目录

[1 一级标题 2](#_Toc11615486)

[1.1 二级标题 2](#_Toc11615487)

[1.2 二级标题 2](#_Toc11615488)

[2 EMD算法概述 2](#_Toc11615489)

[2.1 EMD的概念和基本算法 2](#_Toc11615490)

[2.1.1 EMD算法的基本概述 2](#_Toc11615491)

[2.1.2 EMD算法的停止准则 3](#_Toc11615492)

[2.1.3 EMD算法的端点问题 4](#_Toc11615493)

[2.2 二级标题 4](#_Toc11615494)

[3 第三章 4](#_Toc11615495)

[3.1 二级标题 4](#_Toc11615496)

[3.2 二级标题 5](#_Toc11615497)

[3.2.1 三级标题 5](#_Toc11615498)

1 一级标题

1.1 二级标题

1.2 二级标题

2 EMD算法概述

经验模式分解(Empirical Mode Decomposition, EMD)是由Huang(1998) [73]等于1998年提出的一种处理非线性、非平稳性信号的方法, 其本质是对时间序列进行平稳化处理。与小波分解不同的是EMD是由数据驱动的一种序列分解方式，其更具原序列的极值特征直接对原序列进行分解不依赖于基函数。EMD本质上可以被看作是一种滤波，将不同尺度的波动逐级进行分解。经验模式分解因为其前提假设少，适用性广，提出之后很快就被用于许多理工科领域[59]。

2.1 EMD的概念和基本算法

2.1.1 EMD算法的基本概述

首先Huang在其文章中定义本征模态函数(Intrinsic mode functions)。当函数满足以下两个条件时，我们视其为一个本征模态函数：

（1）在整个数据集中，极值点的个数和过零点的个数必须相等或最多相差1个；

（2）在任意时间点，由局部极大值定义的包络线和由局部极小值定义的包络线的均值为零。

EMD分解是一种直观的、后验的、自适应的分解，完全有数据驱动。该方法的实质是对数据中固有振荡模态的特征时间尺度进行经验识别，然后对数据进行相应的分解。EMD分解基于以下假设：

（1）信号序列至少有两个极值，一个极大值一个极小值；

（2）时间尺度的特征由极值之间的时间间隔来定义；

（3）如果数据没有极值只有拐点，那么可以通过对数据差分来获取极值点。

从本质上讲经验模式分解过程是一个筛选过程，从该过程中依次筛选出频率由高到低的本征模态函数，并最终剩余一个不能再分解的单调残差序列，也称为趋势项。也就是说EMD可以自适应地把一个时间序列分解成有限个IMFs和一个残差项。算法描述如下：

Step1：输入原始序列信号；

Step2：获取原始序列的所有极大值点和极小值点，并分别对极大值点和极小值点用2条3样条插值曲线拟合，得到一条上包络线（upper envelope）和下包络线（lower envelope）。

Step3：计算包络线的均值；

Step4：计算和的差别并令；

Step5：检查是否满足上述的两个IMF的条件，如果是一个IMF或满足停止准则，则转到Step6；否则令，并重复Step1-4直到是一个IMF或满足停止准则。

Step6：计算残差项（residue），如果是一个单调函数或者只有一个极值点则分解完成；否则令，重复Step1-5。

至此，我们将原始信号分解为若干个本征模态函数和一个趋势项，用公式（2-1）表示：

 公式（2-1）

在上述的分解过程中，分解得到的本征模函数很容易满足第一个条件，即极值点的个数与过零点的个数相等或最多相差1个，然而第二个条件即局部上包络线和下包络线的均值为0却不一定能通过变换满足。而且当第二个条件发挥到极致时，可能会消除物理意义上的振幅波动，因此在筛选过程中要格外小心，因为如果将筛选进行到极致，可能使得到的IMF成为恒定的纯调频信号。为了确保IMF各组成部分保持足够的振幅和频率调制的物理意义，我们必须确定一个停止筛选过程的标准[73]。

2.1.2 EMD算法的停止准则

根据2.1.1中IMF的定义，严格IMF应满足局部极大值定义的包络函数和局部极小值定义的包络函数的均值恒为0，而在实际分解过程中很难做到分解出的上下包络之和严格等于0，只能接近于0。针对这种情况，一些学者通过实验给出了一些相对较为理想的停止准则也被称为局部包络条件。

EMD的提出者Huang(1998)[73]在其论文中给出了标准偏差(SD)准则，通过限制SD的值来判断是否停止筛选。SD是由两个连续的筛选结果计算得到的，计算过程用公式（2-2）表示[73]：

 公式（2-2）

其中为第个IMF第次迭代的值，计算公式如见2..1.1。当SD小于研究者给定的某个值时，就认为该为第个本征模态函数。Huang在其文章中通过大量实验给出取值在0.2和03之间时分解效果较好。后来Huang等人又给出了另一个相似的停止准则用公式（2-3）表示：

 公式（2-3）

其中为第个IMF第次迭代的上下包络线的均值（计算见2.1.1）。上述两种停止准则由柯西收敛性检验给出在数学上似乎严格。然而，由于以下原因很难实现这个标准:首先，SD小到什么程度就足够小，这其实并没什么标准。其次，这个准则不依赖于IMFs的定义，因为平方差很小但不能保证函数会有相同数量的过零点和极值。为了弥补这些缺点，Huang(1999,2003)等人提出了第二种类型的标准，称为S停止准则。使用这种类型的停止准则，筛选过程只有在零点个数和极值个数相等或最多相差1且连续S次保持相同时才停止。Huang(2003)等人的大量试验表明，S的最佳范围应该在3到8之间，但更倾向于较低的数字。显然，任何选择都是临时的，需要一个严格的理由[74]。

法国学者Gabriel Rilling(2003)[75]等基于Huang等的研究提出了一个较为常用评价函数来控制筛分过程的停止

 公式（2-4）

其中为上下包络线的均值，。并设置三个阈值、、，当满足以下两个条件：

1. 当里面小于的比率达到；
2. 不存在大于的值。

则认为分解结束，中止筛选，一般取，和[75]。这也是本文中所采用的停止准则。

2.1.3 EMD算法的端点问题

在EMD相关算法的应用过程中，端点效应的处理是数据预处理过程中一个重要的步骤。所谓端点效应，是指在运用EMD对信号进行分解时，需要去“筛选”出本征模函数，而“筛选”过程需要对极值点使用三样条插值构造上下包络线，如果数据在端点处不是极值，那么在对极值点插值拟合时，将会出现误差从而产生端点效应。在端点附近时，三样条插值将出现发散现象，而随着本征模函数的一步一步提取，端点效应向内扩散最终导致污染数据[59]。Huang在其文章中指出在信号的两端，根据端点信号的振幅和频率，分别加两个特征波，可以有效抑制端点效应，但没有给出具体的做法。

处理端点问题最简单的方法是在迭代过程中不断地删除非极值的部分，但是对于预测问题来讲抛弃右端点的值也就意味着抛弃了近期的数据，然而近期的数据包往往是含着信息最大并且最重要的数据。所以如果是对时间序列值做简单的分析分解的话可以使用此方法，但是对于预测问题，往往不适合此方法。本文主要研究的是对股价指数的预测问题，不适合使用此方法。

