

# 软件可靠性混沌神经网络模型

张 柯 张德平 汪 帅

(南京航空航天大学计算机科学与技术学院 南京 210016)

**摘 要** 基于经验模态分解算法、混沌分析和神经网络理论提出了一种软件可靠性建模及预测的混沌神经网络模型。首先应用经验模态分解算法把软件失效数据序列分解成不同尺度的基本模态分量,并在此基础上进一步分析,表明软件失效数据是否存在混沌特性;再经神经网络进行组合预测,提高模型对目标函数的学习能力,有效提高预测精度;最后基于两组真实软件失效数据集,将所提出的方法与基于支持向量回归机以及单纯使用神经网络的软件可靠性预测模型进行比较分析。结果表明,基于混沌分析、结合经验模态分解和神经网络的软件可靠性预测模型具有更为显著的模型拟合能力与精确的预测效果。

**关键词** 经验模态分解,软件可靠性模型,软件可靠性预测,神经网络模型,混沌分析

**中图法分类号** TP311

**文献标识码** A

## Chaotic Neural Network Model for Software Reliability

ZHANG Ke ZHANG De-ping WANG Shuai

(College of Computer Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China)

**Abstract** A forecasting method based on Empirical Mode Decomposition (EMD), chaos analysis and neural network theory was presented to model and forecast software reliability forecasting. First, using EMD theory, the software failure data time serial is decomposed into many intrinsic modal functions (IMF) which are an significantly represent potential information of original time serial, and the further analysis of IMF indicates whether software failure has a chaos feature. Then, by using chaos theory and neural network, the forecasting models are established to forecast the IMF respectively. By these means, the model can be improved to learn various objective functions and more precious prediction can be obtained. After comparing the results forecasted by means of combination of SVR and neural network, it is proved that the effect of the forecasting method of EMD&GEP in software reliability forecasting is better.

**Keywords** Empirical mode decomposition (EMD), Software reliability model, Software reliability prediction, Neural network mode, Chaos analysis

## 1 引言

现代计算机软件系统的规模日益庞大、功能越来越复杂,导致其可靠性保障的难度也越来越大,因此计算机软件系统的可靠性已成为社会广泛关注的焦点<sup>[1]</sup>。软件可靠性分析是实施软件可靠性工程、保证软件产品质量的重要实际过程,它贯穿软件开发的所有阶段。软件可靠性分析以数学为工具,综合运用随机过程、数理统计和数据分析等方面的数学知识,对软件的各种质量指标进行各种评估或预测。软件可靠性是评价软件质量的一个重要指标,如何准确无误地预测软件可靠性是当前软件可靠性工程研究中的热点问题之一<sup>[23]</sup>。

从 20 世纪 70 年代开始,人们在软件可靠性建模、分析、评估和预测等方面进行了大量的研究,提出了近百种软件可靠性模型(Software Reliability Models, SRMs)。软件可靠性模型旨在根据软件失效数据,通过建模给出软件的可靠性估计值或预测值。它不仅是软件可靠性分配、预测与评价的最强有力工具,而且为改善软件质量提供指南。软件可靠性模

型一般可分为分析模型(Analytical Model)和数据驱动模型(Data-driven Model)两类<sup>[3]</sup>。分析模型即为传统的软件可靠性增长模型(Software Reliability Growth Model, SRGM)<sup>[1]</sup>,主要包括 NHPP 模型、Markov 模型等。这类模型都将软件的失效行为作为一个随机系统看待,对软件内部错误、失效与排除过程的特性做出相应的假设后,再利用某种随机过程进行建模,分析软件可靠性。这些假设包括:软件失效相互独立,立即完全排错,不引入新的软件缺陷,每个软件内部错误对软件失效率的贡献相同等。一方面,尽管各种可靠性增长模型的假设条件不尽相同,但这些假设都或多或少与软件失效以及排错过程的实际情况有所不符。另一方面,这类软件可靠性增长模型都基于使用剖面进行可靠性分析,认为使用剖面只具有随机性,因此现有模型完全采用随机分析法。然而,在实际的软件测试中,测试用例的选择并不一定完全是随机的,往往带有一定的确定性,如:常用的测试方法的功能测试中,尽管有些近似用户使用剖面,但测试用例的选择具有一定规律性,它并不完全随机,同样,模块测试依据程序逻辑选

到稿日期:2013-04-16 返修日期:2013-10-28 本文受中央高校基本科研业务费专项资金(NS2012072)资助。

张 柯(1983—),男,硕士生,主要研究领域为软件测试与软件可靠性建模,E-mail:zk@cpu.edu.cn;张德平(1973—),男,博士,主要研究领域为软件测试与软件可靠性建模;汪 帅(1982—),男,硕士生,主要研究领域为软件测试与软件可靠性建模。

取测试用例,也具有确定性。另外,测试人员富于理性,有确定的测试思维轨迹,测试过程中的内在或外在事件可能会影响其思维轨迹,使得软件失效行为既具有随机性又具有混沌性<sup>[4]</sup>。因此,仅仅使用随机分析法的分析模型在实际软件可靠性评估与预测中的适用性和准确性都受到了较大的影响。

