缺陷预测模型真的有效吗?

Umamaheswara Sharma B 计算机科学与工程系 印度瓦朗加尔国家理工学院
uma.phd@student.nitw.ac.in

Ravichandra Sadam 计算机科学与工程系 印度瓦朗加尔国家理工学院
ravic@nitw.ac.in

arXiv:2301.05411v1 [cs.SE] 13 Jan 2023

摘要——“你可以开发一个潜在的预测模型，但我如何相信你的模型能为我的软件带来益处？”我们将软件缺陷预测（SDP）模型用作工具，探讨机器学习研究中的这一根本问题。这是一项初步研究，旨在分析所开发的二元软件缺陷预测模型在实时工作环境中的表现。

索引词——软件缺陷预测、机器学习、概率界、实时分析。

I. 引言

由于复杂关键软件系统的快速发展，测试已成为一项艰巨挑战。软件缺陷预测（SDP）模型的开发旨在缓解这一挑战 [1]–[6]。开发 SDP 模型的主要目标是减少新开发软件项目[7]所需投入的测试时间、成本和精力。SDP 模型的任务是预测新开发软件模块的缺陷倾向性。

一旦开发出高效的 SDP 模型，任何组织都可以利用其服务。然而，机器学习（ML）文献表明，通常情况下，开发出的预测模型可能会在未见数据上产生误分类 [8]。由于预测模型产生误分类或测试无效，软件模块发生任何故障都可能引发从不便到生命损失等问题 [9]。当预测模型错误地将某个模块预测为有缺陷模块时，软件更有可能发生故障。

为了了解 SDP 模型在实时工作环境中的可行性，我们使用概率界进行理论分析。简而言之，我们通过计算随机变量（将其建模为采用 SDP 模型的软件的危险率）与人工测试软件的估计危险率之间的偏差，来证明相关结论。此外，我们还通过称为可靠性的度量给出证明。

II. 预备知识

我们首先根据 SDP 模型的预测，探究系统发生故障的可能性。系统可能以多种方式发生故障 [9], [10]。其中，最主要的一种可能情况是将有缺陷模块预测为无缺陷。在这种情况下，在实时工作环境中，测试人员可能会漏掉有缺陷模块。接下来，

以下假设可确保测试集中的每个假阴性模块都会导致故障事件：

假设 1. 对每个有缺陷模块的错误分类都可能导致软件发生一次故障。

这一假设使我们能够统计测试集和新开发项目中的故障总数。由于我们知道常规测试流程无法发现所有缺陷 [10], ，因此，以下假设确保任何使用 SDP 模型进行测试的软件中都存在故障：

假设 2. 集成测试、系统测试或验收测试无法发现被错误分类的有缺陷模块中的缺陷。

现在，为了衡量测试集上故障案例的发生比例，我们采用一种称为假遗漏率（FOR）的度量。假遗漏率是指假阴性总数与预测为无缺陷模块的总数之比。定义如下：

$$FOR = \frac{\text{False Negatives}}{\text{Predicted Cleans}} = \frac{\text{FN}}{\text{FN+TN}} \tag{1}$$

由于只有被预测为无缺陷模块的模块可能含有潜在缺陷，因此，假阴性率（FOR）非常适合用于估算测试集上故障发生的比例。然而，在实时测试中，假阴性（FN）无法提供有关软件故障的充分信息，因为被预测为无缺陷模块的实际类别标签未知。因此，我们将每个被预测为无缺陷模块的实际类别建模为一个随机变量。对于任何包含 n 个模块的新开发软件$S = \{M_1, M_2, \cdots, M_n\}$，假设$l$ 个模块被预测为属于无缺陷类别。现在，使用以下随机变量来表示被错误预测为有缺陷模块所导致的故障情况， M_i:

$$X_i = \begin{cases} 1, & \text{if the module } M_i \text{ is classified wrongly as clean} \\ 0, & \text{if the module } M_i \text{ is classified correctly as clean} \end{cases} \tag{2}$$

