Modified Jelinski-Moranda Software Reliability Model with Imperfect Debugging Phenomenon

G. S. Mahapatra
Department of Engineering Sciences and Humanities,
Siliguri Institute of Technology,
P.O.- Sukna, Siliguri-734009, West Bengal, India

P. Roy
Department of Engineering Sciences and Humanities,
Siliguri Institute of Technology,
P.O.- Sukna, Siliguri-734009, West Bengal, India

ABSTRACT
In this paper, we have modified the Jelinski-Moranda (J-M) model of software reliability using imperfect debugging process in fault removal activity. The J-M model was developed assuming the debugging process to be perfect which implies that there is one-to-one correspondence between the number of failures observed and faults removed. But in reality, it is possible that the fault which is supposed to have been removed may cause a new failure. In the proposed modified J-M model, we consider that whenever a failure occurs, the detected fault is not perfectly removed and there is a chance of raising new fault/faults due to wrong diagnosis or incorrect modifications in the software. In this paper, we develop a modified J-M model which can describe the imperfect debugging process. The parameters of our modified J-M model are estimated by using maximum-likelihood estimation method. Applicability of the model has been shown on the failure data set of Musa.

Keywords
Software reliability, Jelinski-Moranda model, Failure, Maximum likelihood estimation, Imperfect debugging.

1. INTRODUCTION
Over the last two decades, measurement of software reliability has become increasingly important because of rapid advancements in microprocessors and software. Today computer systems have been widely used for control of many complex systems. For critical systems failure of a computer system may result in disaster. The quality of software system can be described by many metrics such as complexity, portability, maintainability, availability, reliability, etc. Software reliability is a user oriented metric. The software failure is the departure of the software output from the system requirement and specification. There are many reasons for software to fail but usually these are attributed to the design problems resulting from new or changed requirements, revisions, corrections, etc. The software failures are introduced by the system analysts, designers, programmers and managers during different phases of the software development life cycle. To detect and remove these errors, the software system is tested. The quality of software system in terms of reliability is measured by the removal of these errors. Reliability is defined in terms of operational performance that one cannot measure before the product development is finished. In order to provide reliability indicators before the system is completely built, a reliability model is developed on the factors that affect reliability and the reliability predictions are made based on one's understanding of the system while it is under development [1, 2].

Software reliability is defined as the probability of failure-free operation of a computer program for a specified time in a specified environment [2]. Over the years, efforts made to estimate and measure software reliability has led to the development of many software reliability models. The J-M model [3] has a simple structure and assumptions. Use of the J-M model always yields an over optimistic reliability prediction [4]. Musa [5] proposed the basic execution time model with similar assumptions to the J-M model but introduced many important refinements. Goel and Okumoto [6] proposed the first Non Homogenous Poisson Process (NHPP) model. They assumed that the failure removal phenomenon follows NHPP. Yamada et al. [7] described the s-shapedness to the time delay between the failure observation and corresponding error removal. Ohba [8] attributed the s-shapedness to the mutual dependency between the software errors. Most of the software reliability models assume that the error removal process (debugging) is perfect, i.e. when an attempt is made to remove a fault (cause of failure) the fault is removed with certainty. This assumption may be unrealistic, due to the complexity of software systems and vague understanding of the software requirements or specification. The testing team may not be able to remove the faults perfectly and the original error can be replaced by another error. The new fault may generate new failures when this part of the software system is traversed during the testing phase. The concept of imperfect debugging was first introduced in software reliability models by Goel [9] by introducing probability of imperfect debugging to the J-M model. Kapur and Garg [10] introduced the imperfect debugging process in the Goel and Okumoto model. They assumed that the error removal rate per remaining error is reduced due to the imperfect debugging. Chang and Liu [11] proposed a non-Gaussian state space model to formulate an imperfect debugging phenomenon in software reliability. Shyur [12] developed the software reliability growth model with both imperfect debugging and change-point problem. Kapur et al. [13] presented a discrete software reliability growth model and the concept of two types of imperfect debugging during software fault removal phenomenon with Logistic Fault removal rate. Prasad et al. [14] used imperfect debugging and change-point problem into the software reliability growth model based on the well-known exponential distribution. Raju [15] discussed how to integrate a log-logistic testing-effort function into inflection s-shaped NHPP growth models to get a better description of the software fault detection phenomenon under imperfect debugging environment.

In this paper, we shall examine the J-M model, possibly the earliest and certainly one of the most well known black-box models. A modified J-M model is proposed in this paper assuming that the fault removal process is imperfect. The J-M model was developed assuming the debugging process to be perfect i.e. the detected fault is removed with certainty, this assumption is highly unrealistic. In reality, it is possible that the detected fault may not be removed perfectly and the fault, supposed to have been removed may cause a new failure. In our modified J-M model, we consider that whenever a failure occurs, the detected fault is not perfectly removed and there is a chance of raising new fault/faults, due to wrong diagnosis or incorrect modifications in the software. We extend the J-M model by relaxing the assumptions of perfect debugging process and considering imperfect debugging process in fault removal activity. We consider that the probability of perfect debugging, the probability of imperfect debugging and the probability of raising new fault/s are independent of the testing time. We estimate the parameters of our modified J-M model using maximum likelihood estimation method. To check the validity of our modified J-M model, the model has been tested on the Musa system 1 failure data set. We have shown how the failure rate varies on failure number for the two models. Finally, the prediction analysis is presented and some conclusions are drawn.

The rest of the paper is organized as follows. Section 2 presents the classical J-M model, its assumptions and estimation of this model parameters. The modified J-M model with imperfect debugging phenomenon is presented in section 3. This section discusses the assumptions, formulation and parameter estimation of the proposed model. Section 4 gives numerical results showing the data and prediction analysis of the J-M model and the modified J-M model using failure data set. Sensitivity analysis of the proposed model is also presented in this section. Finally, conclusions are drawn in section 5.

2. THE J-M MODEL

The J-M model [1,3,16] is one of the earliest and most widely cited software reliability models to describe the failure behavior of a software system. It belongs to the exponential failure time class of models [1].

2.1 Model Assumptions

The assumptions made in the J-M model include the following:

(i) The number of initial software faults is unknown but fixed and constant.