近年来数据驱动的软件可靠性预测方法受到了越来越多的重视,国内外学者已提出了一些基于数据驱动的软件可靠性模型。这类模型不需要对软件的失效和排错过程做任何假设,而是将观测到的软件失效数据视为一个时间序列,借助于机器学习技术与时间序列分析方法,对失效数据序列中蕴含的失效过程信息进行挖掘和分析,进而得到软件可靠性评估和预测结果。由于基于数据驱动的软件可靠性模型不需要对软件内部错误、失效和排错过程做任何不符合实际的假定,直接对失效数据本身建模,因此其适用范围较传统的软件可靠性增长模型更为广泛,具有很好的自适应与自学习能力,并且不存在实际应用中的模型选择问题。相对于传统的可靠性增长模型,基于数据驱动的可靠性模型的适用性和预测精度增均有显著的提高,已经得到了广泛的关注和研究,成为当前软件可靠性建模研究的热点。目前,基于数据驱动的软件可靠性模型主要包括基于时间序列分析的 ARIMA 模型<sup>[5,6]</sup>、支持向量机(Support Vector Machine, SVM)模型<sup>[7,8]</sup>、灰色理论(Grey Model, GM)模型<sup>[9]</sup>以及基于基因表达式编程算法的模型<sup>[10]</sup>等,特别是人工神经网络模型<sup>[11-15]</sup>在软件失效数据序列分析、建模以及预测等方面得到了广泛应用。由于已有的这些数据驱动模型并没有关注软件失效数据序列中的混沌现象,这类模型大多数仅对于平稳的失效数据序列具有较高的模型精度,对于非平稳的失效数据有时并不尽如人意<sup>[23]</sup>。

对于此类复杂的非线性、非平稳时间序列,如果不考虑数据的混沌现象,仅靠单一的可靠性模型将很难对其进行准确的预测。因此,对于软件失效数据序列预测问题,需要采用一种适当的数据处理方法,将蕴含多种成分的软件失效时间序列数据分解为若干个规律性较强的不同频率子时间序列分量,并针对其时频特性选择适当的数学工具建立预测模型,通过预测子序列使预测风险分散化,则可进一步提高预测精度。本文将经验模态分解(Empirical Mode Decomposition, EMD)算法和递归神经网络模型(DRNN)相结合,对软件失效时间间隔进行预报。首先,利用 EMD 方法将软件失效数据分解成若干个固有模态函数(Intrinsic mode function, IMF),再对每个 IMF 分量利用递归神经网络进行预测,然后利用各个分量的预测值重构出原始信号的预测序列,从而提高预测精度。实验表明,该方法较几类常见的预测模型有更好的预测效果。

## 2 基于 EMD 分解和混沌分析的神经网络结构

### 2.1 EMD 分解算法

EMD 分解算法是一种基于信号局部特征的信号分解方法。该方法吸取了小波变换多分辨的优势,同时克服了小波变换中需选取小波基与确定分解尺度的困难,因此更适于非线性、非平稳信号分析,是一种自适应的信号分解方法,已被成功应用于诸多研究领域<sup>[16]</sup>。经验模态分解作为一种新的时频分析方法,从本质上讲是对一个信号进行平稳化处理,其

结果是将信号中不同尺度的波动或趋势逐级分解开来,产生一系列具有不同特征尺度的数据序列,每一个序列代表一个 IMF。分解出的各个 IMF 分量突出了数据的局部特征,对其进行分析可以更准确有效地把握原始数据的特征信息。本征模态函数必须满足两个条件:①它的零点数与极点数相等或至多相差 1;②由它的极大值和极小值确定的上下包络线关于时间轴局部对称。EMD 方法可以将多分量信号的各阶 IMF 一一筛选出来,具体步骤如下:

设原始软件失效数据序列为  $x(t)$ ,

1) 确定  $x(t)$  的所有极大值和极小值。

2) 对所有局部极大值点和局部极小值点分别用三次样条函数拟合出数据序列的上、下包络线  $u_0(t)$  和  $v_0(t)$ 。

3) 记上、下包络线的平均包络线为  $m_0(t)$ :

$$m_0(t) = \frac{u_0(t) + v_0(t)}{2} \quad (1)$$

记原数据序列  $x(t)$  与  $m_0(t)$  之差为:  $h_0(t) = x(t) - m_0(t)$ 。

4) 判断  $h_0(t)$  是否满足 IMF 的两条性质。若满足,则  $h_0(t)$  为 IMF; 否则,记  $h_0(t)$  为  $x(t)$ , 重复步骤 1)~3), 直到得到一个 IMF, 记为  $c_1(t)$ 。

记  $r_1(t) = x(t) - c_1(t)$  为新的数据序列, 重复步骤 1)~4), 得到第 2 个 IMF, 记为  $c_2(t)$ , 这样一直重复下去直到余项  $r_n(t)$  是一个单调数据序列或  $r_n(t)$  的值小于预先给定的阈值, 分解结束, 得到原始数据序列  $x(t)$  分解形式如下:

$$x(t) = \sum_{i=1}^n c_i(t) + r_n(t) \quad (2)$$

非平稳的失效数据序列经过 EMD 分解以后, 各 IMF 分量都基本趋于平稳, 这对预测是有利的; 而且每一 IMF 分量都是对软件失效样本数据特征的一种真实反映, 通过对失效数据序列的各个特征分别进行预测处理, 然后再对各预测结果分量进行重构, 就可以得到更加理想的预测效果。

