第 38 卷 第 12 期
2016 年 12 月

系统工程与电子技术
Systems Engineering and Electronics

Vol. 38 No. 12
December 2016

文章编号:1001-506X(2016)12-2869-07

网址:www.sys-ele.com

# 基于 APDE-RBF 神经网络的网络安全态势预测方法

李方伟,张新跃,朱 江,黄 卿

(重庆邮电大学移动通信技术重庆市重点实验室,重庆 400065)

**摘 要**:为了提高径向基函数(radical basis function, RBF)神经网络对网络安全态势的预测精度,提出了一种基于吸引力传播(affinity propagation, AP)聚类和差分进化(differential evolution, DE)优化 RBF 神经网络的算法。首先,利用 AP 聚类算法对样本数据进行划分聚类,从而获得 RBF 的中心和网络的隐含层节点数;其次,利用 AP 聚类得出种群差异度,自适应地改变 DE 算法的缩放因子和交叉概率,对 RBF 的宽度和连接权值进行优化;同时为了避免陷入局部最优以及跳出局部极值点,对每一代种群的精英个体和种群差异度中心进行混沌搜索。通过仿真实验表明,此算法在泛化能力增强的同时,对网络安全态势也达到了较高的预测精度。

**关键词**:径向基函数;吸引力传播聚类;差分进化;种群差异度;混沌搜索

**中图分类号**:TP 393 **文献标志码**:A **DOI**:10.3969/j.issn.1001-506X.2016.12.28

# Network security situation prediction based on APDE-RBF neural network

LI Fang-wei, ZHANG Xin-yue, ZHU Jiang, HUANG Qing

(*Chongqing University of Posts and Telecommunications, Chongqing Key Lab of Mobile Communications Technology, Chongqing 400065, China*)

**Abstract**: In order to improve the prediction accuracy of network security situation based on radical basis function (RBF) neural network, an optimization algorithm of RBF neural network based on affinity propagation (AP) clustering and differential evolution (DE) is proposed. Firstly, the AP clustering is used to optimize the center and the number of the hidden layer. Secondly, AP clustering is used to get the population diversity (PD), the scaling factor and the crossover probability of DE are adaptively changed with the PD for the optimized width and connection weights of RBF neural network. In order to avoid falling into the local optimum and jump out of the local extreme point, the elite individual and PD' centers of each generation population are searched by chaotic search. The simulation results show that the APDE-RBF algorithm can enhance the generalization ability, and it also has high prediction accuracy for the network security situation.

**Keywords**: radical basis function (RBF); affinity propagation (AP) clustering; differential evolution (DE); population diversity (PD); chaotic search

## 0 引 言

随着各种网络安全问题层出不穷,网络安全问题越来越受到重视。网络安全态势预测是目前网络安全领域的一个研究热点,不同于以往的入侵检测和防火墙等被动防御手段,网络安全态势预测是主动防御机制[1]。网络安全态势预测主要是为了在网络受到攻击损失前网络管理员采取相对应的措施,根据当前和以往的网络安全态势值,建立合理的数学模型对未来一段时间的网络安全状态进行预测。由于网络攻击是随机和不确定的,所以对态势值的预测是一个复杂的非线性过程[2]。

目前研究人员提出了很多预测的方法,如自回归滑动平均(auto-regressive moving average, ARMA)模型、灰色预测模型(gray model, GM)、马尔可夫链(Markov)、支持向量机(support vector machine, SVM)等[3-6]。但是大量研究发现,上述方法都存在各自的缺点:ARMA 模型会假设时间序列是平稳线性的,但是网络安全态势时间序列是非线性的;GM 模型适用单调变化的时间序列,对于波动较大的时间序列难以预测;Markov 模型需要大量复杂的数学公式推导,难以建立准确的预测模型;SVM 对大规模训练样本难以实施,收敛速度慢。

为更加精确地预测网络安全态势,本文采用吸引力传

**收稿日期**:2015-01-15; **修回日期**:2016-07-28; **网络优先出版日期**:2016-08-31。

**网络优先出版地址**:http://www.cnki.net/kcms/detail/11.2422.TN.20160831.1252.004.html

**基金项目**:国家自然科学基金项目(61271260,61301122);重庆市科委自然科学基金项目(cstc2015jcyjA40050)资助课题

播(affinity propagation, AP)聚类和改进的差分进化(differential evolution, DE)对径向基函数(radical basis function, RBF)神经网络的结构和参数进行优化。为了加强泛化能力,利用 AP 聚类可以依据样本数据自适应确定 RBF 网络的结构;为了避免陷入局部最优解,根据种群差异度动态调整 DE 算法中的缩放因子和交叉概率,并对每代的精英个体和差异度中心进行混沌搜索。利用该方法对网络安全态势预测仿真,并进行了误差分析,结果表明本文方法有更好的泛化能力和预测准确性。

## 1 RBF 神经网络

RBF 神经网络是一种 3 层前向型网络,由第 1 层的输入层、第 2 层的隐含层和第 3 层的输出层组成,网络结构如图 1 所示。

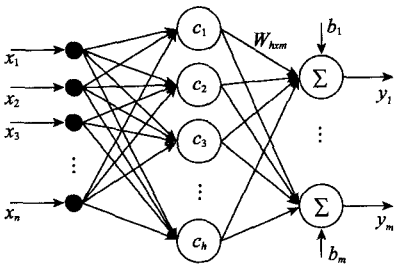



图 1 $n$-$h$-$m$ 结构的 RBF 神经网络

Fig. 1 RBF neural network of $n$-$h$-$m$ structure

由于输入层到隐含层是非线性变换,隐含层到输出层是线性变换,所以 RBF 神经网络本质上是通过非线性基函数的线性组合来实现输入 $R^n$ 到输出 $R^m$ 的映射关系。

