学号 20164020

密级_____

东北大学本科毕业论文

基于改进随机森林的钒钛高炉 Ti 含量预测方法

学 院 名 称 ：信息科学与工程学院

专 业 名 称 ：自动化

学 生 姓 名 ：周科成

指 导 教 师 ：王显鹏 教授

二〇二〇年六月

郑 重 声 明

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名：  日期：2020.6.6

摘 要

钢铁工业作为装备制造业的上游工业，一直以来在国民经济中占据重要地位。高炉炼铁作为钢铁工业的基础工序之一，其技术优劣直接影响到钢铁制造以及装备制造的质量高低。高炉铁水钛含量不仅是生产某些特殊钢材需要控制的关键指标，也经常被用于表征炉温和高炉运行状态。因此，对铁水钛含量的预测方法是高炉冶炼科研中一项有着重要价值的课题，对控制铁水品质、预测表征高炉运行状态都有着重大的意义。同时，由于高炉内部反应的复杂性和黑盒性，如何建立准确的预测模型对高炉反应机理进行解释也成为行业中的一个难题。

本文对从高炉采集的实际数据进行预处理，使用预处理后的数据训练经过改进的随机森林模型算法，进而实现对铁水钛含量的回归预测。

本文首先细致分析了高炉冶炼工艺以及钒钛高炉内部反应原理，在此基础上分析了相关高炉参数的意义，并对数据进行预处理，包括数据归一化、缺失值填充、噪声数据清洗、特征选择以及耦合特征剔除等操作。

随后基于经典的随机森林模型理论建立了基础随机森林模型，并具体分析了各模型参数对模型性能可能造成的影响。在此基础上进一步分析了基础模型的局限性与不足之处；针对算法在采样环节和结合策略上的不足，提出了树间加权随机森林模型(Trees Weighted Bootstrap Random Forest Model)。针对模型参数选取的问题，使用改进粒子群优化算法对模型参数进行优化，并具体研究了不同改进策略对粒子群算法寻优能力的影响，进而形成了粒子群优化的树间加权随机森林模型(PSO Trees Weighted Bootstrap Random Forest Model)。

最后，对以上研究中得出的数据和模型进行实验，测试比较其性能优劣。经过实验仿真测试，本文改进的最佳算法在大幅波动的测试集上预测结果较为理想，预测趋势错误样本占比为 1.9%，30 次平均 RMSE 值为 0.05471，30 次 RMSE 方差为1.162×10^{-5}，10%误差带预测准确率为 72.6%。实验结果表明，本文提出的改进随机森林模型可以克服剧烈的炉况波动，具有较强的鲁棒性同时具备较高的预报精度和命中率。

关键字：铁水钛含量、数据预处理、随机森林、粒子群算法

ABSTRACT

As the upstream industry of equipment manufacturing, iron and steel industry has always occupied an important position in the national economy.As one of the basic processes of iron and steel industry, the technology of blast furnace ironmaking directly affects the quality of steel manufacturing and equipment manufacturing.The content of titanium in molten iron of blast furnace is not only a key index to control the production of some special steel products, but also is often used to characterize the furnace temperature and the operation state of blast furnace.Therefore, the prediction method of titanium content of molten iron is an important topic in the research of blast furnace smelting, which is of great significance for controlling the quality of molten iron and predicting and characterizing the running state of blast furnace.At the same time, due to the complexity and black box of the internal reaction of blast furnace, how to establish an accurate prediction model to explain the reaction mechanism of blast furnace has become a difficult problem in the industry.

In this paper, the actual data collected from the blast furnace are preprocessed, and the modified random forest model algorithm is trained with the pre-processed data, so as to realize the regression prediction of titanium content of molten iron.

First of all, the smelting process of blast furnace and the internal reaction principle of vanadium and titanium blast furnace are studied in detail, then the significance of relevant blast furnace parameters is analyzed, and the data are preprocessed, including data normalization, missing value filling, noise data cleaning, feature selection and coupling feature elimination.

For next, the basic random forest model is established based on the classical random forest model theory, and the influence of each model parameter on the performance of the model is analyzed.On this basis, the limitations and shortcomings of the basic model are further analyzed.In view of the shortcomings of the algorithm in sampling and combining strategies, a Trees Weighted Bootstrap Random Forest Model (TWB-RF) was proposed.Aiming at the problem of Model parameter selection, the improved particle swarm optimization (PSO) algorithm was used to optimize the Model

parameters, and the influence of different improvement strategies on the optimization ability of PSO was specifically studied, and then the PSO Trees Weighted Bootstrap Random Forest Model (PSOTWB-RF) was formed.

Finally, the data and model obtained in the above research are tested and compared.Through the experimental simulation test, the improved optimal algorithm in this paper has a better prediction result on the test set with large fluctuations, with the prediction trend error sample proportion of 1.9%, the average RMSE value of 30 times is 0.05471, the RMSE variance of 30 times is 1.162×10^{-5}, and the prediction accuracy of 10% error band is 72.6%.The experimental results show that the improved stochastic forest model proposed in this paper can overcome the severe furnace condition fluctuation, has strong robustness and has high prediction accuracy and hit ratio.

Keywords: titanium content, data preprocessing, random forest, particle swarm optimization

目 录

1 绪论

1.1 课题研究背景及意义

1.1.1 课题背景

钢铁制品的利用在人类文明中有着悠久的渊源，上可追溯到铁器时代，人类就懂得利用这种在自然界中广泛存在的材料。然而只有自人类进入工业革命以来，人们开始追求制造业的效率和质量，更多的冶炼技术开始普及，人类具有了大规模冶炼制造钢铁制品的能力，钢铁工业才开始走向成熟。直到今天，钢铁由于具有价格低廉、资源储备丰富、强度韧性优越，已经成为了广泛应用于建筑、交通、军工等行业不可或缺的基本材料，而且在未来较长的一段时间内都很难有任何一种材料可以撼动钢铁在工业中的地位。

在今天，人类进入了一个综合产业的时代，各种产业共同构成了人类复杂的社会体系和经济成分。其中实体经济与制造工业实力无疑是衡量一个国家综合国力的重要指标，而钢铁工业作为装备制造业的上游工业，承担着制造原材料的重任，在国民经济中占据着十分重要的地位。对比当今世界公认的发达国家，诸如美国、日本、德国等，没有哪一个不是钢铁强国和制造强国。

我国的钢铁工业从上世纪五六十年代开始起步，近四十年来取得了巨大的进展。其中钢铁产量连年占据世界第一。截至 2018 年，我国的生铁总产量达 77105 万吨，占世界总产量的 62.23%[1]。中国生铁总产量占世界总产量的比例如图 1.1 所示。

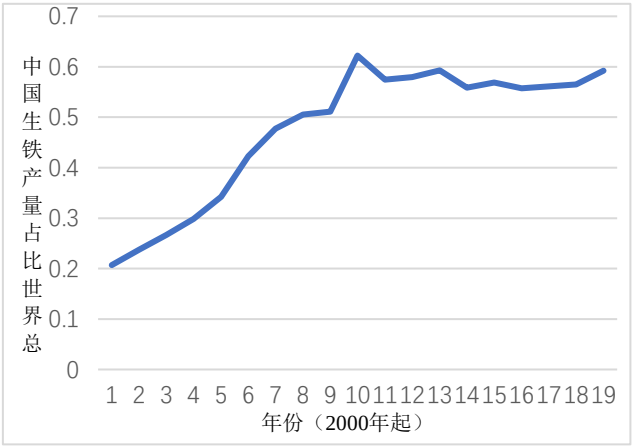

图 1.1 中国生铁年产量占世界的比值

Figure1.1 The ratio of China's annual iron production to that of the world

由上图可见，我国的生铁产量占比全球总体呈上升趋势，是名副其实的钢铁

大国。在产量逐年提高的同时，我国生产的钢铁质量、品种、节能指标等也在逐渐提高和丰富，正逐步向着钢铁强国迈进。

随着中国制造 2025 国家战略的提出，生产与制造的智能化日益被人们重视，提升到了前所未有的新高度[2]。在钢铁工业中，智能化更是体现在诸多方面，如何通过预检验精确控制产品质量？如何通过智能化手段排查潜在生产风险，从而减少杜绝生产事故？如何通过准确控制高炉运行状态（如高炉温度等炉况指标）实现节能减排？如何结合自动化与智能化提高生产效率与车间无人化程度？等等。而高炉作为整个钢铁制造的上游工序，同时也是名副其实的耗能大户，针对高炉的自动化技术、检测技术以及智能分析更是钢铁技术中的关键，对于减少炉况波动、提高铁水质量以及减少能源消耗等方面都有着积极的作用。

1.1.2 课题意义

高炉冶炼是钢铁工业的基础工序之一，高炉冶炼的质量高低直接影响了钢铁制造的优劣。因此研究高炉内部机理，对高炉内的动态过程进行建模分析显得尤为重要。高炉技术作为一项成熟复杂的金属冶炼技术，其内部大量的化学反应、各物质之间繁杂的耦合作用是阻碍学者们对其直接进行机理建模的主要因素[3]。

而铁水钛含量不仅是影响钢铁产品质量的关键指标，而且在钒钛高炉中经常用于表征炉温及高炉的运行状态，在帘线钢、电工钢、轴承钢等特殊钢材的生产中都需要对其含量进行严格的控制[4-7]。于是，掌握并精准预测高炉铁水中钛的百分比在判断高炉是否顺行、控制产品质量以及节约成本等方面都有着极为重大的意义。