### 2.2 混沌序列的递归神经网络结构算法

在混沌时间序列中, 系统中的每个分量的变化都是由与其相互作用的其他分量所决定的。因此, 每个分量的演化过程都包含着整个系统的信息, 系统中的单分量实测序列  $X(t)$  是一维的, 系统的全部动力学本质特征都隐藏在这一时间序列中。将系统的动力学本质特征从这一维时序中提取出来是求解特征值和特征向量的重要内容。为了能够从时间序列中得到系统的动力学相空间的几何结构, Packard 等人采用时间延滞技术, 把一维时间空间序列嵌入到  $m$  维空间中, 则

$$X(t) = \{x(t), x(t-T), \dots, x(t-(m-1)T)\} \quad (3)$$

其中,  $X(t)$  表示  $t$  时刻系统的动力学状态,  $T$  为延迟时间,  $m$  为嵌入空间矩阵的维数。由此可以建立相空间到嵌入空间的映射, 则在相空间中必然存在函数  $f(\cdot)$  使得延时后的状态  $\hat{X}(t+T)$  和当前状态  $X(t)$  之间满足  $\hat{X}(t+T) = f(X(t))$ , 其中  $f(\cdot)$  是待确定的预测函数, 延迟时间  $T$  和嵌入维数  $m$  可以通过不同的方法获得。

对上述模型, 可用神经网络进行模拟。首先利用混沌相空间重构理论, 获得有关神经网络结构的先验知识, 以便合理构建神经网络模型。当进行混沌时间序列预测时, 若时间序列为一个有限离散序列, 可使用内插法对序列进行扩充, 而后再进行混沌参数分析。设  $\eta=1, 2, \dots$  在  $R^m$  空间中存在光滑

函数,满足 $\hat{x}(t'+\eta T)=f'(X(t'))$ ,式中, $\eta T$ 为预测步数<sup>[17]</sup>,本文中取 $\eta T=1$ 进行单步预测。最佳延迟和嵌入维数可使用互信息量法和 G-P 算法<sup>[22]</sup>得到,直接单步预测最大的特点是预测模型的输入数据均为样本观测值,得到的单步模型可由下式表示:

$$\hat{x}(t+1)=f(x(t), x(t-T), \dots, x(t-(m-1)T)) \quad (4)$$

式中, $m$ 为嵌入维数, $\{x(i)|x \leq t+1\}$ 为样本测量值。

### 3 基于 EMD 的递归神经网络单步预测模型

根据给定的软件失效数据,对其进行混沌特征分析<sup>[4]</sup>,辨识数据是否具有混沌特性。针对具有混沌特性的软件失效数据将 EMD 分解法和神经网络结合起来,建立一个新的神经网络集成结构,以下记为 EMD&DRNN。首先对软件失效数据进行 EMD 分解,得到包含各个时间尺度特征的基本模式分量 IMFs;再通过一个分离的神经网络对每一个 IMF 进行预测;然后用前一阶段的每一个 IMF 的预测结果作为输入,对原始序列的下一个样本进行预测。整个预测过程分为 3 个阶段(如图 1 所示),第一阶段中,包括 EMD 分解单元和插值扩充单元,  $\ln X(t)$  和  $\ln imf$  为插值后数据;第二阶段中,  $EDNN_n$  表示基于 EMD 的递归神经网络;第三阶段中,把各 EDNN 的预测数据合成为最终预测结果,  $q$  为原始数据经 EMD 算法分解后得到的 IMF 分量的个数。

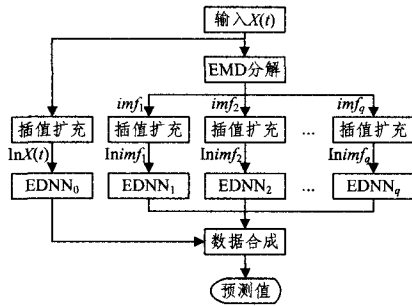


图 1 基于 EMD 的递归神经网络模型结构

给定软件失效数据预测算法的具体实现如下:

1) 利用 EMD 算法分解软件失效时间序列。在第一步中,利用 EMD 算法将软件失效时间序列  $\{X(t)|t=1,2,\dots,n\}$  进行分解,得到有限个基本模式分量  $\{imf_i(t)|t=1,2,\dots,n; i=1,2,\dots,q\}$ 。

2) 利用相空间重构理论对神经网络重要参数进行预测。若对序列进行内插,则称为插值倍数,失效时间序列经 EMD 分解后,在进入神经网络之前,使用内插算法对原始序列和每个基本模式分量进行扩展。因此可得到序列  $X'(t')$  和  $imf'_i(t')$ :

$$\begin{aligned} X'(t') &= \{x'(t')|t'=1,2,\dots,n'\} \\ \{imf'_i(t')|t'=1,2,\dots,n'; i=1,2,\dots,q\} \end{aligned} \quad (5)$$

