ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20220910110537
Original scientific paper

A Crowdsourced Test Defect Number Prediction Model Based on Test Labor and Test Reports

Shanling LI, Yi YAO*, Changyou ZHENG

Abstract: Software reliability growth model is widely used in measurement, prediction and reliability assurance. The uncertainty of software potential defects and the unpredictability of the distributed crowdsourced test process make the crowdsourced test platform crave software reliability modeling techniques to predict the number of potential defects of the software, to evaluate the progress of the test task. This paper puts forward a crowdsourced test defect number prediction model (CTDNPM) that considers both the quantity of test labor and the number of test reports as two test cost elements. A new reliability modeling framework is established based on element correlation, and three existing test work functions are combined to solve the equations, to predict the number of potential defects and the cumulative number of defects detected. The experimental results of four groups of real crowdsourced test data sets show that CTDNPM can predict the number of defects. The error of defect number estimation in the model is less than 10%, which has important guiding significance for monitoring the progress of the test task in the actual crowdsourced test.

Keywords: crowdsourced test; defect number prediction; test labor; test reports

1 INTRODUCTION

With the further study of Internet swarm intelligence technology, the efficiency of defect detection brought by the diversity and complementarity of personnel in the test is improved, which makes the crowdsourced test develop rapidly [1]. After a crowdsourcing test platform publishes a test task, project managers typically plan the end of a crowdsourcing test task based only on their personal experience. However, making reasonable decisions is challenging, these experience-based decisions can lead to an ineffective crowdsourced test process. This demonstrates the real need and potential opportunity to improve current crowdsourcing testing practices. Since it is impossible to calculate the total number of defects before system testing, it is necessary to establish an estimation model of the number of real-time defects in software testing. By constantly comparing the number of predicted and detected defects to evaluate the progress of testing tasks, it can reasonably determine the end of the task and avoid wasting a lot of testing time and resources.

Software reliability growth model (SRGM) [2-3] is widely used in measurement, prediction, and reliability assurance, and provides important decision supporting for software development activities such as optimal release time selection. SRGM conducts reliability modeling from the perspective of software failure and uses the mathematical method of differential equations to establish a quantitative function model among several random parameters in the software testing process, such as test time, cumulative number of defects detected, and test effort. Based on the function expression of the cumulative detection defect number, the reliability of the test phase can be obtained [4]. In addition to the types of perfect debugging model [5-7] and imperfect debugging model [8-9], SRGM also takes into account the testing resources and testing environment. The SRGM based on testing-effort (TE) and change-point (CP) has been successively proposed [10-14]. Various types of TE began to appear widely in SRGM, making the cost of integrating test resources into modeling become the norm.

Different from traditional testing methods, crowdsourced tests make software products more reasonably and adequately tested due to their random diversity of testers and large number of test reports. Based on these features, this paper proposes a defect number prediction model for crowdsourced test. This paper makes the following contributions:

- The traditional reliability growth model framework based on test-effort function (TEF) is improved in this paper, and the quantity of test labor and the number of submitted test reports are taken into consideration.
- This paper analyzes the correlation between two test cost elements and establishes a general piecewise reliability modeling framework based on the degree of correlation and TEF.
- In the cost estimation, this paper introduces three TEFs to select the optimal parameter to predict the defect number, which can effectively improve the accuracy of the results.

The rest of this article is organized as follows. The second section outlines the general model of SRGM based on TEF. Section 3 proposes a defect number prediction model based on the correlation of test cost elements. Three TEFs are discussed in section 4. The next section discusses and analyzes the experimental process and experimental results. And the analysis of the threats to validity is also presented in the same section. The last section is the conclusion.

2 BASIC MODEL DESCRIPTION

Nonhomogeneous Poisson process (NHPP) models have become the most widely studied SRGMs due to their excellent properties [15]. The detection and repair of defects is carried out at the expense of TE, which effectively describes the work profile of the software development process and should be taken into account in SRGM.

Zhang et al. [8] proposed a relatively unified SRGM framework for TE for the first time, which was based on the following assumptions.

(1) The defect detection process follows the NHPP process.

(2) The average number of defects detected within the time $(t, t + \Delta t)$ is proportional to the ratio of the current TE

consumption rate and the average number of remaining defects in the software. This ratio is the current defect detection rate, which can be either a constant b or a time-dependent function $b(t)$.

(3) Once a defect is detected, it will be repaired immediately and perfectly, and the repair will only take a little time and no new defects will be introduced. Therefore, the total number of defects in the software is a constant value a.

(4) All defects are independent and can be detected.

Based on the above assumptions, the NHPP software reliability modeling framework considering TE can be obtained as shown in Eq. (1).

$$\frac{\mathrm{d}m(t)}{\mathrm{d}t}\times\frac{1}{w(t)}=b(t)\left(a-m(t)\right) \tag{1}$$

where $m(t)$ represents the expected mean value function of the number of detected defects in the time interval $[0, t]$. $b(t)$. represents the defect detection rate at that time, and a represents the total number of defects in the software. $W(t)$ represents the TE consumption rate function, which is the derivative of TEF which is defined as $W(t)$.

When the boundary conditions are $m(0) = 0$, $W(0) = 0$, the TEF $W(t)$ of different forms is substituted into the above differential equation, then various cumulative detection defect number function expressions $m(t)$ can be obtained.

3 CTDNPM

CTDNPM considers the diversity of crowdsourced testers and the complementarity of test reports. The model takes the quantity of test labor and the number of test reports submitted as the test cost elements, and builds a crowdsourced test defect number prediction model based on TEF-SRGM. The flow of the entire model is shown in Fig. 1.

Under the assumption of the model, CTDNPM firstly uses the correlation function to analyze the correlation between the quantity of test labor and the number of test reports. Secondly, a general piecewise reliability modeling framework is established to select the appropriate differential equation according to the degree of correlation between two elements. Then the parameters of the three TEFs are estimated and compared with the actual test cost data, and the optimal workload function is selected and substituted into the equation. Finally, the number of potential defects in the software and the cumulative number of detected defects can be predicted.