另外一种解决端点问题的思想就是人为的添加一些数据，也称为延拓法。随着EMD算法的广泛应用，一些学者陆续的提出了一些较为经典的延拓方法，这些延拓方法大致可以概括为两种直接延拓法和预测延拓法[59]。

直接延拓法就是通过一定的技术手段将原有数据向外延拓。目前主要的延拓方法有，镜像延拓[76]、偶延拓、周期延拓[80]、极值延拓、平行线延拓、加窗口延拓[81]、、Coughlin法（Coughlin and Tung 2004）[79]、斜率法(Datig and Schlurmaim 2004)[77]和Rato法(Rato 2008)[78]等。

预测延拓法主要是指根据各种预测模型或方法对未来一些点进行外推预测的方法。常见的预测延拓方法有时间序列预测延拓，神经网络预测延拓，以及一些利用机器学习方法进行端点预测的方法。与直接延拓相比预测延拓往往需要更大的计算量，而且要去根据具体情况选择合适的预测方法，这样给延拓带来个一些不确定性。

本文主要研究的是模型的简化和优化问题，并不关注这些延拓方法的区别。所以本文取最为常用并且简单的延拓方法-镜像延拓法来处理端点问题。镜像延拓法通过对原始数据序列邻近边缘的极值点添加镜像,以对称的方式增加n个极值点的方式来抑制分解过程中的端点效应。对于图2.1镜像延拓法过程如下：

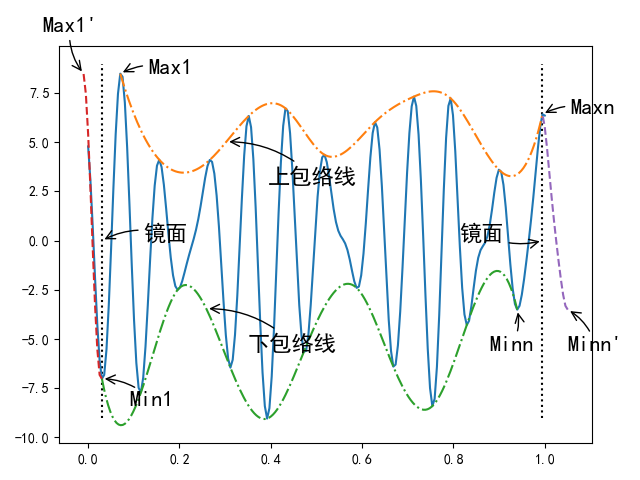


图2.1 镜像延拓法过程

Step1：找到原始序列最左端的极值点Min1（第一个极小值点），以及最右端的极值点Maxn（最后一个极大值），如图2.1中所示；

Step2：别以Min1和Maxn为镜面得到Max1的镜面对称点Max1’和Minn的镜面对称点Minn’,如图2.1所示；

Step3：最后根据数据原有的极大值点和极小值点加上有镜面对称得到的点，分别对运用三样条插值构建上包络线和下包络线。这样可以在一定程度上一直端点的发散（图2.1给的上、下包络线为未进行延拓下的包络线）。

2.2 EMD算法的改进研究

因为停止条件的限制，在经验模态分解过程中可能会出现模态混叠，使得不同特征的序列分解成一个子序列。针对此类问题一些学者提出了一系列的改进算法。其中在处理信号序列中最常用的是一下介绍的EEMD算法和CEEMD算法。在其它序列数据处理中EMD算法和EEMD算法较为常用。本文研究的是有序列分解和简单神经网络混合模型的预测框架的优化问题，并不注重于不同的分解算法对结果的影响。所以本文选择最简单而且效率最高的EMD算法。

2.2.1 集成经验模态分解

针对经验模态分解存在的模态混叠现象，Wu and Huang(2009)[82] 等人提出集成经验模态分解(Ensemble Empirical Mode Decomposition, EEMD)算法，显著地提高了经验模态分解算法的应用价值。

集成经验模态分解（EEMD）算法的主要思想是：向原始的序列数据添加一个白噪声，然后对白噪声和原始序列的混合序列进行分解，重复N次此过程，将所得到的N组IMF和N个余项分别取平均值作为最后的分解的结果。集成经验模态分解过程中认为经过N次集成平均，序列中添加的白噪声可以相互抵消，同时克服了经验模态分解中出现的模态混叠现象，能够更好的实现信号序列的分解，是经验模态分解算法发展中的一个极其重要的改进。

在2.1.1章节EMD算法的基础上，EEMD算法步骤如下：

Step1：在原始信号中添加一个白噪声信号，得到下列混合信号；



其中，表示第次信号混合，，为信号混合的总次数，是一个人为设定参数，通常取[82]。

Step2：对混合后的信号序列进行经验模态分解，得到第次分解的子序列和其中为分解的个IMF（为了确保分解信号个数相同一般要求添加的白噪声为低信噪比的白噪声序列）。重复该步骤，直至。

Step3：对每个IMF和余项取平均，作为最终分解结果，得到第项IMF和余项：





Step4：得到最终的分解结果：



其中，为分解的到的本征模函数的个数。

2.2.2 完整集成经验模态分解

根据2.2.1中的介绍，我们知道集成经验模态分解算法是通过添加噪声项来解决模态混叠问题。在引入噪声项后我们假设经过多次添加噪声项，噪声之间可以相互抵消。但实际上在信号重构时，引入的噪声项并不能完全抵消，一般都是通过增加集成的次数来减弱残留噪声项的相对大小，，但是这样做往往会大大的增加计算量。为解决经验模态分解过程中的模态混叠问题，同时消除集成分解过程中添加的白噪声对信号重构的影响, Yeh J R and Shieh J S(2010)[83]提出了完整集成经验模态分解(Complete Ensemble Empirical Mode Decomposition, CEEMD)。该算法的主要思想是成对的添加噪声，并且使得所添加的白噪声和为零。基于EMD和EEMD算法，CEEMD算法分解过程如下。

Step1：将白噪声成对的添加到原始信号中，得到一组待分解信号，并且该组信号的白噪声序列和为零：



其中，表示原始信号序列，表示加入的白噪声，表示与正的白噪声叠加后的信号序列，表示与负的白噪声叠加后的信号序列。

Step2：分别对、进行EMD分解，重复N次。得到分解后的IMF子序列和余项：





其中，、和、分别表示第次混合信号、分解得到的子序列和余项。

Step3：最终，由CEEMD算法将原始信号分解为



其中，为分解的到的本征模函数的个数。

2.2.3 具有自适应噪声的完整集成经验模态分解

EEMD和CEEMD都是通过对原序列添加噪声来处理模态混叠问题。有些学者考虑在分解过程中添加噪声来处理模态混叠问题，进而提出了具有自适应噪声的完整集成经验模态分解(Complete Ensemble Empirical Mode Decomposition with Adaptive Noise, CEEMDAN)算法，该算法的主要思想是：在序列分解过程中的每个阶段都添加自适应白噪声，通过计算唯一的余量信号获得IMF子序列，在较少的集成次数后就能够使重构误差几乎为零[84]。CEEMDAN算法过程如下（详见Humeau-Heurtier(2105)[85]）：