(ii) Each fault in the software is independent and equally likely to cause a failure during a test.

(iii) Time intervals between occurrences of failure are independent, exponentially distributed random variables.

(iv) The software failure rate remains constant over the intervals between fault occurrences.

(v) The failure rate is proportional to the number of faults that remain in the software.

(vi) A detected fault is removed immediately and no new faults are introduced during the removal of the detected fault.

(vii) Whenever a failure occurs, the corresponding fault is removed with certainty.

2.2 Model Formulation

If the time between failure occurrences are $T_i = t_i - t_{i-1},\ i = 1,2,....,N,$ then by the assumptions, the T_i 's are exponentially distributed random variable with parameter λ and mean is $1/\lambda$ [1].

From the assumptions, the software failure rate at the i^{th} failure interval i.e. the time between the $(i-1)^{th}$ and i^{th} failure is given by

$$\lambda(t_i) = \phi[N-(i-1)], \qquad i = 1,2,....,N \tag{1}$$

where

$\phi =$ a constant of proportionality denoting the failure rate contributed by each fault

$N =$ the initial number of faults in the software

$t_i =$ the time between $(i-1)^{th}$ and i^{th} failure.

The failure density function and distribution function are as follows:

$$f(t_i) = \phi[N-(i-1)]\exp(-\phi[N-(i-1)]t_i) \tag{2}$$

and

$$F_i(t_i) = 1-\exp(-\phi[N-(i-1)]t_i) \tag{3}$$

The reliability function at the i^{th} failure interval is given by

$$R(t_i) = 1-F_i(t_i) = \exp(-\phi[N-(i-1)]t_i) \tag{4}$$

and mean time to failure $(MTTF)$ for the i^{th} fault = $1/\phi[N-(i-1)]$.

2.3 Parameter Estimation

If the failure data set $\{t_1, t_2,....,t_n; n>0\}$ is given, the parameters N and ϕ in the J-M model can be estimated by using the maximum likelihood estimation method as follows:

$$\hat{\phi} = \frac{n}{\hat{N}\left(\sum_{i=1}^{n} t_i\right) - \sum_{i=1}^{n}(i-1)t_i} \tag{5}$$

and

$$\sum_{i=1}^{n}\frac{1}{\hat{N}-(i-1)} = \frac{n}{\hat{N} - \left(\frac{1}{\sum_{i=1}^{n} t_i}\right)\left(\sum_{i=1}^{n}(i-1)t_i\right)} \tag{6}$$

The maximum likelihood estimate of N i.e. $\hat{N}$ can be obtained by solving the equation (6). Substituting the estimated value of $\hat{N}$ from equation (6) into equation (5), we get the maximum likelihood estimate of ϕ i.e. $\hat{\phi}$ [1,16].

Then the current value of software reliability can be calculated by (4) as follows:

$$R(t_{n+1}) = 1-\hat{F}_{n+1}(t_{n+1}) = \exp(-\hat{\phi}(\hat{N}-n)t_{n+1}) \tag{7}$$

3. MODIFIED J-M MODEL WITH IMPERFECT DEBUGGING PHENOMENON

The imperfect debugging is a common practical situation and the J-M model does not take this into account. The assumptions (vi) and (vii) of the J-M model state that whenever a failure occurs, the detected fault is removed with certainty and no new faults are inserted during the removal of the detected fault. These are highly unrealistic assumptions for the J-M model. We extend the J-M model by relaxing the assumptions of perfect debugging process and considering the imperfect debugging process in fault removal activity. During an imperfect debugging process, there can be two types of imperfect removal: (i) the fault is not removed successfully while no new faults are introduced and (ii) the fault is not removed successfully while new faults are created due to incorrect diagnoses. We consider the second type of imperfect removal and this type of debugging process is known as a birth-death Markov process [17]. This is the most practical situation in fault removing activity. We allow the imperfect debugging process to introduce new faults into the software due to incorrect modifications or diagnoses.

3.1 Model Assumptions

The assumptions for our modified J-M model are similar to the J-M model except that it does not consider the perfect debugging process in fault removal activity. Our modified J-M model assumes that the debugging process is truly imperfect. In order to modify the J-M model with imperfect debugging, we replace the assumptions (vi) and (vii) of the J-M model by the following new assumption:

Whenever a failure occurs, the detected fault is removed with probability p, the detected fault is not perfectly removed with probability q and the new fault is generated with probability r. So it is obvious that $p+q+r=1$ and $q \geq r$.

3.2 Model Formulation

The software failure rate function between the $(i-1)^{th}$ and i^{th} failure for our modified J-M model with imperfect debugging is given by

$$\lambda(t_i) = \phi[N - p(i-1) + r(i-1)] = \phi[N-(i-1)(p-r)] \qquad (8)$$

where ϕ, N and t_i have the same meaning as defined in the J-M model.

The failure density and distribution functions are as follows

$$f(t_i) = \phi[N-(i-1)(p-r)]\exp(-\phi[N-(i-1)(p-r)]t_i) \qquad (9)$$

and

$$F_i(t_i) = 1-\exp(-\phi[N-(i-1)(p-r)]t_i) \qquad (10)$$

The reliability function at the i^{th} failure interval is given by

$$R(t_i) = 1-F_i(t_i) = \exp(-\phi[N-(i-1)(p-r)]t_i) \qquad (11)$$

and $MTTF$ for the i^{th} fault = $\frac{1}{\phi[N-(i-1)(p-r)]}$.

Note that if $p=1$ and $r=0$, then the failure behavior of the modified model becomes the same as the J-M model. Thus, the J-M model may be regarded as a special case of this modified model.