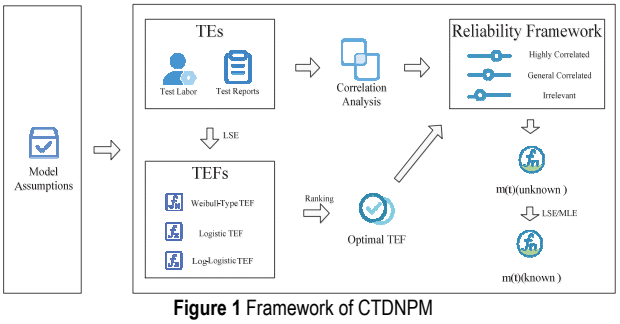

Figure 1 Framework of CTDNPM

3.1 Model Assumptions

First, since the crowdsourced test process involves only defect detection behavior and does not include repair process, the model needs to set up a virtual repair process. This process is a perfect repair in this article, and it does not need to be discussed in practice. Second, crowdsourced test software is usually too small to be suitable for system-level analysis, and therefore requires refinement of the time. Third, considering that the test reports have the characteristics of high repetition rate and high complementation rate, we need to integrate and deal the report. Therefore, CTDNEM modified the assumptions of SRGM base on TEF to meet the characteristics of crowdsourced test. The model needs to meet the following assumptions.

(1) The defect detection process follows the NHPP process in the test process.

(2) All defects are independent and can be detected.

(3) All test calendar time is refined and processed according to the time stamp, and the test starting time is guaranteed to be zero.

(4) Virtual repair will be carried out after each test report is submitted. That means defects detected will be repaired immediately and no new defects will be introduced. Therefore, the total number of defects in the software is a constant a.

(5) Repeated defects in the test report are only recorded for the first time. If the defect is reported in the test report after virtual repair, the defect will be ignored.

(6) The average number of defects detected in $(t, t + \Delta t)$ time is proportional to the average number of remaining defects in the software.

(7) The average number of defects detected within the time of $(t, t + \Delta t)$ is proportional to the consumption rate of the integration effect of two elements. The integration effect of test cost elements will be discussed later.

3.2 Test Cost Elements

The reliability growth model is generally suitable for defect number prediction at the system level [4]. However, the software tested in the crowdsourced test is small in scale, and the simple test effort of TEF-SRGM is difficult to guarantee the accuracy of the model. Therefore, multiple test cost elements can be considered based on the characteristics of crowdsourced test. CTDNPM takes into account the quantity of test labor and the number of test reports submitted.

Actually, test cost elements are not completely independent of each other, nor are they completely consistent. But they are more or less related to each other. According to the study Fig. 2 shows that there is a positive correlation between the quantity of test labor and the number of test reports. In order to improve the accuracy of defect number prediction results, it is necessary to minimize the influence of two elements on each other. Therefore, CTDNPM carries out correlation analysis on two elements.

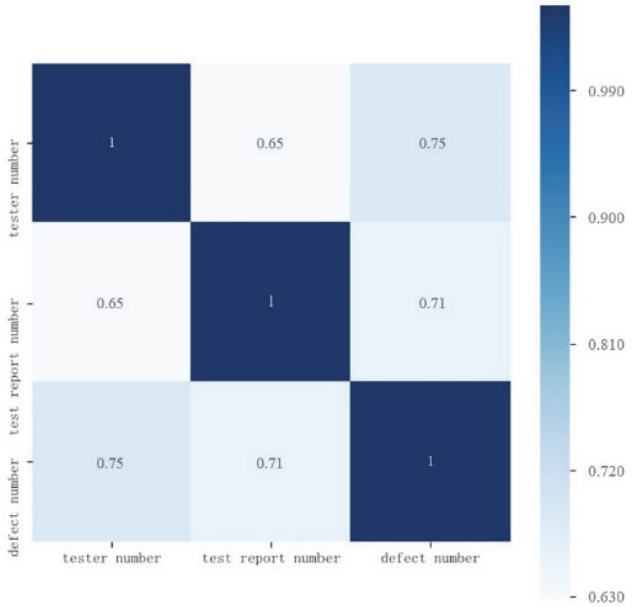

Figure 2 Correlation Results between Test Labor and Test Reports

The model uses correlation coefficients to represent the correlation between two elements. Correlation coefficient is the statistical index correlation designed by statistician Carl Pearson at the earliest. And its calculation formula is as follows.

$$(P,R)=\frac{cov(P,R)}{\sqrt{Var|P|\sqrt{RVar|R|}}},\ cov(P,R)= \\ =E\left(\left(P-E(P)\right)\cdot\left(R-E(R)\right)\right) \tag{2}$$

where P is the variable of the test labor involved, R is the variable of the test reports submitted. *Var* stands for the variance function, and E stands for the expected value function.

Pearson believed that P and R were strongly correlated if the absolute value of the correlation coefficient was greater than 0.7. Between 0.4 and 0.7, two elements can be considered generally correlated. Below 0.4, two elements are considered to have no correlation [16].

3.3 Reliability Framework

TEF-SRGM only considers the case of single test cost element, so two or more test cost elements cannot be directly substituted into Eq. (1). CTDNPM proposes a reliability modeling framework that can contain two test cost elements. Based on Eq. (1), two test cost elements are introduced into the model to calculate the mean number of defects finally detected. Considering the correlation between test cost elements, the following reliability modeling framework is established.

$$\begin{cases} \dfrac{dm(t)}{dt}\times\dfrac{1}{w_1(t)}=b(t)\left(a-m(t)\right)\rho(P,R)\ge 0.7 \\ \dfrac{dm(t)}{dt}\times\dfrac{1}{\lambda w_1(t)+\mu w_2(t)}=b(t)\left(a-m(t)\right)\quad 0.4\le\rho(P,R)<0.7 \\ \dfrac{dm(t)}{dt}\times\dfrac{1}{w_1(t)}\times\dfrac{1}{w_2(t)}=b(t)\left(a-m(t)\right)\rho(P,R)<0.4 \end{cases} \tag{3}$$

where $w_1(t)$ is the test labor occupancy rate at time t, and $w_2(t)$ is the test reports consumption rate at time t. a stands for the total number of defects, λ and μ are the Correlation harmonic coefficients. $b(t)$ represents the defect detection rate.