为了保证对于任意模块 M_i， X_i 都以相同的概率取 $\{0, 1\}$ 中的值，必须满足以下假设。

假设3。软件缺陷预测模型是在该软件项目的历史数据上训练的。

一般而言，SDP 模型基于软件项目的历史数据开发，并假设数据

分布相似，适用于训练集、测试集和总体集合[1]–[7]。根据假设3，由于软件缺陷预测模型不会动态变化，因此每个被预测为无缺陷的模块都以相似的假阴性率被归入错误类别。也就是说，假阴性率被视为每个被预测为无缺陷的模块被错分为有缺陷模块的概率。表示如下：

$$p = FOR \tag{3}$$

该概率用于定义软件项目的故障分布。因此，根据公式 2 和 3，随机变量 X_i 的概率分布表示为：

$$\mathbf{Pr}[X_i = 1] = p, \text{ and, } \mathbf{Pr}[X_i = 0] = 1 - p, \text{ for } 1 \leq i \leq l.$$

现在，为了统计预测模型中的故障实例总数，以下假设确保每个受测模块之间相互独立：

假设 4. 软件缺陷预测模型对相互独立的观测（软件模块）进行预测。

事实上，所有 SDP 模型都假设数据点 [1]–[7] 之间相互独立。由于每个预测为无缺陷的模块具有相同的概率，因此它构成一次伯努利试验 [11]。现在，l 次相同的伯努利试验之和服从二项分布 [11], [12]。表示如下：

$$X = \sum_{i=1}^{l} X_i \tag{4}$$

现在，随机变量 X 的均值推导如下（利用期望的线性性质）：

$$\mathbb{E}[X] = \mathbb{E}\Big[\sum_{i=1}^{l} X_i\Big] = \sum_{i=1}^{l} \mathbb{E}[X_i] = \sum_{i=1}^{l} \big[1 * \mathbf{Pr}[X_i = 1] + 0 * \mathbf{Pr}[X_i = 0]\big] = \sum_{i=1}^{l} p = lp \tag{5}$$

到目前为止，我们已将故障的发生（本文后文也称为危险率）建模为随机变量，并估算了软件中的预期故障数（对有缺陷模块的错误预测）。值得注意的是，不失一般性，预测为有缺陷的模块将由测试人员进行测试。以下假设确保软件发布后，部分软件中会出现故障：

假设5. 对于软件中除预测为无缺陷模块之外的某些部分，其危险率服从威布尔分布。

这里，危险率被定义为软件系统中故障的瞬时失效率 [9]。根据 Hartz 等人（见 [13]）的说法，软件（或软件的一部分）中的危险率可能无法用单个函数估算。因此，为了拟合各种危险率曲线，有必要研究一种称为威布尔分布的危险率模型形式。这里，我们假设威布尔分布适用于其余软件模块（预测为无缺陷模块的模块除外）的危险率。现在，对于同时使用 SDP 模型和测试人员进行测试的软件，其估算的总危险率（$\hat{z}(t)$）计算如下：

$$\hat{z}(t) = X + \hat{K}t^{\hat{m}}, \text{ for some } \hat{K} > 0, \hat{m} > -1, \text{ and time } t > 0 \tag{6}$$

这里，$\hat{z}(t)$表示使用SDP模型进行测试的软件的危险率（之后，预测为有缺陷的$\hat{}$模块由测试人员进行维护）。这里，Kt^m 假定为除预测为无缺陷的$\hat{}$模块之外的软件模块的危险率，并以威布尔分布表示。这里，参数 K、$\hat{m}$和 t 取实数值，这些参数的不等式约束直接采用Lyu的研究 [9]中的设定。因此，根据假设1和假设5，对于全部软件模块，所得的危险率模型($\hat{z}(t)$)是软件各子部分危险率之和。现在，推导出软件的预期危险率如下：

$$\mathbb{E}[\hat{z}(t)] = \mathbb{E}[X + \hat{K}t^{\hat{m}}] = \mathbb{E}[X] + \hat{K}t^{\hat{m}} = lp + \hat{K}t^{\hat{m}} \tag{7}$$

为证明SDP模型在实时场景中的可行性，必须满足以下假设：

假设6. 在使用 SDP 模型进行测试和手动测试这两种情况下，使用的是相同的软件。

这是证明 SDP 模型在实时场景中可行性的一项重要假设。对于某个软件，根据公式 6，我们知道其危险率为 $\hat{z}(t) = X + \hat{K}t^{\hat{m}}$。假设测试人员测试的是同一软件，且其危险率服从威布尔分布，如下所示：

$$z(t) = Kt^m, \text{ for some } K > 0, m > -1, \text{ and time } t > 0 \tag{8}$$