自 20 世纪 80 年代以来，随着人工智能技术在工业界的大范围应用[8]，一些黑盒模型开始用来分析高炉的运行状态以及各项炉况指标。其中典型的有模糊控制[9]、专家系统[10]、支持向量机[11]以及神经网络等[12-15]。而随机森林算法作为一种数据驱动的模型学习算法[16]，被认为是代表集成学习最高水平之一的学习算法[8]。随机森林本质上运用 bagging 并行学习机制对若干个决策树子学习机进行集成学习，其中每个子学习机的训练数据来源于 bootstrap 重采样从原始数据集中抽取的若干样本[17-18]。理论和大量实验研究表明，随机森林解决了单个决策树算法精度不高、易过拟合等问题，有较好的泛化能力。[19-24]

因此，针对高炉内部反应的机理模型难以准确预测铁水钛含量的难题，使用

数据驱动的随机森林算法对高炉中的钛转移过程进行建模，并预测铁水中钛含量是十分具有科研价值和应用价值的一项课题。

1.2 高炉钛含量相关课题国内外研究现状

首先，在高炉自动化技术领域，国外团队在上世纪 50 年代就已开始研究。这些研究大多从冶炼机理出发，从而建立评价高炉运行状态的数学模型。十分著名的有，前苏联的拉姆配料模型；法国钢铁研究院的 RIST 操作线模型[27]以及 Wu 模型[28]；荷兰、卢森堡、比利时的 Ec 指数模型；日本的 Tc 模型，以及模糊理论等其他的高炉运行状态经验模型[9]。

其中，RIST 模型由法国 Rist 教授在重多复杂耦合的炉内氧化还原反应中提炼出其中的最主要体系，即 Fe-C-O 氧化还原体系，用操作线的形式给出三者反应的质量分数关系；Wu 模型定义了一种表征高炉温度的热力学指标，并给出了 Wu 的解析表达式，可以反应炉内 Si、Mn、P 的还原情况以及硅含量的变化趋势；Ec 指数模型将炉顶煤气成分和高炉风量风温参数作为输入，可以计算出用于表征高炉温度的状态指数 Ec；Tc 模型与 Ec 指数模型类似，其利用重油喷吹量、高炉风口温度与铁水温度之间的关系建立数学模型，从而根据已掌握的输入量信息预报出产铁水的温度以及炉内硅含量百分数等信息。

在这些经典的数学机理模型的基础上，20 世纪 80 年代以来，越来越多的数据驱动模型开始涌现，通过数据中蕴含的高炉内部潜在机理，这些模型可以很好的给出目标量的回归预报，这也是目前高炉模型技术的热点研究方向之一。

为了建立数据驱动模型以预测高炉产出铁水中钛的含量，需要弄清楚高炉中含钛化合物的存在形态、基本化学反应原理以及最终影响到钛含量的相关因素。针对高炉中的钛转移情况以及钛最终在铁水中含量的影响因素，已有很多学者专家致力研究并取得成果：

刘壮壮等人[4]针对脱钛铁水建立了 SiO2-TiO2-CaO-MgO-FeO-MnO 共存模型，并依此计算了 TiO2 的活度。喻爱国等人[25]通过测量实验得出了铁水中的 Ti 与 Si 含量成线性正相关关系，并据此提出了一种利用控制铁水中 Si 含量来间接控制 Ti 含量的方法。梁振华等人[26]系统地分析了高炉中钛元素的来源和还原机理，并提出了冶炼原料配比、硅含量、炉渣碱度和风温等因素可能对铁水中 Ti 含量有较大影响。雷家柳和薛正良[6]基于热力学理论具体计算分析了铁水中的 Si-Ti

平衡，并结合转炉脱钛和精炼增钛等具体工艺需求给出了控制钛质量分数的具体方法。范和华等人[7]具体分析了高炉铁水中钛的来源、影响因素及还原率，最后给出了在工业中控制铁水钛含量的具体方法。李胜杰等人[29]通过高炉数据和线性回归方法论证了铁水中 Ti 和 Si 的线性相关性，并建立了钛百分比与钛负荷、炉温之间的二元线性回归数学模型。杨志昌[9]全面地研究了透气性指数、料速、风量、风温等因素对铁水硅含量的影响。文光远等人[30]讨论了铁水中钛的比重对铁水黏度及冷却性能的影响并提出了冶炼中控制铁水成分的建议。

大量研究表明，铁水中硅与钛的含量成很好的正相关性，因此有关硅含量的预测研究对钛含量预测也具有参考价值。关于铁水中硅和钛含量的预测，有很多学者利用高炉采集数据建立学习模型，取得成果：

杨志昌[9]在分析了各种因素对铁水中硅含量影响的基础上，利用模糊理论构建了模糊系统用于对铁水硅含量的预测。马世文[5]等人在利用样条函数对数据进行等间距化预处理的基础上，构建 ARIMA 模型对铁水中钛含量进行预测。李胜杰等人[29]根据铁水 Ti 含量与钛负载、炉温之间有较好的线性回归关系，据此提出利用线性回归方法预测铁水的含钛量。温继勇[31]分析了风压、炉顶温度、料速等高炉操作参数与铁水硅含量之间皮尔逊相关系数的滞后性关系，由此得出每一操作参数对铁水中硅含量的最强相关性所对应的滞后时间。刘忻梅和石琳[32]提出了一种考虑时滞因素的 RBF 神经网络用以预测铁水硅含量，相较于不考虑时滞因素的相同算法有更高的预报准确率。李军朋[3]围绕实际钢厂高炉背景，开发了“高炉铁水预测系统”，同时提出了一种门控极限学习机模型，当硅含量波动较大时仍能实现准确预测。闫冲[33]采用了量子编码的遗传算法来优化神经网络模型中的权重，获得了更佳的铁水硅含量预测结果。祁鹏[34]建立了偏最小二乘回归模型用以预测铁水硅含量。徐循进[35]基于时差法提出了一种时序神经网络模型预测高炉炉温。蒋朝辉等人[15]构建了基于神经网络的并行集成学习模型实现了可信度-硅含量预测值的二维预报。孙冠群[11]在单个支持向量机的基础上，构建了基于 k-均值聚类的支持向量机群模型，对铁水硅含量进行预测。于涛等人[36]讨论了 CART 决策树和支持向量机模型在预测铁水硅含量时的准确性，并发现在相同条件下 CART 的回归精度要高于 SVM。罗世华和陈坤[37]采用偏态深度分类方法将 11 个特征的高炉数据划分成稳定类和离群类，并分别采用 Elman 神经

网络模型和逻辑回归模型对二者进行回归预测。庄田[38]分别用 Elman 神经网络和 Adaboost 模型对高炉数据进行硅含量回归预测和硅含量变化趋势分类分析，并采用模糊推理理论，通过设置适当的隶属度函数将二者融合，作为结果反馈。黄陈林[13]使用主成分分析对高炉数据进行降维处理而后采用 PSO 优化算法改进的极限学习机模型对铁水硅含量进行预测，提高了预测工作的效率和准确率。

对于随机森林及决策树算法的改进研究，大概可以分为决策树剪枝、随机森林中子学习机的加权投票和运用优化算法对随机森林参数的调整这几种研究方向。随机森林作为一种新颖而高效的学习算法，一直以来因其出色的可解释性、超参数较少等优点广受国内外学者的关注，在工业数据、图像处理以及金融信息等领域都有着广泛全面的应用：

Meng 等人[22]将 C4.5 决策树算法用于电网在线电压稳定性测试(Online Voltage Stability Assessment)任务中，并讨论了 C4.5 决策树相较于 CART 决策树在处理连续特征数据的效率和过拟合问题的优势；同时提到了决策树算法可以看作一种白箱模型(White Box Model)，具有较好的可解释性。张俊玉等人[39]基于传统 CART 决策树原理提出将关联规则与 CART 相结合的改进策略，用于学习煤电厂节能降耗过程中的关键变量及所起作用。Ganaie 等人[19]首先讨论了随机森林(Random Forest)和旋转森林(Rotate Forest)在分类任务中的精度，随后运用一个双边界的支持向量机算法(Twin Bounded SVM)对决策树的决策边界进行“细化”，尽管这种做法使得单颗决策树可以更为精准地找到分类问题的边界，但这也增大了整体算法过拟合的风险。廖明生[40]等人使用以 CART 决策树为子学习机的 Boosting 集成学习算法估算城市不透水层百分比。Wang 等人[41]提出了一种针对 C4.5 决策树的剪枝策略(Pruning Strategy)，并在一个分类数据上进行测试比较。许允之和王舒平[42]利用随机森林算法搭建徐州雾霾回归预测模型，实现预测徐州的 AQI 值；同时粗略讨论了决策树规模和特征子集容量对随机森林预测精度的影响。闫云凤[43]针对计算机视觉领域中小样本图像数据集问题，提出了一种级联随机森林回归模型，并提出利用概率函数控制决策树节点分裂以达到训练目的；理论与实验证明该方法相较于传统神经网络模型具有超参数较少，易收敛以及回归精度高等优点。Gamze 等人[20]将决策树原理用于搭建经济学模型，并采用敏感性分析(Sensitive Analysis)方法来决定输入决策树的候选特征集。Cuadrado 等人