3.3 Parameter Estimation

Maximum likelihood estimation method has been used to estimate the parameters N and ϕ of our modified J-M model. The parameters N and ϕ are estimated as follows:

Suppose that the failure data set $\{t_1, t_2, ..., t_n; n>0\}$ is given as in the J-M model. The likelihood function of the parameters N and ϕ is given by

$$L(t_1, t_2, ..., t_n; \hat{N}, \hat{\phi})$$
$$= \prod_{i=1}^{n} f(t_i) = \prod_{i=1}^{n}\left[\hat{\phi}(\hat{N}-(i-1)(p-r))\exp(-\hat{\phi}(\hat{N}-(i-1)(p-r))t_i)\right]$$
$$= \hat{\phi}^n \prod_{i=1}^{n}[\hat{N}-(i-1)(p-r)]\exp\left(-\hat{\phi}\sum_{i=1}^{n}[\hat{N}-(i-1)(p-r)]t_i\right) \qquad (12)$$

Taking the natural logarithm of the above likelihood function, we get

$$\ln L$$
$$= \ln\left[\hat{\phi}^n \prod_{i=1}^{n}[\hat{N}-(i-1)(p-r)]\exp\left(-\hat{\phi}\sum_{i=1}^{n}[\hat{N}-(i-1)(p-r)]t_i\right)\right]$$
$$= n\ln\hat{\phi} + \sum_{i=1}^{n}\text{In}\,[\hat{N}'-(i-1)(p-r)] \qquad (13)$$
$$-\hat{\phi}\sum_{i=1}^{n}[\hat{N}-(i-1)(p-r)]t_i$$

By taking the first partial derivative of the above log-likelihood function with respect to $\hat{N}$ and $\hat{\phi}$, respectively, and equating them to zero, we get the following likelihood equations:

$$\frac{\partial}{\partial\hat{N}}\text{In}\,L = \sum_{i=1}^{n}\frac{1}{[\hat{N}-(i-1)(p-r)]} - \hat{\phi}\sum_{i=1}^{n}t_i = 0 \qquad (14)$$

and

$$\frac{\partial}{\partial\hat{\phi}}\text{In}\,L = \frac{n}{\hat{\phi}} - \sum_{i=1}^{n}[\hat{N}-(i-1)(p-r)]t_i = 0 \qquad (15)$$

From equation (15), we get

$$\hat{\varphi} = \frac{n}{\sum_{i=1}^{n}[\hat{N}-(i-1)(p-r)]t_i} = \frac{n}{\sum_{i=1}^{n}\hat{N}t_i - \sum_{i=1}^{n}(i-1)(p-r)t_i}$$
$$= \frac{n}{\hat{N}\sum_{i=1}^{n}t_i - \sum_{i=1}^{n}(i-1)(p-r)t_i} \qquad (16)$$

Now putting the value of $\hat{\phi}$ from equation (16) into equation (14), we obtain

$$\sum_{i=1}^{n}\frac{1}{[\hat{N}-(i-1)(p-r)]} = \frac{n\sum_{i=1}^{n}t_i}{\hat{N}\sum_{i=1}^{n}t_i - \sum_{i=1}^{n}(i-1)(p-r)t_i}$$

or,

$$\sum_{i=1}^{n}\frac{1}{[\hat{N}-(i-1)(p-r)]} = \frac{n}{\hat{N}-\left(\frac{1}{\sum_{i=1}^{n}t_i}\right)\left(\sum_{i=1}^{n}(i-1)(p-r)t_i\right)} \qquad (17)$$

We get the maximum likelihood estimate $\hat{N}$ by solving the equation (17) and putting this estimated value into equation (16) to obtain the maximum likelihood estimate $\hat{\phi}$.

The software reliability function can be obtained from (11) as follows:

$$R(t_{n+1}) = 1-\hat{F}_{n+1}(t_{n+1}) = \exp(-\hat{\phi}[\hat{N}-n(p-r)]t_{n+1}) \qquad (18)$$

The estimated mean time to failure for the $(n+1)^{th}$ fault is $\hat{MTTF} = \frac{1}{\hat{\phi}[\hat{N}-n(p-r)]}$.

4. NUMERICAL EXAMPLE

4.1 Model validation using Musa Data Set

In this section, we have concentrated on analysis of the software reliability data set published by Musa [18]. To check the validity of our modified J-M model, it is tested on the Musa system 1 failure data set. The values of p and r are supposed to be known. In all the existing software failure data sets, these values are not provided. The estimation of p and r from the failure data is also not possible since the parameters estimation tend to be unstable. Thus the values of p and r are assumed and for example, we consider as $p=0.93$ and $r=0.02$.

4.1.1 Data analysis

The data and estimates of parameters of both the J-M model and the modified J-M model with imperfect debugging are shown in Table 1.

