**电磁发射技术中基于RBF神经网络的初速度预测**

**摘要**

本文提出一种基于径向基函数（Radial Basis Function ，RBF）算法的电磁发射速度预测策略，将机器学习技术中的径向基函数神经网络应用于电磁发射的速度预测。具体实现过程如下：首先建立电磁发射过程的电路模型，使用该模型仿真求解发射过程，得到包含电枢质量m，加速距离s，电流时间积分和初速度，4个特征参数的460组样本数据；然后建立基于RBF算法的神经网络模型包含输入层、隐藏层、输出层，数据集中的数据进行预处理后用于对模型的训练和测试，通过将目标变量分别设为初速度和电流，得到了能够预测初速度和电流时间积分的两个模型。用3组测试集分别测试两个预测模型，测试结果显示误差均在1%左右。因此该方法能够精确预测电磁发射中电枢的初速度，以及某一初速度所对应的电流对时间的积分值。通过遗传算法将该积分值作为目标，可以得到使电枢达到该初速度的触发时序和电流波形[16]。因此该方法具有能够精确控制发射速度的潜力。且该方法基于机器学习框架易应用、易扩展、预测速度快，相比通过数值计算预测速度的复杂性和计算时间长有非常大对优势。

## Introduction

电磁发射技术是一项先进的驱动技术。通过导轨传递电流并通过电枢闭合电流路径会产生磁场，导致施加在电枢上沿轨道方向的力[1]。它在科学实验，武器，导弹防御系统，发射火箭和卫星以及航空弹射器等领域有着广泛的应用前景。[2][3]

电磁发射是一个毫秒的过程。磁场扩散，热传导，结构变形和材料磨损耦合在这个高速发射过程和这些参数都是非线性变化。因此，电枢的初始速度无法实现实时反馈和控制。脉冲电源网络由多个并联的脉冲电源模块组成，被广泛用作电磁发射的主要能量源[4]。功率模块电气参数是在同一实验环境下确定的，而可以通过调节每个模块的放电时间改变直接影响初始速度的电流波形。通过电流精确控制初速度，是电磁发射相对于传统发射方法的优势。因此，找到通过发射条件预测初速度、以及想达到某初速度所需要的电流的方法[5]，是绝对有意义的，因为这些是精确控制速度的基础。

初始速度可以通过数值程序求解耦合变量和非线性微分方程来模拟发射过程来计算。[6]中，使用Micro-Cap VI（一种电路分析软件包）模拟轨道导弹系统和轨道导弹瞬态性能的瞬态电路模型，在[1]和[7]中，研究人员提高了轨道电感的计算精度和轨道上的力分布。然而，在所有上述方法中，工作条件和计算有不同程度的简化。在这些模拟中，它往往是单向耦合而不是双向耦合，忽略了一些问题，如轨道上的相位变化。尽管这些仿真平台可以准确地解决发射过程，但也存在很多限制。例如，所有PFU需要预先建立，所有系统参数需要手动设置，必须事先估计上限，并且如果考虑更多因素或计算双向耦合，则不可接受的模拟持续时间可能是几小时。

本文将机器学习技术应用于EML的初始速度预测中，提出了一种基于径向基函数（RBF）神经网络算法的新的预测策略。建立了基于径向基函数的机器学习模型，并对模拟得到的预处理数据进行训练，实验验证了其准确性。增加参数和训练样本的数量可以提高模型的准确性，因此在进一步的研究中可以提高该方法的准确性。该方法基于机器学习框架易于应用和扩展。一旦模型完成了训练，每个初始速度预测就可以瞬间完成，考虑到数值模拟的复杂性，这无疑具有优势。

## Obtain Data Set

为了便于获取训练数据，建立了图1所示的电磁发射系统的电路模型。输入不同的发射条件并计算初始速度。电枢的初始速度实际上是到达身管末端时的速度。基于电磁轨道发射系统的工作原理和组成部分，在电源和发射器两个方面建立了整个系统的电路模型。几个主要假设包括晶闸管和硅堆栈的压降在导通电流时大约为零，并且在发射期间导轨电感梯度不变。使用离散时间步长和电源解耦的思想用MATLAB解决电路问题。针对多模块解耦处理的相互作用，各模块放电的放电过程被认为是独立的，它们包含一阶和二阶放电过程。脉冲电源网络包括许多脉冲形成单元（PFU），其中一个如图2所示。