[21]将决策树用于自适应软件测试技术(Computerized Adaptive Test)中，从而大幅度的提升了传统计算机软件测试的速度。Yong 等人[23]将决策树用于社区长期吸烟者戒烟可能性估计任务中，并提到用交叉验证(Cross Validation)方法来训练并测试学习模型。Gohari 等人[24]分别将决策树模型和 K-近邻学习模型(k-Nearest Neighbor Learning)用于多盘转子轴不平衡(Shaft Unbalance in multi-discs rotors)模型特征预测，通过实验得出，K-近邻学习的结果优于决策树模型。

随机森林作为一种优秀的集成学习模型，其较少的超参数和出色的解释性使之在同类学习算法中脱颖而出，这一点在文献[22、42、43]中均有论证。而在随机森林的改进方面也有很多学者做过相似研究：

李贞贵[44]在关于随机森林算法改进方面，证明了随机森林的收敛性，同时阐明了随机森林的泛化误差主要取决于两个因素：单颗决策树的决策精度和树之间的关联程度。因此针对随机森林决策精度的改进无非在于优化这两个方面。周天宁等人[45]比较了遗传算法和网格法两种算法对参数的寻优能力，分别优化了随机森林的决策树个数和特征子集个数两个超参数。温博文等人[46]尝试了网格法优化随机森林决策树个数和候选分裂特征子集个数两个参数。谢诗雨等人[47]提出了一种加权随机森林模型训练方法，并利用粒子群算法（PSO）对加权随机森林的剪枝阈值（在进行决策树剪枝的情况下）、决策树个数以及随机特征子集个数进行优化。马骊[48]提出决策树个数、特征子集个数以及每颗决策树形成叶节点的最小包含样本数对随机森林的性能有较大影响，并分别尝试了遗传算法、粒子群算法和鱼群算法来优化上述三个参数。马晓军等人[49]用粒子群算法优化了剪枝阈值、决策树个数、随机属性个数以及预测试样本数四个参数。王杰等人[50]利用粒子群算法优化了决策树的剪枝阈值、决策树个数。

综上所述，不难发现，尽管高炉铁水硅含量的预测任务已有大量的学者专家做出研究，但在铁水中钛含量的预测方面国内的研究并不十分充足。因此钛含量预测的相关研究在高炉数据领域需求迫切。此外，在预测铁水硅、钛元素含量时，国内大部分学者使用了改进的极限学习机等神经网络模型，然而神经网络的使用增大了调参的难度，往往使得结果精度不高；而随机森林恰好具有超参数较少的特点，可以较好地解决当前研究的不足。在改进随机森林算法的研究层面，很多学者选择粒子群算法对超参数进行优化。也有研究表明，粒子群算法在调整随机

森林算法参数方面相较于其他优化算法有着明显的优越性。

由以上分析可以看出，将改进的随机森林算法运用到高炉铁水钛含量预测的研究对于弥补当前研究不足以及提高实际工程中产品质量控制水平等方面都有着重要的意义。

1.3 论文主要研究内容

针对目前课题研究现状中的不足和问题，本文在了解学习了高炉冶炼工艺、钢铁制造工艺、高炉机理模型的基础上，主要侧重围绕高炉铁水 Ti 含量的预报模型上做出一些工作和思考。本文先从数据入手，详细分析了高炉工艺，对采集数据的物理意义进行研究分析，并利用数学方法分析数据间的耦合性与相关性，利用分析结论处理数据，建立数据驱动的随机森林模型。同时在经典模型的基础上，对模型参数与预报精度之间的关系进行研究、并提出了可行的改进方法。最后，为了提高学习模型的泛化能力，从优化算法领域对模型结构做出优化，避免因手工经验设定的参数对模型造成的过拟合风险，从而提高模型最终的预测精度。

本文的主要工作方法及研究内容如下：

（1） 在开始设计前，首先进行理论知识储备与补充，阅读机器学习、概率论与数理统计、高炉生产操作等相关书籍。做好相关知识储备后，阅读大量相关领域的文献，包括高炉铁水 Ti 元素转移过程分析、高炉数据预测模型研究、改进随机森林模型研究以及优化算法理论等等。在充分阅读文献后，总结学者专家研究思路，从而开拓自己的想法，找到模型改进的创新点。

（2） 数据预处理；首先进行缺失值填充，同时比较邻近学习方法与平均值法的特点，最终选择用平均值法填充数据中的缺失值。随后，利用对所有数据利用 3σ 准则进行异常值剔除。对剔除异常值后的数据计算每个特征与钛含量之间的相关性系数，从中选出相关性最大的若干个特征，进一步计算这些特征之间的相关系数分析数据耦合情况，综合给出可以使预测精度最大化的数据。

（3） 搭建随机森林模型；先搭建出基于 CART 决策树的随机森林模型，在没有加入剪枝策略、随机森林加权及参数优化算法的情况下对（2）中得到的数据进行预测评估，并记录此时预测的均方根误差。

（4） 进行模型改进尝试；在搭建好的经典随机森林模型基础上，添加决策树间加权投票策略。最后选定算法参数并利用 PSO 优化算法对模型调参，以实

现对随机森林结构的优化。同时利用不同的策略对 PSO 的优化能力进行改进比较，最终得出最高效的优化模型以及最准确的预测模型。

2 高炉铁水钛含量相关参数分析与数据预处理

对任何一个数据驱动的学习模型而言，数据的质量都将会对模型的预测能力造成影响，可以说数据的好坏直接决定了模型建立工作的成败。对于高炉这个及其复杂的工艺对象而言，其特征数据往往有着低相关、高耦合、大时滞等特点，加之从冶炼现场采集回的数据通常具有噪声干扰、传感器故障数据等情况。综合这些问题，导致得到的数据往往具有部分离群点、缺失数据、低质量特征等不利因素。如果不考虑这些问题而直接对原始数据进行建模，其效果明显是十分糟糕的。（实验分析详见章节 5.3）因此，在建立具体模型前先对原始数据进行深入的分析和适当的预处理是十分有必要的。

2.1 高炉重要参数分析

在高炉反应中，变量间的变化过程是一个复杂的非线性过程。同时，在高炉数据中，数据间经常存在繁杂的耦合关系。于是在数据预处理任务中，详细分析各特征数据的物理意义以及它们中间的相关性及耦合性对于最佳输入模型特征的选择来说是非常重要的步骤。

2.1.1 现代高炉冶炼工艺介绍

从总体上讲，高炉炼铁实际上是在由喷吹热风、重油或者天然气形成的高温炉体环境中，利用焦炭、煤等还原剂与各种铁矿石发生氧化还原反应，制成生铁以及高炉煤气、炉渣等副产品的过程。

而高炉作为高炉炼铁的核心设备，是由多种设备组合而成，配合完成这一项庞大复杂的任务。一般的高炉系统由以下几部分组成：高炉炉体、布料系统、风口系统、产物处理系统、检测系统、控制系统等[9]。高炉系统的物理结构如图 2.1 所示。

高炉炉体是冶炼的核心场所，整个炉体的结构又可细分为若干结构，其中与冶炼过程直接相关，密切影响着出产铁水成分及质量的结构是炉喉、炉身、炉腰、炉腹和炉缸。炉喉位于炉体的最上部，呈圆筒形。作为炉料的加入口和煤气的引出口，对高炉反应的进程和煤气成分起着控制和调节作用；炉身是炉料向下运动的主要区域，成梨形，炉身的形状设计至关重要，直接影响着料速和煤气气流的分布情况；炉腰位于炉身和炉腹之间，是高炉炉腔直径最大的部分，在此处会有粘稠的炉渣累积，增大了煤气和炉料运行的阻力，从而影响着高炉运行的状态和

氧化还原反应的进程；炉腹紧邻风口和喷吹口，温度最高可达 1400℃到 1800℃，是炉料融化和造渣的主要区域，也是炉内发生氧化还原反应的主要区域，呈倒锥台形；炉缸是燃料燃烧、渣铁反应的主要区域，出铁口和出渣口都设置在这里，各种料物的密集分布，使这里最易受到腐蚀和磨损，对出产生铁的质量和成分都有直接影响。

布料系统包括传送胶带机、给料卷扬机、称重漏斗、滤料漏斗等设备。负责传送、称重并向炉喉运送各种冶炼用原材料。一般包括焦炭、矿石、特殊辅料和煤粉等。对于钒钛高炉，还需要对铁矿石进行一定预处理操作。

风口系统一般包括气体加压设备、鼓风机、热风炉等。风口系统可以将来自煤气管网的气体加热到 1000℃至 1500℃，用于向风口鼓入大量热风以提供炉内反应所必须的温度。有时还会根据需要向高炉内喷吹适量重油或者天然气等其他高热值气体燃料助燃。风口系统的效能将直接决定影响铁水质量的两个关键特征变量：鼓风动能和风温。（具体介绍详见章节 2.1.3）这两个指标也常被用来衡量一个国家的高炉技术水平。

产物处理系统通常包括鱼雷送铁罐车、水渣处理设备、除尘清洗设备等。其中鱼雷罐车用于将出铁口出产的铁水运送到后续相关工序车间进行下一步加工处理；水渣处理设备用来处理炉渣，通过将炉渣颗粒化从而加以利用；除尘清洗设备可以处理高炉炉顶煤气，经过重力除尘管、洗涤塔和文氏管等设备后可以向用户提供煤气燃料。