The defect detection rate $b(t)$ will eventually rise and become constant with the detection and elimination of defects. Literature [17] defines $b(t)$ as follows.

$$b(t)=b\left[r+(1-r)\frac{m(t)}{a}\right] \tag{4}$$

where r is referred to as the bending factor and represents the proportion of unrelated defects in the software. When $r=1$, the $m(t)$ obtained is concave, and the rest is S-shaped.

Three correlations of two test cost elements are described as follows.

(1) When $\rho(P,R)\ge 0.7$, P and R are strongly correlated. In this case, the influence of two elements can be replaced by any one of them, and the differential equation is the same as Eq. (1).

(2) When $0.4\le\rho(P,R)<0.7$, the correlation between P and R is weak. At this point, it is determined that P and R influence the number of defects detected in a certain proportion, and the average number of defects detected in $(t,\ t+\Delta t)$ time is proportional to the sum of the consumption rates of two elements under a certain weight.

(3) When $\rho(P,R)<0.4$, P and R are basically independent. At this point, it is determined that P and R are independent of each other and jointly affect the number of defects detected. The average number of defects detected within $(t,\ t+\Delta t)$ time is proportional to the consumption rate of both elements.

4 TEF ANALYSIS

It is clearly pointed out in literature [18] that the total number of defects, defect detection rate and testing effort are important parameters affecting SRGM. The first two factors have been analyzed in the previous part, so they will not be repeated here. This section outlines the third factor, TE.

During the software testing phase, a lot of TE is consumed, such as the number of test reports, the quantity of test labor, and CPU time. The consumed TE indicates how defects can be effectively detected in the software, so resource consumption or labor resource allocation can be modeled using different distributions. TE is usually represented by the TEF, which is defined as $W(t)$. Three commonly used TEFs are listed below.

(1) Weibull-Type TEF: TE per unit time is not a constant in the whole testing phase. In fact, the instantaneous TE eventually decreases during the test lifecycle. So, the cumulative TE approaches a finite limit. This analysis makes sense because no software company spends unlimited resources on software testing. Therefore, Yamada et al. [19] proposed the Weibull distribution, and the formula of the cumulative TE consumed in (0, t] is as follows.

$$W(t)=W\left(1-e^{\left(-\beta t^{\delta}\right)}\right)^{\theta},\ W>0,\ \beta>0,\ \delta>0,\ \theta>0 \tag{5}$$

The instantaneous TE consumed at the time t is defined.

$$w(t)=\frac{\mathrm{d}W(t)}{\mathrm{d}(t)}=W\cdot\beta\cdot\delta\cdot\theta\cdot t^{\delta-1}\cdot e^{\left(-\beta t^{\delta}\right)}\left(1-e^{\left(-\beta t^{\delta}\right)}\right)^{\theta-1} \tag{6}$$

where W is the total amount of TE, and β is the size parameter. δ and θ are the shape parameters.

The Weibull-type curve has several special cases. When $\theta = 1$, $\delta = 1$, the cumulative TE is an exponential curve. When $\theta = 1$, $\delta = 2$, the cumulative TE is the Rayleigh curve. When $\theta = 1$, the cumulative TE is Weibull curve.

(2) Logistic TEF: when $\delta > 3$, the Weibull-type curve has an obvious peak. This phenomenon is inconsistent with the actual software development/testing process. To this end, Huang et al. [20] proposed the use of logistic TEF to describe the test work effort model. The cumulative TE consumed in (0, t] is as follows:

$$W(t)=\frac{W}{1+Ae^{(-\alpha kt)}},\ W>0,\ A>0,\ \alpha>0,\ k>0 \tag{7}$$

The instantaneous TE consumed at the time t is defined.

$$w(t)=\frac{\mathrm{d}W(t)}{\mathrm{d}(t)}=\frac{W\cdot Ae^{(-\alpha t)}}{\left[1+Ae^{(-\alpha t)}\right]^{2}} \tag{8}$$

(3) Log-Logistic TEF: The rate of TE loss per unit time may also be increasing or decreasing as the test progresses. This trend cannot be captured by the existing model. Therefore, Gokhale and Trivedi [21] proposed log-logistic TEF to describe this trend. The cumulative TE consumed in (0, t] is as follows.

$$W(t)=W\left[\frac{(\theta t)^{\delta}}{1+(\theta t)^{\delta}}\right],\ W>0,\ \theta>0,\ \delta>0 \tag{9}$$

The instantaneous TE consumed at the time t is defined.

$$w(t)=\frac{\mathrm{d}W(t)}{\mathrm{d}(t)}=\frac{W\cdot\delta\cdot(\theta t)^{\delta}}{\left[1+(\theta t)^{\delta}\cdot t\right]^{2}} \tag{10}$$

5 EXPERIMENTAL VERIFICATION

5.1 Experimental Setup

(1) Data Sets: We chose four groups of real defect data set DS1 to DS4, including information of TE. The four groups of data sets are from mooctest crowdsourced test platform [22]. After the crowdsourced test of these projects, we conducted statistics and integration of test reports, test labor and defect information according to their complementary characteristics, and formed a defect data set. See Tab. 1 for details.

(2) Control Group: TEF-SRGM was selected as the baseline method, and the TEF used for baseline method was selected as the optimal distribution function after experimental comparison.

Table 1 Details of the Crowdsourced Test Defect Data Sets

Data Sets	Test Time (Time stamp)	Test Labor	Test Report	Defects Defected
ALU	39660	430	1215	172
Anagram	39660	264	1227	45
Jipa	39660	702	3695	87
Sudoku	39660	680	4387	130

(3) Experimental Group: CTDNPM was used to establish defect number prediction models suitable for real data sets, and a comparative study was conducted with the baseline method.

(4) Parameter Estimation Method: Maximum likelihood estimation (MLE) and least squares estimation (LSE) are used for parameter estimation. LSE was used for TEF parameter estimation, while MLE and LSE were used for parameter estimation of $m(t)$.