Table 1. Analysis of Musa system 1 failure data

		J-M Model					Modified J-M Model				
N	t_n	$\hat{N}$	$\hat{\phi}$	$\hat{\lambda}$	$M\hat{T}TF$	$\hat{F}_{n+1}(t_{n+1})$	$\hat{N}$	$\hat{\phi}$	$\hat{\lambda}$	$M\hat{T}TF$	$\hat{F}_{n+1}(t_{n+1})$
1	3	1	0.3333	0	∞	0	1	0.3333	0.0300	33.3333	0.5934
2	30	2	0.0556	0	∞	0	2	0.0517	0.0093	107.5000	0.6505
3	113	3	0.0165	0	∞	0	3	0.0146	0.0040	253.1358	0.2738
4	81	4	0.0098	0	∞	0	4	0.0088	0.0032	315.2153	0.3057
5	115	6	0.0046	0.0046	218.6000	0.0403	5	0.0060	0.0027	372.1378	0.0239
6	9	11	0.0021	0.0105	95.2333	0.0208	10	0.0023	0.0105	95.3142	0.0208
7	2	∞	0	0.0198	50.4286	0.8354	∞	0	0.0198	50.4286	0.8354
8	91	28	0.00074218	0.0148	67.3688	0.8103	25	0.00083370	0.0148	67.6903	0.8088
9	112	16	0.0014	0.0099	100.7460	0.1383	15	0.0015	0.0102	98.2283	0.1416
10	15	46	0.00042405	0.0153	65.5056	0.8784	42	0.00046426	0.0153	65.4698	0.8785
11	138	20	0.0011	0.0098	102.1818	0.3870	18	0.0012	0.0097	103.1262	0.3842
12	50	27	0.00075572	0.0113	88.2167	0.5822	25	0.00081211	0.0114	87.4542	0.5854
13	77	29	0.00069496	0.0111	89.9327	0.2342	27	0.00074148	0.0112	88.9023	0.2366
14	24	61	0.00030036	0.0141	70.8359	0.7823	56	0.00032682	0.0141	70.7293	0.7828
15	108	39	0.00049358	0.0118	84.4167	0.6474	35	0.00055186	0.0118	84.8730	0.6454
16	88	38	0.00050881	0.0112	89.3352	0.9994	35	0.00055060	0.0113	88.8557	0.9995
17	670	18	0.0014	0.0015	686.2353	0.1604	17	0.0015	0.0022	449.2957	0.2344
18	120	20	0.0012	0.0023	429.9444	0.0587	18	0.0013	0.0021	470.3628	0.0538
19	26	23	0.00089852	0.0036	278.2368	0.3362	21	0.00098070	0.0036	274.8461	0.3395
20	114	25	0.00078204	0.0039	255.7400	0.7194	23	0.00084144	0.0040	247.5921	0.7309
21	325	24	0.00084378	0.0025	395.0476	0.1300	22	0.00091234	0.0026	379.2691	0.1350
22	55	27	0.00068427	0.0034	292.2818	0.5631	25	0.00072668	0.0036	276.3307	0.5835
23	242	28	0.00063944	0.0032	312.7739	0.1954	25	0.00073062	0.0030	336.2883	0.1831
24	68	31	0.00054131	0.0038	263.9107	0.7979	28	0.00060325	0.0037	269.1065	0.7916
25	422	29	0.00060835	0.0024	410.9500	0.3547	27	0.00063636	0.0027	369.7528	0.3854
26	180	31	0.00053751	0.0027	372.0846	0.0265	28	0.00060006	0.0026	383.9882	0.0257
27	10	35	0.00043850	0.0035	285.0602	0.9821	32	0.00047767	0.0035	281.7638	0.9829
28	1146	30	0.00057648	0.0012	867.3393	0.4993	28	0.00059192	0.0015	670.3998	0.5914
29	600	31	0.00052915	0.0011	944.9138	0.0157	29	0.00053854	0.0014	711.4468	0.0209
30	15	33	0.00046201	0.0014	721.4778	0.0487	30	0.00050901	0.0014	727.6242	0.0483
31	36	35	0.00041217	0.0016	606.5403	0.0066	32	0.00044795	0.0017	589.0268	0.0068
32	4	38	0.00035361	0.0021	471.3229	0	34	0.00040303	0.0020	508.4395	0
33	0	41	0.00031202	0.0025	400.6098	0.0198	38	0.00033081	0.0026	379.2781	0.0209
34	8	46	0.00025893	0.0031	321.8382	0.5061	42	0.00028285	0.0031	319.6620	0.5084
35	227	47	0.00025115	0.0030	331.8048	0.1779	43	0.00027335	0.0030	328.0981	0.1797
36	65	52	0.00021546	0.0034	290.0747	0.4549	47	0.00023949	0.0034	293.2293	0.4513
37	176	54	0.00020439	0.0035	287.8045	0.1825	49	0.00022567	0.0035	289.0592	0.1818
38	58	60	0.00017613	0.0039	258.0778	0.8298	55	0.00019135	0.0039	255.9210	0.8323
39	457	55	0.00019958	0.0032	313.1522	0.6163	50	0.00021970	0.0032	313.6941	0.6157
40	300	56	0.00019362	0.0031	322.7922	0.2596	51	0.00021248	0.0031	322.3448	0.2599
41	97	60	0.00017518	0.0033	300.4454	0.5833	54	0.00019608	0.0033	305.5672	0.5771
42	263	61	0.00017088	0.0032	308	0.7695	55	0.00019071	0.0032	312.4837	0.7646
43	452	58	0.00018471	0.0028	360.9240	0.5066	53	0.00020147	0.0028	357.8521	0.5096
44	255	60	0.00017494	0.0028	357.2656	0.4239	55	0.00018977	0.0028	352.2385	0.4284
45	197	62	0.00016669	0.0028	352.8824	0.4213	57	0.00017993	0.0029	346.2687	0.4273
46	193	65	0.00015501	0.0029	339.5275	0.0175	59	0.00017109	0.0029	341.0078	0.0174
47	6	71	0.00013666	0.0033	304.8927	0.2283	65	0.00014872	0.0033	302.4713	0.2299
48	79	77	0.00012203	0.0035	282.5769	0.9443	70	0.00013431	0.0035	282.8895	0.9441
49	816	67	0.00014865	0.0027	373.7313	0.9731	61	0.00016321	0.0027	373.3741	0.9732
50	1351	59	0.00018299	0.0016	607.1933	0.2163	54	0.00019859	0.0017	592.4077	0.2211
51	148	61	0.00017286	0.0017	578.5157	0.0356	56	0.00018647	0.0018	559.2125	0.0369
52	21	65	0.00015463	0.0020	497.4630	0.3740	59	0.00017078	0.0020	501.3182	0.3717
53	233	67	0.00014710	0.0021	485.5741	0.2412	61	0.00016149	0.0021	484.8984	0.2414
54	134	70	0.00013704	0.0022	456.0729	0.5429	64	0.00014927	0.0022	450.8378	0.5470
55	357	71	0.00013390	0.0021	466.7511	0.3387	65	0.00014548	0.0022	459.7839	0.3428
56	193	74	0.00012518	0.0023	443.8036	0.4124	67	0.00013886	0.0022	448.9809	0.4088
57	236	76	0.00012015	0.0023	438.0646	0.0683	69	0.00013259	0.0023	440.2864	0.0680
58	31	81	0.00010898	0.0025	398.9678	0.6034	73	0.00012180	0.0025	406.0499	0.5970
59	369	81	0.00010912	0.0024	416.5716	0.8340	74	0.00011908	0.0024	413.4820	0.8362
60	748	78	0.00011550	0.0021	481.0083	0	71	0.00012685	0.0021	480.6770	0