Step1：在原始信号序列中添加一些自适应白噪声：



其中，表示第次加入白噪声后的信号，表示噪声系数（噪声标准差），表示第次添加的白噪声，表示集成次数。

Step2: 对每个信号，分别采用EMD算法分解获得第一个IMF分量，然后对这个分量取均值：



其中，表示分解得到的第一个IMF信号。则余项信号为：



Step3：定义为使用EMD算法分离出第个本征模函数的过程，分解，，直到分解出一个IMF。(在该步)在每一步允许选择的信噪比（SNR）。然后，得到第二个IMF：



Step4: 计算第个余项：



使用EMD算法分解，直到分解出一个IMF，定义第IMF：



Step5：回到Step4，并令，直到余项不能再被分解。得到最终的余项：



其中为分解得到的IMF个数。最终原信号序列被写作：



2.3 本章小结

本章对经验模态分解算法进行了详细的介绍。介绍了经验模态分解算法的思想、基本原理以及算法实现过程。而且本章也对经验模态分解存在的问题以及其改进算法做了一些简单的介绍。本文并没有深入探讨不同的停止准则、不同的端点处理手段、不同的经验模态分解改进算法对结果的影响。本文主要的研究内容是经验模态分解与简单神经网络混合预测模型结构的改进。所以在此说明本文所采用的停止准则为2.1.2中提到的法国学者Gabriel Rilling(2003)[75]提出的停止准则，端点问题的处理依据2.1.3中详细介绍的而且较为常用的镜像延拓法，分解算法采用最简单并且效率最高的EMD算法分解序列。更多细节问题见代码链接。

3 基于经验模态分解和神经网络的混合预测模型

3.1 相关算法介绍

3.1.1 ANN算法介绍

人工神经网络(ANN)是一种模拟生物神经网络结构和功能的数学模型。因为其在变量间非线性关系强大的识别能力，在过去的几十年里得到了广泛的应用。在众多不同类型的人工神经网络中，反向传播人工神经网络(BP-ANN)由于其结构简单、易于实现，已被证明是解决许多问题的有力工具[66]。本文中使用三层神经网络预测股票指数，网络结构大致如图3.1所示。假设输入层有n个节点，隐藏层有p个节点，输出层有一个节点。



假设输入层有n个节点，隐藏层有p个节点，输出层有一个节点。则b1是p维向量，b2是一维向量。

对于隐藏层的节点hi来说



其中，为个隐藏层的第个节点，为输入变量的权重值，为第个隐藏层节点的偏置，为激活函数。



其中(图3.1只有一个输出，可以有多个输出)，为个输出层的第个节点，为隐藏变量的权重值，为第个输出节点的偏置，为激活函数。

3.2 EMD与ANN混合预测模型及改进

自1998年经验模态分解法提出以来，经验模态分解算法已经取得了较大的发展，并且理论知识日趋完善。经验模式分解因为其前提假设少，适用性广，提出之后很快就被用于许多理工科领域[59]。正如第一章节文献综述所介绍，使用EMD分解算法，然后对其子序列分别进行预测，最后在把预测结果相加这一预测模型框架在许多领域的研究都表明能取得较好的表现性能。然而这一预测模型框架，往往存在模型过多、计算量过大、参数难以择优等一系列问题。特别是使用EMD算法与机器学习算法结合时，例如EMD-ANN[59,61,62,63,66,68,71,72,84,86]、EMD-SVM[58,61,63,64,67,]、EMD-LATM[60]等都会出现模型过多所带来的计算量大、超参数过多难以对模型择优等问题。本文针对这些问题提出了一种改进的预测模型结构，并结合实证实验研究模型的表现性能。人工模型由于具有非线性、非局限性的优良性质，在进行组合预测模型中是最常用的基本模型之一，本章主要针对EMD与ANN模型的混合提出一种基于EMD的单模型ANN预测模型。

3.2.1 基于EMD与ANN的预测模型

现实中许多时间序列都具有高度复杂的特征，其高度非线性非平稳性性等一系列性质，使得直接对序列分析变得比较困难，也难以用具体模型去捕捉时间序列的特性，进而难以建立精确的预测模型。近年来对时间序列的研究越来越广泛，一系列对时间序列的处理方法相继被提出，如单尺度分析、多尺度分析、DFA趋势分析、MF-DFA趋势分析[57]、先分解后组合[84]分析等一些方法，本文多研究的就是基于“先分解后组合”思想的一种方法。先分解后组合的模型框架即首先将复杂的时间序列使用一些分解算法分解成若干子序列，然后再对其子序列进行建模分析预测。已有文献所给出的预测模型结构大多和本文使用的组合预测模型EMD-ANN相似如图===所示，只是基本预测模型和分解算法略有不同。

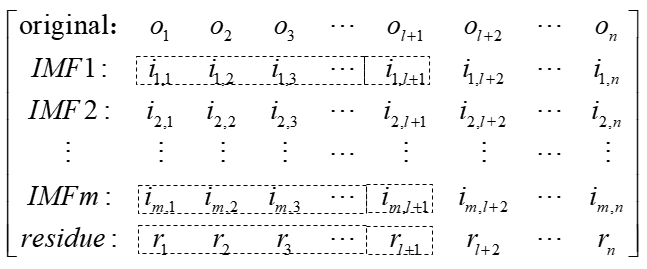


EMD-ANN的预测框架结构如图===所示，其主要算法步骤如下：

Step1：对时间序列进行使用EMD算法进行分解，等到一系列的子序列IMF1，IMF2,……IMFm和一个余项RES。

Step2：对所有的子序列建立ANN模型，用前个序列值预测第项的值，具体变量预测方式如图====所示。

Step3：对所有子序列的预测结果进行叠加等到最终预测值。



如图===所示绝大多数学者都是对每一个IMF和余项都进行建模，最后把每个模型得到的结果叠加作为最后的预测值，如图====虚线标识所示，对每一个子序列（IMF和residue）分别用前个序列值预测第项，并使用滑动窗口法预测余下各项。

3.2.2 基于EMD与ANN的单一模型预测结构

传统的基于EMD的预测模型框架都是对于每一个分解出来的IMF进行建模预测，然而对于每个IMF进行建模代价往往要面临极大的计算量这一问题。特别是在使用机器学习模型对每个IMF进行预测时往往需要很大的计算量，特别是现在使用支持向量机模型、人工神经网络等算法在构建模型的时候往往有一定量的超参数要进行试验调优，所以所组合的模型个数越多也就会有越多的超参数需要去确定。而且在对每个IMF结果进行叠加的时候往往会造成大量的累积误差，如果有个模型叠加作为最后结果就会有项的模型误差累积。再者有学者研究（H. Zhou(2019)[62]、张文风（2017）[59]）表明经过EMD分解的各项IMF往往具有一定的短期相关性，各IMF的特征之间具有一定的“交互作用[62]”，所以如果只是对各个子序列进行建模往往会忽略各子序列之间的相互影响。考虑到这些问题本文提出了一种基于EMD算法的单一模型预测框架并基于此框架建立S(single)-EMD-ANN模型。模型框架如图===所示。