其中C是储能电容器，TH是普通晶闸管，D是续流硅堆栈，L是PFU的电感值和电缆电感的总和，R是电抗器电阻，电缆和一个PFU中的引线电阻;

***i***

***B***

***F***

**Pulse Power**

**Supply Network**

system diagram of electromagnetic launch

为了便于获取训练数据，建立了图1所示的电磁发射系统的电路模型。输入不同的发射条件并计算初始速度。电枢的初始速度实际上是到达身管末端时的速度。基于电磁轨道发射系统的工作原理和组成部分，在电源和发射器两个方面建立了整个系统的电路模型。几个主要假设包括晶闸管和硅堆栈的压降在导通电流时大约为零，并且在发射期间导轨电感梯度不变。使用离散时间步长和电源解耦的思想用MATLAB解决电路问题。针对多模块解耦处理的相互作用，各模块放电的放电过程被认为是独立的，它们包含一阶和二阶放电过程。脉冲电源网络包括许多脉冲形成单元（PFU），其中一个如图2所示。

其中C是储能电容器，TH是普通晶闸管，D是续流硅堆栈，L是PFU的电感值和电缆电感的总和，R是电抗器电阻，电缆和一个PFU中的引线电阻;

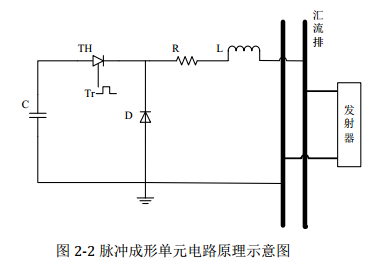
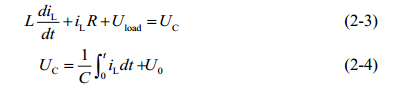


Fig. 2. Circuit schematic of Pulse Forming Unit

The relationship expression is



When the capacitor stops discharging, its model expression is



iL 为 PFU 中流过电感形成脉冲的电流， UC 为电容两端电压，U0 为电容器初始充电电压, Uload 为电源侧汇流排两端电压（或称负载电压）。



I current into the track from all the PFU；ik （k= 1,2, , n）current value of the kth PFU

In the circuit model, the influence of the velocity skin effect [8][9], the contact resistance[10] and the friction between the armature and rail[11] are taken into account.

发射器也就是负载端的负载电压和运动控制方程分别为：

The governing motion equations and load voltage at the launcher side are:











Rsum是回路中轨道电阻、电枢电阻、枢轨间的接触电阻、速度趋肤效应产生的电阻和电缆电阻之和；Lr Lcable分别为轨道和电缆的电感；a v s分别为电枢在t时刻的加速度、速度和位移；

从电路模型中可以看出影响电枢初速度的主要变量包括质量、加速距离和电流。通过该仿真得到的电流数据是发射过程中的电流波形，而训练时要求数据集中每一个特征参数是一维的，因此使用电流对时间的积分值（也就是波形对时间横轴所围的面积）作为电流特征参数。应用MATLAB的语言，并用此模型模拟发射过程。，随机给出不同的质量、加速距离和电流计算得到初速度。以获得训练RBF神经网络所需的数据集。

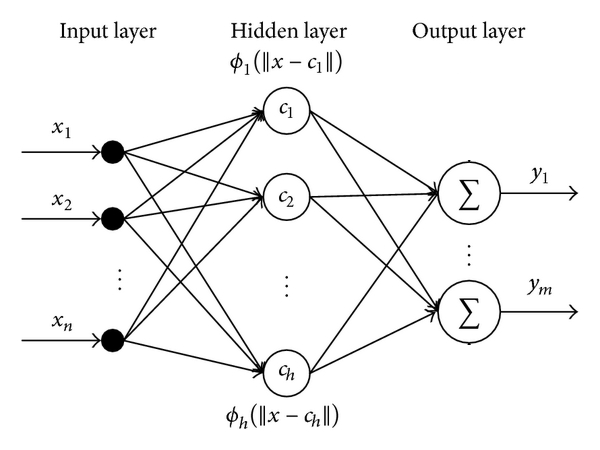
## RBF Neural Network Building and Training