RBF 神经网络结构如图 1 所示,具有 $n$ 个输入节点,$h$ 个隐含层节点和 $m$ 个输出节点。对于一个 $n$ 维的网络输入矢量 $\boldsymbol{X}=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}\in R^n$,第 $i$ 个隐含层节点的输出为

$$\varphi_i=\exp\left(-\frac{\|\boldsymbol{X}-C_i\|^2}{\sigma_i^2}\right),\ i=1,2,\cdots,h \tag{1}$$

式中,$C_i$ 是隐含层节点中心;$\|\cdot\|^2$ 是 2 范数,表示 $\boldsymbol{X}$ 与 $C_i$ 的欧氏距离;$\sigma_i$ 是第 $i$ 个基函数的宽度;$h$ 是隐含层节点个数。

RBF 网络输出层中第 $j$ 个神经元的输出为

$$y_j=\sum_{i=1}^{h}w_{ji}\varphi_i(\boldsymbol{X})+b_j,\ j=1,2,\cdots,m \tag{2}$$

式中,$w_{ji}$ 是隐含层第 $i$ 个神经元与输出层第 $j$ 个神经元的连接权值;$m$ 是输出神经元个数;$b_j$ 是输出层第 $j$ 个神经元的阈值。

## 2 基于 APDE 优化的 RBF 神经网络

从 RBF 网络结构可以看出,影响预测精度和泛化能力的参数主要有 4 个:隐含层节点个数 $h$;隐含层节点中心 $C_i$;基函数宽度 $\sigma_i$;隐含层到输出层的连接权值 $w_{ji}$。本文使用 AP 聚类对隐含层个数和节点中心进行优化,利用改进的 DE 算法对基函数宽度和连接权值进行自适应寻优,并用混沌搜索进行二次寻优,避免陷入局部最优解。

### 2.1 AP 聚类

对基函数中心的选取,无论是随机选取还是自组织选取都要预先设定隐含层节点个数[7],这种人为设置隐含层节点个数的方法在样本训练期间很容易导致 RBF 神经网络的过拟合或者欠拟合,从而影响网络的泛化能力和后期的预测精度,所以本文采用了 AP 聚类的方法,根据数据样本的实际情况,对隐含层节点个数和节点中心进行自适应选取。

AP 聚类是一种基于相似度矩阵的新型无监督聚类算法[8],不同于以往的 $k$-means 和模糊 C 均值聚类算法,该算法不必预先设定聚类个数,所有样本数据点都被默认为潜在的聚类中心,通过不断地竞争迭代,达到最优聚类结果。具体算法步骤如下。

**步骤 1** 计算相似度矩阵 $\boldsymbol{S}$

$$\boldsymbol{S}(i,k)=-x_i-x_k^2 \tag{3}$$

**步骤 2** 初始化吸引矩阵 $\boldsymbol{R}$,归属度矩阵 $\boldsymbol{A}$,$\boldsymbol{R}(i,k)=0$,$\boldsymbol{A}(i,k)=0$。

**步骤 3** 确定偏向参数 $p_k(k=1,\cdots,N)$

$$p_k=\underset{i\neq j,i,j=1,\cdots,N}{median}\boldsymbol{S}(i,j) \tag{4}$$

**步骤 4** 计算与更新吸引度矩阵

$$\boldsymbol{R}(i,k)=\boldsymbol{S}(i,k)-\max_{k\neq k'}\{\boldsymbol{A}(i,k')+\boldsymbol{S}(i,k')\}$$

$$\boldsymbol{R}(i,k)=\lambda\times\boldsymbol{R}(i,k)_{\mathrm{old}}+(1-\lambda)\times\boldsymbol{R}(i,k)_{\mathrm{new}} \tag{5}$$

**步骤 5** 计算与更新归属度矩阵

$$\boldsymbol{A}(i,k)=\min\left\{0,\boldsymbol{R}(k,k)+\sum_{i'\notin(i,k)}\max\{0,\boldsymbol{R}(i',k)\}\right\}$$

$$\boldsymbol{A}(i,k)=\lambda\times\boldsymbol{A}(i,k)_{\mathrm{old}}+(1-\lambda)\times\boldsymbol{A}(i,k)_{\mathrm{new}} \tag{6}$$

**步骤 6** 如果满足以下条件之一:①选择的类中心保持稳定,②超过最大迭代次数,转步骤 7;否则转步骤 4。

**步骤 7** 输出聚类结果。

### 2.2 改进的 DE 算法

DE 算法是一种基于群体差异的高效并行搜索方法[9],在收敛速度和稳定性方面已被证明优于其他进化算法[10],DE 算法的参数包括种群规模、缩放因子和交叉概率。标准 DE 算法中缩放因子和交叉概率都是固定的,缩放因子过大或者交叉概率过小,虽然有助于种群的多样性,增强全局搜索能力,但是在进化后期易导致收敛速度慢,最优解精度低等问题;反之,缩放因子过小或者交叉概率过大,易对局部过度开发,易出现早熟的现象。所以本文对 DE 算法进行改进,使得缩放因子和交叉概率可以根据当前收敛情况进行动态调整。