参数 K、m 和 t 的定义与公式 6 中参数的定义类似。请注意，在时间 t，两个风险函数 $z(t)$ 和 $\hat{z}(t)$ 分别描述了手动测试和使用 SDP 测试时软件的瞬时失效率。对于任意软件，第III节给出的证明为随机变量相对于手动测试估算出的相应危险率大幅偏低时的偏差提供了紧界。类似地，另一种可能的方法是求出随机变量（以可靠性表示）相对于人工测试软件可靠性大幅偏高时的偏差。这里，可靠性定义为软件在指定环境 [9] 中指定时间段内无故障运行的概率。

可靠性与危险率之间的关系如下 [9]:

$$R(t) = e^{-\int_0^t z(x)dx} \tag{9}$$

其中，$R(t)$ 是软件在时间 t 的可靠性，而 $z(t)$ 是危险率。利用公式9，我们可以推导出某个时间间隔 [0,t] 内多种风险模型的可靠性值。这里，我们假设这两个相同的软件系统

（测试场景不同——一个采用人工测试，另一个使用 SDP 进行测试）在时间 0 部署。

现在，手动测试软件的可靠性由危险率的威布尔模型 ($z(t)$) 定义。

引理1. 对于软件$z(t) = Kt^m$的威布尔危险率模型，存在某个 $K > 0$、$m > -1$ 以及时间 $t > 0$，其可靠性为：

$$R(t) = e^{\frac{-Kt^{(m+1)}}{m+1}}$$

证明. 将 $z(t)$的值（由公式8给出）代入 $R(t)$（即公式9）可得：

$$R(t) = e^{-\int_0^t Kx^m \, dx} \tag{10}$$

化简公式10即可满足该引理。 □

通过确保概率值 (p) 位于区间 (0,1) 内，第III节中的证明才成立。因此，必须满足以下假设：

假设7. SDP模型应在测试集上至少产生一个假阴性和一个真阴性。

III. 证明

A. 关于危险率的紧下界

在第II节中，我们将使用软件缺陷预测模型进行测试的软件中的危险（故障）实例数量建模为一个随机变量（即 $\hat{z}(t) = X + \hat{K}t^{\hat{m}}$）。现在，下面的定理定义了随机变量相对于人工测试软件的危险率值的偏差，$X + Kt^m$，该值低于危险率（事实上，远低于期望值，μ）。

定理 1. 设 $X_1, X_2, \ldots, X_l$ 为独立伯努利试验，使得对于 $1 \le i \le l$，$Pr[X_i = 1] = p$，其中 $0 < p < 1$。再设 $\exists$ 参数为 $K > 0, m > -1$，$K > 0, \hat{m} > -1$，时间为 $t > 0$。那么，对于 $\mathbf{X} = \sum_{i=1}^{l} X_i$，$\mathbb{E}[X + Kt^m] = Kt^m + lp$

$(t) = X + \hat{K}t^{\hat{m}}$，$\hat{K}\hat{K}\hat{K}$，以及人工测试软件的 Weibull 危险率模型，Kt^m：

$$Pr[X + \hat{K}t^{\hat{m}} < Kt^m] < e^{\frac{-(lp - Kt^m + 2\hat{K}t^{\hat{m}})^2}{2[\hat{K}t^{\hat{m}} + lp]}}$$

证明。和之前一样，$Pr[X + \hat{K}t^{\hat{m}} < Kt^m]$ 可以改写为：

$$Pr[X + \hat{K}t^{\hat{m}} < Kt^m] = Pr[X < Kt^m - \hat{K}t^{\hat{m}}] \tag{11}$$

我们知道，对于某个 $0 < \delta \le 1$ 和 μ，利用切尔诺夫界，偏离期望 μ 较远的独立伯努利试验之和 X 的下尾界[14]为：

$$Pr[X < (1 - \delta)\mu] < e^{\frac{-\mu\delta^2}{2}} \tag{12}$$

这里，数值 $(1 - \delta)\mu$ 表示区间长度为 δ 时，期望 μ 左侧的边际值。

现在，我们希望得到随机变量（即 $X + Kt^m$）的紧下界，该变量远低于人工测试的软件的危险率 Kt^m。在方程 11 中，对于某些 $K > 0$、$K > 0$、$m > -1$、$m > -1$ 和 $t > 0$，值 $Kt^m - Kt^m$ 被假定低于期望，μ，在给定的时间段 $[0, t]$ 内。现在，根据方程 11 和 12：

