{ik}-X_i)$;

[0080] 2-2-4)重复以上步骤中的2-2-1),2-2-2),2-2-3),直到缺陷预测数据集中的多数类与少数类直接的比值为2:1终止循环。

[0081] 所述步骤(2)中采用差分演化算法按照以下步骤进行:

[0082] 3-1)随机初始化种群。以神经网络为缺陷预测模型的分类器,需要随机初始化学习速率learningRate,候选取值为0.1到0.4,隐藏层的层数hiddenLayers,候选取值范围为3到10。迭代次数epoch,候选取值为100到300。选择代理的规模NP>=4。根据确定的NP数目初始化得到一组染色体。初始化交叉因子CR,设定CR等于0.5,设定缩放因子F等于2。以(learningRate,hiddenLayers,epoch)作为神经网络的超参数向量染色体,差分演化算法过程参见图2。

[0083] 3-2)计算种群中每个染色体的适应值。依据该染色体在神经网络中使用十折交叉验证法计算当前训练数据集上的相应因子的AUC值,该值作为该染色体的函数值。随机初始化得到四个染色体为(0.2,5,200),(0.1,6,100),(0.4,8,300),(0.3,9,200),计算各个因子对应的验证集中的AUC值分别为0.6125,0.687,0.786,0.623。适应值AUC具体计算过程参

见图3。

[0084] AUC值具体包括如下步骤：

[0085] 步骤3-2-1)将数据集借助分层抽样方法划分为10份，i＝1，对缺陷预测数据集D进行切分，按照层次抽样的方式进行分割，划分为10个数据集，随机选择其中9个数据集合并作为训练数据集，剩余的1个数据集作为验证数据集，分别对训练数据集和验证数据集使用少数类过采样算法进行预处理，获得待处理数据集；

[0086] 步骤3-2-2)将第i份设置为测试集，将剩余数据设置为训练集；

[0087] 步骤3-2-3)在训练集上使用少数类过采样方法，使用染色体对应的超参数取值训练模型，并在测试集上得到测试结果，记为AUC_i；

[0088] 步骤3-2-4)将i取值加1，跳转到步骤2-2-2，当i取值大于10的时候，跳转到步骤2-2-5)；

[0089] 步骤3-2-5)将10次预测结果取均值，$AUC＝(AUC_1+AUC_2+AUC_3+AUC_4+AUC_5+AUC_6+AUC_7+AUC_8+AUC_9+AUC_{10})/10$，作为该染色体对应的适应值。

[0090] 3-3)染色体变异。从代理候选集合里面随机选择一个染色体X，假如说选中的是(0.1,6,100)，并随机选择其他三个染色体，分别为A,B,C，这三个染色体与染色体X都互不相同，其他三个染色体则分别为A＝(0.2,5,200)，B＝(0.4,8,300)，C＝(0.3,9,200)。

[0091] 3-4)染色体交叉。当前染色体的维度为3，随机选择一个整数n作为当前染色体交叉的因子，n的取值范围为0到2。如果n等于0，或者均匀分布随机变量小于交叉因子CR，那么就计算分量，

[0092] $y_0＝a_0+2*(b_0-c_0)＝0.2+2*(0.4-0.3)＝0.4$

[0093] 如果n等于1，或者均匀分布随机变量小于交叉因子CR，那么就计算分量，

[0094] $y_1＝a_1+2*(b_1-c_1)＝5+2*(8-9)＝3$

[0095] 如果n等于2，或者均匀分布随机变量小于交叉因子CR，那么就计算分量，

[0096] $y_2＝a_2+2*(b_2-c_2)＝200+2*(300-200)＝400$

[0097] 通过染色体交叉以后，生成了新的染色体为(0.4,3,400)。

[0098] 3-5)染色体选择。当前变异交叉后产生的染色体为(0.4,3,400)，依据该染色体在神经网络中使用十折交叉验证法计算当前训练数据集上的相应因子的AUC值，该值作为该染色体的函数值，将该染色体与X染色体的函数值比较，如果新产生染色体的AUC值大于X染色体的AUC值，则删除染色体种群的中的X染色体，添加当前染色体到种群中去。如果当前染色体的AUC输出值为0.8，0.8大于原有的染色体X的输出值0.687，则删除X，添加染色体(0.4,3,400)到当前染色体库中去。

