

ニューラルネットワークに基づく分散ソフトウェア開発 環境における信頼性評価法に関する適合性評価の一考察

02302815 鳥取大学 *田村 慶信 TAMURA Yoshinobu
01702425 鳥取大学 山田 茂 YAMADA Shigeru
02101865 法政大学 木村 光宏 KIMURA Mitsuhiro

1 はじめに

現在における情報化社会の公共基盤として、また、経営と情報を一体化している企業の情報化技術としてコンピュータシステムへの依存度はますます大きくなってきている。特に、ソフトウェア開発を取り巻く環境は、クライアント/サーバ・システム (Client/Server System, 以下 CSS と略す) 開発やネットワーク環境での分散開発といった新しい開発形態が多用されるようになってきている。開発されたシステムの故障による長時間のシステムダウンは、我々の社会生活に重大な影響を及ぼす可能性がある。その原因は、開発工程で潜した欠陥や誤り、いわゆるソフトウェアフォールト (software fault) によるものが多い。この対策には、ネットワークで相互接続された分散開発環境の下で開発されたソフトウェアシステムの信頼性評価が重要となる。

本論文では、こうした分散開発環境を対象としたニューラルネットワークに基づくソフトウェア信頼性評価法として、従来からの時系列分析に基づく手法とソフトウェアコンポーネント間の相互作用を考慮した手法との適合性比較を行う。

2 ニューラルネットワークに基づくソフトウェア信頼性評価法

2.1 時系列分析に基づくソフトウェア信頼性評価法

時系列 (time series) とは、特定の対象についての状態の時間的な推移を時間の経過に沿って観測・記録したデータ系列であり、その状態の時間的な変動・振る舞いを記述するものである。いま、特定の対象の状態を x とすると、その時系列は離散的な時間のインデックスを用いて ($x_i: i = 1, 2, \dots, n$) のように表現することができる。ここで、 i は状態を観測された時点を表す。従来のニューラルネットワークに基づくソフトウェア信頼性評価法は、ソフトウェア開発のテスト工程で一定のテスト時間間隔 $(0, t_k]$ において発見された総フォールト数 y_k に関する K 組のフォールト発見数データ $(t_k, y_k) (k = 1, 2, \dots, K)$ をニューラルネットワークの入力層に適用する手法であり、パラメトリックモデルと比較してソフトウェアのフォールト発見過程に対する物理的な意味付けを必要としないという特徴をもつ。

2.2 分散開発環境下におけるソフトウェア信頼性評価法

クライアント/サーバ処理単位としての単体テスト工程においては、コンポーネントの規模が小さいことから信頼性評価のための十分な量のフォールト発見数データが採取されにくい。したがって、フォールト発見数データを使用した信頼性評価を行うにあたり、本章では、単体プログラムを結合した後の段階であるサブシステム内での統合テスト工程以降を対象とした信頼性評価法を示す [1]。

2.2.1 サブシステム内でのテスト工程

従来から、ソフトウェアの信頼性を定量的に評価する手法として、ソフトウェア信頼度成長モデルによる方法がとられている。中でも非同次ポアソン過程 (nonhomogeneous Poisson process, 以下 NHPP と略す) モデルは、実利用上極めて有効でありモデルの簡潔性が高いゆえにその適用性も高く、実際のソフトウェア信頼性評価に広く応用されている。サブシステム内でのテスト工程においては、各サブシステムについて累積フォールト発見数データの成長曲線の形状により、NHPP に基づく指数形ソフトウェア信頼度成長モデルと遅延 S 字形ソフトウェア信頼度成長モデルを用いた信頼性評価法の適用を前提とする [2]。

2.2.2 システム全体としての総合テスト工程

分散開発環境の総合テスト工程に対してニューラルネットワークを適用する。入力層には、NHPP モデルから導出されるソフトウェア信頼度の値を適用する。ソフトウェア信頼度は、テスト時刻 t までテストが進行しているときに、時間区間 $(t, t+x]$ ($t \geq 0, x \geq 0$) においてソフトウェア故障の発生しない条件付き確率と定義され、それぞれ各モデルに対して、

$$R_e(x|t) = \exp[E(t) - E(t+x)], \quad (1)$$

$$R_d(x|t) = \exp[D(t) - D(t+x)], \quad (2)$$

により与えられる [2]。ここで、 $E(t)$ および $D(t)$ は NHPP モデルにおける平均値関数であり、時間区間 $(0, t]$ において発見される期待累積フォールト数を表す。さらに、テスト時刻 t はサブシステム内におけるテスト終了時刻を、テスト時刻 x は総合テスト工程におけるテスト時刻を表す。

また、ネットワークの学習を行うために、誤差逆伝播法 [3] を用いる。ここで、教師パターンには、実際の総合テスト工程から得られる累積発見フォールト数データの正規化された値を採用する。すなわち、総合テスト工程において逐次に得られる累積フォールト発見数データから、各ソフトウェアコンポーネントの結合状態の特徴をニューラルネットワークの結合係数に蓄積させ、その都度その結合係数のもとにおいて、将来の累積発見フォールト数の推定・予測が可能なモデルを考える。

3 数値例と適合性評価

実際のテスト工程 (総合テスト) において観測されたデータを適用して数値例を示す。ここに示す数値例は、実際にある企業で開発されたソフトウェアシステムにおけるソフトウェアプロジェクトデータに基づいている。本論文で用いたデータは、9つのソフトウェアコンポーネントから構成されたソフトウェアシステムのテスト工程から採取され