大型钢铁工厂的高炉系统中一般配有检测系统和控制系统。高炉流程繁多工艺复杂，因而往往需要时刻监测原料、产物、燃料以及各项高炉指标的状态以便更好地帮助高炉操作者了解实时的炉况信息；而控制系统则用来收集相关的炉况信息，根据收集到的信息分析决策，通过现场总线实现对高炉各执行机构的动作控制。

通过以上对高炉系统各组成部分的介绍，高炉冶炼工艺的具体过程可以简单描述为：由布料系统向按层交替向炉喉输送矿石、焦炭和辅料（铁矿层和焦炭层交替排列），同时在高炉底部向炉内鼓如高热富氧风和其他助燃剂以达到炼铁所需温度。铁矿和焦炭等还原剂在热风的作用下在炉腹处发生氧化还原反应，铁矿中的高价铁被还原成铁元素，呈液态到达位于炉缸的出铁口；同时，铁矿中的脉

石与石灰石等助燃剂结合形成炉渣，从出渣口排除。这个过程产生的煤气比重较轻，经过引流至炉顶，通过烟道产出[33]。高炉冶炼工艺的系统结构如图 2.2 所示。

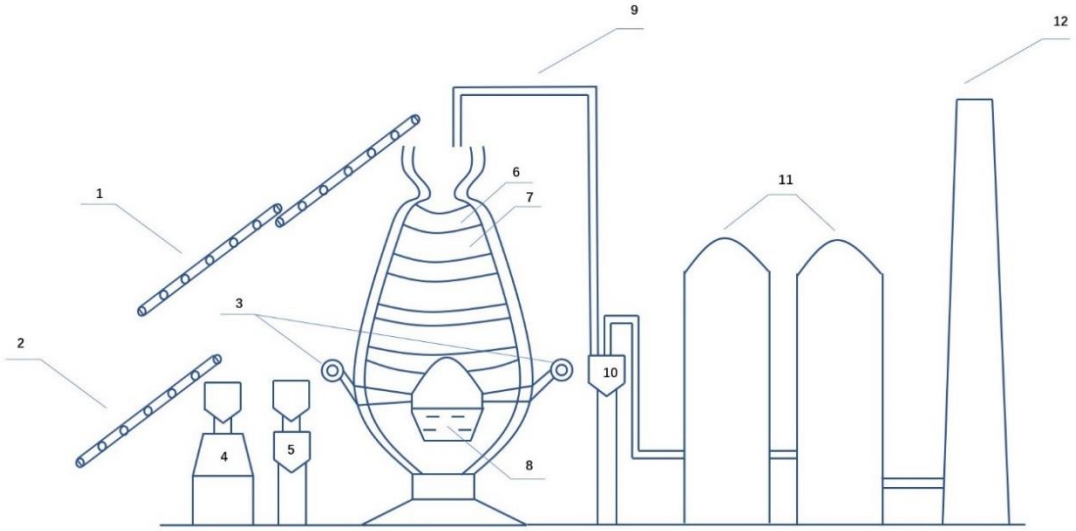

图 2.1 高炉系统物理结构示意图

Figure2.1 Schematic diagram of physical structure of blast furnace system

1-矿石输送皮带机；2-焦粉输送皮带机；3-环炉热风管；4-原煤仓；5-煤粉仓；6-焦炭层；7-铁矿层；8-铁水；9-烟道；10-重力除尘器；11-热风炉；12-热风炉废气烟囱

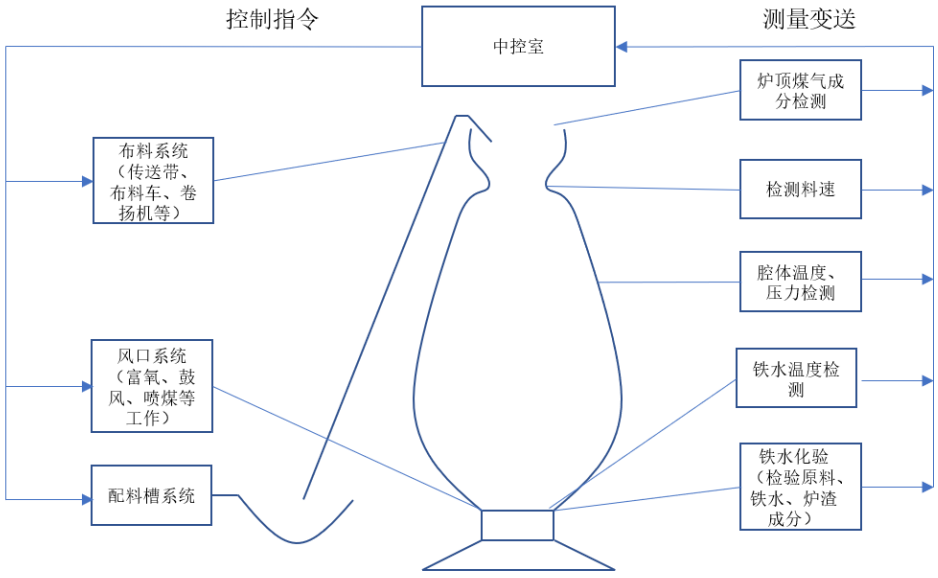

图 2.2 高炉工艺系统结构框图

Figure2.2 Structural block diagram of blast furnace process

2.1.2 钒钛高炉中含钛化合物的还原分析

在化学层面上，高炉的炉内反应是十分复杂的，有上百个化学反应在同时进行。尽管完全弄清楚这些化学反应之间的关系并对其中的系统进行建模定量分析十分困难，但对于分析数据之间的耦合性而言，仍有必要分析其中的主要反应机理。本节从化学层面上分析钒钛高炉中钛元素的还原路径，并从热化学角度分析可能与铁水中 Ti 含量直接相关的因素。

在钒钛高炉中，选用的铁矿石为钒钛磁铁矿，其中的高价可还原元素主要以铁（Fe）、钒（V）、钛（Ti）为主。而钒钛磁铁矿中的 Ti 元素主要以二氧化钛(TiO_2)形式存在。比较 Fe、V、Ti 发生还原反应的自由能与温度的关系[51]发现，Ti 的还原反应相较于另两种元素更难发生，即含钛化合物的夺取氧的能力较弱，因此很难像铁的还原一样按照逐级还原进行，铁元素的典型逐级还原式为：

$$Fe_2O_3 \xrightarrow{C、H_2、CO} Fe_3O_4 \xrightarrow{C、H_2、CO} FeO \xrightarrow{C、H_2、CO} [Fe] \tag{2.1}$$

在这个过程中铁元素发生的主要化学反应有：

C 对Fe_2O_3的还原作用：

$$3Fe_2O_3+C=2Fe_3O_4+CO \tag{2.2}$$

$$Fe_3O_4+C=3FeO+CO \tag{2.3}$$

$$FeO+C=Fe+CO \tag{2.4}$$

CO 对Fe_2O_3的还原作用：

$$3Fe_2O_3+CO=2Fe_3O_4+CO_2 \tag{2.5}$$

$$Fe_3O_4+CO=3FeO+CO_2 \tag{2.6}$$

$$FeO+CO=Fe+CO_2 \tag{2.7}$$

H_2对Fe_2O_3的还原作用：

$$3Fe_2O_3+H_2=2Fe_3O_4+H_2O \tag{2.8}$$

$$Fe_3O_4+H_2=3FeO+H_2O \tag{2.9}$$

$$FeO+H_2=Fe+H_2O \tag{2.10}$$

有研究表明TiO_2的还原路径如下：

$$TiO_2 \xrightarrow{C、S、Si} Ti_3O_2 \xrightarrow{C、S、Si} TiC_{0.07}O_{0.33} \xrightarrow{C、S、Si} TiC_xO_y \xrightarrow{C、S、Si} [Ti] \tag{2.11}$$

与 Fe 不同，TiO_2的还原产物中并不都是简单的低价氧化物，而是出现了一

些低价复合中间产物。值得说明的是，这里的 0 价[Ti]主要有两种存在形式：游离的 Ti、与 N、C 形成的结合物 TiN、TiC。

由于H_2和 CO 的还原能力更强，这两种还原剂优先还原氧化性更强的 Fe 和 V。而多数的TiO_2直接被还原性更弱的 C、Si 和N_2还原。这个过程中发生的总化学方程式有：

$$TiO_2+2C=[Ti]+2CO \tag{2.12}$$

$$TiO_2+3C=TiC+2CO \tag{2.13}$$

$$TiO_2+2C+0.5N_2=TiN+2CO \tag{2.14}$$

$$TiO_2+[Ti]=[Ti]+SiO_2 \tag{2.15}$$

以上反应生成的 *TiN*、*TiC* 以固溶体形态存在于渣相中，生成的[*Ti*]部分进入铁相中，而存在于这两个体系中的 0 价 *Ti* 有如下形式的动态平衡：

$$[Ti]+C=TiC_{(s)} \tag{2.16}$$

$$[Ti]+0.5N_2=TiN_{(t)} \tag{2.17}$$

因此，炉内铁水的 *Ti* 溶解量 *Ti*(%)可由式(2.16)、式(2.17)表达。

由以上分析可以看出，影响铁水中 *Ti* 百分数的因素应该有渣系的状态、炉腹中TiO_2含量、还原剂含量（如 Si、C、S、N_2等）和反应温度。

由大量高炉实验归纳显示，铁水中的 Ti(%)主要取决于炉渣碱度、渣中TiO_2含量、反应温度和铁水中的 Si 含量。

由于可逆化学反应发生的条件因素限制[51]，炉渣碱度对 Ti 百分数的影响具有双面性：当TiO_2含量处于较高值时，[Ti]随碱度增加而减少；而当TiO_2含量处于较低值时，[Ti]随碱度增加而增加；同时，由于炉渣碱度与TiO_2含量具有一定相关性，因此在碱度与TiO_2含量二者都不固定时，[Ti]含量与这两者的相关度并不高。（章节 2.2.4 中的数据可以印证这一点）温度对[Ti]含量的影响体现在温度的提高可以提高活化能，使铁系、渣系中的平衡反应更容易向正向移动。因此在一定温度范围内，[Ti]含量随温度的升高而增加。与温度类似，还原剂对[Ti]含量的影响同样体现在促进了可逆反应的反应程度，其中最具有代表性的就是铁水中的 Si 百分数对 Ti 百分数的影响，很多研究表明，二者具有非常好的正相关性。根据本文数据所进行的实验可知，二者的皮尔逊相关系数达到了 0.8191；同时，渣系中的还原剂 C、S 含量与 Ti 含量之间的相关系数分别为 0.5814 和 0.7083，