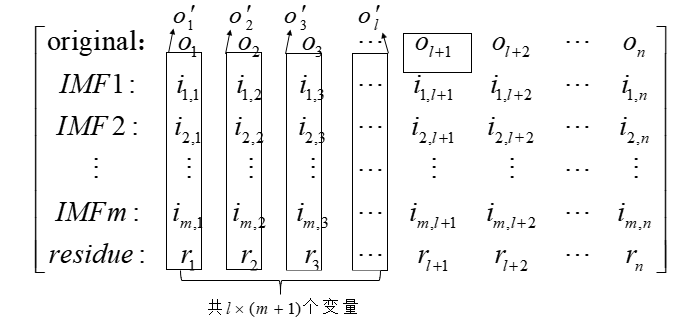


S-EMD-ANN的预测框架结构如图===所示，其主要算法步骤如下：

Step1：对时间序列进行使用EMD算法进行分解，等到一系列的子序列IMF1，IMF2,……IMFm和一个余项RES。

Step2：对所有的子序列建立一个ANN模型，用每一个子序列的前个序列值预测原始序列第项的值，具体变量预测方式如图====所示。

Step3：直接预测最终的结果值。



如图===所示S-EMD-ANN算法设计思想是把经过EMD分解出来的子序列看做是不同时域和频域的特征子序列，然后在每个子序列上选择一些滞后变量作为子序列的特征直接预测最后的结果。如图===所示把每个时间点的所有分解值作为的特征向量，最后用这些特征向量组成变量集，直接去预测最终值。用每个子序列前个序列值组成一个维的变量去预测第项，并使用滑动窗口法预测余下各项。

3.3 实证分析研究

3.3.1 数据介绍

本文实证研究主要选择上海证券综合指数简称“上证指数”和标准普尔500指数。上证指数选择从1990年12月19日（基准日）-2019年4月2日的数据共6916个时间点序列数据，为了数据的一般性对于标准普尔500指数我们也选择相同日期的数据即1990年12月19日-2019年4月2日的数据共7123个时间点序列数据。考虑到要与线性模型做对比，并且对于股票指数的预测研究在实际中都是经常根据新的数据来更新参数所以大部分研究都做的是短时间内的预测如：Zhou(2019)[62]对几个不同地区的股指预测选择250个左右的作为测试集，Carnelossi Furlaneto(2017)[70]选择250（大约一年的交易日数）各值最为测试集。对于本文中每一个数据集我们都选择留下最后300个数据点即一年多的股票指数作为测试数据。数据情况描述见表===。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据名称 | 训练数据 | 测试数据 | 均值 | 方差 | 数据日期范围 |
| 上证指数 | 6616 | 300 | 1924.18 | 1071.92 | 1990.12.19-2019.04.02 |
| 标普500 | 6823 | 300 | 1253.42 | 601.77 | 1990.12.19-2019.04.02 |

3.3.2 基于EMD的数据分解

本小节主要是依据一定的数据变换规则，对数据进行预处理然后再进行分解。我们知道EMD分解算法最初提出是解决复杂的信号问题，信号序列一般都是高频、趋势性不强，然而如图==所示，本文处理的股票指数时间序列具有较强的趋势性并且频率较低。所以本文对数据进行一步差分预处理。文献[87,88]认为差分运算可以放大高频信息，显示高频信息引起的极值点，从而为提高频率分辨率提供了可能[88]。还有就是差分数据可以对数据进行缩放，在学习类算法中对数据缩放是加快学习速度的一个重要方法。

另外由第二章内容我们知道检验模态分解是有数据驱动的序列分解方式，再进一步讲经验模态分解是由序列的极值点驱动的序列分解方式，因为它主要是依据由极值点拟合的上下包络线分解的一种分解算法。所以我们可以知道当一个序列的极值点越多的话则上下包络线越接近真实的序列。所以我们比较了股票指数的原序列和差分后序列的极值点个数，认为本文中的两个股票指数序列差分后有会有更多的极值点。结果如表==所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据名称 |  | 极大值点个数 | 极小值点个数 | 极值点个数 |
| 上证指数 | 原序列 | 1666 | 1666 | 3332 |
|  | 差分后序列 | 2265 | 2264 | 4529 |
| 标普500 | 原序列 | 1827 | 1828 | 3655 |
|  | 差分后序列 | 2380 | 2379 | 4759 |

再者，我们可以从另外的一些评估标准来分析对原序列进行差分处理的一些优越性。文献[89,90]中从能量的角度给出了一个端点效应的评价指标。可以比较EMD分解前后的能量来评价分解的效果和端点效应的影响[89]。指标如公式===所示



其中为信号有效值，为信号序列，为信号的采样点数。

按式子===比较原信号有效值和各子序列（所有IMF和一个余项）有效值的总和，得到一个评价指标



其中为原信号有效值，为第个子序列的有效值，为IMF和余项的共个数。根据定义可知，的值越小，表示端点效应的影响越小[90]。

因为差分后的序列值要远小于原序列值，为了避免量纲造成的影响，所以对两个序列分别进行线性归一化处理后再计算。归一化方法如式子===所示。



其中为的最大值，为的最小值。

|  |  |  |
| --- | --- | --- |
| 数据名称 |  | 值 |
| 上证指数 | 原序列 | 0.0096728 |
|  | 差分后序列 | 0.0006603 |
| 标普500 | 原序列 | 0.1478892 |
|  | 差分后序列 | 0.0004604 |

综上我们介绍了对数据进行差分的一些好处：第一，差分序列可以加快模型的学习速度；第二，差分可以提高序列的频率使得序列的极值点显著的增加，可以使上下包络线更加接近需要分解的真实序列，再者说极值点个数的增加可以减弱端点效应的影响++++++；第三，差分后序列的值远小于没有差分后的序列的值，可以认为差分后端点效应对序列的影响较小。（因为在[89,90]并没有对评价标准进行严格的论证，这里只做参考。）

从图+++可以大致看出差分后的序列的极值点比原序列极值点要多。对差分序列分解如图===所示。对上证指数分解得到====个IMF和一个余项，对标普500指数分解得到===个IMF和一个余项。

3.3.3 模型性能评价指标

本文实证结果分别从总体预测精度。显著性检验等多准则对预测效果进行综合评价。

3.3.3.1 整体预测准确度评估指标

本研究采用平均绝对误差(MAE)、平均绝对百分误差(MAPE)和均方根误差(RMSE)三个常用的总体误差评价指标。他们的计算公式如下：









其中为观测值，为预测值，为样本数量。

为了定量评价不同模型的性能，定义指标RMSE、MAE和MAPE的改进百分比。



其中为RMSE、MAE或MAPE，和分别是模型1和模型2的评价指标值。

3.3.3.2 显著性检验指标

此外，本文还采用Diebold Mariano (DM)检验[90,91]和Wilcoxon Signed rank (WS)检验来验证了两种模型预测精度的等价性，即表明两种模型的测试数据的预测值之间是否存在统计学上显著的差异。