た 26 組のフォールト発見数データ ($K = 26, t_{26} = 26$) である。また、テスト時間 t_k の測定単位は日である。

分散開発環境を対象としたニューラルネットワークによる信頼性評価法 (以下 DDE と略す) と、従来から用いられてきた時系列分析に基づくニューラルネットワークによる手法 (以下 TS と略す) との実測データに対する適合性比較を行う。また、適合性評価基準としては平均偏差 2 乗和 (mean squared errors, 以下 MSE と略す) および予測相対誤差を用いた [2, 4]。さらに、適合性比較を行うにあたり、短期・長期的な推定値のふるまいを比較するためにテスト期間を以下のように分類する。

Case 1: 20 日目までのデータを学習データとし、21 日目以降の推定フォールト数および実測値との誤差を比較する。

Case 2: 10 日目までのデータを学習データとし、11 日目以降の推定フォールト数および実測値との誤差を比較する。

まず、MSE の実測データに対する比較結果を表 1 に示す。表 1 から、DDE による手法の実測データに対する適合性が良いことが確認できる。特に、Case 1 よりも Case 2 における MSE の差が大きいことから、DDE による長期的なフォールト数の推定精度が良いことが確認できる。

表 1: 累積発見フォールト数に関する平均偏差 2 乗和の比較。

Compared methods	Case 1	Case 2
DDE	0.68346	2.4449
TS	1.1170	19.106

次に、Case 1 における予測相対誤差を用いた適合性比較結果を図 1 に示す。図 1 より、両者ともテスト時間の経過とともに推定値と実測値の差は大きくなるが、TS よりも DDE による推定値の方が安定していることが確認できる。さらに、Case 2 における予測相対誤差を用いた適合性比較結果を図 2 に示す。図 2 より、テスト時間の経過とともに両者の手法による推定値の差が大きくなる様子が確認できる。特に、テスト進捗率 60% まではある程度同等の推定精度を示すが、DDE についてはテスト進捗率 70% 以降においても推定値が安定している様子が確認できる。

以上の適合性比較結果から、従来の TS よりも DDE の実測データに対する適合性が良いことが確認できる。特に、長期的な予測に関しても高い精度が得られた。これは、従来の TS においては、時系列の外形としての特徴を結合係数に蓄積させているのに対して、DDE は各ソフトウェアコンポーネントの結合状態の特徴をニューラルネットワークの結合係数に蓄積させたものである。すなわち、システム全体の状態を包括した手法であることに起因するのではないかと考えられる。

4 おわりに

本論文では、ソフトウェアの部品化により複雑化したネットワーク環境の下で分散開発されたソフトウェアシステムの信頼性評価法として、従来からの TS と DDE について議論した。さらに数値例として、その具体例を実測データ

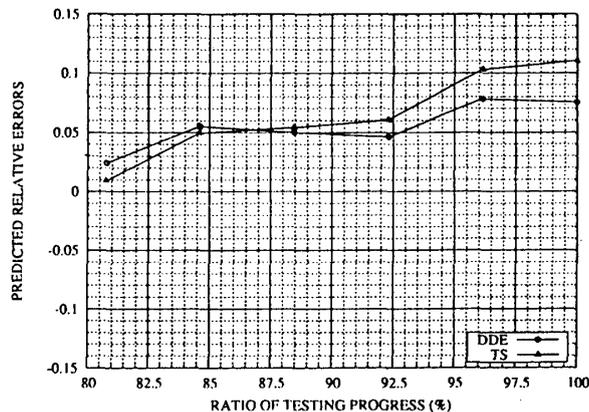


図 1: Case 1 における予測相対誤差の推定結果。

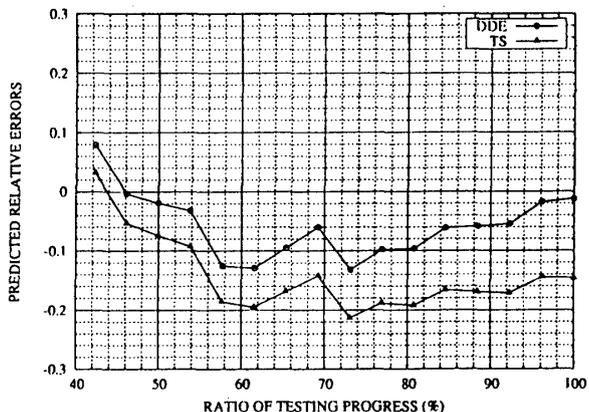


図 2: Case 2 における予測相対誤差の推定結果。

に適用して示した。その結果、従来の TS に比べて DDE の実測データに対する適合性が良いことを確認した。これは、TS が時系列の外形としての特徴を結合係数に蓄積させているのに対して、DDE はソフトウェアシステム全体の状態を包括した手法であることに起因する。したがって、従来の TS と比較して、より高い精度で将来の累積発見フォールト数を推定することが可能となる。

参考文献

- [1] 田村 慶信, 山田 茂, 木村 光宏, “分散開発環境におけるニューラルネットワークに基づくソフトウェア信頼性評価法,” 電子情報通信学会論文誌, Vol. J85-A, No. 11, pp. 1236-1243, 2002 年 11 月。
- [2] 山田 茂, ソフトウェア信頼性モデル—基礎と応用—, 日科技連出版社, 1994。
- [3] 吉富 康成, ニューラルネットワーク, 朝倉書店, 2002。
- [4] A. Iannino, J. D. Musa, K. Okumoto, and B. Littlewood, “Criteria for software reliability model comparisons,” *IEEE Trans. Software Engineering*, vol. SE-10, no. 6, pp. 687-691, Nov. 1984。