61	0	82	0.00010707	0.0022	444.7502	0.4065	75	0.00011659	0.0023	440.0930	0.4097
62	232	85	0.00010115	0.0023	429.8527	0.5359	77	0.00011205	0.0023	433.6545	0.5328
63	330	86	0.00009942	0.0023	437.3230	0.5660	78	0.00010990	0.0023	440.2042	0.5636
64	365	87	0.00009763	0.0022	445.3546	0.9357	79	0.00010770	0.0022	447.2760	0.9349
65	1222	81	0.00010917	0.0017	572.5192	0.6127	73	0.00012232	0.0017	590.2688	0.6014
66	543	81	0.00010925	0.0016	610.2101	0.0163	74	0.00011910	0.0017	602.3228	0.0165
67	10	85	0.00010071	0.0018	551.6600	0.0286	77	0.00011165	0.0018	558.7317	0.0282
68	16	89	0.00009357	0.0020	508.8922	0.6464	81	0.00010280	0.0020	508.7435	0.6465
69	429	89	0.00009352	0.0019	534.6428	0.5078	81	0.00010275	0.0019	534.4750	0.5079
70	379	90	0.00009192	0.0018	543.9800	0.0777	82	0.00010077	0.0018	542.2709	0.0779
71	44	94	0.00008581	0.0020	506.6552	0.2248	86	0.00009338	0.0020	500.6751	0.2271
72	129	98	0.00008038	0.0021	478.5080	0.8160	89	0.00008865	0.0021	480.4346	0.8147
73	810	95	0.00008436	0.0019	538.8064	0.4162	86	0.00009362	0.0018	545.7881	0.4122
74	290	97	0.00008162	0.0019	532.6786	0.4306	88	0.00009021	0.0019	536.5569	0.4283
75	300	99	0.00007903	0.0019	527.2417	0.6333	90	0.00008700	0.0019	528.4436	0.6325
76	529	99	0.00007902	0.0018	550.1894	0.3999	90	0.00008700	0.0018	551.5263	0.3992
77	281	101	0.00007659	0.0018	544.0097	0.2548	92	0.00008401	0.0018	542.7668	0.2553
78	160	105	0.00007195	0.0019	514.7588	0.7998	95	0.00007990	0.0019	521.0477	0.7959
79	828	102	0.00007540	0.0017	576.6483	0.8268	93	0.00008255	0.0017	573.8421	0.8283
80	1011	99	0.00007926	0.0015	664.0276	0.4884	91	0.00008536	0.0016	643.6806	0.4991
81	445	101	0.00007643	0.0015	654.1981	0.3639	92	0.00008382	0.0015	652.2555	0.3648
82	296	103	0.00007399	0.0016	643.6336	0.9346	94	0.00008084	0.0016	638.2868	0.9360
83	1755	98	0.00008059	0.0012	827.2104	0.7237	89	0.00008896	0.0012	834.5513	0.7206
84	1064	97	0.00008221	0.0011	935.6319	0.8513	88	0.00009098	0.0011	950.7763	0.8467
85	1783	95	0.00008559	0.00085593	1168.3	0.5210	87	0.00009260	0.00089357	1119.1	0.5363
86	860	96	0.00008362	0.00083622	1195.9	0.5604	87	0.00009285	0.00081149	1232.3	0.5496
87	983	96	0.00008379	0.00075414	1326	0.4133	88	0.00009038	0.00079807	1253	0.4312
88	707	97	0.00008201	0.00073805	1354.9	0.0241	89	0.00008824	0.00078709	1270.5	0.0256
89	33	100	0.00007680	0.00084479	1183.7	0.5197	91	0.00008439	0.00084479	1183.7	0.5197
90	868	100	0.00007703	0.00077027	1298.2	0.4275	91	0.00008464	0.00077027	1298.2	0.4275
91	724	101	0.00007547	0.00075472	1325	0.8268	92	0.00008273	0.00076031	1315.3	0.8290
92	2323	100	0.00007689	0.00061510	1625.8	0.8351	92	0.00008205	0.00067937	1472	0.8634
93	2930	99	0.00007850	0.00047102	2123.1	0.4975	93	0.00007875	0.00065918	1517	0.6183
94	1461	99	0.00007877	0.00039382	2539.2	0.2825	94	0.00007643	0.00064659	1546.6	0.4202
95	843	101	0.00007463	0.00044779	2233.2	0.0054	95	0.00007453	0.00063722	1569.3	0.0076
96	12	102	0.00007325	0.00043947	2275.4	0.1084	96	0.00007315	0.00063198	1582.3	0.1521
97	261	104	0.00006988	0.00048917	2044.3	0.5854	97	0.00007171	0.00062606	1597.3	0.6760
98	1800	104	0.00006997	0.00041980	2382.1	0.3045	98	0.00006960	0.00061388	1629	0.4120
99	865	106	0.00006658	0.00046604	2145.7	0.4877	99	0.00006798	0.00060573	1650.9	0.5807
100	1435	106	0.00006680	0.00040079	2495.1	0.0120	100	0.00006618	0.00059561	1678.9	0.0177
101	30	108	0.00006387	0.00044710	2236.7	0.0619	101	0.00006502	0.00059103	1691.9	0.0810
102	143	110	0.00006120	0.00048959	2042.5	0.0515	102	0.00006388	0.00058638	1705.4	0.0614
103	108	112	0.00005879	0.00052906	1890.1	0	103	0.00006280	0.00058215	1717.8	0
104	0	114	0.00005662	0.00056622	1766.1	0.8281	104	0.00006181	0.00057859	1728.3	0.8346
105	3110	113	0.00005762	0.00046092	2169.6	0.4372	105	0.00005976	0.00056476	1770.7	0.5055
106	1247	114	0.00005641	0.00045129	2215.9	0.3466	106	0.00005839	0.00055704	1795.2	0.4086
107	943	115	0.00005532	0.00044256	2259.6	0.2664	107	0.00005716	0.00055044	1816.7	0.3198
108	700	116	0.00005432	0.00043459	2301	0.3163	108	0.00005604	0.00054476	1835.7	0.3791
109	875	118	0.00005207	0.00046861	2134	0.1085	109	0.00005492	0.00053877	1856.1	0.1237
110	245	119	0.00005128	0.00046152	2166.7	0.2857	110	0.00005401	0.00053474	1870.1	0.3228
111	729	121	0.00004929	0.00049291	2028.8	0.6074	111	0.00005301	0.00052959	1888.3	0.6338
112	1897	121	0.00004932	0.00044388	2252.9	0.1800	112	0.00005173	0.00052149	1917.6	0.2079
113	447	123	0.00004748	0.00047481	2106.1	0.1675	113	0.00005086	0.00051723	1933.4	0.1810
114	386	124	0.00004679	0.00046786	2137.4	0.1883	114	0.00005003	0.00051330	1948.2	0.2046
115	446	126	0.00004514	0.00049652	2014	0.0588	115	0.00004922	0.00050940	1963.1	0.0603
116	122	128	0.00004368	0.00052411	1908	0.4048	117	0.00004745	0.00054282	1842.2	0.4157
117	990	129	0.00004298	0.00051575	1938.9	0.3867	118	0.00004660	0.00053735	1861	0.3992
118	948	131	0.00004148	0.00053930	1854.2	0.4421	119	0.00004579	0.00053212	1879.3	0.4377
119	1082	132	0.00004082	0.00053072	1884.2	0.0116	120	0.00004498	0.00052668	1898.7	0.0115
120	22	134	0.00003963	0.00055479	1802.5	0.0408	122	0.00004349	0.00055672	1796.2	0.0409
121	75	136	0.00003851	0.00057759	1731.3	0.2430	124	0.00004211	0.00058497	1709.5	0.2457
122	482	138	0.00003738	0.00059810	1672	0.9629	126	0.00004075	0.00061043	1638.2	0.9654
123	5509	134	0.00003966	0.00043622	2292.4	0.0427	123	0.00004258	0.00047135	2121.6	0.0460
124	100	136	0.00003841	0.00046096	2169.4	0.0046	124	0.00004200	0.00046870	2133.6	0.0047