此时式(1)用公式表示为:

$$\hat{x}(t'+(\lambda+1)p)=f(x'(t'), x'(t'-T), \dots, x'(t'-(M-1)T))$$

在神经网络中,关于时间延迟,由  $S(m, N, r, T)$ <sup>[18]</sup>为基础构造统计量来确定最佳时间延迟  $T'$ ,在  $S$  和  $T$  的关系图上第一个极小值对应最佳延迟  $T'$ ;嵌入维数的选择由  $\ln(C_m(t))$  与  $\ln r$  关系求得<sup>[19]</sup>。在选取相空间重构中的最佳延迟  $T'$

为神经网络中的  $T$  后,网络输入层节点数  $M=m$ 。因此,神经网络模型可表示为<sup>[20]</sup>:

$$\begin{aligned} S(t'+(\lambda+1)) &= \Omega(U^{[2 \rightarrow 2]} s(t') + U^{[1 \rightarrow 2]} x'(t') + a^{[2]}) \\ \hat{x}(t'+\lambda+1) &= U^{[2 \rightarrow 3]} s(t'+(1+\lambda)) + a^{[3]} \end{aligned} \quad (6)$$

其中, $\Omega(\cdot)$ 是一个  $N$  维的向量集合,包括所有隐藏层神经元的激活函数,其中神经网络学习算法选用正向传播算法来实现。

3) 对不同的分量预测完后,得到预测结果  $\hat{x}_k(t+1)$ ,  $k=1,2,\dots,q+1$ 。然后通过一个单层的感知器进行线性组合来预测原始失效时间序列的下一个样本点  $\hat{x}(t+1)$ ,即

$$\hat{x}(t+1) = \sum_{j=1}^q \omega_j^3 \hat{x}_k(t+1) \quad (7)$$

该层网络没有隐层,输入层数目等于第 2) 步中的多层神经网络的个数,即  $imf$  的个数  $q$ ;其输出单元是线性的,对于权值向量  $\omega_j^3$  ( $j=1,2,\dots,q$ ) 的计算,为了计算最小范数解,它的训练可以通过文献<sup>[21]</sup>中所述算法来完成。

### 4 实例分析与讨论

实验中采用 4 种性能评价指标进行比较:

(1) 均值误差平方和 (mean square error, MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - y_i')^2 \quad (8)$$

(2) 回归曲线方程的相关指数 (R-Square 或 R 值)

$$R\text{-Square} = 1 - \frac{\sum_{i=1}^n (y_i - y_i')^2}{\sum_{i=1}^n (y_i - y_{ave})^2} \quad (9)$$

(3) 均值误差 (average error, AE)

$$AE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - y_i'}{y_i} \right| \times 100 \quad (10)$$

(4) 均方百分比误差 (MSPE)

$$MSPE = \frac{1}{n} \sqrt{\sum_{i=1}^n \left( \frac{y_i - y_i'}{y_i} \right)^2} \quad (11)$$

以上各式中,  $y_i$  表示数据的实际值,  $y_i'$  表示数据的预测值,  $y_{ave}$  表示观测数据  $y_i$  的均值。显然, AE 值、MSE 值和 MSPE 值越小, R-Square 值越接近 1, 则表明预测值与实际值越接近, 模型拟合或预测性能越好。

实例 1 本文选取的实验数据是 Musa 的数据集, 如图 2 所示, 它们是由 IBM 收集的正交缺陷分类项目过程中的软件失效数据。

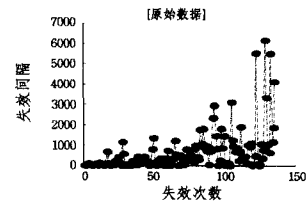


图 2 Musa 数据集

所选取数据的样本数量为 136 个, 利用 EMD 神经网络算法对数据进行建模和预测。因为对软件失效的混沌时序只能做短期预测, 所以把前 130 个数据作为训练集, 后面 6 个作为观察值来检验模型的预测效果。

首先, 对 Musa 数据集进行混沌性判断, 先在低维空间重构吸引子, 并计算吸引子的相关维数, 将嵌入空间的维数加

1,再在该新空间重构吸引子,并重新计算吸引子的相关维数。反复进行该过程,直至相关维数到达某个极限值。

根据 Takens 定理,嵌入空间与吸引子维数关系为  $D \geq 2+1$ ,Musa 数据集吸引子大约为 1.49 维,所以 Musa 数据集嵌入空间维数取 4。图 3 表明 Musa 数据所表示的系统随着空间维数的增加,相关维数趋于平稳,无明显增加,即表明该数据具有混沌性<sup>[4]</sup>。