首先介绍一下DM检验，DM检验是一种非参数检验方法。

假设一对预测值的误差，。为测试样本数。模型预测结果的质量依据一些指定的损失函数来判断。假如两个模型预测效果相等则两个模型的的差的期望为0.即：



常用的损失函数，，等。并令



很自然的根据观察到的样本均值进行检验：



其中为样本的观测值，为第个模型的第项样本的预测值。

假设向前步（本文为向前一步预测也就是提前一天预测）预测。





则得到DM（Diebold-Mariano）统计量：



其中有式====和=====得到。在零假设下DM统计量服从渐进标准正态分布（证明）。即：



学者Harvey, Leybourne 和 Newbold[92]给出了更严谨的修正统计量，即DM检验的修正也称为HLN检验。



并有HLN服从自由度为的分布（证明见[92]）。即



现在常用的DM检验一般指DM检验的修正，即HLM检验。本文DM统计量的检验也是使用HLM检验值。

其次介绍一下WS检验。Wilcoxon signed-rank test（**威尔科克森符号秩检验**）也是一种非参数假设检验方法，它成对的检查2个数据集中的数据（即 paired difference test），判断 2个数据集是不是来自相同分布的总体。如果可以得到精确的有限样本检验，则可以根据零中值损失差的零假设检验检验观察到的损失差；即有的中位数为零。如果采样的损失差异具有对称的分布，则零假设与上述DM测试一致，表示相同的准确度[1]。WS统计检验值定义如下：







其中如公式====所示，为样本数量

3.3.4实证结果分析

实证研究数据依据3.3.1,3.3.2小节给出的数据，测试数据预测表现结果如表===和表===所示。对于模型自变量的选择，文献[62]分别选择了窗口大小为3,4,5的滞后变量进行建模分析。本文为了研究模型直接的差异分别选择了窗口大小为3-9的滞后变量（见表格==Lag=3,…,9）。模型结构是一个输入层、一个隐藏层、一个输出层，输出层节点为1即向前一步预测，其他网络结构设置见附录（）和代码===。

不同模型下上证指数的预测表现结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model |  | Lag | | | | | | | |
|  | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Mean |
| ARIMA | MAPE | 3.120 | | | | | | | |
|  | MAE | 89.90 | | | | | | | |
|  | RMSE | 117.2 | | | | | | | |
| ANN | MAPE (%) | 1.381 | 1.377 | 1.378 | 1.377 | 1.378 | 1.381 | 1.378 | 1.379 |
|  | MAE | 39.58 | 39.45 | 39.49 | 39.44 | 39.49 | 39.57 | 39.47 | 39.50 |
|  | RMSE | 52.82 | 52.65 | 52.70 | 52.62 | 52.68 | 52.81 | 52.65 | 52.70 |
|  | （1,T） | 0.071 | 0.072 | 0.074 | 0.076 | 0.071 | 0.074 | 0.074 | 0.073 |
| E-A | MAPE (%) | 0.687 | 0.673 | 0.690 | 0.688 | 0.695 | 0.696 | 0.696 | 0.689 |
|  | MAE | 19.77 | 19.39 | 19.88 | 19.83 | 20.06 | 20.09 | 20.07 | 19.87 |
|  | RMSE | 26.04 | 25.95 | 26.45 | 26.30 | 26.44 | 26.53 | 26.42 | 26.30 |
|  | （13,T） | 0.943 | 0.932 | 0.963 | 1.002 | 0.968 | 0.956 | 0.954 | 0.96 |
| S-E-A | MAPE (%) | 0.673 | 0.684 | 0.683 | 0.678 | 0.676 | 0.671 | 0.673 | 0.677 |
|  | MAE | 19.39 | 19.72 | 19.68 | 19.53 | 19.50 | 19.40 | 19.46 | 19.53 |
|  | RMSE | 25.29 | 25.59 | 25.52 | 25.91 | 26.06 | 26.01 | 26.27 | 25.81 |
|  | （1,T） | 0.082 | 0.085 | 0.089 | 0.094 | 0.095 | 0.096 | 0.097 | 0.091 |

不同模型下标普500指数的预测表现结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model |  | Lag | | | | | | | |
|  | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Mean |
| ARIMA | MAPE (%) | 1.334 | | | | | | | |
|  | MAE | 35.98 | | | | | | | |
|  | RMSE | 48.96 | | | | | | | |
| ANN | MAPE (%) | 1.039 | 1.036 | 1.039 | 1.038 | 1.035 | 1.038 | 1.037 | 1.038 |
|  | MAE | 28.06 | 27.98 | 28.05 | 28.02 | 27.96 | 28.04 | 28.02 | 28.02 |
|  | RMSE | 39.86 | 39.75 | 39.81 | 39.79 | 39.68 | 39.78 | 39.76 | 39.78 |
|  | （1,T） | 0.072 | 0.073 | 0.072 | 0.072 | 0.072 | 0.072 | 0.072 | 0.072 |
| E-A | MAPE (%) | 0.519 | 0. 528 | 0.529 | 0.525 | 0.527 | 0.528 | 0.530 | 0.527 |
|  | MAE | 14.02 | 14.27 | 14.28 | 14.19 | 14.24 | 14.27 | 14.32 | 14.23 |
|  | RMSE | 19.38 | 19.88 | 19.81 | 19.79 | 19.81 | 19.88 | 19.89 | 19.78 |
|  | （12,T） | 0.868 | 0.869 | 0.873 | 0.868 | 0.869 | 0.870 | 0.876 | 0.879 |
| S-E-A | MAPE (%) | 0.511 | 0.514 | 0.516 | 0.521 | 0.523 | 0.518 | 0.524 | 0.518 |
|  | MAE | 13.84 | 13.93 | 13.96 | 14.11 | 14.16 | 14.04 | 14.19 | 14.03 |
|  | RMSE | 19.21 | 19.10 | 19.12 | 19.28 | 19.45 | 19.40 | 19.64 | 19.31 |
|  | （1,T） | 0.083 | 0.086 | 0.087 | 0.089 | 0.091 | 0.094 | 0.095 | 0.089 |

E-A表示EMD-ANN，S-E-A表示S-EMD-ANN，（N，T）表示模型个数和训练模型所需时间。

表格==和==中的数据说明：ARIMA模型的平稳性和白噪声检验结果已经对应的自回归项数和移动平均项数见附录==，其中、值的确定依据AIC信息准则选取。ANN、EMD-ANN和S-EMD-ANN中每个窗口下（自变量Lag的值）对应的MAPE、MAE、RMSE、T评估指标值为20次实验的平均值。Mean列表示的是所有选取的窗口下的模型表现评估指标的平均值即140（20\*7）次实验的预测结果评估指标的均值。