2.2.1 *初始化*

为了能够实现对 RBF 神经网络基函数宽度 $\sigma_i$ 和隐含层到输出层连接权值 $w_{ji}$ 的同时优化，本文采用实数编码，种群中每个个体结构如图 2 所示。

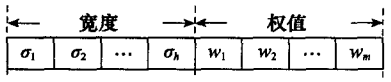



图 2　个体编码结构

Fig. 2　Individual coding structure

设初始种群 $\boldsymbol{S}=\{X_1, X_2, \cdots, X_N\}$，$N$ 是种群规模，$X_i(X_i \in R^D)$ 是种群中的个体，具体初始化过程如下：

$$\begin{cases}\sigma_i = \sigma_{\min} + \text{rand}(0,1) \times (\sigma_{\max} - \sigma_{\min}) \\ w_i = \text{rand}(0,1)\end{cases} \tag{7}$$

式中，$\sigma_{\max} = \underset{i\neq j, i,j=1,\cdots,h}{\text{argmax}}\,(abs(c_i - c_j))$，$\sigma_{\min}$ 也是同理可得；rand(0,1) 表示(0,1)间均匀分布的随机数。

2.2.2　*变异*

DE 算法差分策略来实现种群中个体的变异，通常是随机选取 3 个个体，将其中两个个体的差分缩放后与第 3 个个体进行合成，如式(8)所示：

$$V_i(g+1) = X_{r1}(g) + F \times (X_{r2}(g) - X_{r3}(g)),$$
$$i \neq r1 \neq r2 \neq r3 \tag{8}$$

式中，$X_i(g)$ 是第 $g$ 代种群中第 $i$ 个个体；$F$ 为缩放因子，标准 DE 算法中设为固定值，但是 $F$ 的设置不当会导致种群的早熟和慢收敛，为了加快收敛速度以及求得全局最优解，本文提出依据种群差异度动态调整 $F$ 的策略。

种群差异度指对种群空间中所有个体进行聚类所得到的聚类个数。当种群差异度越大时，个体在种群空间中分布越均匀，求得全局最优解可能性越大。所以在进化前期，为了尽可能使种群个体多样化，使用式(9)对 $F$ 进行动态调整：

$$F(g) = \begin{cases} F_{\max} - (F_{\max} - F_{\min})\left(\dfrac{g-1}{g_{\max}}\right) PD(g) > \\ PD(g-1) \text{ and } \dfrac{g}{g_{\max}} < \tau_1 \\ F(g-1)\ PD(g) \leqslant PD(g-1) \text{ and } \dfrac{g}{g_{\max}} < \tau_1 \\ F_{\min}, \text{其他} \end{cases} \tag{9}$$

式中，$F_{\max}$ 和 $F_{\min}$ 是缩放因子的上下界；$\tau_1$ 是设置的迭代阈值。

2.2.3　*交叉*

交叉操作也是 DE 算法的核心操作，对种群中目标个体 $X_i$ 和变异个体 $V_i$ 进行交叉，生成新的个体 $U_i$：

$$u_{ij}(g+1) = \begin{cases} v_{ij}(g+1), \text{ rand} < CR \text{ 或 } j = j_{\text{rand}} \\ x_{ij}(g), \text{其他} \end{cases} \tag{10}$$

式中，rand 是(0,1)间均匀分布的随机数；$j_{\text{rand}}$ 是[1,$n$]间的随机整数，这是为了保证新个体一定会有变异成分；$CR$ 是交叉概率，控制个体各维度参量对交叉的参与程度，因为 $CR$ 越小，收敛越快，所以为了平衡局部和全局搜索能力，使 $CR$ 可以自适应调整，在迭代前期减缓收敛速度，保持较大的种群多样性，在后期加快收敛速度，加强局部搜索能力。

$$CR(g) = \begin{cases} CR_{\min} \cdot 2^{e^{\left(1-\frac{g_{\max}}{1+g}\right)}}, \dfrac{g}{g_{\max}} < \tau_2 \\ CR_{\max}, \text{其他} \end{cases} \tag{11}$$

式中，$CR_{\min}$ 和 $CR_{\max}$ 是交叉概率的上下界；$\tau_2$ 是设置的迭代阈值。

2.2.4　*选择*

DE 算法采用一对一竞争策略实现选择。候选个体 $U_i(g+1)$ 和对应个体 $X_i(g)$ 按照适应度进行竞争，优胜者进入下一代种群，即

$$X_i(g+1) = \begin{cases} U_i(g+1), f(U_i(g+1)) \geqslant f(X_i(g)) \\ X_i(g), \text{其他} \end{cases} \tag{12}$$

式中，$f(\cdot)$ 是个体的适应度函数，本文使用均方误差(mean square error, MSE)作为适应度函数。

### 2.3　混沌搜索

混沌是一种非线性现象，具有随机性和遍历性，可在一定范围内进行不重复遍历搜索，因此可以作为进化算法跳出局部最优的一种方法[11-12]。本文采用的混沌搜索对每一代的最优个体以及种群差异度中心进行 $T$ 次搜索，如果搜索到更优个体则进行取代，从而提高 DE 算法的全局搜索能力。

本文采用一维 Logistic 映射混沌模型，即

$$\boldsymbol{Z}^{t+1} = \mu \boldsymbol{Z}^t (1 - \boldsymbol{Z}^t) \tag{13}$$

式中，$\mu$ 是控制参数，随机生成一个 $D$ 维向量 $\boldsymbol{Z}_i^0 = [z_1^0, z_2^0, \cdots, z_D^0]$，然后对初始值进行 $T$ 次混沌迭代。对种群中最优个体和差异度中心迭代过程为

$$\boldsymbol{X}_i^{t+1} = X_i + \alpha \boldsymbol{Z}^{t+1}$$
$$\alpha = \begin{cases} 1, r \geqslant 0.5 \\ -1, \text{其他} \end{cases} \tag{14}$$