$$(1 - \delta)\mu = Kt^m - \hat{K}t^{\hat{m}}$$

$$\Rightarrow \delta = 1 - \frac{Kt^m - \hat{K}t^{\hat{m}}}{\mu} \tag{13}$$

根据公式 6，我们知道使用 SDP 模型的软件中的预期危险事件为：

$$\mu = \mathbb{E}[X + \hat{K}t^{\hat{m}}] = \hat{K}t^{\hat{m}} + lp$$

现在，将 δ（来自公式 13）和 μ 代入公式 12，可得随机变量（即 $X + Kt^m$）低于人工测试软件危险率的偏差紧下界。如下所示：

$$Pr[X + \hat{K}t^{\hat{m}} < Kt^m] < e^{\frac{-[\hat{K}t^{\hat{m}} + lp]\left[1 - \frac{Kt^m - \hat{K}t^{\hat{m}}}{\hat{K}t^{\hat{m}} + lp}\right]^2}{2}} \tag{14}$$

化简上述公式即可完成证明。 □

因此，根据定理 1，在使用 SDP 的软件中，发生的危险事件少于由人工测试的同一软件中的危险事件总数，其概率关于 l、K、$\hat{m}$ 和 t 呈指数级小，这意味着这些参数的取值越大，上界就越紧。

B. 可靠性方面的紧上界

本节给出一个引理，用于计算使用软件缺陷预测模型进行测试的软件的可靠性。

引理 2. 设 X_1、X_2、...、X_l 为相互独立的伯努利试验，另设 $\exists$ 个参数 $K > 0$、$\hat{m} > -1$、时间 $t > 0$。那么，对于 $\mathbf{X} = \sum_{i=1}^{l} X_i$ 和 $\hat{z}(t) = X + \hat{K}t^{\hat{m}}$，其可靠性为：

$$\hat{R}(t) = e^{-\left[Xt + \frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]}$$

证明. 根据公式 6，我们得到使用软件缺陷预测模型进行测试的软件中的危险事件。现在，将 $\hat{z}(t)$ 的值（来自公式 6）代入公式9，即得：

$$\hat{R}(t) = e^{-\int_0^t [X + \hat{K}x^{\hat{m}}]dx} \tag{15}$$

这里，$\hat{R}(t)$ 是一个随机变量，用于表示根据 SDP 模型的预测进行测试的软件的可靠性。现在，化简公式 15 将得到使用 SDP 模型进行测试的软件的可靠性。

□

现在，使用 SDP 模型的软件的期望可靠性 $\mu_{\hat{R}}$ or $\mathbb{E}[\hat{R}(t)]$ 推导如下：

$$\mathbb{E}[\hat{R}(t)] = \mathbb{E}\left[e^{-\left[Xt + \frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]}\right] = \mathbb{E}\left[e^{-Xt}\right] e^{\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}} \tag{16}$$

我们观察到：

$$\mathbb{E}\left[e^{-Xt}\right] = \mathbb{E}\left[e^{-t\sum_{i=1}^{l} X_i}\right] = \mathbb{E}\left[\prod_{i=1}^{l} e^{-tX_i}\right] \tag{17}$$

由于 X_i 相互独立，随机变量 e^{-tX_i} 也相互独立。因此，$\prod_{i=1}^{l}\mathbb{E}[e^{-tX_i}]$ 。现在将这些事实代入方程，得到：

$$\mathbb{E}[\hat{R}(t)] = e^{\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}}\prod_{i=1}^{l}\mathbb{E}\left[e^{-tX_i}\right] \tag{18}$$

这里，随机变量 e^{-tX_i} 以概率 p取值 e^{-t}，以概率 $1-p$取值 1。现在根据这些值计算 $\mathbb{E}\left[e^{-tX_i}\right]$，可得：

$$\prod_{i=1}^{l}\mathbb{E}\left[e^{-tX_i}\right] = \prod_{i=1}^{l}\left[pe^{-t}+1-p\right] = \prod_{i=1}^{l}\left[1+p(e^{-t}-1)\right] \tag{19}$$

现在，我们使用不等式 $1+x<e^x$，其中 $x=p(e^{-t}-1)$，来求得期望可靠性。

$$\mu_{\hat{R}} = \mathbb{E}[\hat{R}(t)] < e^{\left[lp(e^{-t}-1)+\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]} \tag{20}$$