从表===和===我们可以得出以下结论

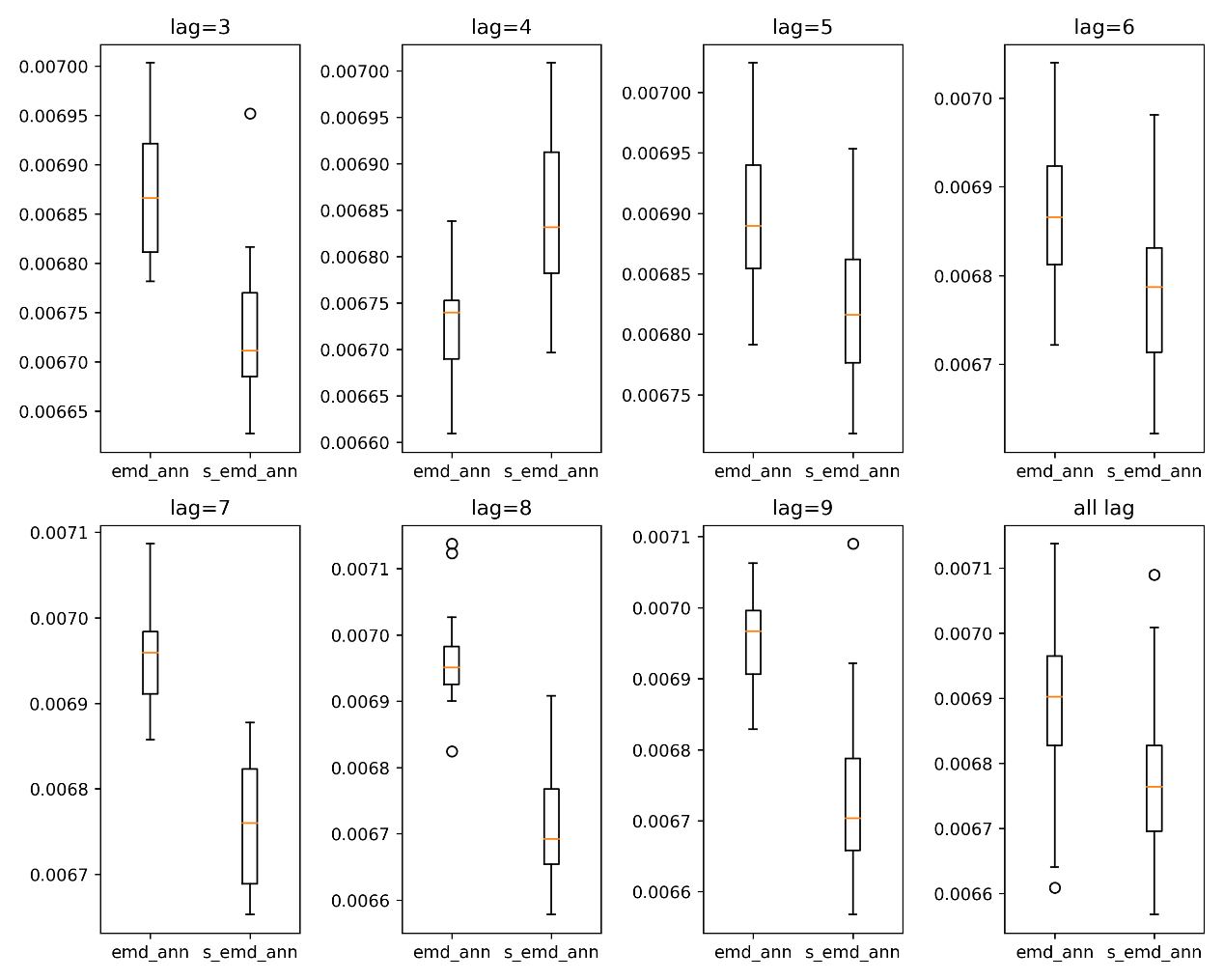
对于上证指数和标普500指数时间序列数据集来说，ANN、EMD-ANN、S-EMD-ANN模型的预测结果要好于ARIMA模型，表现评估指标以MAPE为例，在上证指数数据集上分别改进55.8%（(3.12-1.379)/3.12）、77.9%、78.3%，在标普500指数数据集上分别改进22.2%、60.5%

61.2%。

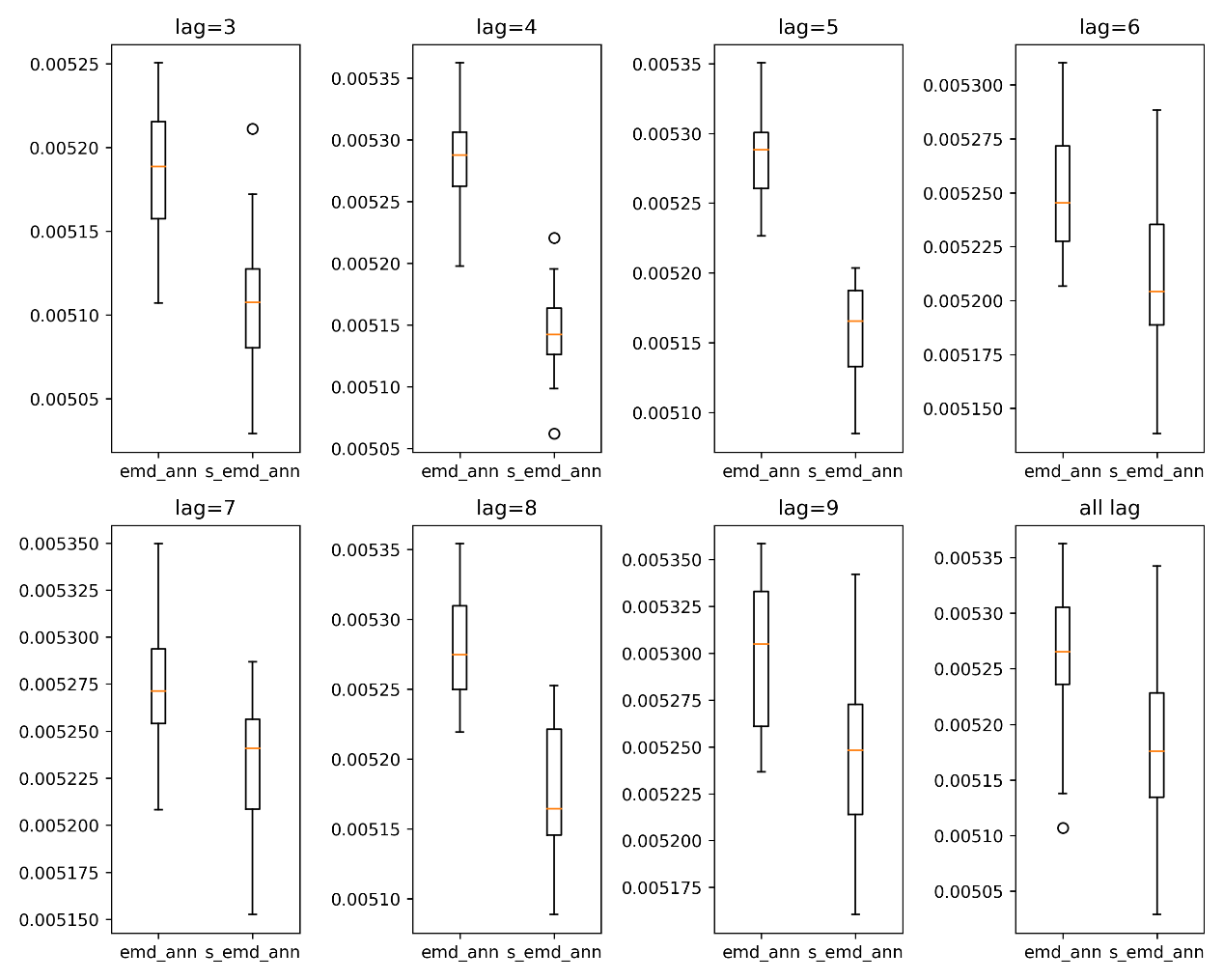
正如大多数文献[=====]研究的结论一样，在本文的数据集上EMD-ANN模型预测结果优于单独的ANN模型，表现评估指标以MAPE为例，在上证指数数据集上改进50.0%（(1.379-0.689)/1.379），在标普500指数数据集上改进49.2%。本文所提出的S-EMD-ANN也同样优于单独的ANN模型，表现评估指标以MAPE为例，在上证指数数据集上改进50.9%（(1.379-0.689)/1.379），在标普500指数数据集上改进50.1%，同时我们也发现但就结果来看本文所提出的EMD-ANN的改进模型S-EMD-ANN的表现结果略优于EMD-ANN的表现结果。