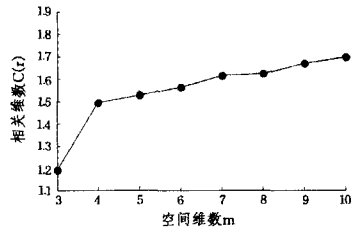


图 3 Musa 数据集嵌入空间维数与相关维数

Musa 数据集经 EMD 分解后,产生 8 个基本模式分量以及余项,如图 4 所示。

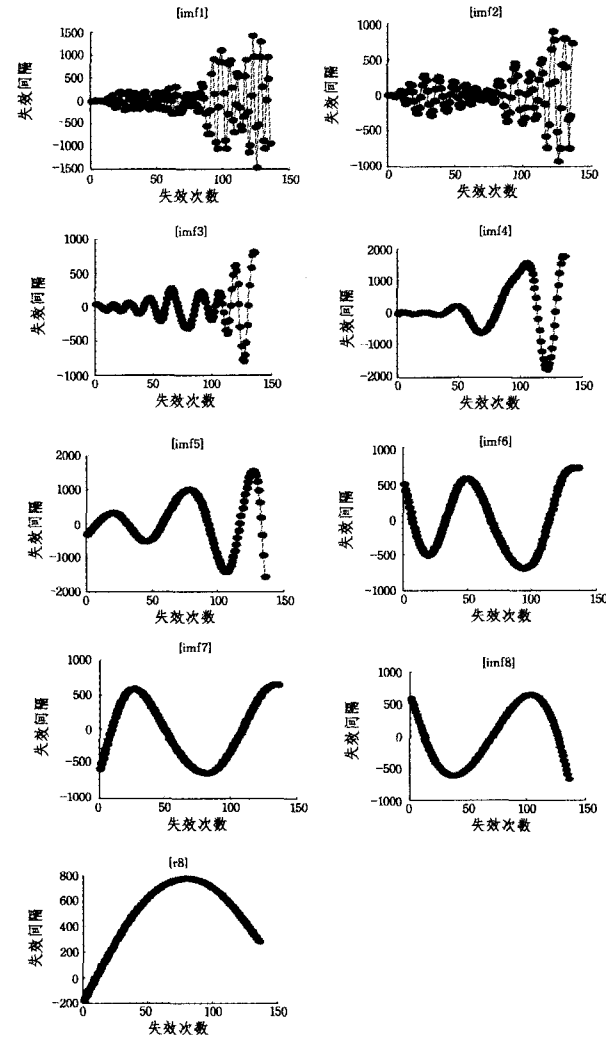


图 4 EMD 分解结果

根据原始数据以及各个分解后的数据设计神经网络的拓扑结构,需要计算原始数据和这 8 个 IMF 的最佳延迟时间与嵌入维数。由于软件失效数据是有限的离散数据,可对其进行不同数量的插值处理,软件可靠性增长行为本身已自然形成离散值,它是行为的较好体现,所以可取延迟时间为 1。然

后利用 G-P 算法<sup>[22]</sup>,经计算可得到空间维数与相关维数的关系图,各序列的嵌入维数为:4,4,5,6,6,4,4,5,4。在本文的 EMD 神经网络中输入节点个数等于序列的嵌入维数,隐藏层节点数的确定可以采用试凑法,根据 Kolmogorov 定理<sup>[21]</sup>,分别得到 9 个网络的结构为:4-9-1,4-9-1,5-11-1,6-13-1,6-13-1,4-9-1,4-9-1,5-11-1,4-9-1。

表 1 给出了分别利用 ARIMA、EMD& DRNN、DRNN 与 SVR 方法对 Musa 数据进行预测的结果及相应评价指标值。

表 1 Musa 数据集上各模型的预测结果

| 数据集      |      | 模型      |       |           |           |
|----------|------|---------|-------|-----------|-----------|
| 失效间隔     | 失效次数 | ARIMA   | SVR   | DRNN      | EMD&DRNN  |
| 1045     | 131  | 1189.37 | 1387  | 1231.85   | 1161.42   |
| 648      | 132  | 804.23  | 785   | 884.65    | 704.56    |
| 5485     | 133  | 5312.82 | 5290  | 5274.97   | 5288.22   |
| 1160     | 134  | 1104.6  | 972   | 1398.49   | 1286.19   |
| 1864     | 135  | 1804.61 | 1945  | 2228.35   | 1758.43   |
| 4116     | 136  | 4210.53 | 3667  | 4397.98   | 4264.02   |
| R-Square |      | 0.9816  | 0.976 | 0.97894   | 0.994557  |
| AE       |      | 11.43   | 13.69 | 17.5311   | 7.26583   |
| MSPE     |      | 0.05892 | 0.064 | 0.0836424 | 0.0323402 |
| MSE      |      | 5421.9  | 67166 | 67361.6   | 17409     |

EMD 算法能够完成对原始数据的特征提取。由于神经网络可以无限接近训练结果,使得 EMD&DRNN 算法能够在绝大多数数据集上获得不错的拟合和预测结果。由表 1 可得,在本组失效数据集上,EMD&DRNN 模型的预测结果均优于对比模型:AE 值、MSPE 值和 MSE 值最小,R-Square 值最接近于 1。ARIMA 模型次之。

实例 2 图 5 中的数据是 NTDS(the naval tactical data system,海军战术数据系统)数据集,记录了软件的失效次数以及失效间隔。