该不等式背后的直觉是，提供一种简便的计算方法，同时不影响下述定理中的最终界。现在，利用引理 1 和 2，下述定理给出了随机变量 $e^{-\left[Xt+\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]}$ 相对于人工测试的软件 $e^{-Kt^{m+1}}$ 的可靠性的偏差界，其中 $e^{\frac{-Kt^{m+1}}{m+1}}$。

定理 2. 设 X_1、X_2、…、X_l 为独立伯努利试验，满足对于 $1\le i\le l$，$Pr[X_i=1]=p$，其中 $0<p<1$ $\exists$ $K>0$ $m>-1$ $K>0$ $m>-1$ $t>0$ 参数、、、、时间。那么对于 $R(t) = e^{-\left[Xt+\frac{Kt^{m+1}}{m+1}\right]}$ $\mathbb{E}[R(t)] = \mu_R < e^{\left[lp\left(e^{-t}-1\right)+\frac{Kt^{m+1}}{m+1}\right]}$，以及威布尔分布下的可靠性函数 $e^{-Kt^{m+1}}$：

$$Pr\left[e^{-\left[Xt+\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]} > e^{\frac{-Kt^{m+1}}{m+1}}\right] < e^{-\left[\left[e^{\left[lp(e^{-t}-1)+\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]}-\left[\frac{Kt^m}{m+1}-\frac{\hat{K}t^{\hat{m}}}{\hat{m}+1}\right]\right]^2\frac{1}{2e^{\left[lp(e^{-t}-1)+\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]}}\right]}$$

证明。这个上尾的证明与下尾的证明非常相似，正如我们在定理 1 中所见。和之前一样，

$$Pr\left[e^{-\left[Xt+\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]} > e^{\frac{-Kt^{m+1}}{m+1}}\right] = Pr\left[Xt < \left[\frac{Kt^{m+1}}{m+1}-\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]\right] = Pr\left[X < \left[\frac{Kt^m}{m+1}-\frac{\hat{K}t^{\hat{m}}}{\hat{m}+1}\right]\right] \tag{21}$$

现在，我们希望得到一个紧下界，用于描述随机变量 $e^{-\left[Xt+\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]}$ 偏离数值 $e^{\frac{-Kt^{m+1}}{m+1}}$ 21, $K>0$ $K>0$ $m>$ 较远 -1 $m>-1$ $t>0$ 的程度。在方程中，对于某些，，，和，假设数值 $\left[\frac{Kt^m}{m+1}-\frac{\hat{K}t^{\hat{m}}}{\hat{m}+1}\right]$ 低于给定时间段内的期望 $\mu_{\hat{R}}=\mathbb{E}[R(t)]$，时间段 $[0,t]$。现在，将方程 12 和方程 21 相等，可得：

$$(1-\delta)\mu_{\hat{R}} = \frac{Kt^m}{m+1}-\frac{\hat{K}t^{\hat{m}}}{\hat{m}+1}$$

$$\Rightarrow \delta = 1-\left[\frac{Kt^m}{m+1}-\frac{\hat{K}t^{\hat{m}}}{\hat{m}+1}\right]\frac{1}{\mu_{\hat{R}}} \tag{22}$$

现在，将 δ （来自公式 22）的值代入公式 12，即可得到随机变量（即 $e^{-\left[Xt+\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]}$）相对于手动测试软件 $e^{\frac{-Kt^{m+1}}{m+1}}$ 的可靠性（该可靠性由危险率的威布尔模型推导得出）的偏差的紧上界（以下界表示）。表达如下：

$$Pr\left[e^{-\left[Xt+\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]} > e^{\frac{-Kt^{m+1}}{m+1}}\right] < e^{\frac{-\mu_{\hat{R}}\left[1-\frac{1}{\mu_{\hat{R}}}\left[\frac{Kt^m}{m+1}-\frac{\hat{K}t^{\hat{m}}}{\hat{m}+1}\right]\right]^2}{2}} \tag{23}$$

化简后，我们得到：

$$Pr\left[e^{-\left[Xt+\frac{\hat{K}t^{\hat{m}+1}}{\hat{m}+1}\right]} > e^{\frac{-Kt^{m+1}}{m+1}}\right] < e^{-\left[\mu_{\hat{R}}-\left[\frac{Kt^m}{m+1}-\frac{\hat{K}t^{\hat{m}}}{\hat{m}+1}\right]\right]^2\frac{1}{2\mu_{\hat{R}}}} \tag{24}$$