125	10	138	0.00003727	0.00048457	2063.7	0.4049	126	0.00004061	0.00049749	2010.1	0.4130
126	1071	139	0.00003672	0.00047740	2094.7	0.1623	127	0.00003994	0.00049285	2029	0.1671
127	371	141	0.00003563	0.00049886	2004.6	0.3257	129	0.00003863	0.00051880	1927.5	0.3363
128	790	143	0.00003455	0.00051822	1929.7	0.9587	130	0.00003806	0.00051457	1943.4	0.9578
129	6150	139	0.00003672	0.00036720	2723.3	0.7046	129	0.00003822	0.00044373	2253.6	0.7709
130	3321	139	0.00003666	0.00032992	3031	0.2916	130	0.00003727	0.00043605	2293.3	0.3660
131	1045	140	0.00003608	0.00032470	3079.8	0.1897	131	0.00003664	0.00043193	2315.2	0.2441
132	648	141	0.00003555	0.00031996	3125.4	0.8271	132	0.00003607	0.00042852	2333.6	0.9047
133	5485	140	0.00003613	0.00025293	3953.6	0.2543	133	0.00003494	0.00041827	2390.8	0.3844
134	1160	141	0.00003553	0.00024870	4020.9	0.3710	134	0.00003433	0.00041408	2415	0.5378
135	1864	142	0.00003489	0.00024423	4094.4	0.6341	135	0.00003367	0.00040906	2444.6	0.8143
136	4116	142	0.00003489	0.00020934	4777		136	0.00003278	0.00040126	2492.1	

Since the relationship between the reliability function $R(t_i)$ and the failure distribution function $F_i(t_i)$ are related by $R(t_i)=1-F_i(t_i)$, the quality of reliability prediction of the models may be judged by examining the estimated failure distribution functions of the models.
From Table 1, we can see that normally the failure rate for our modified J-M model with imperfect debugging is greater than or equal to the failure rate of the J-M model. It is also noted that the MTTF is infinite for some intervals of times between failures in J-M model. So this model assumes that there are no more faults in the software although faults are present in the software at that time. In our modified J-M model, the MTTF is not infinite for every interval between failures. So this is more realistic in practical sense as it implies that till then fault is remaining in the software. This is the most practical situation in software fault debugging process. When $p \to 1$ and $r \to 0$ i.e. for perfectly debugging procedure, our considered imperfect debugging modified J-M model approaches to the J-M model.

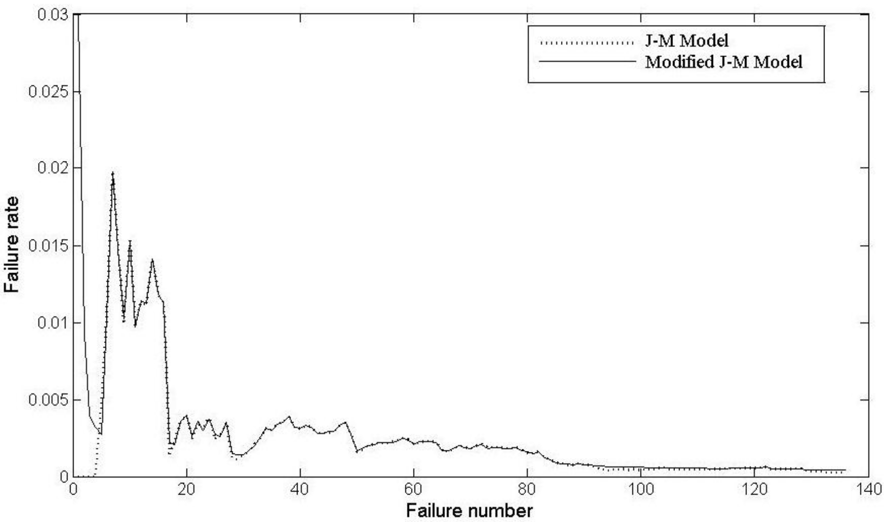

Fig 1: Failure rate as a function of failure number

A plot of failure rate against failure number for the J-M model and the modified J-M model is shown in Figure 1. From the graph of Figure 1, we get the clear idea about the failure behavior of the two models. The failure rate is maximum at the beginning of the testing process. As the fault content of the software is decreasing i.e. the failure number is increasing, the failure rate of our model is decreasing. This practical event occurs at the time of debugging.