1. **Algorithm Introduction**

径向基函数是一种传统的多维空间插值技术，由Powell于1985年提出.RBF神经网络属于BP神经网络的类型。网络结构类似于多层前向网络，是一个三层前向网络[12]。一般的BP神经网络在函数逼近中花费更多的训练时间，容易陷入局部最小值，并且获得的网络表现不佳。而RBF神经网络具有良好的函数逼近能力，结构简单，学习速度更快。

1. **The structure and parameters of the RBF Neural Network**

RBF 神经网络模型结构如图 5-2 RBF



第一层是输入层。输入矢量x用作所有径向基函数的输入。第二层是具有非线性RBF激活函数的隐藏层且径向基函数通常使用高斯函数。每个RBF单元的输出如下[13-14]：。



x是输入向量ci是第i个高斯函数中心，可以使用k均值聚类算法来确定[15]，为欧式范数， n为隐藏层的个数，σi是RBF单元的宽度



网络的输出是输入向量的标量函数,

p表示第p个样本，j表示输出层中的第j个神经元，ypj是第p个样本和第j个输出神经元的输出，wij是第i个接收字段到第j个输出的权重或强度，w0j是偏差的第j个输出。

为了减少网络的复杂性，在以下分析中不考虑偏差。

输出值的目标函数期望Ep：



其中表示相对于第 k 个输入向量的理想输出，表示相对于第 k个输入向量的实际输出



其中为学习率，权值修正公式为：



通过对网络权值的不断循环调整计算出最优的权值。

1. **Data preprocessing and Model Training**

在第二部分中我们已经获得了包括质量、加速距离、电流和初速度的数据集。为了使模型有较好的训练效果，在使用这些数据训练模型之前需要进行预处理：异常值检测和归一化处理，由于该数据集来自仿真计算，理论上不会出现异常值，因此只进行归一化处理。对数据集中的每一个特征进行如下转换：



对于特征i，是转换后的数据，x是原始数据，μ和σ分别是该特征数据的均值和标准差。

为了通过电枢质量、加速位移、电流三个特征预测初速度、以及预测达到某初速度所需要的电流，我们训练了以下两个模型,两个模型的参数配置分别如表1表2

1. 将电枢质量、加速位移、电流对时间的积分值作为输入特征向量x，预测目标是初速度v
2. 将电枢质量、加速位移、初速度作为输入特征向量x，预测目标是电流对时间的积分值

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 样本序号 | 特征 | | | 目标 |
| 电枢质量g | 加速位移m | 电流时间积分kA·ms | 初速度V |
| 1 | 9.64 | 2.47 | 209.58 | 2303.60 |
| 2 | 8.11 | 2.17 | 162.43 | 2996.28 |
| 3 | 12.54 | 2.42 | 221.11 | 3220.46 |
| … |  |  |  |  |
| 460 | 11.20 | 2.24 | 232.64 | 1744.46 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 样本序号 | 特征 | | | 目标 |
| 电枢质量g | 加速位移m | 初速度V | 电流时间积分kA·ms |
| 1 | 9.64 | 2.47 | 2303.60 | 209.58 |
| 2 | 8.11 | 2.17 | 2996.28 | 162.43 |
| 3 | 12.54 | 2.42 | 3220.46 | 221.11 |
| … |  |  |  |  |
| 460 | 11.20 | 2.24 | 1744.46 | 232.64 |

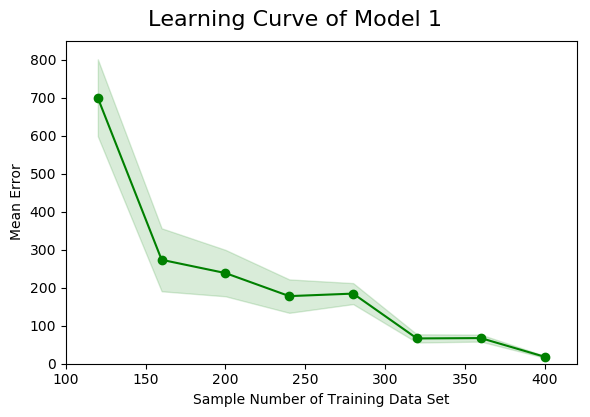
我们将表中数据集分割为含有400个样本的训练集和3个含有20个样本的测试集，测试集用于测试训练得到的最优模型。