（详见章节 2.2.4）可见还原剂对铁水中 Ti 含量的影响程度是非常大的。在机理分析之后合理挖掘并利用这些高质量特征会对数据的处理有着启发式的引导作用。

2.1.3 高炉参数意义分析

在掌握了必备的高炉工艺知识和高炉反应机理后，本节利用工艺和原理对所有备选特征进行初步筛选。

实验用于钛含量预报的数据共有采集自高炉炉体和铁水的 61 维特征变量，由于特征的数量过多且存在复杂的耦合关系，所以必须对特征初步进行筛选。根据章节 2.1.2，铁水中的钛百分数主要受炉渣碱性情况、反应温度和还原剂三个大方面影响。现分别列举出可能影响到这三个方面的 40 个特征如表 2.1（尽管有些特征可能不止影响一个方面，但都按照直接影响的方面来统计），并依次介绍。

表 2.1 不同特征影响 Ti 含量的方面

Table2.1 The influence of different characteristics on Ti content

影响方面	特征变量
炉渣碱度	R2、R3、TiO_2、镁铝比
反应温度	风温、实际风速、鼓风动能、透气性指数、富氧量、喷吹速率、煤气利用率、炉缸温度、炉缸中心温度、炉腰温度、炉身下二段温度、炉喉温度、炉顶温度、铁水温度、探尺差
还原剂	M10、M40、CSR、CRI、Ad、St、SiO_2、Si、C、S、喷吹煤 Vdaf/Ad/St、炉顶煤气 $CO/H_2/CO_2$、炉腹煤气量指数、焦炭负荷、焦比、煤比、燃料比

（1） 可能影响炉渣碱度的特征：

（a） R2：钙硅比。需测量炉渣中CaO与SiO_2的百分含量。炉渣中的CaO计数可以反应炉渣中钙离子浓度从而影响炉渣碱度情况。数据中的计算公式为：

$$R2=\frac{CaO(\%)}{SiO_2(\%)} \tag{2.18}$$

（b） R3：钙镁硅比。需测量炉渣中CaO、MgO与SiO_2的百分含量。与 R2 类似，钙镁铝可以反应炉渣中钙离子镁离子铝离子浓度从而影响碱度情况。数据中的计算公式为：

$$R3=\frac{CaO(\%)+MgO(\%)}{SiO_2(\%)} \tag{2.19}$$

（c） TiO_2：二氧化钛百分含量。需测量炉渣中 TiO2 的百分计数。在章节 2.1.2 已经简要分析过，二氧化钛的浓度会对炉渣碱度造成影响。

（d） 镁铝比：指炉渣中氧化镁与三氧化二铝百分含量的比值。同样可以反应炉渣中的镁离子与铝离子的浓度，表征炉渣碱度的情况。

（2） 可能影响反应温度的特征：

（a） 风温：风温是高炉数据的重要参数之一，代表从风口鼓入炉缸上部的热风温度。提高风温是现代高炉技术追求的关键技术之一，风温的提高可以对高炉反应带来多方面的优势：第一，风温的提高可以直接影响炉缸中进行的化学反应温度，由章节 2.1.2 的 TiO_2还原机理可知，反应温度的提高可以促进含钛化合物的还原，使得铁水中钛含量增加，这对于有些特殊钢材是有必要的；（尽管大多数的场合需要低钛铁水）第二，适当提高风温可以促进焦炭燃烧，如果焦炭燃烧更充分，就可以大大节省每一炉焦炭的用量，从而达到节约成本的目的；第三，由于温度更高，焦炭燃烧更充分，因此降低了焦比以及单位铁水产生的煤气量，使得因煤气带走的热量减少，反应更加充分。热风温度从风口测量得出，由于此测量点距离炉缸较近，因此可以近似表示炉缸内的反应温度。

（b） 实际风速：实际风速由标准风速、风压和风温计算得出，可以作为代表炉内反应激烈程度的指标。数据中的计算公式为：

$$Vs=\frac{V_d\times\left(\frac{0.101325}{273}\right)\times(10+0.101325)\times(273+T)}{P} \tag{2.20}$$

式中：$Vs(m/s)$代表实际风速、$V_d(m/s)$代表标准风速、$T(℃)$代表风温、$P(kPa)$代表热风压力。

（c） 鼓风动能：由风压和实际风速计算得出，是高炉风口系统的重要指标之一，常用的计算公式为：

$$E=\frac{0.5nPQV_s^2}{g} \tag{2.21}$$

式中：$E(kg\cdot m/s)$代表鼓风动能、$V_s(m/s)$代表实际风速、$g(m/s^2)$代表重力加速度、$P(kPa)$代表热风压力、$Q(m^3/s)$代表热风量、n 代表风口个数。

（d） 透气性指数：用来表示高炉在冶炼过程中接受风量的情况，也可以间接表征反应的进行程度，经常用来判断高炉是否顺行。当出现崩料、悬料、炉温骤降等异常时透气性指数可能会体现出现异常从而完成预警。数据中的计算公式为：

$$Tz=\frac{Q}{p_1-p_2} \tag{2.22}$$

式中：Tz代表透气性指数、$p_1(kPa)$代表热风压力、$p_2(kPa)$代表炉顶煤气压力、$Q(m^3/s)$代表热风量。

（e） 富氧量：在数据中指每单位鼓入炉缸的气体中，纯氧的体积。富氧量的提高可以使焦炭燃烧更充分，使得反应更剧烈。

Ti 百分数与富氧量的散点分布情况如图 2.3。

（f） 喷吹速率：指由风口喷吹入炉缸的重油或天然气等助燃剂的速率，单位为：m^3/s。

Ti 百分数与喷吹速率的散点分布情况如图 2.4。

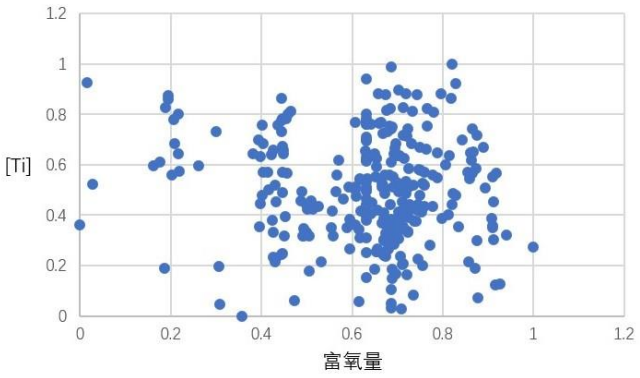

图 2.3 [Ti]-富氧量分布散点图
Figure2.3[Ti] - Oxygen value scatter plot

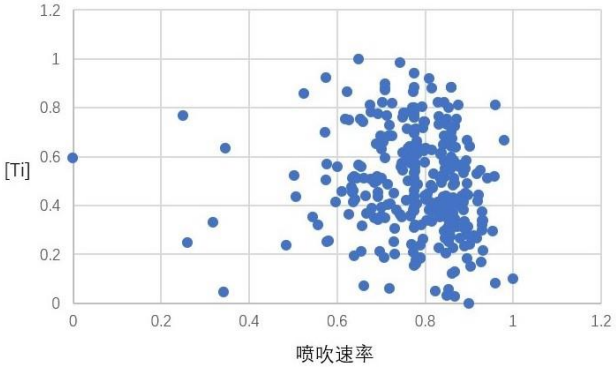

图 2.4 [Ti]-喷吹速率分布散点图

Figure2.4 [Ti] - Injection rate scatter plot

（g） 煤气利用率：指炉顶煤气中二氧化碳气体的百分计数占煤气重要成分的比率。数据中的计算公式为：

$$k=\frac{CO_2(\%)}{CO_2(\%)+CO(\%)} \tag{2.23}$$

Ti 百分数与煤气利用率的散点分布情况如图 2.5：

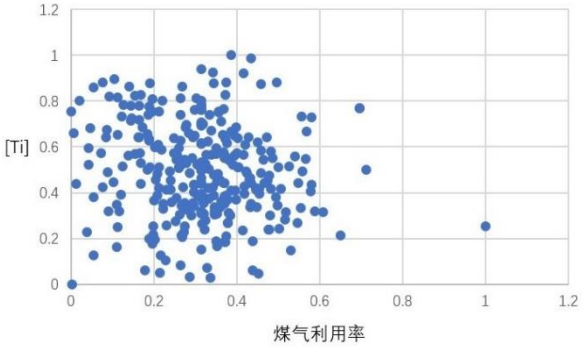

图 2.5 [Ti]-煤气利用率分布散点图

Figure2.5 [Ti] - Gas efficiency scatter plot

（h） 炉缸温度&炉喉温度&炉缸中心温度&炉腰温度&炉顶温度&炉身下二段温度&铁水温度：在数据中测量了六个不同位置的高炉温度以及铁水温度。高炉过程中的温度具有大滞后性的特点，有很多文献都对高炉温度时间滞后的特性做了细致地研究，本文对滞后性不做重点分析，采取对各时间序列的温度采样值取平均的方法来计算得到各位置的温度值。Ti 百分数与炉内各温度的散点分布情况如图 2.6：