4.1.2 Prediction analysis

In order to compare the quality of prediction of the two models, we consider the sequences of probabilities for $T_i = t_i$. Let $u_i = \hat{F}_i(t_i)$. If each of the estimated $\hat{F}_i$ is equal to the true F_i, the sequence $\{u_i\}$ should look like a realization of independent uniform $U(0,1)$ random variables [19,20,21]. We can examine the quality of each model by plotting the sample distribution function of the u_i's and comparing it with the distribution function of $U(0,1)$, which is the line of unity slope through the origin. We use the quantile-quantile plot [4] which is a plot of the order set of n number of u_i's against i/n, where n is the sample size and i is the rank of the sample point. The closeness of the plot to the line of unity slope is an indication of the closeness of the u_i's to uniformity and so an indication of the quality of prediction of the model. The closer the plot to the line of unity slope, the more likely the u_i's is from a random sample of the uniform distribution, and so the better the model. The goodness-of-fit test can be performed by formal statistical tests. In this paper, we have used the Kolmogorov-Smirnov (KS) test. KS distance is the maximum vertical deviation between the plot and the line of unity slope.

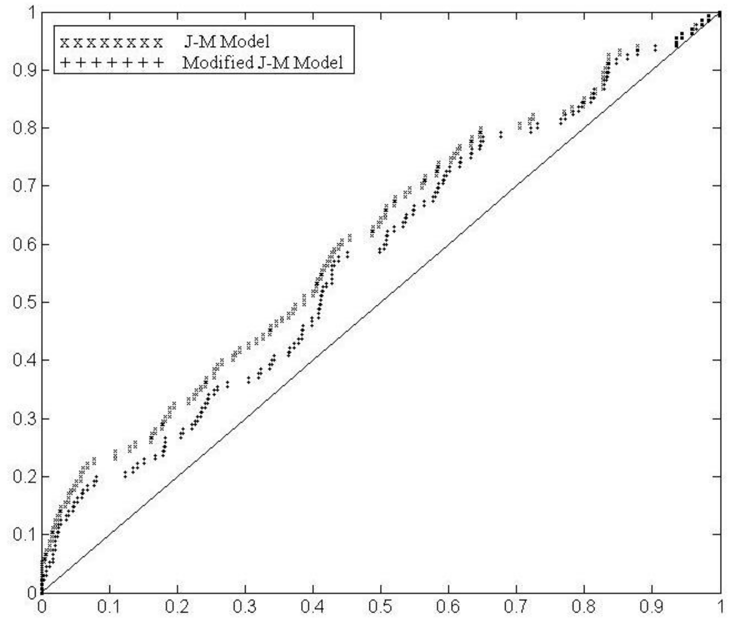

Fig 2: Quantile-quantile plot of the data in Table 1

The quantile-quantile plot of the data in Table 1 is shown in Figure 2. The KS distance is 0.165 for the J-M model, and 0.14 for the modified model. The results show that the modified model yields a better prediction in this case.

4.2 Sensitivity Analysis of the Proposed Model

To study the performance of our modified J-M model with imperfect debugging, sensitivity analyses have been presented graphically on the different values of p and r keeping q as constant. First, we consider the fixed value of q at 0.05 and the values of p and r are taken as follows:

$p = 0.94$ and $r = 0.01$; $p = 0.92$ and $r = 0.03$; $p = 0.9$ and $r = 0.05$

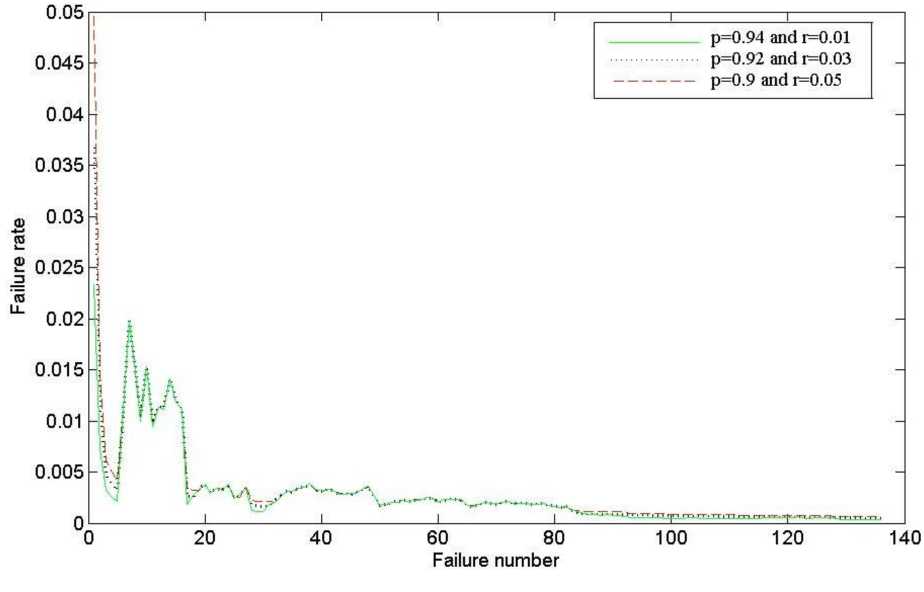

Fig 3: Failure rate against failure number for different values of p and r

A plot of failure rate versus failure number for the above different values of p and r of the modified J-M model is shown in Figure 3. From the graph, we can see that the failure rate is maximum for $p = 0.9$ and $r = 0.05$. But as p is increasing and r is decreasing, we are able to remove more and more faults perfectly. So the failure rate is decreasing which can be found from the Figure 3.

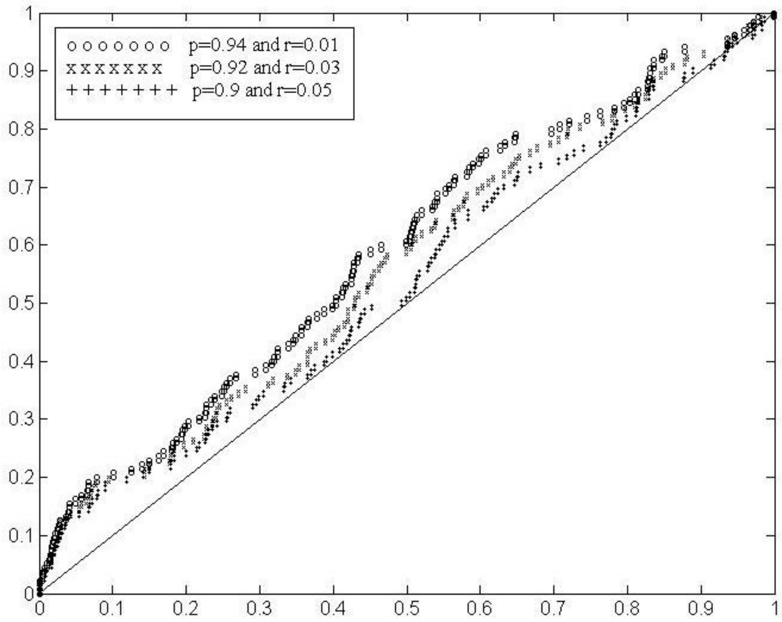

Fig 4: Quantile-quantile plots for different values of p and r

Quantile-quantile plots for the different values of p and r on the fixed value of q at 0.05 of our modified J-M model are shown in Figure 4.