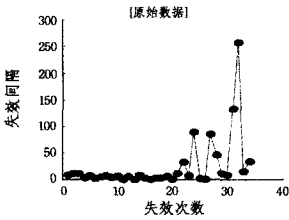


图 5 NTDS 数据集

图 6 表明 NTDS 数据在空间维数逐渐增大的情况下,其相关维数无明显趋于平稳,不存在极限维数,即不具有混沌性<sup>[4]</sup>。

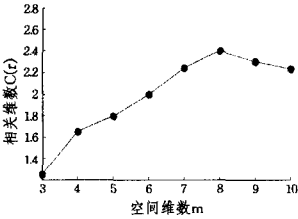


图 6 NTDS 数据集嵌入空间维数与相关维数

利用经验模态分解方法对上述数据集进行处理,得到的分解结果如图 7 所示。

由图 7 可知,通过 EMD 将原始数据分为 6 个本征模式分量和 1 个剩余分量,相较于原始数据,每个本征模式分量明显趋于平稳,频率依次降低,且剩余分量也反映了软件失效数据序列的变化趋势。

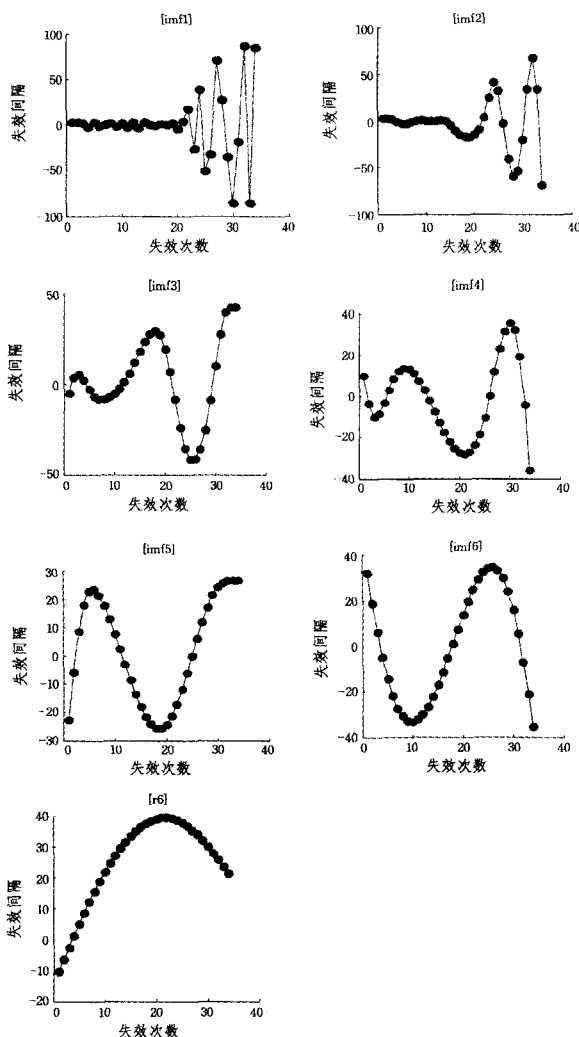


图7 NTDS数据集的经验模态分解结果

接着设计神经网络的拓扑结构,计算原始数据以及这6个IMF的最佳延迟时间与嵌入维数,形成最终的预测结果。

由表2可以看出,本文提出的EMD&DRNN模型的预测效果好于单一的DRNN模型、SVR模型以及ARIMA模型,但由于数据训练集较短,且不具有混沌性,其预测参数比实例1中的Musa数据集的预测精度稍差。

表2 NTDS数据集上各模型的预测结果

| 数据集      |      | 模型     |              |          |                 |
|----------|------|--------|--------------|----------|-----------------|
| 失效间隔     | 失效次数 | ARIMA  | SVR          | DRNN     | EMD&DRNN        |
| 47       | 28   | 49.34  | 52.1         | 66.74    | 48.84           |
| 12       | 29   | 19.52  | 18.0         | 24.53    | 23.87           |
| 9        | 30   | 17.69  | 15.0         | 19.04    | 6.94            |
| 135      | 31   | 118.42 | 129          | 103.12   | 122.3           |
| 258      | 32   | 278.33 | 210.9        | 303.11   | 225.45          |
| 16       | 33   | 27.12  | 22.0         | 42.55    | 11.03           |
| 35       | 34   | 29.63  | 29.0         | 56.72    | 29.71           |
| R-Square |      | 0.9731 | 0.953        | 0.839836 | <b>0.988236</b> |
| AE       |      | 28.44  | 29.266       | 76.7055  | <b>26.654</b>   |
| MSPE     |      | 0.1689 | <b>0.136</b> | 0.343463 | 0.154083        |
| MSE      |      | 137.45 | 346.346      | 1911.4   | <b>74.2862</b>  |

**结束语** 本文应用EMD算法和混沌分析解决了小波变换中需选取小波基与确定分解尺度的困难,它是一种自适应的信号分解方法,可以对非线性、非稳定的失效数据序列进行逐层分解,完成对原始数据的特征提取。然后针对软件可靠性建模,提出了一种动态递归神经网络结构,从而提高了单步预测模型的精度,并应用该方法对软件失效数据进行了预测。