(5) Task closing rule: The cumulative defect detection number function converges, and the actual detection number and the predicted software potential defect number reach a certain proportion (such as 90%), then the task can be determined to be completed.

5.2 Evaluation Criteria

In order to check the performance of TEF and CTDNPM, and better evaluate and compare the defect number prediction ability of the model, we use the following three evaluation criteria.

(1) Accuracy of Estimation (AE).

The accuracy of estimation can be calculated by the initial estimated number of defects in the software and the actual cumulative number of defects detected. The accuracy of software defect estimation is reflected by the estimated number of potential defects and the error of detection results.

$$AE=\left|\frac{M_a-a}{M_a}\right| \tag{11}$$

where M_a is the cumulative number of defects actually detected after testing, and a is the total number of defects estimated in the software. And the estimate a can be used to compare the number of defects detected to measure test task progress.

(2) Mean square Error (*MSE*).

$$MSE=\frac{1}{k}\sum_{i=1}^{k}\left[m(t_i)-m_i\right]^{2} \tag{12}$$

where m_i is the actual number of defects detected at time t_i, and $m(t_i)$ is the expected number of defects predicted by the model at time t_i. The smaller the *MSE*, the smaller the prediction error, the better the performance.

(3) Predictive Validity (PV).

Predictive validity is defined as the ability of a model to predict future failure behaviors based on current and past failures. Suppose we observe defect number q at the end of

test time t_q. We can use defect data up to time $t_e(t_e \le t_q)$ to estimate the parameter values of the model. The parameter estimation value is substituted into model $m(t)$, and we can get the number of defects $\hat{m}(t_q)$ at time t_q. The relative error formula is used to compare the predicted value with the actual defect number q.

$$RE = \frac{\hat{m}(t_q) - q}{q} \tag{13}$$

The relative error (RE) can be obtained by taking different time points $t_e(t_e \le t_q)$ and repeating the above process. In general, we make PV diagram according to different RE values to visually check the predictive performance of the model. The more points in the PV diagram that are close to the horizontal axis, the better the model predicts performance.

(4) Defect Coverage (DC).

Defect coverage can reflect the degree of defect coverage detected by calculating the ratio of the number of detected defects to the number of software defects initially predicted in real time [23].

$$DC = \frac{M_a}{a} \times 100\% \tag{14}$$

where M_a is the cumulative number of defects actually detected after testing, and a is the total number of defects estimated in the software. Ideally, when DC = 100% occurs, all defects are detected. However, DC is always less than the ideal value in the actual test, so a certain proportion of DC can be set to determine the completion of the task.

5.3 Results Analysis

In order to verify the proposed model CTDNPM and compare its performance with the baseline method, we conducted experiments on four actual crowdsourced test defect data sets. The experimental results of each data set are as follows.

(1) DS1.

In order to estimate the TE, we used LSE to estimate the parameters of three TEFs. Fig. 3 shows the cumulative quantity of test labor and test reports by using three TEFs mentioned above. The fitting curve and actual software data are respectively represented by solid and dotted lines. Among the test labor TE curves, the logistic TEF is the closest to the actual data, and its parameters are $W = 391.70$, $A = 7.870$, $\alpha = 0.107$, $k = 3.5 \times 10^{-3}$. In the test report TE curve, Weibull-type TEF is the closest to the actual data, and its parameters are $W = 1221.31$, $\beta = 9.48 \times 10^{-12}$ $\delta = 2.5640$, $\theta = 1.915$.

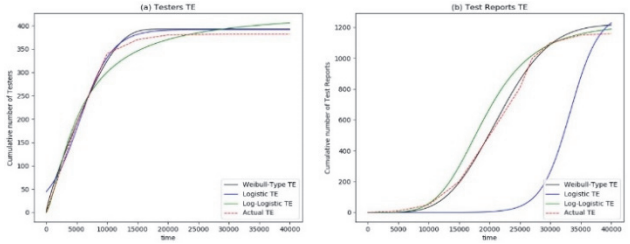

Figure 3 Forecast/Actual Quantity of Test Labor and Test Report of DS1

The correlation coefficient ρ = 0.65 was obtained through the correlation coefficient calculation formula. The second differential equation in the reliability model framework was selected to solve according to ρ. And other parameters in $m(t)$ can be numerically solved by MLE and LSE. Then the model CTDNPM was compared with the baseline method. The parameter estimation and comparison results are shown in Tab. 2.

Table 2 Comparison Results of Predicted Performance of DS1

Models	Methods	a	b	r	λ	μ	AE / %	MSE	DC / %
CTDNPM	MLE	164.61	0.0362	23.275	0.371	0.812	4.30	22.73	104.49
	LSE	186.73	0.0765	29.239	0.339	0.944	8.56	49.62	92.11
TEF-SRGM	MLE	195.26	0.1462	7.392			13.52	82.49	88.09
	LSE	203.55	0.1546	10.983			18.34	126.51	84.50