式中，$X_i$ 是种群的最优个体或者差异度中心；$\boldsymbol{X}_i^{t+1}$ 是混沌搜索后的新个体；$\alpha$ 是混沌调节参数，可向待搜索个体的正反两个方向进行遍历；$r$ 是[0,1]间的随机数。

### 2.4　算法步骤

本文主要利用 AP 聚类得到 RBF 神经网络的隐含层节点数和节点中心，并用改进的 DE 算法对 RBF 的基函数宽度和连接权值进行寻优，同时为了提高全局搜索能力，使用混沌搜索对最优个体和差异度中心进行二次搜索。本文算法流程如图 3 所示。

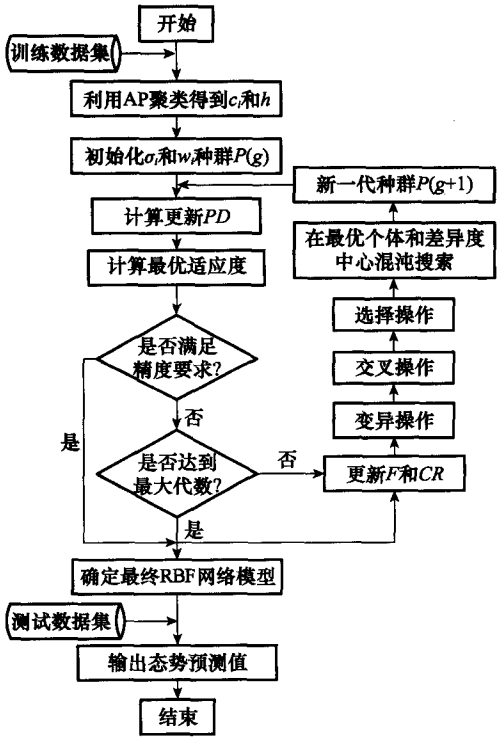



图 3 基于 APDE-RBF 算法态势预测流程图

Fig. 3 Flow chart of situation prediction based on APDE-RBF algorithm

### 2.5 算法收敛性分析

文献[13]证明了在缩放因子 $F$ 限定的条件下 DE 算法能以概率 1 收敛，文献[14]通过随机压缩应映射定理论证了 DE 是渐近收敛的。在上述研究的基础上，对 APDE 算法的收敛性进行分析。

**引理** DE 迭代形成的随机映射 $\Psi:\Omega\times S\to S$ 为随机压缩算子，根据随机压缩定理可得 $\Psi$ 具有唯一随机不动点，即 DE 是渐近收敛的。

**证明** 参考文献[14]。

从引理 1 可得 DE 算法具有渐近收敛性，APDE 算法在 DE 算法方面的改进主要体现在 $F$ 和 $CR$ 的自适应调整和混沌搜索。虽然 $F$ 和 $CR$ 是根据种群差异度动态变化，但最终是收敛的，没有影响 DE 算法的收敛性，所以混沌搜索是影响 DE 算法的唯一因素。根据 DE 算法的渐近收敛性可以证明 APDE 算法的收敛性。 证毕

**定理** DE 算法具有渐近收敛性，则在 $1<\mu<3$ 的情况下，APDE 算法也具有渐近收敛性。

**证明** 假设 DE 算法收敛的全局最优解 $X_{\text{best}}$，则对第 $t$ 次迭代的个体 $X_i(t)$ 有：

$$\lim_{t\to\infty}X_i(t)=X_i^{\text{best}},\ i\in(1,2,\cdots,n) \tag{15}$$

对 APDE 算法得出的变异解 $X_i^*$ 则有：

$$\begin{aligned}\lim_{t\to\infty}X_i^*&=\lim_{t\to\infty}(X_i+\alpha(\mu\boldsymbol{Z}_i(1-\boldsymbol{Z}_i)))=\\&\lim_{t\to\infty}X_i+\lim_{t\to\infty}\alpha\mu\boldsymbol{Z}_i(1-\boldsymbol{Z}_i)=\\&X_i^{\text{best}}+\alpha\lim_{t\to\infty}\mu\boldsymbol{Z}_i(1-\boldsymbol{Z}_i)\end{aligned} \tag{16}$$

由式(16)可知，如果 $\lim_{t\to\infty}\mu\boldsymbol{Z}_i(1-\boldsymbol{Z}_i)$ 渐近收敛，则 APDE 算法渐近收敛。根据非线性方程平衡点稳定性判定法，解下列方程

$$z=f(z)=\mu z(1-z) \tag{17}$$

便可得到式(13)的平衡点为：$z_1=0$ 和 $z_2=1-1/\mu$，不论 $\mu$ 是何值，$z_1=0$ 都是式(13)的不稳定平衡点。对于 $z_2=1-1/\mu$，由于 $f'(z_2)=\mu(1-2z_2)=2-\mu$，根据稳定性条件 $|f'(z_2)|=|2-\mu|<1$ 可得到当 $1<\mu<3$ 时，$z_2=1-1/\mu$ 才是差分方程式(13)的稳定平衡点，由此可得式(16)是渐近收敛的，即 APDE 算法具有渐近收敛性。 证毕