实验结果表明:该方法在预测时效果较好,与一些常用预测算法相比效果有一定改善。

由于本模型使用神经网络对数据进行训练,因此数据的训练集应尽可能大,以使神经网络得到充分训练,这样才能达到更好的预测效果。另外,本文算法是进行单步预测,每预测一步后,再将其加入训练集来预测下一个点,导致随着预测点数的增加,在后面点数的预测中误差会越来越大,所以此模型更适合于进行短期预测。

## 参考文献

- [1] Lyu M R. Handbook of software reliability engineering [M]. New York: McGraw Hill, 1996: 56-60
- [2] 赵亮,王建民,孙家广. 统计测试的软件可靠性保障能力研究[J]. 软件学报, 2008, 19(6): 1379-1385
- [3] Yang B, Li X, Xie M, et al. A generic data-driven software reliability model with model mining technique[J]. Reliability Engineering and System Safety, 2010, 95: 671-678
- [4] 邹丰忠,李传湘. 软件可靠性混沌模型[J]. 计算机学报, 2001, 24(3): 281-291
- [5] Raja U, Hale D P, Hale J E. Modeling software evolution defects; a time series approach[J]. Journal of Software Maintenance and Evolution-research and Practice, 2009, 21: 49-71
- [6] 贾治宇,康锐. 软件可靠性预测的ARIMA方法研究[J]. 计算机工程和应用, 2008, 44(35): 17-19
- [7] Moura M C, Zio E, Lins I D, et al. Failure and reliability prediction by support vector machines regression of time series data[J]. Reliability Engineering and System Safety, 2011, 96: 1527-1534
- [8] Lo J H. A study of applying ARIMA and SVM model to software reliability prediction[C]// 2011 International Conference on Uncertainty Reasoning and Knowledge Engineering. 2011: 141-144
- [9] 李海峰,陆民燕,王智新. 基于灰色系统理论的软件可靠性综合评价框架[J]. 北京航空航天大学学报, 2008, 34(11): 1261-1265
- [10] 李海峰,陆民燕,曾敏,等. 基因表达式编程在软件可靠性建模中的应用[J]. 计算机科学与探索, 2011, 5(6): 534-546
- [11] Su Y S, Huang C Y. Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models[J]. The Journal of Systems and Software, 2007, 80: 606-615
- [12] Jin C, Jin S W, Ye J M, et al. Software reliability prediction based on discrete wavelet transform and neural network[C]// International Conference on Computational Intelligence and Software Engineering. 2009: 1-4
- [13] Kiran N R, Ravi V. Software reliability prediction using wavelet neural networks[C]// International Conference on Computational Intelligence and Multimedia Application. 2007: 195-199
- [14] Hu Q P, Xie M, Ng S H. Software reliability predictions using artificial neural networks[J]. Computational Intelligence in Reliability Engineering, 2007, 40: 197-222
- [15] Lo J H. The implementation of artificial neural networks applying to software reliability modeling[C]// IEEE Control and Decision Conference, 2009. CCDC' 09. China, 2009: 4349-4354
- [16] 玄兆燕,杨公训. 经验模态分解法在大气时间序列预测中的应用[J]. 自动化学报, 2008, 34(1): 97-101
- [17] Han M, Xi J H, Xu S G, et al. Prediction of chaotic time series based on the recurrent network[J]. IEEE Trans on Signal Pro-

cessing, 2004, 52(12): 3409-3416

[18] 张旭海, 贺国光, 卢宇. 一种在线实时快速地判定交通流混沌的组合算法[J]. 系统工程, 2005, 23(9): 42-45

[19] 顾圣士, 王志谦, 程极泰. 太阳黑子数时间序列的分形研究及预测[J]. 应用数学与力学, 1999, 20(1): 81-86

[20] Weigend A S, Huberman B A, Rumelhart D E. Predicting the future: A connectionist approach[J]. International Journal of Neu-

ral System, 1990, 1(3): 1993-209

[21] 伍春香, 刘琳, 王葆元. 三层 BP 网隐层节点数确定方法的研究[J]. 计算机学报, 1998(6): 2-5

[22] 于青. 关联维数计算的分析研究[J]. 计算机学报, 2004(12): 1-2

[23] 张德平, 汪帅, 周昊杰. 基于 EMD 和 GEP 的软件可靠性预测模型[J]. 计算机科学, 2013, 40(4): 164-168

(上接第 154 页)

根据新鲜性定义, 攻击者不允许进行 MasterReveal 询问, 不允许同时揭示测试会话的长期私钥和短期私钥, 攻击者询问 SessionReveal(sid), 如果攻击者能以不可忽略的优势赢得安全性游戏, 那么模拟者就成功解决了判定性  $q$ -parallelBDHE 假设。

4.4 效率分析

本文提出的 ABAKA 协议中整个算法仅有 3 个双线性运算, 具有较高的计算效率。 $n$  表示用户属性集个数, 本文 ABAKA 与其他密钥协商协议的比较如表 2 所列。