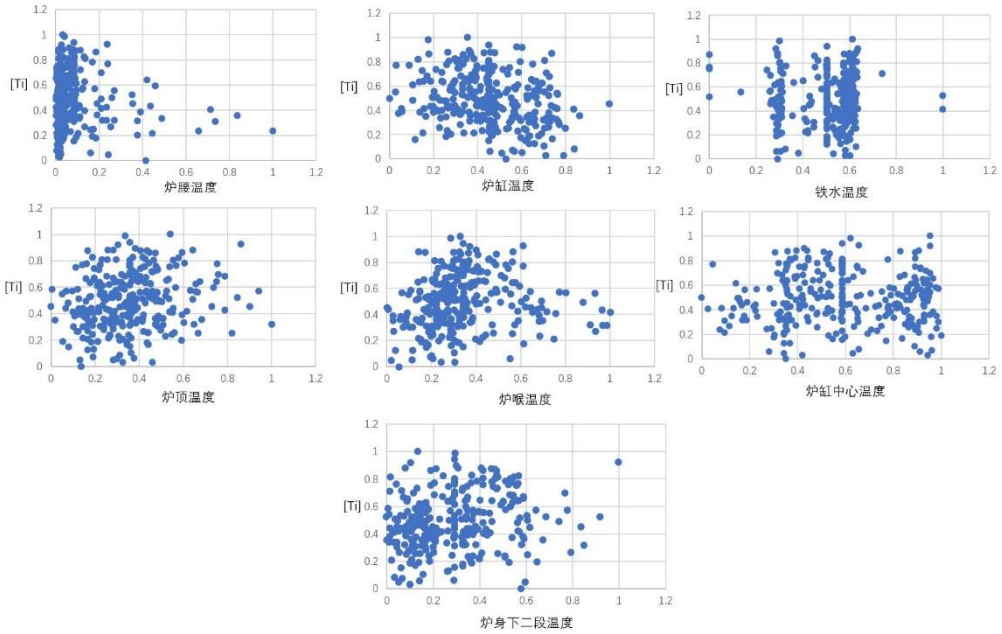

图 2.6 [Ti]-炉内各部分温度分布散点图

Figure2.6 [Ti] - Temperature of parts in furnace scatter plot

（i）探尺差：高炉探尺是一种常见高炉测量设备。探尺一般放在高炉顶部，向下伸出若干个分尺，用来测量炉料距高炉顶端的距离，以实时监测料位的下降情况。探尺差，就是指安装在不同位置的探尺测量值的差值。用于反应高炉目前是否有塌料异常，也可以显示高炉反应是否均匀、平稳。数据中采用如下计算方法：先筛选出三个分探尺都有返回值的时刻，然后计算这些时刻的极差，最后再对应出铁的时刻将这些数据取平均值。

（3） 可能影响还原剂的特征：

（a） M10＆M40：焦炭的两个常用硬度指标，分别指高炉用焦炭的耐磨强度指标和抗碎指标。

（b） CSR&CRI：焦炭的两个常用热性能指标，分别指焦炭反应性和反应后强度。焦炭的热性能指标（b）和硬度指标（a）会因存放时间、温度、气压等因素产生一些变化。因此，在给料前对焦炭性能的采样测试可以直接显示炉内反应的焦炭状态情况。

（c） Ad＆St：焦炭工分指标，反应焦炭的成分构成。分别表示焦炭的灰分和硫分，用含杂质量和含硫量来计算。

（d） SiO_2 ＆Si＆C＆S：铁水中各物质成分含量的百分计数。统计这些还

原剂在铁水中的计数可以表征发生氧化还原反应时还原剂的参与情况。Ti 百分数与铁水中各物质百分数的散点分布情况如图 2.7。

（e） 喷吹煤 Vdaf & Ad& St：指从风口喷吹入高炉的煤炭质量指标。分别代表煤炭的挥发性、灰分和硫分百分含量。作为助燃剂的煤炭中也含有大量的焦炭和单质硫。在影响反应温度的同时也为反应提供了还原剂。

（f） 炉顶煤气 $CO/H_2/CO_2$：分别测量炉顶煤气中的 CO、H_2、CO_2成分的百分含量。可以表示在炉缸反应中还原剂的利用情况以及反应的激烈程度。Ti 百分数与炉顶煤气成分的散点分布情况如图 2.8。

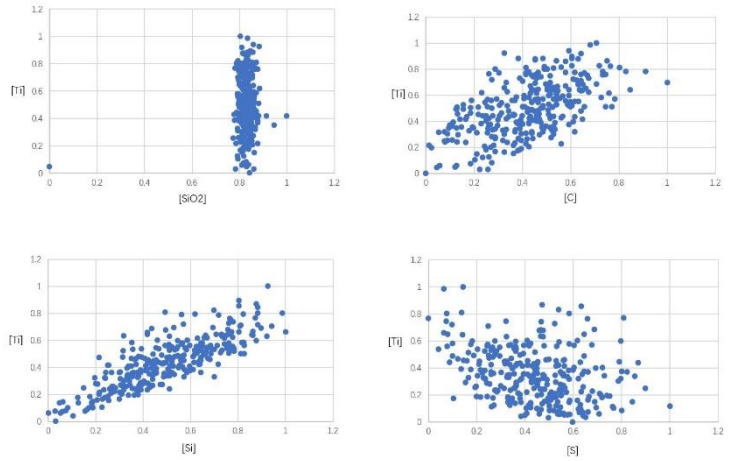

图 2.7 [Ti]-铁水组分百分数分布散点图

Figure2.7 [Ti] - Composition of the molten iron scatter plot

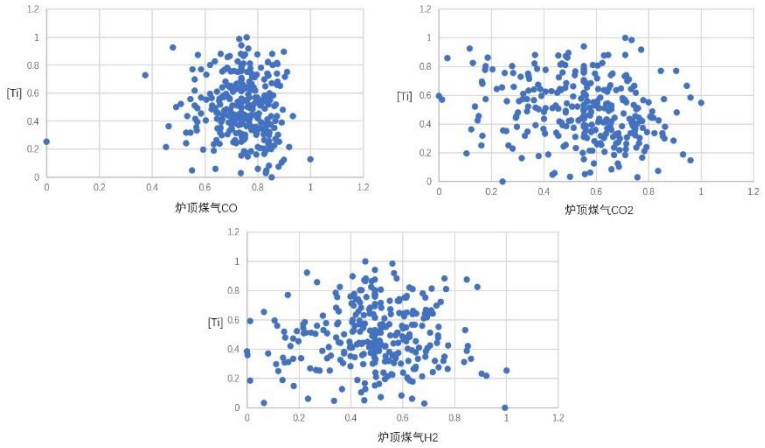

图 2.8 [Ti]-炉顶煤气成分百分数分布散点图

Figure2.8 [Ti] - The gas composition scatter plot

（g） 炉腹煤气量指数：用来表示炉腹的煤气浓度，可以表示炉缸反应的进行情况。

（h） 焦炭负荷：一般指在装料时，矿石量与焦炭量的比值。焦碳负荷直接近似反映了氧化剂与还原剂之间的比例，通过对焦碳负荷的控制，可以实现对氧化还原的进程的掌握。

（i） 焦比：出铁对应时间段的焦炭质量与本次铁水质量的比值。又是一个重要的高炉指标，不仅表征还原剂量的多少，还是一个反应高炉节能性能的指标。

（j） 煤比：出铁对应时间段的喷煤量与本次出铁铁水质量的的比值。本数据中的计算公式为：

$$q=Vp\times\frac{T}{M} \tag{2.24}$$

式中：q 代表煤比、Vp 代表喷吹速率、T 代表本次受铁时间、M 代表本次出铁量。

（k） 燃料比：燃料比综合了各种形式的燃料的质量与出铁质量的比值。在本文中的燃料来源只有焦炭和喷吹煤两种，因此数据中的燃料比计算方法为将焦比和煤比相加。Ti 百分数与燃料比的散点分布情况如图 2.9：