[0099] 3-6)重复执行步骤3-3，3-4)及3-5)，直到满足算法的终止条件为止。算法的终止条件是或者迭代次数达到100次，或者是最大的AUC值大于0.9。在一次循环过程中，步骤3-3，3-4)及3-5)重复执行染色体的规模NP次，完成对所有染色体的随机变异。

[0100] 以上所述，仅是本发明的较佳实施例而已，并非对本发明作任何形式上的限制，虽然本发明已以较佳实施例揭露如上，然而并非用以限定本发明，任何熟悉本专业的技术人员，在不脱离本发明技术方案范围内，当可利用上述揭示的技术内容作出些许更动或修饰为等同变化的等效实施例，但凡是未脱离本发明技术方案的内容，依据本发明的技术实质对以上实施例所作的任何简单修改、等同变化与修饰，均仍属于本发明技术方案的范围内。

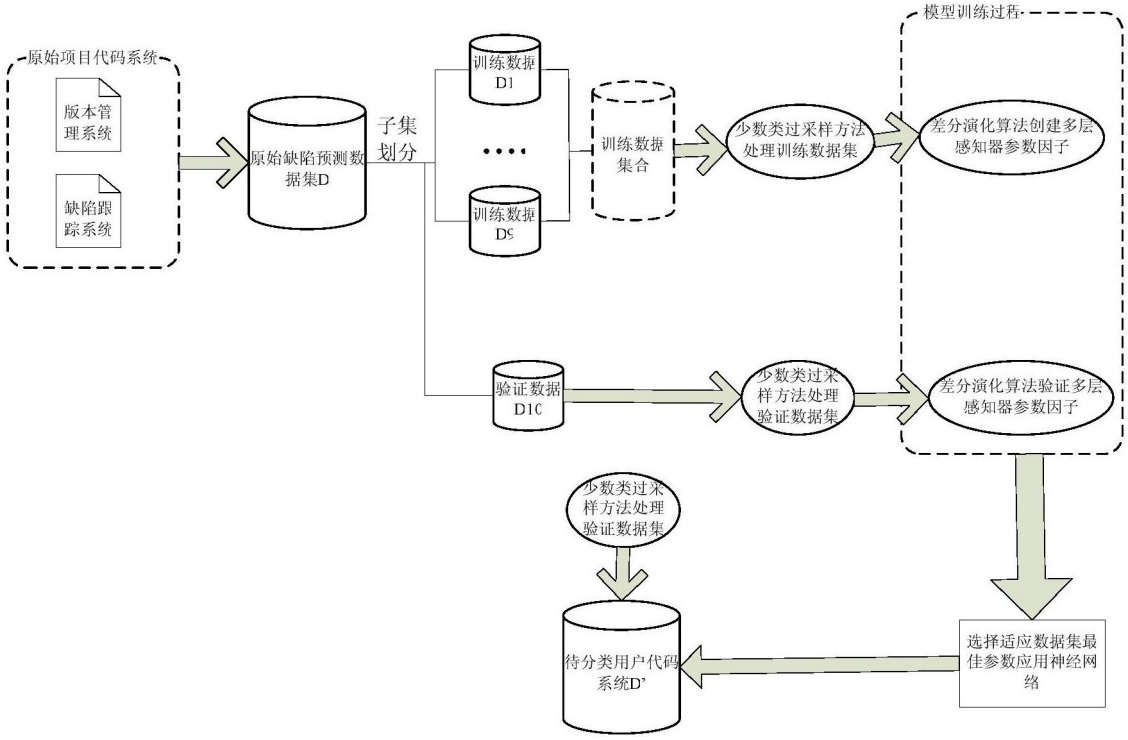

原始项目代码系统
版本管理系统
缺陷跟踪系统
原始缺陷预测数据集D
子集划分
训练数据D1
训练数据Ds
训练数据集合
少数类过采样方法处理训练数据集
模型训练过程
差分演化算法创建多层感知器参数因子
验证数据D1c
少数类过采样方法处理验证数据集
差分演化算法验证多层感知器参数因子
少数类过采样方法处理验证数据集
待分类用户代码系统D'
选择适应数据集最佳参数应用神经网络