将X（m, x, I）作为特征，初速度作为目标时，神经网络输入400\*3的特征矩阵， 400\*1的目标矩阵，得到模型1；

将X（m, x, V）作为特征，电流时间积分值作为目标时，神经网络输入400\*3的特征矩阵，400\*1的目标矩阵，得到模型2。

## Results

Model1在测试集上的预测误差随训练样本数量的变化如图1：

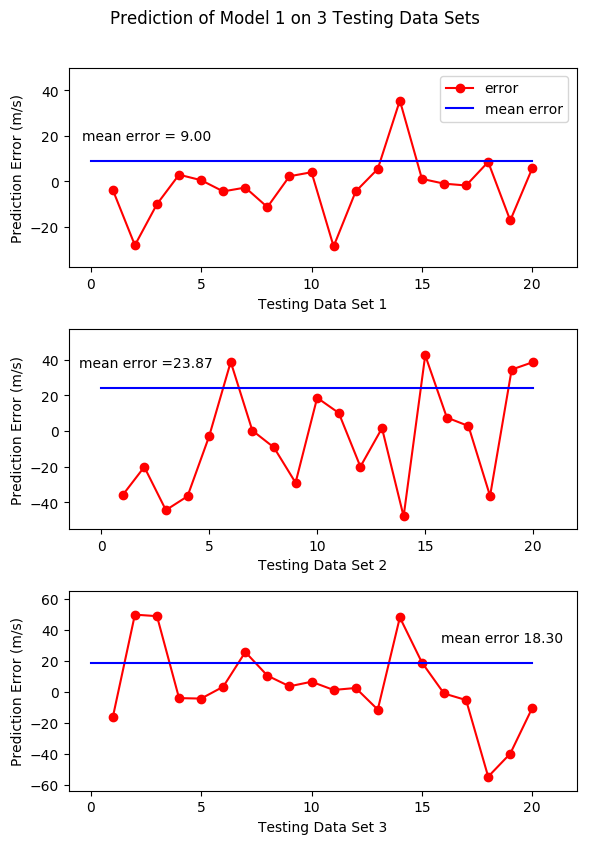


|  |  |  |
| --- | --- | --- |
| sample | test\_diffmean | test\_std |
| 40 | 406.06 | 501.13 |
| 80 | 468.69 | 550.99 |
| 120 | 699.26 | 1012.86 |
| 160 | 273.31 | 828.12 |
| 200 | 238.35 | 610.42 |
| 240 | 177.78 | 439.56 |
| 280 | 184.4 | 276.13 |
| 320 | 66.51 | 108.12 |
| 360 | 67.41 | 92.09 |
| 400 | 18.3 | 25.76 |

测试集含有20个通过仿真随机得到的样本。可以看出随着训练样本数量增多，预测误差逐渐减小。本研究中平均误差为误差绝对值的平均值，因此都是正数。阴影部分为预测误差的标准差。训练样本较少时标准差极大，为便于显示图中使用了真实标准差的0.1倍。真实的平均误差绝对值和标准差如表1.

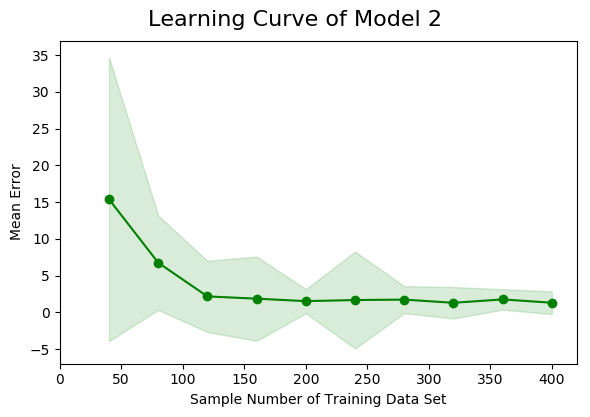
训练样本数量达到400时在训练集上的平均预测误差达到最小为18.3m/s, 标准差为25.76m/s。因此选择样本数量400训练后的模型为最优模型，该模型有160个隐藏层神经元，Spread of radial basis functions is 0.93。

图2是该模型在3个不同的测试集上的表现，具体的平均误差和标准差见表2.



|  |  |  |
| --- | --- | --- |
| 测试集 | Mean error | Standard error |
| 1 | 9.00 | 13.27 |
| 2 | 23.87 | 28.33 |
| 3 | 18.30 | 25.76 |