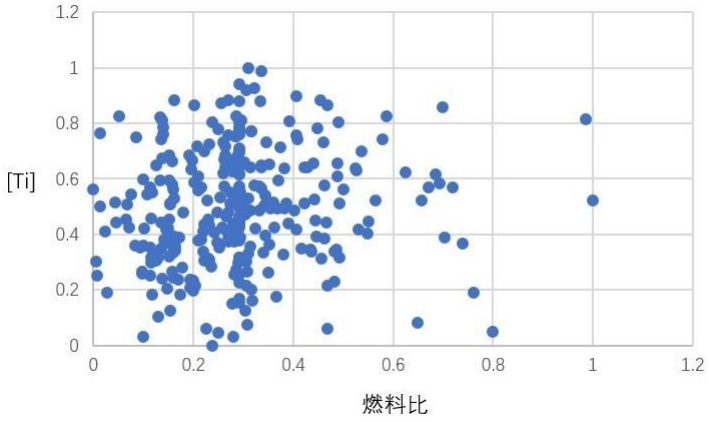

图 2.9 [Ti]-燃料比分布散点图

Figure2.9 [Ti] - Fuel rate scatter plot

需要指出的是，以上的数据散点图全部是经过归一化、缺失值填充和离群点剔除操作之后得到的数据散点，具体操作方法详见章节 2.2。

由这些散点图可以清楚地观察根据机理筛选的各属性与铁水钛含量之间的相关性程度，其中铁水中的其他成分明显与 Ti 含量有非常强的线性关系。为章

节 2.2.4 进一步用统计学方法进行特征选择提供了一种参考。

2.2 高炉数据的预处理

经过上一节对高炉工艺和相关高炉参数的物理意义分析之后，根据机理初步选定了 40 个备选参数特征，见表 2.1。然而这些数据还存在着各方面的问题，导致这样的数据无法直接用来训练模型，必须在章节 2.1 的基础上做进一步的数据预处理。总结数据中仍存在的问题及对应解决方案如下：

（1） 各特征数据的量纲不同，数据区间范围不统一，因此需要对所有特征归一化处理。章节 2.2.1 将介绍数据归一化的几种方法以及对本文数据的归一化处理。

（2） 本文数据采集自高炉车间现场传感器，由于传感器故障、线路异常、工作人员记录数据遗漏等不可抗因素，导致数据集中有相当数量的缺失值。直接对缺失值补零处理显然是不合理的，因此需要一种合理填充缺失值的策略。章节 2.2.2 提出两种填补缺失值的策略，并通过简单的数学推导分析比较两种策略的优劣。

（3） 由于高炉生产过程中的参数存在较大噪声，数据中可能存在部分离群点，对实验结果造成干扰，因此需要用合适的策略对数据进行离群点剔除。章节 2.2.3 中介绍了两种常见的数据离群点清洗方法，选择了其中一种对钛含量数据进行离群点检查。

（4） 章节 2.1 初步筛选的特征有 40 个，而且很有可能仍然存在低相关性的糟糕特征。章节 2.2.4 将从统计学角度介绍两种十分经典的量化相关性的方法。并利用数据的统计学指标进一步分析出数据之间的耦合情况，最终确定模型的输入特征。

2.2.1 数据归一化处理

归一化有时又称作标准化，数据的归一化对大部分数据挖掘、机器学习建模、最优化任务而言都是十分必要的工作，已有实验数据表明：对大多数学习模型而言，使用归一化之后的数据对比不使用归一化的数据来训练模型，预测精度有着明显地提高[52]。其原因在于以下两个方面：

第一，增加了特征之间的可比性。对原始数据而言，数据各特征的量纲不同、区间范围不同，甚至存在几个数量级的差异。因此难以直接对各特征进行比较。

而经过归一化之后的数据，各特征将钳位在同一标准区间内，避免了量纲不同对模型的干扰，方便对特征进行直接比较，增加了可比性。

第二，对于权重驱动的学习模型（如神经网络、逻辑回归等）以及最优化算法而言，归一化有利于模型的收敛，会使收敛过程会变得平滑、各特征变量在每次迭代过程中的步长可以做到均匀。这样更有利于算法以最小的迭代次数找到一组最优权重或参数。另外，关于最优化算法中的归一化问题将在章节 4.3.4 详细介绍。

在数据挖掘领域，目前有三种常用的归一化方法：分别是线性归一化(MinMaxScaler)、标准差归一化(StandardScaler)、最大绝对值归一化(MaxAbsScaler)，这三种方法各有特点和适用场合。

线性归一化的公式为：

$$x^{*}=\frac{x\text{-}x_{min}}{x_{max}\text{-}x_{min}} \tag{2.25}$$

式中，x表示特征中的任一点数据，x^{*}表示归一化后的x值，x_{min}、x_{max}分别表示此特征数据中的最小值和最大值。此方法可以将特征钳位到区间[0,1]之间，而且由于是线性变换，将不改变原始特征的分布比例性质。

标准差归一化的公式为：

$$x^{*}=\frac{x\text{-}\mu}{\sigma} \tag{2.26}$$

式中，x表示特征中的任一点数据，x^{*}表示归一化后的x值，μ、σ分别表示原始数据该特征的均值和标准差。经过这种方法处理后的数据将服从标准正态分布。

最大绝对值归一化的公式为：

$$x^{*}=\frac{x}{x_{max}} \tag{2.27}$$

最大绝对值归一化处理后数据的范围将被钳位到[-1,1]之间，使用这种方法之前需要先对数据进行中心均值化。由于稀疏数据的分布不符合一般规律且其比例性不明显，因此不适用前两种方法，一般用这种方法将稀疏数据集做密集处理。

在本文中，所有的特征数据均采用线性归一化方法，在做到密集化的同时不改变数据原有的比例性质，对本文数据应用这种方法归一化是有利的。变化前后

具体的数据分布情况本文不做重点分析。

2.2.2 缺失值填充

缺失值处理是数据挖掘领域常见的工作之一，由于各种条件限制，几乎所有用于训练的数据集都会有一定数量的缺失值。因此，这也是数据预处理不可回避的问题，是否能够尽可能准确地填补所有的缺失值直接影响着数据集的好坏。

缺失值的填充从本质上说可以看成一个参数估计问题，而针对参数估计问题，在统计学领域已有很多经典的方法：比如矩估计方法不需了解样本服从的分布，以不改变估计前后样本的 k 阶原点矩为准则；极大似然估计需要了解样本服从的分布，以将似然函数最大化为准则；最小二乘估计则计算使估计后样本中各采样点的平方误差和函数最小的一组参数解，通过求解矩阵方程来得到待估计的参数。

以上这些传统方法尽管在数百年间已经被无数次地证明了它们的优越性，但学习模型数据集的缺失数据特征往往具有缺失点多、稀疏性强、分布不标准、缺失点具体个数不定等特点，应用传统方法时往往受到局限而很少被用来做缺失值填充。

本文主要介绍两种缺失值处理方法并分析比较二者的优劣性：平均值法和 k-近邻学习(k-Nearest Neighbor Learning)。

首先，平均值法是一种方法简单且计算复杂度非常小的方法。它计算某一特征所有数据的平均值，并将缺失的数据用该特征的平均值填充即可。

k-近邻学习是一种计算代价很小的学习算法。它的预测原理十分简单：对于给定的测试样本，基于某种距离度量找到与测试样本最邻近的 k 个样本点，将这 k 个样本点的值取均值即可得出该测试样本的预测结果。用 k-近邻学习进行预测或者缺失值填充需要数据集满足以下两个条件：

第一，该数据集必须便于找到合适的距离度量策略。如果对一个数据集而言很难找到适合的距离度量策略，那么 k-近邻学习的精度就会大打折扣。

第二，数据集在基于某种距离度量下，不应过于稀疏。否则找到的 k 个样本点其实并不邻近于被测样本，使得预测结果糟糕。

下面来通过公式来推导两种方法的误差大小。

首先设定一些变量：

假设数据集中共有 t 条数据：D_1，D_2 ... D_t；

每条数据各有 s 个特征；比如，用D_{12}表示第一条数据的第二个特征对应的值。

设第 m 个特征有若干个缺失值：

其中对应第 n 条数据缺失值的实际估计值为 $\hat{D}_{nm}$；

对应第 n 条数据缺失值的最优估计值为 $\hat{y}_{nm}$；

如此，那么：

第 p 条数据与第 q 条数据之间的欧式距离可表示为：

$$d_{pq}=\sqrt{\sum_{j=1}^{s}\left(D_{pj}-D_{qj}\right)^2} \tag{2.28}$$

计算所有的$d_{p\cdot}$，选择 k 个与第 p 条数据欧氏距离最小的数据，记为：

D_{q1}，D_{q2} ... D_{qk}；

若第 p 条数据的第 m 个特征为缺失值，易得该缺失值的估计值为：

$$\hat{D}_{pm}=\frac{\sum_{i=0}^{k}D_{q_i m}}{k} \tag{2.29}$$

得出$\hat{D}_{pm}$估计值后，下面考虑最优估计值的求解：

在计算实际估计值时，使用了经典的欧氏距离作为度量。但这样做的前提是数据各维度的坐标轴刻度应该均匀分布，而由于本数据中特征之间与钛含量的相关性存在较大差异，因此在各维度上坐标轴刻度的分布显然并不均匀。这时，如果仍然使用典型的欧氏距离衡量两个点之间的距离就会产生一个偏差向量：$\vec{e}=(e_1,e_2,e_3,\dots,e_s)$，应该采用一个调整向量$\vec{\delta}=(\delta_1,\delta_2,\delta_3,\dots,\delta_s)$来弥补产生的误差，这里不具体给出$\vec{e}$与$\vec{\delta}$之间的关系。调整后的泛化欧式距离为：

$$d'_{pq}=\sqrt{\sum_{j=1}^{s}\delta_j\left(D_{pj}-D_{qj}\right)^2} \tag{2.30}$$

式中，δ_j为调整向量$\vec{\delta}$的第 j 个分量，即为第 j 个特征的调整系数。

类似的，计算所有的d'_{pq}，选择 k 个与第 p 条数据泛化欧氏距离最小的数据，记为：

D'_{q1}，D'_{q1} ... D'_{qk}；

则最优估计值可以表示为：

$$\hat{y}_{pm}=\frac{\sum_{i=0}^{k} D'_{q_i m}}{k} \tag{2.31}$$

需要说明的是调整向量$\vec{\delta}$在实际数据中是很难得到的，即所谓的最优估计值只是一个用于与实际估计值比较理想化的概念。这里引入这个概念是为了后面便于推导。