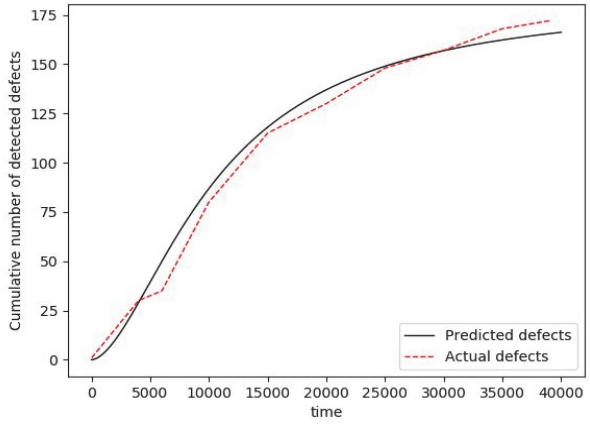

Figure 4 Predicted/Actual Cumulative Number of Defects Defected of DS1

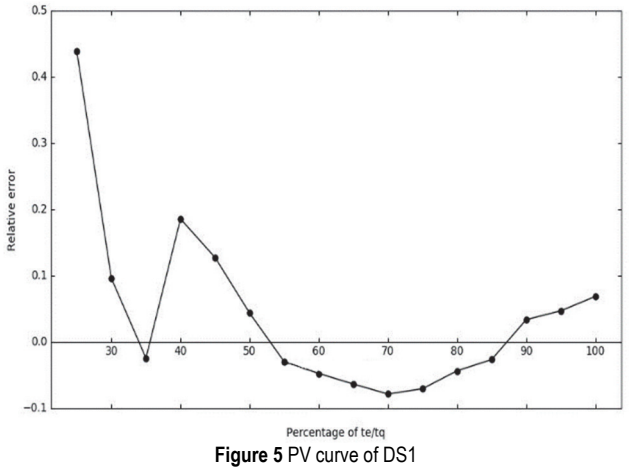

Figure 5 PV curve of DS1

In Tab. 2, the *AE* value of CTDNPM using *MLE* is 4.30%, the *MSE* value is 22.73, the *AE* value of CTDNPM using *LSE* is 8.56%, the *MSE* value is 49.62. Compared with the baseline method, the prediction performance of CTDNPM is greatly improved. Based on the above results. Fig. 4 plots the predicted cumulative number of defects, and the final predicted value of defects began to converge. The *DC* value in Tab. 2 had exceeded 100%, indicating that the test had been completed and was in line with the actual situation.

Finally, the PV of this data set was calculated. The results are shown in Fig. 5. We observed that the *RE* values gradually approached 0 with the increase of time. And when t_e approached time t_q, the *RE* curve was usually concentrated within 10%. It can be seen that CTDNPM has good prediction.

(2) DS2.

Fig. 6 shows the fitting curve of the cumulative quantity of test labor and test reports by using LSE. The logistic TEF is the closest to the actual data in the test labor TE curves, and its parameters are $W = 233.99$, $A = 4.719$, $\alpha = 1.38 \times 10^{-2}$, $k = 2.72 \times 10^{-2}$. In the test report TE curve, the logistic TEF is the closest to the actual data, and its parameters are $W = 1253.88$, $A = 7673.40$, $\alpha = 6.27 \times 10^{-2}$, $k = 6.34 \times 10^{-3}$.

According to the calculation, the correlation coefficient is $\rho = 0.59$, and the second differential equation in the model framework was selected. The comparison results between CTDNPM and baseline method are shown in Tab. 3.

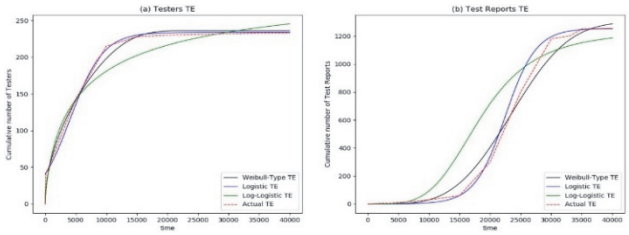

Figure 6 Forecast/Actual Quantity of Test Labor and Test Report of DS2

Table 3 Comparison Results of Predicted Performance of DS2

Models	Methods	*a*	*b*	*r*	*λ*	*μ*	*AE* / %	*MSE*	*DC* / %
CTDNPM	MLE	51.82	0.6302	34.247	0.787	0.327	3.16	28.58	86.84
	LSE	55.35	0.7193	38.860	0.965	0.272	5.00	42.02	81.30
TEF-SRGM	MLE	46.63	0.3862	13.082			3.02	22.10	96.50
	LSE	49.70	0.4046	12.639			4.41	30.37	90.54

In Tab. 3, the *AE* value of CTDNPM using MLE is 3.16%, and the *MSE* value is 28.58. Under the same method, the *AE* value predicted by the baseline method is only 3.02%, and the *MSE* value is 22.10. Thus, it can be seen that CTDNPM did not play its advantages well in this data set. According to the parameter results. Fig. 7 plots the cumulative number of defects predicted, which is basically consistent with the actual data submitted. At this point, the number of detected defects basically converges, and the *DC* value in the baseline method exceeds 90%, which can also determine that the test task is completed and is consistent with the actual situation.

Finally, the result of PV is shown in Fig. 8. The RE values gradually approach 0 with the increase of time. When t_e approaches time t_q, the error curve is usually concentrated within 10%.

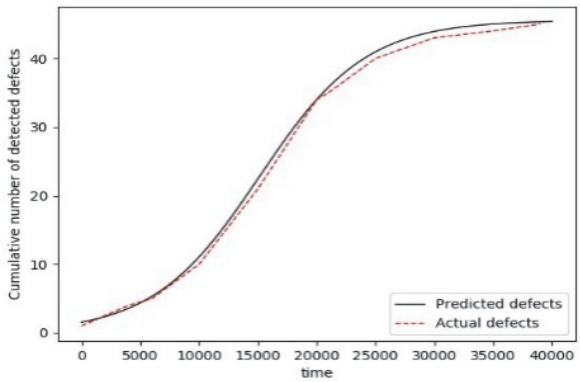

Figure 7 Predicted/Actual Cumulative Number of Defects Defected of DS2

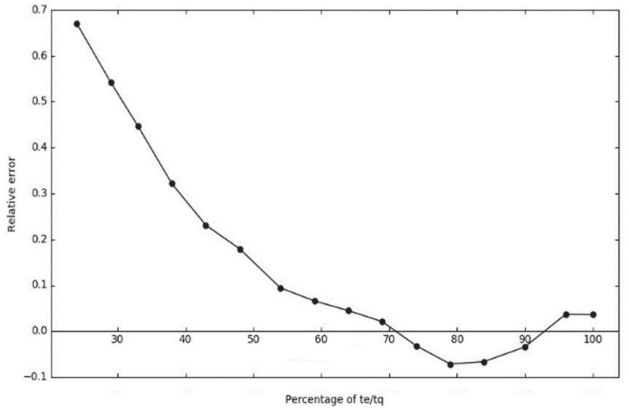

Figure 8 PV curve of DS2

(3) DS3.