Model2在测试集上的预测误差随训练样本数量的变化如图3：

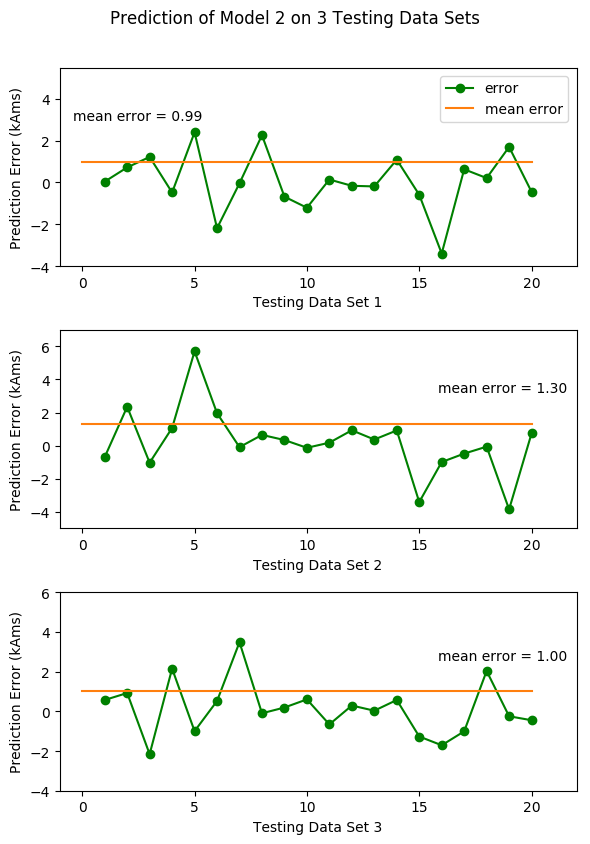


|  |  |  |
| --- | --- | --- |
|  | test\_diffmean | test\_std |
| 40 | 15.388 | 19.32 |
| 80 | 6.752 | 6.43 |
| 120 | 2.163 | 4.85 |
| 160 | 1.856 | 5.73 |
| 200 | 1.518 | 1.65 |
| 240 | 1.669 | 6.59 |
| 280 | 1.727 | 1.84 |
| 320 | 1.296 | 2.14 |
| 360 | 1.753 | 1.39 |
| 400 | 1.302 | 1.55 |

图3中的标准差为实际值，具体平均误差和标准差见表3。从图3可以看出训练样本数量达到120时，训练集上预测误差绝对值平均接近于0，但标准差较大。

训练样本数量达到320时在测试集上的平均预测误差达到最小为1.3kA·ms, 标准差为2.14 kA·ms。选择该模型为最优模型，该模型有140个隐藏层神经元，Spread of radial basis functions is 1。当训练样本数量增加至360和400时，平均误差反而上升，可能的原因是模型出现了过拟合[16]。

图4是该模型在3个不同测试集上的表现，具体的平均误差和标准差见表4.



|  |  |  |
| --- | --- | --- |
| 测试集 | Mean error | Standard error |
| 1 | 0.99 | 1.35 |
| 2 | 1.30 | 1.93 |
| 3 | 1.00 | 1.32 |

对比模型1和2，可以看出模型1的误差和标准差较大，但模型1预测的量是初速度，模型2预测的是电流对时间积分，考虑前者比后者实际值要大得多（见Part3-C表1表2），且预测误差与真实值的比值始终在1%左右，该范围内的误差是完全可以接受的。模型1的训练耗时为6.3秒，模型2的训练耗时为2.4秒。完成测试集20个样本的预测耗时0.3秒，表明使用rbf 神经网络模型用于电磁发射系统初速度与电流的预测是十分准确且高效的。

## Conclusion and Discussion

本研究使用rbf neural network通过电枢质量m，加速距离s，电流时间积分值预测电磁发射中电枢的初速度，精确度能够达到18.3m/s，通过电枢质量m，加速距离s，和电枢初速度预测发射过程中电流对时间的积分值，精确度能够达到1.3kA·ms，同时该方法仅分别使用了400组和320组样本数据进行训练，意味着使用真实的实验数据用作训练集是可行的。

同时，通过该方法可以得到某一初速度所对应的电流对时间的积分值，那么根据文献[16],通过遗传算法将该积分值作为目标，可以得到使电枢达到该初速度的触发时序和电流波形。因此该方法具有能够精确控制发射速度的潜力。