将公式 20 中的期望可靠性 $\mu_{\hat{R}}$ 代入公式 24，即可完成证明。□

因此，根据定理 2，对于使用 SDP 的软件而言，其可靠性优于由人工测试的同一软件的可能性在 l、K、m 和 t 中呈指数级小。这表明，与定理 1 的结果类似，这些参数的值越大，上界就越紧。

IV. 未来计划

定理 1 和定理 2 为实时工作环境中的二分类模型（SDP 模型）提供了事后分析的初步界限。我们认为，对实时工作环境中开发的二分类模型进行批判性分析，在机器学习理论中是一项新颖的工作，并有望为其他应用的可行性提供见解（例如安全关键应用，如用于医学诊断的肿瘤预测系统、在线欺诈检测等）。在本工作的范围内，定理 1 和定理 2 可以有许多扩展。以下是几个例子：1）如果在构造证明时采用最先进的危险模型（和可靠性模型），所得界限将更贴合具体应用；2）假设随机变量 X 是时间 t 的函数时，可以推导出新的界限；3）假设随机变量之间存在依赖关系（放宽假设 4）时，可以推导出新的界限。在这种情况下，我们通过假设软件因 SDP 模型而出现级联故障来推导界限。

参考文献

[1] T. M. Khoshgoftaar 和 J. C. Munson，“使用软件复杂度度量预测软件开发错误”，《IEEE 通信领域精选期刊》，第 8 卷，第 2 期，第 253–261 页，1990 年。[2] S. Lessmann、B. Baesens、C. Mues 和 S. Pietsch，“软件缺陷预测分类模型的基准测试：一种拟议框架及新发现”，《IEEE 软件工程汇刊》，第 34 卷，第 4 期，第 485–496 页，2008 年。[3] S. Herbold、A. Trautsch 和 J. Grabowski，“跨项目缺陷预测方法的基准比较研究”，《IEEE 软件工程汇刊》，第 44 卷，第 9 期，第 811–833 页，2017 年。[4] Y. Zhou、Y. Yang、H. Lu、L. Chen、Y. Li、Y. Zhao、J. Qian 和 B. Xu，“我们在这段旅程中走了多远？跨项目缺陷预测研究”，《ACM 软件工程与方法学汇刊》（TOSEM），第 27 卷，第 1 期，第 1–51 页，2018 年。[5] T. Zimmermann、N. Nagappan、H. Gall、E. Giger 和 B. Murphy，“跨项目缺陷预测：关于数据、领域与过程的大规模实验”，载于《第 7 届欧洲软件工程会议与 ACM SIGSOFT 软件工程基础研讨会联合会议论文集》，2009 年，第 91–100 页。[6] S. Amasaki、H. Aman 和 T. Yokogawa，“在同质化跨公司工作量估算情境下，对同质化跨项目缺陷预测方法适用性与性能的扩展研究”，《实证软件工程》，第 27 卷，第 29 期，2022 年，第 1–2 页。[7]U. S. B 和 R. Sadam，“预测决策对软件项目的影响有多大？基于跨项目缺陷预测模型的成本、服务时间与故障分析”，《系统与软件期刊》，第 111522 页，2022 年。 [在线]。网址：https://www.sciencedirect.com/science/article/pii/S0164121222001984
[8]

C. M. Bishop 和 N. M. Nasrabadi，《模式识别与机器学习》。Springer，2006 年，第 4 卷，第 4 期。[9] M. R. Lyu 等，《软件可靠性工程手册》。IEEE 计算机学会出版社，洛斯阿拉米托斯，1996 年，第 222 卷。[10] R. S. Pressman，《软件工程：实践者的方法》。Palgrave Macmillan，2005 年。[11] S. M. Ross，《概率模型导论》。Academic Press，2014 年。[12] R. Motwani 和 P. Raghavan，《随机算法》。剑桥大学出版社，1995 年。[13] M. A. Hartz、E. L. Walker 和 D. Mahar，《软件可靠性导论：现状综述》。The Center，1997 年。[14] H. Chernoff，“基于观测值之和的假设检验的渐近效率度量”，《数学统计年刊》，第 493–507 页，1952 年。

此图“RevisedBathTubDiagram.jpg”提供“JPG”格式，来源：

http://arxiv.org/ps/2301.05411v1

此图“fig1.png”可从以下位置获取 PNG 格式版本：

http://arxiv.org/ps/2301.05411v1