Fig. 9 shows the fitting curve of the cumulative quantity of test labor and test reports by using LSE. The logistic TEF is the closest to the actual data in the test labor TE curves, and its parameters are $W = 636.74$, $A = 2.060$, $\alpha = 3.05 \times 10^{-2}$, $k = 1.05 \times 10^{-2}$. In the test report TE curve, Weibull-type TEF curve is the closest to the actual data, and its parameters are $W = 4001.44$, $\beta = 9.73 \times 10^{-7}$, $\delta = 1.397$, $\theta = 0.276$.

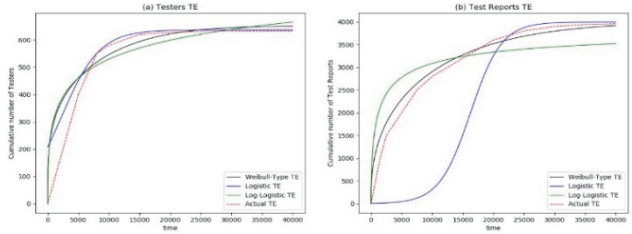

Figure 9 Forecast/Actual Quantity of Test Labor and Test Report of DS3

According to the calculation, the correlation coefficient is $\rho = 0.72$, and the first differential equation in the model framework was selected. The differential equation obtained by CTDNPM is consistent with the baseline method, and the results are shown in Tab. 4.

Table 4 Comparison Results of Predicted Performance of DS3

Models	Methods	*a*	*b*	*r*	*AE* / %	*MSE*	*DC* / %
CTDNPM	MLE	94.2	0.27	49.27	8.31	37.62	92.33
	LSE	97.6	0.21	42.23	12.24	87.51	89.09

In Tab. 4, the MLE used for CTDNPM can achieve better results. The *AE* value of the predicted results is 8.31%, and the *MSE* value is 37.62%. The *AE* value of LSE is 12.24%, and the *MSE* value is 87.51. Based on the parameters. Fig. 10 plots the cumulative number of defects predicted. At this time, the number of tests just begin to level off, while the *DC* value is around 90%, so it is judged that the test is closed too early.

Finally, the result of PV is shown in Fig. 11. The error curve is mainly within 10%. It can be seen that CTDNPM can better predict the number of defects in DS3

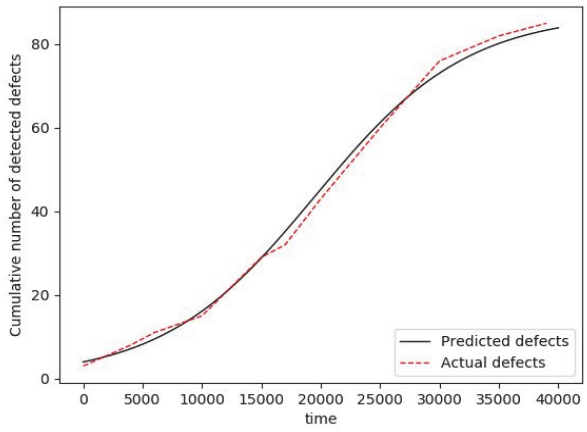

Figure 10 Predicted/Actual Cumulative Number of Defects Defected of DS3

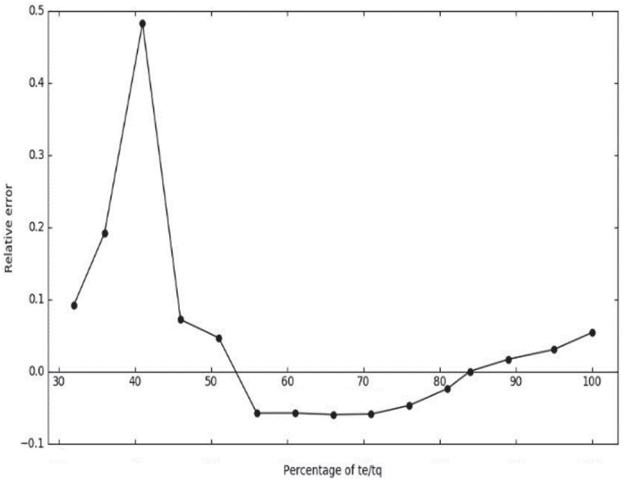

Figure 11 PV curve of DS3

(4) DS4.

Fig. 12 shows the fitting curve of the cumulative quantity of test labor and test reports by using LSE. The logistic TEF is the closest to the actual data in the test labor TE curves, and its parameters are $W = 616.41$, $A = 6.810$, $\alpha = 7.28 \times 10^{-3}$, $k = 5.16 \times 10^{-2}$. In the test report TE curve, Weibull-type TEF curve is the closest to the actual data, and its parameters are $W = 4395.53$, $\beta = 2.185$, $\delta = 1.990$, $\theta = 0.357$.

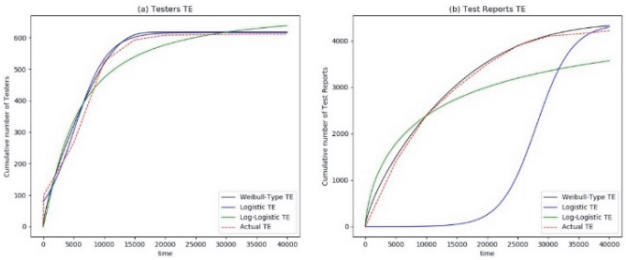

Figure 12 Forecast/Actual Quantity of Test Labor and Test Report of DS4

Table 5 Comparison Results of Predicted Performance of DS4

Models	Methods	*a*	*b*	*r*	λ	μ	*AE* / %	*MSE*	*DC* / %
CTDNPM	MLE	134.69	0.9462	33.278	1.723	0.472	3.61	29.72	96.52
	LSE	141.13	0.8365	29.228	1.572	0.825	8.56	51.55	92.11
TEF-SRGM	MLE	153.40	1.846	13.63			18.00	118.02	84.75
	LSE	181.38	2.858	19.44			39.52	178.41	71.67

According to the calculation, the correlation coefficient is $\rho = 0.66$, and the first differential equation in the model framework was selected. The differential equation obtained by CTDNPM is consistent with the baseline method, and the results are shown in Tab. 5.