Now we show the KS distances for the different values of q keeping it constant for the different values of p and r in Table 2.

Table 2. KS distances for different values of p and r for different value of q

For q=0.07			For q=0.05			For q=0.04		
p	R	KS distance	p	r	KS distance	p	r	KS distance
0.92	0.01	0.140078	0.94	0.01	0.155263	0.95	0.01	0.161781
0.9	0.03	0.110993	0.92	0.03	0.115093	0.94	0.02	0.145678
0.88	0.05	0.093693	0.9	0.05	0.101993	0.93	0.03	0.129978

From Table 2, we can see that for fixed value of q, as p is decreasing and r is increasing, the KS distance is decreasing. So we get better predictions as p is decreasing and r is increasing for some fixed value of q. Again, for fixed value of r, as p is increasing and q is decreasing, the KS distance is increasing. It is also noted that for fixed value of p, as q is decreasing and r is increasing, the KS distance is decreasing. So in this case we also get better predictions. From Table 2, it is clear that we get better predictions in our model for the smaller value of (p-r). In reality, the value of (p-r) is less than 1 and it decreases in most of the cases of debugging process during testing. So our model excellently predicts the software reliability in practical sense.

5. CONCLUSION

We develop a modified J-M model which assumes the imperfect debugging process in fault removal activity during the testing phase. In our proposed model, we assume that whenever a failure occurs, the detected fault is not perfectly removed and there is a chance of raising new fault/s. We consider that the perfect debugging probability, the imperfect debugging probability and the probability of arising new fault are independent of the testing time. A set of failure data is given for illustration. From the experimental results, we have seen that the failure rate is normally high for our modified J-M model. So we need to increase the probability of perfect debugging and to decrease the probability of raising new fault/s during the fault removing process. Again, the difference between the probability of perfect debugging and the probability of raising new fault/s should be decreased to get better software reliability prediction. The experimental results also show that our proposed model yields a better prediction than the J-M model.

6. ACKNOWLEDGEMENTS

The authors wish to acknowledge the financial support given to this work through the research project (No. 25(0191)/10/EMR-II) by the Council of Scientific and Industrial Research, New Delhi, India.

7. REFERENCES

[1] Lyu, M.R. 1996. Handbook of Software Reliability Engineering. McGraw-Hill.

[2] Musa, J.D., Iannino, A., and Okumoto, K. 1990. Software Reliability: Measurement, Prediction, Application. McGraw-Hill.

[3] Jelinski, Z. and Moranda, P.B. 1972. Software reliability research, Statistical Computer Performance Evaluation. Academic Press: New York, 465-484.

[4] Littlewood, B. 1987. How good are software reliability predictions?. Software Reliability: Achievement and Assessment. Blackwell Scientific Publications. 154-166.

[5] Musa J.D. 1975. A theory of software reliability and its application. IEEE T. Software Eng. 1(3), 312-327.

[6] Goel, A.L., and Okumoto, K. 1979. Time dependent error detection rate model for software reliability and other performance measures. IEEE T. Reliab. R-28(3), 206-211.

[7] Yamada, S., Ohba, M. and Osaki, S. 1983. S-shaped reliability growth modeling for software error detection. IEEE T. Reliab. R-32(5), 475-484.

[8] Ohba, M. 1984. Software reliability analysis models. IBM J. Res. Dev. 28(4), 428-443.

[9] Goel, A.L. 1985. Software reliability models: assumptions, limitations and applicability. IEEE T. Software Eng. SE-11(12), 1411-1423.

[10] Kapur, P.K., and Garg, R.B. 1990. Optimal release policy for software reliability growth models under imperfect debugging. Oper. Res. RAIRO. 24(3), 295-305.

[11] Chang, Y.C., and Liu, C.T. 2009. A generalized JM model with applications to imperfect debugging in software reliability. Appl. Math. Model. 33, 3578-3588.

[12] Shyur, H.J. 2003. A stochastic software reliability model with imperfect-debugging and change-point. J. Syst. Software. 66(2), 135-141.

[13] Kapur, P.K., Singh, O.M.P., Shatnawi, O., and Gupta, A. 2006. A discrete NHPP model for software reliability growth with imperfect fault debugging and fault generation. Int. J. Perform. Eng. 2(4), 351-368.

[14] Prasad, R.S., Raju, O.N., and Kantam, R.R.L. 2010. SRGM with imperfect debugging by genetic algorithms. Int. J. Software Eng. Appl. 1(2), 66-79.

[15] Raju, O.N. 2011. Software reliability growth models for the safety critical software with imperfect debugging. Int. J. Comput. Sci. Eng. 3(8), 3019-3026.

[16] Xie, M. Dai, Y.S. and Poh, K.L. 2004. Computing System Reliability Models and Analysis. Kluwer Academic Publisher.

[17] Kremer, W. 1983. Birth-death and bug counting. IEEE T. Reliab. R-32(1), 37-47.

[18] Musa, J.D. 1980. Software Reliability Data. Data & Analysis Center for Software.

[19] Dawid, A.P. 1984. Statistical theory: the prequential approach. J. Roy. Stat. Soc. A. 147, 278-292.

[20] Pham, H. 2006. System Software Reliability. Springer.

[21] Bittanti, S. 1988. Software Reliability Modelling and Identification (Lecture Notes in Computer Science). Springer-Verlag.