### 2.6 算法复杂度分析

按照算法的流程步骤分析时间复杂度，最大迭代次数是 $g_{\max}$，种群规模是 $N$，问题维度是 $D$。每一迭代计算和更新 $PD$ 和最优适应度分别为 $O(N\times D)$，更新 $F$ 和 $CR$ 为 $O(1)$，交叉和变异的操作为 $O(N\times D)$，对最优个体和差异度中心的混沌搜索为 $O(D\log D)$。综上所述，本文算法的时间复杂度为 $O(N\times D\times g_{\max})$。

**表 1 不同算法的复杂度对比**

**Table 1 Complexity comparison of different algorithms**

| 算法 | 复杂度 |
|---|---|
| ARMA | $O(D)$ |
| GM(1,1) | $O(D)$ |
| HMM | $O(T^2\times D)$ |
| SVM | $O(D\times D)$ |
| APDE-RBF | $O(N\times D\times g_{\max})$ |

注：$T$ 是 HMM 的状态数

## 3 实验仿真

为了验证 APDE-RBF 算法的有效性，采用某公司安全部门提供的数据，并用文献[15]的评估方法把原始的多源数据融合为网络安全态势值，预测后与现有的算法进行对比。

### 3.1 数据预处理

因为网络攻击的随机性和连续性，预测时间跨度太小，会使网络学习程度不够，如果时间跨度太大，则会影响网络对近期安全态势的学习，所以本文使用前 80 组数据作为训练集，后 21 组数据作为测试集，并采用滑动窗口方式(窗口为 6，每次滑动为 1 个单元)对 80 组训练数据进行重构，重构结果如表 2 所示。

**表 2 训练数据重构结果**

**Table 2 Reconstruction results of training data**

| 五维输入向量 | 一维输入向量 |
|---|---|
| $x_1,x_2,x_3,x_4,x_5$ | $x_6$ |
| $x_2,x_3,x_4,x_5,x_6$ | $x_7$ |
| ⋮ | ⋮ |
| $x_{75},x_{76},x_{77},x_{78},x_{79}$ | $x_{80}$ |

为了提高收敛效率，加快网络学习速度，本文按照式(18)对网络安全态势值进行归一化处理。

$$\hat{x}=\frac{x-x_{\min}}{x_{\max}-x_{\min}} \tag{18}$$

### 3.2 网络的训练

根据第 3.1 节的数据预处理，对 RBF 神经网络进行和

APDE算法进行初始化设置：

(1) RBF 神经网络的输入向量维度 $n=5$，输出向量维度 $m=1$；通过 AP 聚类的结果得出隐含层节点数为 $h=10$。

(2) APDE 算法的种群设置为 $N=80$，缩放因子的初始值 $F_0=0.9$，交叉概率的初始值 $CR_0=0.3$，最大迭代次数为 $g_{max}=500$，误差精度设置为 $1.2\times10^{-3}$。

为了验证 APDE 算法在收敛速度和全局搜索能力上比其他改进的 DE 算法有优势，将本算法与 DE 算法，jDE 算法[16]，SDE 算法[17]，EPSDE 算法[18]，CoDE 算法[10]进行对比。从表 3 可以看出 APDE 不仅收敛速度快，而且均方误差最小，从而证明了 APDE 算法的高效性。

**表 3　不同 DE 算法的误差收敛对比**

**Table 3　Error convergence comparison of different DE algorithms**

| 对比算法 | 迭代次数 | 均方误差 |
|---|---|---|
| DE | 500 | 0.013 2 |
| SDE | 354 | 0.012 9 |
| EPSDE | 135 | 0.009 0 |
| jDE | 243 | 0.010 2 |
| CoDE | 247 | 0.004 5 |
| APDE | 92 | 0.001 2 |

### 3.3　网络的预测

本文采用平均绝对误差(mean absolute percent error，MAPE)、均方根误差(root mean square error，RMSE)、相对均方误差(relative mean square error，RE)作为各算法预测准确度的指标。3 个指标的公式如下：

$$\mathrm{MAPE}=\frac{1}{Nr}\sum_{i=1}^{Nr}\frac{|y_i-y'_i|}{y_i} \tag{19}$$

$$\mathrm{RMSE}=\sqrt{\frac{1}{Nr}\sum_{i=1}^{Nr}(y_i-y'_i)} \tag{20}$$

$$\mathrm{RE}=\frac{\sum_{i=1}^{Nr}(y_i-y'_i)^2}{\sum_{i=1}^{Nr}y_i^2} \tag{21}$$

式中，$Nr$ 是预测集大小；$y_i$，$y'_i$ 分别是网络安全态势的实际值与预测值。

#### 3.3.1　泛化能力仿真分析

RBF 神经网络中的隐含层节点的个数，对整个网络的泛化能力有很大影响。当隐含层节点过多时，虽然训练拟合的程度高，但是易产生“过拟合”的现象，泛化能力降低，预测误差大；而当隐含层节点过少时，虽然泛化能力强，但是易导致“欠拟合”的问题，预测误差同样变大。