结论3也是本文主要的研究内容，从图===和图===我们可以看出除了在上证指数数据上窗口（Lag=4）等于4时S-EMD-ANN模型的MAPE值大于EMD-ANN模型的MAPE值，在其他选取的窗口下S-EMD-ANN模型的MAPE值都要小于EMD-ANN模型的MAPE值。从整体预测结果的MAPE评价指标上看S-EMD-ANN比EMD-ANN在上证指数和标普500指数分别改进了1.74%（(0.689-0.677)/0.689），1.71%，在MAE评价指标上分别改进了1.71%，1.41%，在RMSE评价指标数上分别改进了1.86%,2.38%。虽然本文提出的改进模型在精度上提上比较小，但是在效率上却又很有的提高，从表===中我们看出在上证指数数据上本文所提出的S-EMD-ANN只需要1个模型而一般的EMD-ANN模型需要13（与EMD分解得到的子序列数相等）模型,而且模型训练时间减少了90.5%，在标普500指数数据上S-EMD-ANN只需要1个模型而一般的EMD-ANN模型需要12个模型，模型训练时间减少了89.9%，单个模型还有一点优势在于像ANN等一些学习类模型往往需要大量的超参数需要要去搜索确定，模型数量的减少也大大的提高了超参数择优的效率。为了更进一步说明两个模型结果的差异，我们对两个模型的预测结果进行假设检验，利用DM检验和WX检验检验两个模型的结果差异性是否显著。假设检验结果如表===所示，从表中数据可以看出无论对于绝对百分比误差(APE)、绝对误差(AE)、还是方差(SE)，S-EMD-ANN和EMD-ANN两种模型在任何窗口下预测结果都没有显著的差异，假设检验的值都远大于0.05，所以我们可以认为S-EMD-ANN模型的预测结果和EMD-ANN模型的预测结果没有显著性的差异。从前面的表述也可以看出S-EMD-ANN模型的预测结果相较于EMD-ANN模型的预测结果在三种评估指标上都仅仅改进了1.5%左右。但从模型的训练时间和模型数量上看S-EMD-ANN模型远优于EMD-ANN模型。

上证指数MAPE



标普500指数MAPE



上证指数

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | S-E-A | Lag | | | | | | | |
| E-A | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Mean |
| MAPE | DM | 0.377 | 0.713 | 0.633 | 0.480 | 0.186 | 0.099 | 0.103 | 0.282 |
|  | WX | 0.554 | 0.390 | 0.708 | 0.824 | 0.344 | 0.166 | 0.349 | 0.511 |
| MAE | DM | 0.416 | 0.705 | 0.626 | 0.469 | 0.191 | 0.111 | 0.118 | 0.296 |
|  | WX | 0.623 | 0.394 | 0.707 | 0.846 | 0.338 | 0.180 | 0.358 | 0.552 |
| MSE | DM | 0.225 | 0.560 | 0.189 | 0.507 | 0.426 | 0.328 | 0.520 | 0.299 |
|  | WX | 0.421 | 0.520 | 0.885 | 0.964 | 0.526 | 0.305 | 0.761 | 0.403 |

标普500指数

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model |  | Lag | | | | | | | |
|  | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Mean |
| MAPE | DM | 0.435 | 0.247 | 0.244 | 0.630 | 0.527 | 0.307 | 0.435 | 0.281 |
|  | WX | 0.943 | 0.804 | 0.819 | 0.706 | 0.893 | 0.688 | 0.727 | 0.835 |
| MAE | DM | 0.476 | 0.281 | 0.267 | 0.674 | 0.580 | 0.339 | 0.477 | 0.313 |
|  | WX | 0.940 | 0.790 | 0.819 | 0.696 | 0.885 | 0.699 | 0.732 | 0.837 |
| MSE | DM | 0.626 | 0.135 | 0.161 | 0.274 | 0.343 | 0.275 | 0.482 | 0.220 |
|  | WX | 0.791 | 0.970 | 0.993 | 0.538 | 0.704 | 0.976 | 0.995 | 0.986 |

3.3.5 同类研究对比

为了进一步研究本文所提出模型的预测能力，本小节选取了一些文献并利用文献中的数据使用本文的预测模型，并与文献结果进行对比研究。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 上证 | Zhou (2019) [62] |  |  | 本文模型结果 |  |  |
| Lag | MAPE | MAE | RMSE | MAPE | MAE | RMSE |
| 3 | 0.0123 | 37.1778 | 61.5138 | 0.0074 | 21.83 | 31.45 |
| 4 | 0.0143 | 43.5513 | 66.7576 | 0.0076 | 22.42 | 32.11 |
| 5 | 0.0152 | 54.8499 | 89.9065 | 0.0075 | 22.35 | 32.74 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 标普500 | Zhou (2019) [62] |  |  | 本文结果 |  |  |
| Lag | MAPE | MAE | RMSE | MAPE | MAE | RMSE |
| 3 | 0.0105 | 13.0396 | 17.6591 | 0.00812 | 10.09 | 13.67 |
| 4 | 0.0122 | 15.1386 | 20.3978 | 0.00831 | 10.33 | 13.91 |
| 5 | 0.0130 | 16.1700 | 22.1277 | 0.00843 | 10.47 | 14.09 |

如表格===，为zhou(2019)[62]的研究结果其中表==为上证指数研究结果，数据为2012年1月4日到2016年12月30日，共1214个数据点，表中数据为最后242个数据点的预测结果评价指标值。从表中的结果评价指标来看本文所提出的模型的预测效果要远远优于文献中的预测结果，从窗口为3（文献中选择的最优窗口）时的结果来看MAPE指标改进了39.84%，MAE指标改进了41.28%，RMSE指标改进了48.87%。表==为标普500指数研究结果，数据为2007年1月3日到2011年12月30日，共1260个数据点，表==中数据为最后252个数据点的预测结果评价指标值。从表中的结果评价指标来看本文所提出的模型的预测效果要远远优于文献中的预测结果，从窗口为3（文献中选择的最优窗口）时的结果来看MAPE指标改进了22.67%，MAE指标改进了22.62%，RMSE指标改进了22.59%。同时从本文模型结果中我们也可以进一步证明文献[62]所选的窗口大小时，确实是当窗口等于3时模型的表现结果较好。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 本文模型 |
| 预测模型 | GARCH | ARIMA | RBF(ANN) | EMD-RBF(ANN) | S-E-A |
| MAE | 22.8945 | 20.1335 | 18.6743 | 15.8276 | 11.09 |
| RMSE | 30.2314 | 25.4589 | 24.7865 | 21.3992 | 13.75 |