图1

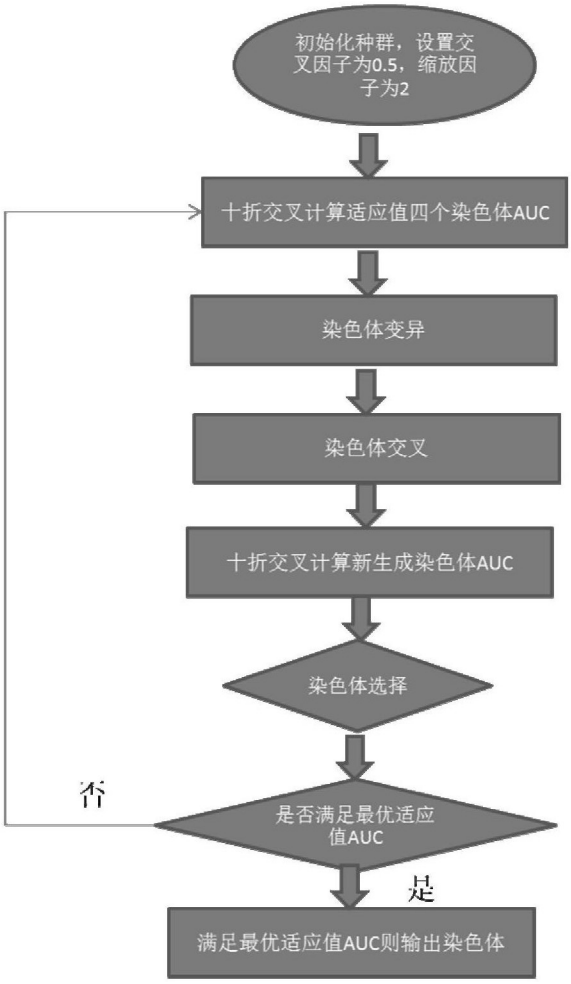
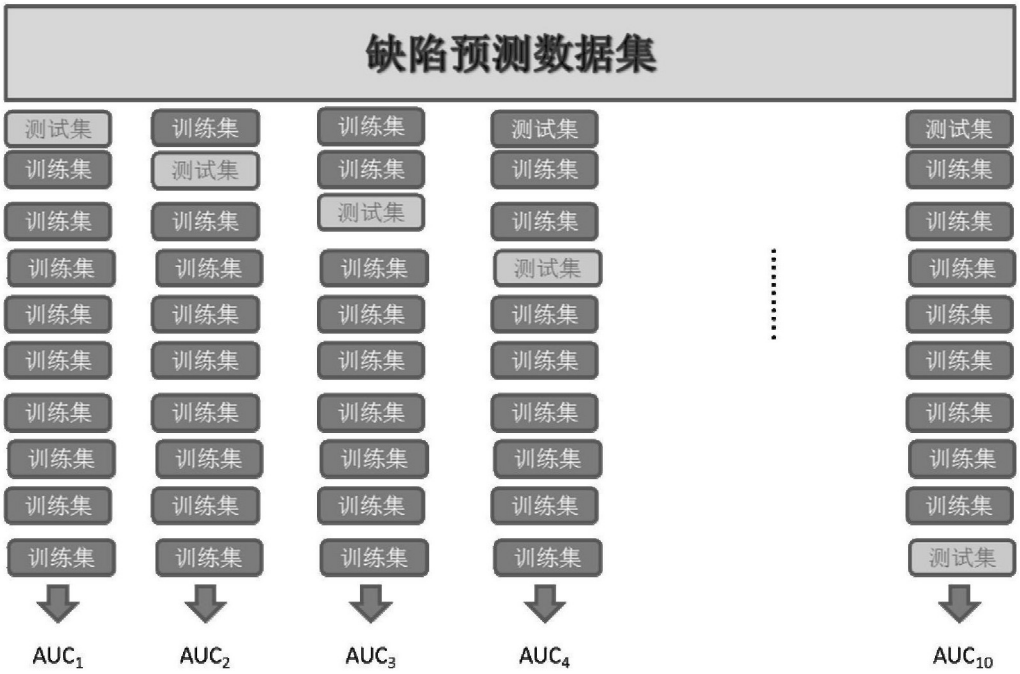

初始化种群，设置交叉因子为0.5，缩放因子为2
十折交叉计算适应值四个染色体AUC
染色体变异
染色体交叉
十折交叉计算新生成染色体AUC
染色体选择
否
是否满足最优适应值AUC
是
满足最优适应值AUC则输出染色体

图2



$$AUC = (AUC_1 + AUC_2 + AUC_3 + AUC_4 + AUC_5 + AUC_6 + AUC_7 + AUC_8 + AUC_9 + AUC_{10}) / 10$$

图3