从表 4 可以看出，本文算法可以使用较少的隐含层节点达到较高的预测精度，从而使其优化过的 RBF 神经网络具有很强的泛化能力。

所以做不同隐含层节点个数对 RBF 神经网络泛化能力影响的仿真，并以平均种群差异度、最大种群差异度和平均绝对误差为评判指标，仿真结果如表 4 所示。

**表 4　隐含层节点数和种群差异度对比**

**Table 4　Comparison of the difference of population under the difference number of nodes in the hidden layer**

| 算法 | 变量 | | | |
|---|---|---|---|---|
| | $h$ | $PD_{max}$ | $PD_{avg}$ | MAPE |
| Kmeans | 5 | — | — | 0.398 5 |
| | 10 | — | — | 0.241 9 |
| | 15 | — | — | 0.271 3 |
| DE | 5 | 14 | 9.196 0 | 0.742 3 |
| | 10 | 15 | 12.584 7 | 0.280 7 |
| | 15 | 18 | 12.897 0 | 0.219 2 |
| SDE | 5 | 15 | 6.318 7 | 0.616 8 |
| | 10 | 16 | 12.661 3 | 0.264 1 |
| | 15 | 18 | 10.905 1 | 0.362 4 |
| EPSDE | 5 | 15 | 8.125 9 | 0.308 9 |
| | 10 | 16 | 11.538 5 | 0.165 5 |
| | 15 | 17 | 11.816 2 | 0.223 8 |
| jDE | 5 | 15 | 9.528 9 | 0.464 5 |
| | 10 | 16 | 12.087 8 | 0.142 2 |
| | 15 | 15 | 11.838 3 | 0.263 1 |
| CoDE | 5 | 16 | 10.546 9 | 0.437 4 |
| | 10 | 18 | 11.423 2 | 0.138 4 |
| | 15 | 18 | 10.285 4 | 0.305 1 |
| APDE | 10 | 21 | 13.447 1 | 0.069 5 |

#### 3.3.2　预测精度仿真分析

为了证明 APDE-RBF 的预测能力高于其他算法，本文分为纵向不同算法对比和横向不同改进 DE 算法对比，从而验证了本文所提算法的优越性。

从图 4 可以看出本文方法预测精度最高，其他方法都有不同程度的误差。ARMA 主要针对随机平稳的时间序列，但是因为网络攻击的随机性和复杂性，网络安全态势序列是非平稳的；GM 对于单调变化的时间序列预测效果好，反之误差大；LSSVM 的支持向量变成了所有数据点，失去了 SVM 的稀疏性特点；Kmeans-RBF 需要预先设定隐含层节点，忽略了数据本身的特点，弱化了 RBF 的泛化能力；表 5 是不同算法预测的绝对误差可以看出本文方法的预测绝对误差全都控制在 0.02 以内，相比其他算法预测精度高。

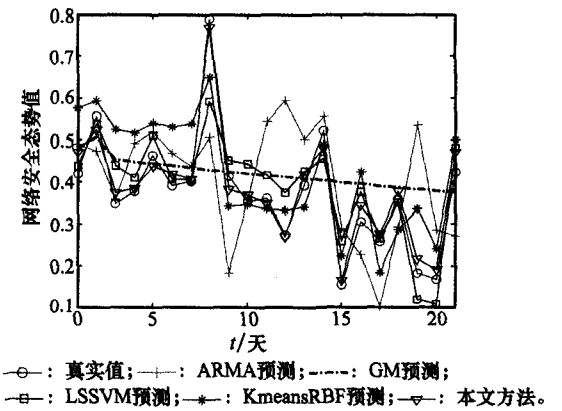



图 4　不同算法态势值预测的对比

Fig. 4　Comparison situation prediction of different algorithms

为了能够从整体上评估不同算法预测的能力，计算不同算法的 3 种误差，结果如图 5 所示。

**表 5 不同算法各时间点的绝对误差对比**

**Table 5 Absolute error comparison of different time points in different algorithms**

| 序号 | 算法 | | | | |
|---|---|---|---|---|---|
| | ARMA | GM | LSSVM | Kmeans | APDE |
| 1 | 0.069 4 | 0.061 5 | 0.015 7 | 0.155 0 | 0.051 3 |
| 2 | 0.085 4 | 0.051 2 | 0.018 2 | 0.035 2 | 0.033 1 |
| 3 | 0.010 6 | 0.103 5 | 0.090 9 | 0.176 1 | 0.026 2 |
| 4 | 0.112 9 | 0.068 0 | 0.030 5 | 0.137 7 | 0.005 5 |
| 5 | 0.055 3 | 0.019 7 | 0.046 8 | 0.079 0 | 0.023 0 |
| 6 | 0.074 5 | 0.045 9 | 0.012 9 | 0.137 8 | 0.026 7 |
| 7 | 0.035 8 | 0.032 8 | 0.005 7 | 0.138 4 | 0.007 4 |
| 8 | 0.282 2 | 0.359 7 | 0.197 1 | 0.139 9 | 0.020 8 |
| 9 | 0.229 9 | 0.011 5 | 0.037 5 | 0.070 7 | 0.029 8 |
| 10 | 0.004 2 | 0.066 4 | 0.087 8 | 0.006 9 | 0.017 8 |
| 11 | 0.180 9 | 0.052 4 | 0.052 5 | 0.027 3 | 0.014 0 |
| 12 | 0.321 6 | 0.139 1 | 0.101 5 | 0.060 3 | 0.003 9 |
| 13 | 0.109 3 | 0.014 2 | 0.033 1 | 0.052 7 | 0.027 2 |
| 14 | 0.042 4 | 0.111 2 | 0.059 9 | 0.025 2 | 0.032 4 |
| 15 | 0.131 9 | 0.243 5 | 0.104 6 | 0.067 6 | 0.009 5 |
| 16 | 0.081 5 | 0.088 7 | 0.070 4 | 0.118 8 | 0.037 8 |
| 17 | 0.157 8 | 0.131 0 | 0.005 7 | 0.076 1 | 0.016 6 |
| 18 | 0.070 2 | 0.025 2 | 0.011 9 | 0.076 6 | 0.016 9 |
| 19 | 0.351 3 | 0.201 6 | 0.062 3 | 0.153 0 | 0.033 5 |
| 20 | 0.116 6 | 0.213 7 | 0.059 1 | 0.073 9 | 0.025 2 |
| 21 | 0.152 7 | 0.046 5 | 0.059 0 | 0.078 2 | 0.047 2 |