In Tab. 5, the *AE* value of CTDNPM using MLE is 3.61%, and the *MSE* value is 29.72. the *AE* value of CTDNPM using LSE is 8.56%, and the *MSE* value is 51.55. Both results are superior to the performance of the baseline method. Based on the parameters. Fig.13 plots the cumulative number of defects predicted. The cumulative number of detected defects has begun to converge, and the *DC* values are all above 90%, indicating a high degree of test completion.

Finally, the result of PV is shown in Fig. 11. The error curve is mainly within 5%, indicating that CTDNPM has a high prediction accuracy.

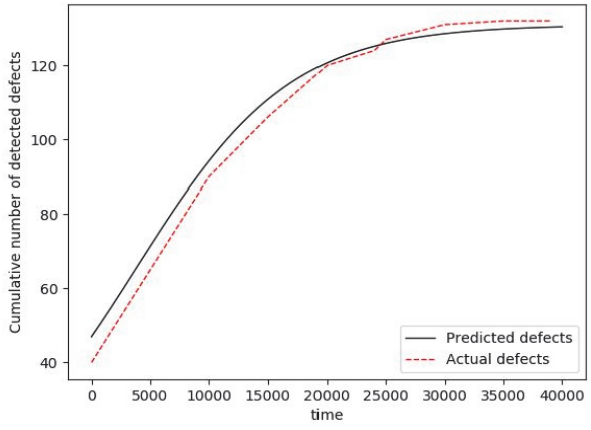

Figure 13 Predicted/Actual Cumulative Number of Defects Defected of DS4

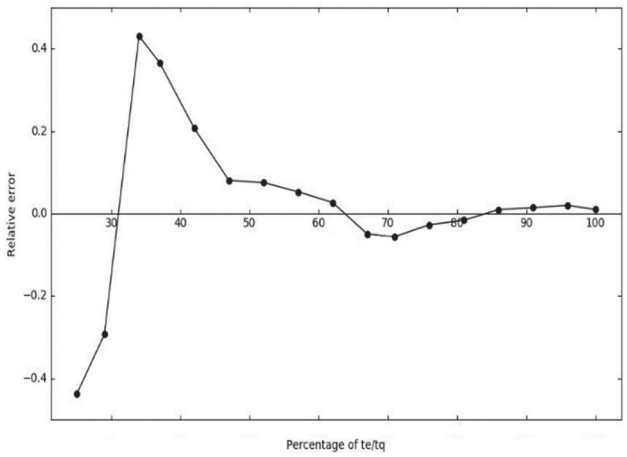

Figure 14 PV curve of DS4

To sum up, CTDNPM conducted experiments on four crowdsourced test data sets. All of test labor TEs are close to the logistic TEF curve, while most of the test report TEs are close to the Weibull-type TEF curve. The predicted results of the cumulative number of defects show good performance in the prediction on three data sets, while is inferior to the baseline method in the DS2 data set. And the RE values of the four experiments are mainly concentrated within 10%. Also, it can accurately evaluate the degree of task completion to a great extent. Therefore, CTDNPM can well predict the defects of the crowdsourced test data sets.

5.4 Threats to Validity

Experiments are always associated with potential threats, which may hinder the discovery of results. Here, we put forward the factors that may affect the validity of the experimental study.

First, the quantity of test labor and the number of test reports is appropriate when considering the cost of test in this article. Because it is perfectly suited to the reward mechanism of the crowdsourcing test platform the platform issues rewards based on the number and quality of testers and reports submitted.

Second, when analyzing the relationship between test cost elements and the number of defects submitted, the paper also considers the interaction between test cost elements. Correlation analysis is therefore used to minimize the impact between elements.

Finally, three TEFs are selected as the test-effort fitting curve of two cost elements, and the fitting curve with the optimal parameters is selected and substituted into the prediction equation, which effectively improved the accuracy of defect number prediction.

6 CONCLUSION

In crowdsourced test, the quantity of test labor and the number of test reports submitted are two important elements in measuring test costs. In order to monitor the progress of the crowdsourced test task by real-time prediction of the number of defects, this paper proposes a crowdsourced test software defect number prediction model CTDNPM which considers two test cost elements simultaneously. On this basis, model analyzes how to use the new reliability modeling framework to model the crowdsourced test data sets in combination with three TEFs. Based on four real crowdsourced test data sets, experiment was conducted using CTDNPM. The typical TEF-SRGM model was used as baseline method to compare the predicted performance. According to the comparison results, the prediction results of CTDNPM performed well in three data sets. The evaluations of the closing time of the four test tasks are consistent with the actual situation. It can be seen that CTDNPM has better defect number prediction ability, which has important guiding significance for monitoring the test task process in the actual crowdsourced test.

Acknowledgments

This work is supported by the National Key Research and Development program of China (No: 2018YFB1403400), the National Natural Science Foundation of China (No: 61702544), and the China Postdoctoral Fund (No: 2016M603031).

7 REFERENCES

[1] Zhang, X. F., Feng, Y., Liu, X., Chen, Z., & Xu, B. (2018). Research progress of crowdsourcing software testing technology.*Journal of Software*, *29*(1), 69-88. https://doi.org/10.13328/j.cnki.jos.005377

[2] Satyaprasad, R., Mohan, K. V. M., & Sridevi, G. S. G. (2014). Burr Type XII Software Reliability Growth Model. *International Journal of Computer Applications*, *108*(16), 16-20. https://doi.org/10.5120/18995-0452

[3] Lee, T. Q., Yeh, C. W., & Fang, C. C. (2014). Bayesian Software Reliability Prediction Based on Yamada Delayed S-Shaped Model. *Applied Mechanics and Materials*, *490*, 1267-1278. https://doi.org/10.4028/www.scientific.net/AMM.490-491.1267

[4] Zhang, C., Meng, F. C.,Kao, Y. G., et al (2017).Survey of Software Reliability Growth Model. *Journal of Software, 28*(9), 2402-2430.