表==为史（2015）[86]的研究结果，其研究的是对沪深300股指期货的预测，其研究结果如表虚线所示，数据为2012年5月22日到2014年9月9日，表中数据为后61个数据点的预测结果评价指标。文献[86]选择的窗口大小也为3但是没给出理由，为了进行对比研究表==中本文的模型也选择窗口大小为3时的结果。从表中的结果评价指标来看本文所提出的模型的预测效果要优于文献中的预测结果，其中 MAE指标改进了29.93%，RMSE指标改进了35.74%。

3.4 本章小结

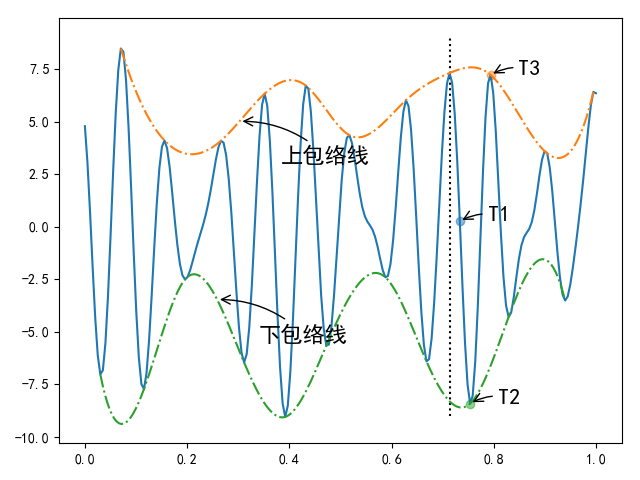
本章主要是提出了一种改进的EMD-ANN预测模型S-EMD-ANN预测模型，通过大量的实证研究并结合其它文献研究的研究结果我们可以得到以下结论。1本文所提出的S-EMD-ANN模型在模型的数量以及复杂程度是上都远小于EMD-ANN模型，并且S-EMD-ANN模型在训练是需要的时间远小于EMD-ANN训练需要的时间，所以S-EMD-ANN的效率更高。2 从两种模型的结果差异性检验上我们可以看出两种预测模型的结果没有显著性差异。但是在各个评价指标上我们可以看出S-EMD-ANN预测模型的结果要优于EMD-ANN预测模型的结果。所以我们可以得出结论S-EMD-ANN预测模型的预测结果要比EMD-ANN预测结果要好，但结果改进不是很大，从3.3.4章节我们知道在各个指标上的改进大约在1.5%左右。3 在3.3.5章节中在与其它学者的研究成果进行对比时，发现S-EMD-ANN模型的预测结果比其它学者的研究结果改进了20%-50%之间，说明本文所提出的模型在其他数据上也能有良好的表现结果。

4 基于经验模态分解与神经网络的自适应预测模型

4.1 EMD算法中的前视偏差分析

前视误差（Look-ahead bias）也有称为前瞻性偏差可以定义为对直到稍后的日期才可用信息的无意使用，换句话说，就是无意中用未来的数据预测未来。Mahfoud和Mani(1996)[93]指出，商业和公开的金融数据可能从一开始就含有前瞻性偏见，例如，与政府经济指标相关的数据在通过审查过程，可能会修改过去的数据[70]。

除了人为的向数据本身添加的前视偏差之外，使用一些技术手段对数据进行预处理时也有可能引入前视偏差。例如，数据标准化技术是非常流行的数据预处理方式也是处理时间序列的常用方式。Min-max和z-score标准化，可以说是两种最流行的标准化技术，利用一些数据的统计信息对数据进行变换处理，然而对于一个数据集来说最大值，最小值以及标准差都是整个数据的统计信息，特别是大部分研究在计算这些统计信息时是用了个数据集来计算的，那么对于我们要用来测试的结果来说，我们在预测之前就已经知道了它的最大值、最小值或者标准差的情况，这显然是引入了前视偏差。在经验模态分解中也是存在这样的前视性误差，而且这种前视误差影响比较大。不幸的是，在文献的研究工作中，这种性质的偏差经常被忽视，而且由于EMD算法的使用包含有前视偏差，在文献中包含前视偏差测试结果似乎很普遍[70]。这也使得诸多学者的研究结果都看起来很“漂亮”。



正如N. Sun (2018)[65] 、Q. Tan(2018)[66] 和X. Zhang(2015)[69]提出的问题，现有的大部分文献[57-64]对于数据集划分都存在一定的瑕疵，现有的文献在处理数据是都会忽略掉“前视性[70]”误差。如图表1所示，现在有大部分文献在划分数据是都是将整个时间序列进行经验模式分解，然后将分解后的序列划分为训练集和测试集。然而这样的序列分解方式考虑了未发生时间点的信息，所以不是真实的预测。如图表1所示对T1点的值分解使用了极值点T2和T3的值，而T2和T3对于T1来说属于未来的信息。所以如果我们要预测T2时刻的值那么对序列分解时只能用T2时刻之前的历史值。在文献[66]中学者Q. Tan将涉及未来信息的实证实验称为后视实验（Hindcast experiment），把不涉及未来信息的预测称为预测实验（forecast experiment）。在基于EMD的一系列混合模型的研究中后视实验是一种及其理想化的实验，是一种不考虑未来极值点对结果影响下的实验，然而绝大多数的研究者都停留在理想化的研究中，显然本文的第3章的实验也是理想化状态的实验。

4.2 EMD与ANN混合的自适应预测模型

4.2.1 基于EMD与ANN的自适应预测模型

为了解决基于EMD的混合模型的前视偏差问题，N. Sun (2018)[65] 、Q. Tan(2018)[66]、Dennis(2017)[70]和X. Zhang(2015)[69]学者都使用了一种自适应方法来构建基于EMD的一系类预测模型。其基本预测流程如图===所示。



图==所示算法的核心思想是：对于测试数据来说当预测T+1时刻的值时，在分解时只使用T+1时刻之前的数据，避免使用到未来的极值信息；然后再把T+1时刻的值添加到训练数据中再重新分解数据，重新训练模型然后再预测T+2时刻的值，依次类推直到所有的测试数据被预测出来。自适应的意思就是每增加一个值，都要重新训练模型的参数，使得模型能够学习到最新数据的特征。其流程如下：

Step1：把原始数据序列分解为训练集和测试集 ，并初始化k=1；

Step2：使用EMD算法把原序列分解成n个IMF和一个余项；

Step3：对每一子序列建立ANN模型，用前个序列值预测第项的值，具体变量预测方式如图====所示。

Step4：每个模型向前预测一步，预测结果求和得到测试数据集的第k个值的预测值；

Step5：如果k=m，则输出所有测试数据的预测值，否则，将加入得到新的训练数据，并令k=k+1重复Step2-Step5。

4.2.2 基于EMD与ANN自适应预测模型的改进

根据3.2章节的分析我们知道传统的EMD与ANN得混合模型，存在模型结构复杂，模型数量大，模型训练效率低等问题，这些问题在自适应模型中同样存在并且更加严重。因为对于测试数据的每个数据点都要重新分解数据和训练模型，所以可想而知传统的自适应混合模型的计算量是非常庞大的，在后面的实证结果中这个问题也充分的显现了出来。基于这些问题我们同样给出EMD与