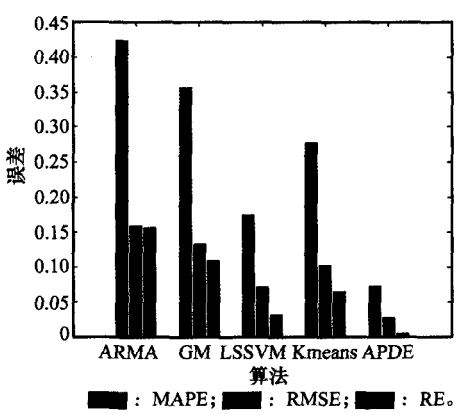


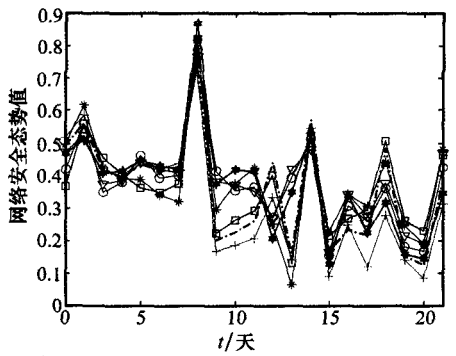

图 5 不同算法的误差对比

Fig. 5 Error comparison of different algorithms

从图 5 可以看出不论是平均相对误差、RMSE 还是 RE，本文方法都保持在较小的误差水平，体现了较高的预测精度。

以上是不同算法之间的纵向对比，为了对比的完整性，将本文算法与不同改进的 DE 算法进行对比。

图 6 显示不同改进 DE 算法在不同时间点表现了不同的预测精度，但是从表 6 可以看出，APDE 算法总体上维持了较低的绝对误差。DE 算法是固定的 $F$ 和 $CR$，易陷入局部最优；SDE 算法的 F 采用简单的随机数；EPSDE 算法利用变异策略池和参数池随机组合进行迭代进化；jDE 算法的 $F$ 和 $CR$ 依赖随机数判别从而得到不同的结果；CoDE 算法是利用 3 种不同的变异策略和参数设置竞争耦合进行迭代进化。上述方法虽然对 DE 算法的变异策略和参数设置进行自适应改进，但是大多都是随机数或依赖随机数进行判别选取，导致进化不稳定。本文算法依赖种群差异度和迭代进化程度对 $F$ 和 $CR$ 进行自适应调整，使种群向有利方向进化，加快了算法的收敛速度。




—○—: 真实值; —+—: DERBF; —·—·—: SDERBF; —✳—: jDERBF; —□—: EPSDERBF; —✱—: CoDERBF; —▽—: APDE。

图 6 不同改进 DE 算法预测对比

Fig. 6 Prediction comparison of different improved DE algorithms

**表 6 不同改进 DE 算法各时间点绝对误差对比**

**Table 6 Absolute error comparison of time point of the different improved DE algorithms** $\times 10^{-4}$

| 序号 | 算法 | | | | | |
|---|---|---|---|---|---|---|
| | DE | SDE | EPSDE | jDE | CoDE | APDE |
| 1 | 976 | 722 | 527 | 756 | 476 | 513 |
| 2 | 274 | 14 | 15 | 596 | 479 | 331 |
| 3 | 655 | 669 | 1057 | 771 | 572 | 262 |
| 4 | 154 | 159 | 244 | 108 | 355 | 55 |
| 5 | 132 | 275 | 1 017 | 749 | 175 | 230 |
| 6 | 274 | 180 | 432 | 524 | 382 | 267 |
| 7 | 364 | 262 | 254 | 810 | 179 | 74 |
| 8 | 743 | 372 | 317 | 290 | 792 | 208 |
| 9 | 2 457 | 2 112 | 1 899 | 1183 | 384 | 298 |
| 10 | 1 697 | 1 292 | 928 | 196 | 632 | 178 |
| 11 | 1 587 | 1 047 | 724 | 505 | 484 | 140 |
| 12 | 602 | 1 685 | 1 275 | 217 | 655 | 39 |
| 13 | 2 537 | 2 360 | 2 644 | 3 286 | 450 | 272 |
| 14 | 322 | 582 | 92 | 187 | 149 | 324 |
| 15 | 654 | 128 | 605 | 721 | 28 | 95 |
| 16 | 574 | 745 | 382 | 369 | 315 | 378 |
| 17 | 1 393 | 414 | 363 | 47 | 349 | 166 |
| 18 | 829 | 1 211 | 1 453 | 778 | 454 | 169 |
| 19 | 421 | 296 | 804 | 667 | 236 | 335 |
| 20 | 807 | 407 | 648 | 25 | 242 | 252 |
| 21 | 112 | 42 | 38 | 56 | 767 | 472 |