[5] Yamada, S., Ohba, M., & Osaki, S. (1984). S-Shaped software reliability growth models and their applications. *IEEE Transactions on Reliability*, *33*(4). https://doi.org/10.1109/TR.1984.5221826

[6] Yamada, S., Ohba, M., & Osaki, S. (1983). S-Shaped Reliability Growth Modeling for Software Error Detection. *IEEE Transactions on Reliability*, *32*(5). https://doi.org/10.1109/TR.1983.5221735

[7] Jin, Y., Wu, Z. B., Shu, Y. J., & Zhang, Z. (2015). Software Reliability Model with Irregular Changes of Fault Detection Rate. *Journal of Software*, *26*(10), 2465-2484.

[8] Zhang, C., Cui, G., Liu, H., et al (2014). A Unified and Flexible Framework of Imperfect Debugging Dependent SRGMs with Testing-Effort. *Journal of Multimedia*, *9*(2), https://doi.org/10.4304/jmm.9.2.310-317

[9] Madhu, J., Manjula, T., & Gulati, T. R. (2014). Cost optimization of a software reliability growth model with imperfect debugging and a fault reduction factor. *The ANZIAM Journal*, *55*,182-196. https://doi.org/10.21914/anziamj.v55i0.7834

[10] Ahmad, N., Khan, M. G. M., & Rafi, L. S.(2010). A study of testing-effort dependent inflection S-shaped software reliability growth models with imperfect debugging. *International Journal of Quality & Reliability Management*, *27*(1). https://doi.org/10.1108/02656711011009335

[11] Ahmad, N., Khan, M. G. M., & Rafi, L. S. (2011). Analysis of an Inflection S-shaped Software Reliability Model Considering Log-logistic Testing-Effort and Imperfect

Debugging. *International journal of computer science and network security*, *11*(1), 161-171.
[12] Li, Q., Li, H., & Lu, M. (2015). Incorporating S-shaped testing-effort functions into NHPP software reliability model with imperfect debugging. *Journal of Systems Engineering and Electronics*, *26*(1), 190-207. https://doi.org/10.1109/JSEE.2015.00024
[13] Madhu, J., Manjula, T., & Gulati, T. R. (2014). Prediction of reliability growth and warranty cost of software with fault reduction factor, imperfect debugging and multiple change point. *International Journal of Operational Research*, *21*(2), 201-220. https://doi.org/10.1504/IJOR.2014.064544
[14] Jain, M., Manjula, T., &Gulati, T. R.(2014). Imperfect debugging study of SRGM with fault reduction factor and multiple change point. *International Journal of Mathematics in Operational Research, 6*(2), 155-175. https://doi.org/10.1504/IJMOR.2014.059526
[15] Hsu, C. J., Huang, C. Y., & Changb, J. R. (2011). Enhancing software reliability modeling and prediction through the introduction of time-variable fault reduction factor. *Applied Mathematical Modelling*, *35*(1), 506-521. https://doi.org/10.1016/j.apm.2010.07.017
[16] Li, H. B., He, G. Z., & Guo, Q. T. (2015). Similarity retrieval method of organic mass spectrometry based on Pearson correlation coefficient. *Stoichiometric Measurement*, *24*(03).
[17] Jha, P. C., Gupta, D., Yang, B., & Kapur, P. K. (2009). Optimal testing resource allocation during module testing considering cost, testing effort and reliability. *Computers & Industrial Engineering*, *57*(3), 1122-1130. https://doi.org/10.1016/j.cie.2009.05.001
[18] Huang, C. Y. & Lyu, M. R. (2011). Estimation and Analysis of Some Generalized Multiple Change-Point Software Reliability Models. *IEEE Transactions on Reliability, 60*(2), 498-514. https://doi.org/10.1109/TR.2011.2134350
[19] Yamada, S. & Hishitani, J. (1993). Software-reliability growth with a Weibull test-effort: a model and application. *IEEE Transactions on Reliability*, *42*(1), 100-106. https://doi.org/10.1109/24.210278
[20] Huang, C. Y. & Kuo, S. Y. (2002). Analysis of incorporating logistic testing-effort function into software reliability modeling. *IEEE Transactions on Reliability*, *51*(3), 261-270. https://doi.org/10.1109/TR.2002.801847
[21] Gokhale, S. S. & Trivedi, K. S. (1998). Log-Logistic software reliability growth model. *Proceedings Third IEEE International High-Assurance Systems Engineering Symposium (Cat. No.98EX231)*, Washington, DC, USA, 34-41. https://doi.org/10.1109/HASE.1998.731593.
[22] Wang, J., Yang, Y., Krishna, R., Tim, M., & Qing, W. (2019). iSENSE: Completion-Aware Crowdtesting Management. *IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, Montreal, QC, Canada, 912-923. https://doi.org/10.1109/ICSE.2019.00097
[23] Pocatilu, P. (2013). A Framework For Test Data Generators Analysis. *Economic Computation & Economic Cybernetics Studies & Research*, *47*(3), 185-198.
[24] Bendovschi, A. C., Ionescu, B. S., & Ionescu, I. M. (2017). Statistical Investigation into Exploring the Use of Cloud Computing Technology by Increasing the Users'trust. *Economic Computation & Economic Cybernetics Studies & Research*, *51*(1), 135-150.
[25] Zhang, D., Sui, J., & Gong, Y. (2017). Large scale software test data generation based on collective constraint and weighted combination method. *Tehnicki Vjesnik-Technical Gazette*, *24*(4), 1041-1049. https://doi.org/10.17559/TV-20170319045945

Contact information:

Shanling LI, PhD
Army Engineering University of PLA,
No.1 Haifu Road, Qinhuai District, Nanjing, China
E-mail: lishanling_nj@sina.com

Yi YAO, Associate Professor
(Corresponding author)
Army Engineering University of PLA,
No.1 Haifu Road, Qinhuai District, Nanjing, China
E-mail: yaoyi226@aliyun.com

Changyou ZHENG, Associate Professor
Army Engineering University of PLA,
No.1 Haifu Road, Qinhuai District, Nanjing, China
E-mail: zhengchy@aeu.edu.cn