表 2 各密钥协商协议的分析与比较

| 协议         | 长期私钥空间      | 安全模型  | 双线性运算个数 |
|------------|-------------|-------|---------|
| $Y^{[11]}$ | $(2n+1) G $ | ABeCK | $2n+1$  |
| $Y^{[13]}$ | $(n+2) G $  | ABCK  | $2n+2$  |
| 本协议        | $(n+2) G $  | ABeCK | 3       |

对本协议与  $Y^{[11]}$  密钥协商阶段的计算进行比较, 仿真如图 1 所示。

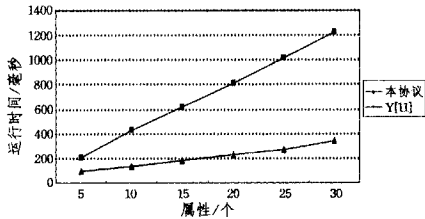


图 1 本协议与  $Y^{[11]}$  密钥协商阶段的计算效率比较

硬件: Intel T6600@2.2GHz, 2GBytes RAM, 平台: Windows, 代码库: Miracl。

文献[8]指出, 群  $G, G_T$  上的一个乘法运算效率是 0.0034 毫秒, 一个双线性运算效率是 8.22 毫秒, 双线性运算效率明显较低。表 2 表明: 方案  $Y^{[11]}$  和  $Y^{[13]}$  中, 双线性运算个数与属性个数相关, 效率为  $O(n)$ , 本文提出的 ABAKA 中整个算法只有 3 个双线性运算, 效率为  $O(1)$ , 且用户私钥空间开销较小。图 1 的仿真结果表明: 本文提出的 ABAKA 由于双线性运算个数只有 3 个, 不再与用户属性个数线性相关, 因此具有更高的运算效率。

结束语 本文分析了多种基于属性的加密方案, 提出了一种新的高效且具有灵活访问结构的密文策略的属性加密方案, 并在该加密方案的基础上提出了一个高效的基于属性的密钥协商协议。该协议采用秘密共享方案, 能够表示任意访问结构, 且解密效率较高, 适用于计算和存储资源有限的网络系统。但该协议占用较大的密文空间, 如何减少密文的长度是今后需要进一步研究的工作。

参考文献

[1] Diffie W, Hellman M. New directions in cryptography [J]. IEEE Transactions on Information Theory, 1976, 22 (6): 644-654

[2] Sahai A, Waters B. Fuzzy identity-based encryption [M]// Advances in Cryptology-EUROCRYPT2005. Springer Berlin Heidelberg, 2005: 457-473

[3] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[C]//Proceedings-IEEE Symposium on Security and Privacy. Berkeley, CA, United states, 2007: 321-334

[4] Cheung L, Newport C. Provably secure ciphertext policy ABE [C]// Proceedings of the ACM Conference on Computer and Communications Security. Alexandria, VA, United states, 2007: 456-465

[5] Waters B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization[C]//14th International Conference on Practice and Theory in Public Key Cryptography, PKC 2011. Taormina, Italy, 2011: 53-70

[6] Ge Ai-jun, Zhang Rui, Chen Cheng, et al. Threshold Ciphertext Policy Attribute-Based Encryption with Constant Size Ciphertexts[C]//17th Australasian Conference on Information Security and Privacy, ACI SP2012. Wollongong, NSW, Australia, 2012: 7372: 336-349

[7] Attrapadung N, Herranz J, Laguillaumie F, et al. Attribute-based encryption schemes with constant size ciphertexts [J]. Theoretical Computer Science, 2012, 422: 15-38

[8] Hohenberger S, Waters B. Attribute-Based Encryption with Fast Decryption[M]//Public-Key Cryptography PKC2013. Springer Berlin Heidelberg, 2013: 162-179

[9] Wang Hao, Xu Qiu-liang, Ban Tao. A provably secure two-party attribute-based key agreement protocol[C]//Intelligent Information Hiding and Multimedia Signal Processing, 2009. IHH-MSP' 09. Fifth International Conference on. IEEE, 2009: 1042-1045

[10] Wang Hao, Xu Qiu-Liang, Fu Xiu. Two-party attribute-based key agreement protocol in the standard model[C]//Proceedings of the 2009 International Symposium on Information Processing (ISIP 2009). 2009: 325-328

[11] Yoneyama K. Strongly secure two-pass attribute-based authenticated key exchange[C]//4th International Conference on Pairing-Based Cryptography, Pairing 2010. Kaga, Japan, 2010, 6487: 147-166

[12] LaMacchia B, Lauter K, Mityagin A. Stronger security of authenticated key exchange[C]//1st International Conference on Provable Security 2007. Wollongong, NSW, Australia, 2007, 4784: 1-16

[13] Yoneyama K. Two-party round-optimal session-policy attribute-based authenticated key exchange without random oracles[C]//14th International Conference on Information Security and Cryptology, ICISC 2011. Seoul, Korea, 2012: 7259: 467-489

[14] 魏江宏, 胡学先, 刘文芬. 多属性机构环境下的属性基认证密钥交换协议[J]. 电子与信息学报, 2012, 34(2): 451-456

[15] Shamir A. How to share a secret [J]. Communications of the ACM, 1979, 22(11): 612-613