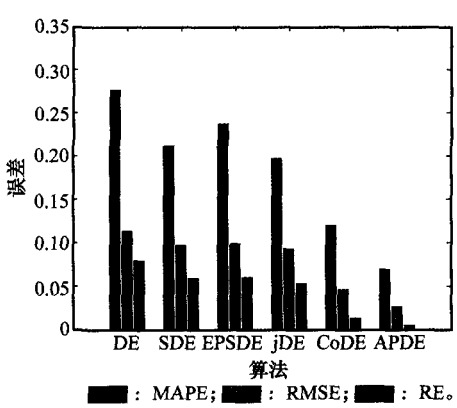



图 7 不同改进 DE 算法的误差对比

Fig. 7 Error comparison of different improved DE algorithms

从图 7 可以得出 APDE 算法在整体的误差上较其他改进 DE 算法有较大的优势，从而说明了本文算法对 DE 算法的改进是有效的。

## 4 结　语

本文提出了一种基于 APDE-RBF 算法的网络安全态势预测算法，利用 APDE 算法对 RBF 神经网络的结构和参数进行优化。利用 AP 聚类加强网络的泛化能力；利用改进的 DE 算法加快收敛速度，并用混沌搜索提高算法的全局搜索能力。通过实验数据和仿真分析，不论是纵向的其他算法还是横向的 DE 算法的不同改进，本文算法对网络安全态势预测都达到了较高的预测精度，从而证明了本文算法的高效性和实用性

## 参考文献：


[1] Ajodia S, Liu P, Swarup V, et al. *Cyber situational awareness*[M]. Germany: Springer, 2010.

[2] Xi R R, Jin S Y, Yun X C, et al. CNSSA: a comprehensive network security situation awareness system[C]// *Proc. of the IEEE International Joint Conference*, 2011, 11(3): 482 - 487.

[3] Box G E P, Jenkins G M, Reinsel G C. *Time series analysis forecasting and control*[M]. 3rd ed. Beijing: Posts & Telecom Press, 2005: 19 - 180.

[4] Dong J F. The building of network security situation evaluation and prediction model based on gery theory[C]// *Proc. of the International Conference on Challenges in Environmental Science and Computer Engineering*, 2010: 401 - 404.

[5] Halilcevic S S, Gubina A F. The short term electricity prices forecasting using markov chains[C]// *Proc. of the Energy Market*, 2011: 198 - 203.

[6] Xing Z, Zhang H. Support vector machine based aircraft ground icing type classification forecast[C]// *Proc. of the Intelligent Control and Automation*, 2012: 4541 - 4544.

[7] Wilamowski B M, Cecati C, Kolbusz J, et al. A novel RBF training algorithm for short-term electric load forecasting and comparative studies[J]. *IEEE Trans. on Industrial Electronics*, 2015, 62(10): 6519 - 6529.

[8] Frey B J, Delbert D. Clustering by passing messages between data points. [J]. *Science*, 2007, 31(3): 972 - 976.

[9] Storn R, Price K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1995, 23(4): 341 - 359.

[10] Wang Y, Cai Z, Zhang Q. Differential evolution with composite trial vector generation strategies and control parameters[J]. *IEEE Trans. on Evolutionary Computation*, 2011, 15(1): 55 - 66.

[11] Caponetto R, Fortuna L, Fazzino S, et al. Chaotic sequences to improve the performance of evolutionary algorithms[J]. *IEEE Trans. on Evolutionary Computation*, 2003, 7(3): 289 - 304.

[12] Alatas B, Akin E, Ozer A B. Chaos embedded particle swarm optimization algorithms[J]. *Chaos Solitons & Fractals*, 2009, 40(4): 1715 - 1734.

[13] Zaharie D. Critical values for the control parameters of differential evolution algorithms[C]// *Proc. of the 8th International Conference on Soft Computing*, 2002: 62 - 67.

[14] He Y C, Wang X Z, Liu K Q, et al. Convergent analysis and algorithmic improvement of differential evolution[J]. *Journal of Software*, 2010, 21(5): 875 - 855(贺毅朝，王熙照，刘坤起，等. 差分演化的收敛性分析与算法改进[J]. 软件学报，2010, 21(5): 875 - 885.)

[15] Li F W, Zhang X Y, Zhu J, et al. Network security situational awareness model based on information fusion[J]. *Journal of Computer Applications*, 2015, 35(7): 1882 - 1887.(李方伟，张新跃，朱江，等. 基于信息融合的网络安全态势评估模型[J]. 计算机应用，2015, 35(7): 1882 - 1887.)

[16] Brest J, Greiner S, Boskovic B, et al. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems[J]. *Evolutionary Computation*, 2006, 10(6): 646 - 657.

[17] Xie X F, Zhang W J, Zhang G R, et al. Empirical study of differential evolution[J]. *Journal of Control and Decision*, 2004, 19(1): 49 - 52(谢晓锋，张文俊，张国瑞，等. 差异演化的实验研究[J]. 控制与决策，2004, 19(1): 49 - 52.)

[18] Mallipeddi R, Suganthan P N, Pan Q K, et al. Differential evolution algorithm with ensemble of parameters and mutation strategies[J]. *Applied Soft Computing*, 2011, 11(2): 1679 - 1696.


## 作者简介：


**李方伟**(1960 -)，男，教授，博士研究生导师，主要研究方向为移动通信技术与理论、信息安全技术。
E-mail: huangq46@163.com

**张新跃**(1990 -)，男，硕士研究生，主要研究方向为网络安全态势感知。
E-mail: zhang_xin_yue@qq.com

**朱　江**(1977 -)，男，副教授，博士，主要研究方向为认知无线电技术。
E-mail: juliuszhu@vip.qq.com

**黄　卿**(1990 -)，男，硕士研究生，主要研究方向为网络安全态势感知。
E-mail: 2474692825@qq.com