虽然电磁发射的初速度的预测可以通过仿真模拟一次发射实验来获得。例如通过仿真软件或编写程序进行仿真。使用仿真软件需要耦合多个物理场，电磁场、温度场、力场；编写程序则需要准确的模拟电源模块的放电、电枢的运动过程以及关键部位的温度变化。想提高预测的准确度和精确度就需要对各影响因素建立准确的数学模型进行描述，建立精细的物理模型和剖分网格进行数值计算，这意味着冗长的代码和长达几小时的计算时间，并且不同的供电设备、不同结构的发射装置、不同结构的电枢都需要不同的模型和代码。但通过应用机器学习的方法，模型易扩展，可以增加更多特征，如轨道特性参数、电枢特性参数、接触电阻、接触面粗糙度等。设计出适用于更多种装置的模型，训练数据的来源也同时增多。因此一旦训练出可用的模型，该方法预测初速度或电流要比一个复杂的仿真系统准确、高效且普遍适用得多，应用前景极具吸引力。

[1]BAYATI M S, KESHTKAR A, KHOSRAVI F等. Analyzing the electromagnetic launcher with combination both FEM-3D and IEM methods in time domain[C]//2012 16th International Symposium on Electromagnetic Launch Technology. 2012: 1–7.

[2] J. S. Bernardes, M. F. Stumborg, and T. E. Jean, “Analysis of a capacitor-based pulsed-power system for driving long-range electromagnetic guns,” IEEE Trans. Magn., vol. 39, no. 1, pp. 486–490, Jan. 2003.

[3] J. D. Wang, “Development and application in military of electromagnetic emission technology,” in Proc. Fire Command Control. 2001, p. 1.

[4] H. D. Fair, “Electric launch science and technology in the United States,” IEEE Trans. Magn., vol. 39, no. 1, pp. 11–17, Jan. 2003.

[5] Ma J, Zhang D D, Yuan W Q, et al. A method of generating timing for a given target current waveform in electromagnetic launch technology [J].IEEE Transactions on Plasma Science. 10.1109/TPS.2017.2706219.

[6] J. S. Bernardes, M. F. Stumborg, and T. E. Jean, “Analysis of a capacitor-based pulsed-power system for driving long-range electromagnetic guns,” IEEE Trans. Magn., vol. 39, no. 1, pp. 486–490, Jan. 2003.

[7] GHASSEMI M, BARSI Y M, HAMEDI M H. Analysis of Force Distribution Acting Upon the Rails and the Armature and Prediction of Velocity With Time in an Electromagnetic Launcher With New Method[J]. IEEE Transactions on Magnetics, 2007, 43(1): 132–136.

[8] Yu X ，Fan Z ．Simulation and two-objective optimization of the electromagnetic-railgun model considering VSEC resistance and contact resistance[J]． IEEE Transactions on Plasma Science,2011,39(1):405-410．

[9] Engel T G,Neri J M,Veracka M J. Characterization of the velocity skin effect in the surface layer of a railgun sliding contact[J]. IEEE Transactions on Magnetics, 2008, 44(7)： 1837-1844.

[10] Hsieh K T,Satapathy S, Hsieh M T．Effects of pressure-dependent contact resistivity on contact interfacial conditions[J]． IEEE Transactions onMagnetics,2009,45(1),313-318．

[11] Aigner S, Igenbergs E. Friction and Ablation Measurement in a Round Bore Railgun[J]. IEEE Transactions on Magnetics. 1989，25(1) :33~39.

[12] POWELL M J D. Radial Basis Function Methods for Interpolation to Functions of Many Variables[M]. 2001.

[13] WANG Z, YANG B, KANG Y等. Development of a Prediction Model Based on RBF Neural Network for Sheet Metal Fixture Locating Layout Design and Optimization[J]. Computational Intelligence and Neuroscience, 2016, 2016: 1–6.

[14] ER M J, WU S, LU J等. Face recognition with radial basis function (RBF) neural networks[J]. IEEE Transactions on Neural Networks, 2002, 13(3): 697–710.

[15] LIKAS A, VLASSIS N, J. VERBEEK J. The global k-means clustering algorithm[J]. Biometrics, 2003, 36(2): 451–461.

[16] DIETTERICH T. Overfitting and Undercomputing in Machine Learning[J]. ACM Comput. Surv., 1995, 27(3): 326–327.