得出实际估计值和最优估计值的表达形式之后，现定义与 k 相关的损失数列:

$$L(k)=\hat{D}_{pm}-\hat{y}_{pm} \tag{2.32}$$

将式 2.29 和式 2.31 代入式 3.32 中，整理后得：

$$\mathrm{L(k)}=\frac{\sum_{\mathrm{i=0}}^{\mathrm{k}} \left|D_{q_i m}-D'_{q_i m}\right|}{k} \quad k\in[1,s],k\in N \tag{2.33}$$

考察数列 L 在 k 的取值域中的单调性：

$$\begin{aligned}\mathrm{L(k+1)-L(k)}&=\frac{\sum_{\mathrm{i=0}}^{\mathrm{k+1}} \left|D_{q_i m}-D'_{q_i m}\right|}{k+1}-\frac{\sum_{\mathrm{i=0}}^{\mathrm{k}} \left|D_{q_i m}-D'_{q_i m}\right|}{k}\\&=\frac{k\sum_{\mathrm{i=0}}^{\mathrm{k+1}}\left|D_{q_i m}-D'_{q_i m}\right|-(k+1)\sum_{\mathrm{i=0}}^{\mathrm{k}}\left|D_{q_i m}-D'_{q_i m}\right|}{k(k+1)}\\&=-\frac{k\left|D_{q_{(k+1)} m}-D'_{q_{(k+1)} m}\right|-\left|D_{q_k m}-D'_{q_k m}\right|}{k(k+1)}\ (k\geq 1)\end{aligned} \tag{2.34}$$

由于调整向量$\vec{\delta}$为非零向量，$D_{q_k m}$与$D'_{q_k m}$一定不相等。而二者都是归一化后的值，即$\left|D_{q_k m}-D'_{q_k m}\right|$<1恒成立。观察式 2.34 可以发现，尽管该式并不严格小于零，但当 k 足够大时，一般情况下该式是小于零的。由此可以说明 L 数列随着 k 值增大总体上呈下降趋势，但不一定严格单调，这取决于实际估值与最优估值之间的差值。据此，为了使估计误差尽可能小，应使 k 值尽量大。不妨令 k 取最大值 s，此时的误差在总体上说是最小的。注意到，当 k 取值为 s 时，此时的 k-近邻学习即为平均值法，平均值法其实正是 k-近邻学习的一种特殊情况。

根据以上理论推导，本文选择使用平均值法对数据进行缺失值填充。平均值法的计算量很小，更重要的是误差并不一定比计算更复杂的 k-近邻学习差。综合估计误差和计算量两个方面来说，平均值法也是最稳妥的一种方法。（基于上述

推导，绝大部分情况平均值法的误差小于 k-近邻学习，此处不做展开）本文中的原始数据规模为 301 条数据，60 个特征，共 18060 个数据点，共计填补缺失数据点 974 个。

2.2.3 噪声数据清洗

噪声数据清洗的过程实际是离群数据点的判断，并将其剔除。数据清洗同样也是数据挖掘项目不可避免的工作，本节简要介绍两种常见的离群点判断方法：3σ 准则判断和聚类分析，并利用 3σ 准则对数据进行清洗。

3σ 准则又称拉依达准则。假设样本近似服从正态分布，且只存在随机误差；那么根据正态分布的规律，求出样本值的均值和方差，划定数据中 3σ 的范围，所有落在 3σ 以外的数据属于小概率事件发生，被视作随机误差，予以剔除。需要注意的是，3σ 准则有两个使用条件：第一，样本容量必须足够大，最少不低于 20；第二，样本分布应近似服从正态分布。

3σ 准则操作简单可行，本文采用这种方法对经过归一化和缺失值填充后的钛含量数据计算其 σ 值，进行清洗：

$$\begin{cases} E_i = x_i - \dfrac{1}{N}\sum_{i=1}^{N} x_i \\ \sigma = \sqrt{\dfrac{\sum_{i=1}^{N} E_i^2}{N-1}} \end{cases} \tag{2.35}$$

若残差$|E_i|>3\sigma$，则认为其为噪声数据。共计样本容量为 301，噪声数据为 0 条，故数据噪声性能良好，不需要进行数据剔除。图 2.10 为钛含量的分布直方图，可以看出钛含量的分布近似呈正态分布，因此用 3σ 准则做数据得到的结果较为可信[53]。

聚类分析是无监督学习策略中最典型的一种。也就意味着在聚类中，事先不清楚样本的标签信息，需要算法通过优化迭代的方法找到样本集中最优的一个分类。聚类分析通过计算样本点之间某种度量揭示数据内在的规律和性质，因此可以用于对异常数据的筛选。

在聚类算法中，如果使用不同的度量准则或者不同的最优化策略，将会形成不同类型的聚类算法；在各种聚类算法中最经典的一种是 k-均值(k-means)算法。图 2.11 简要介绍了 k-means 的工作过程，具体推导过程详见[8]。

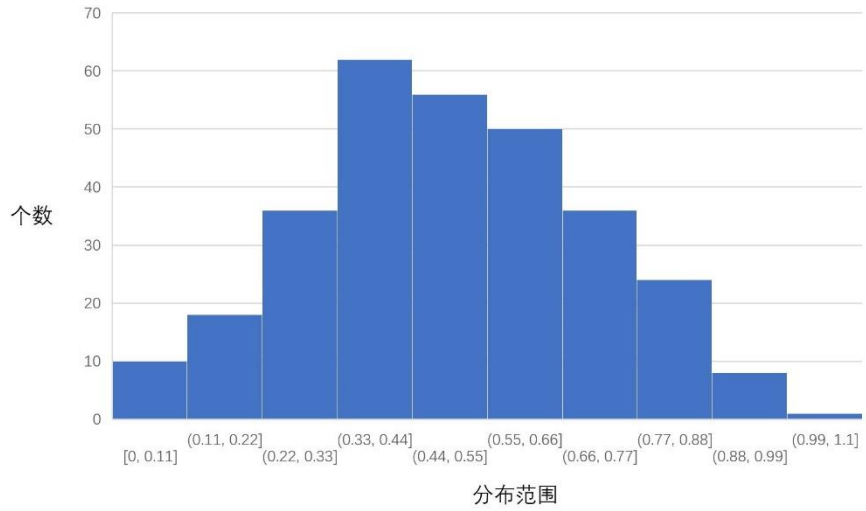

图 2.10 钛含量分布散点图

Figure2.10 Scatter diagram of titanium content distribution

算法 2.1：k-means 聚类算法

输入： 训练集 $\mathbf{D}=\{\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_n}\}$

聚类簇数 **k**

聚类簇 $\mathbf{C}=\{\mathbf{C_1},\mathbf{C_2},\mathbf{C_3},\ldots,\mathbf{C_k}\}=\emptyset$

步骤：

1：从 D 中随机选择 k 个样本作为初始均值向量$\mathbf{U}=\{\boldsymbol{\mu_1},\boldsymbol{\mu_2},\ldots,\boldsymbol{\mu_k}\}$

2：**while** 停止条件 **ω** 不成立，**do**：

3： $\mathbf{C}=\{\mathbf{C_1},\mathbf{C_2},\mathbf{C_3},\ldots,\mathbf{C_k}\}=\emptyset$

4： **for i=1 :m，do**：

5： 计算样本点$\boldsymbol{x_i}$到均值向量各分量$\boldsymbol{\mu_j}$**(1≤j≤k)**的距离：

$$d_{ij}=\left\|x_i-\mu_j\right\|_2 \tag{2.36}$$

6: 寻优以确定样本点$\mathbf{x_i}$的簇标记$\boldsymbol{\lambda_i}$：

$$\lambda_i=argmin\sum_{j=1}^{k}\sum_{x_i\in C_i}d_{ij} \tag{2.37}$$

7： 将样本$\mathbf{x_i}$按照$\boldsymbol{\lambda_i}$标志聚类：$\mathbf{C_{\lambda_i}}=\mathbf{C_{\lambda_i}}\cup\{\mathbf{x_i}\}$

8： **end for**

9： **for i=1 :m，do**：

10: 按照当前聚类重新计算均值向量，并更新。

11： **end for**

12：**end while**

输出：$C=\{C_1,C_2,C_3,\ldots,C_k\}$

图 2.11 k-均值算法工作流程

Figure2.11 Process of k-means algorithm

由于 k-均值算法操作起来相对 3σ 准则复杂，而且参数 k 的确定并不容易。因此文本没有采用聚类方法对数据进行清洗。

2.2.4 特征选择

经过前三节的处理，此时的数据已经可以用来直接进行训练模型了。但由于特征与目标变量钛含量的相关性普遍较低，而且存在大量的耦合，容易导致训练后的模型输出精度下降且鲁棒性较差。已有相关实验表明，对于浅层的神经网络模型，当输入特征过多或者输入特征之间存在较多耦合时，不仅会影响到网络预报的精度，而且会影响模型的鲁棒性，使模型对于同一给定输入的响应变得不稳定。有关于使用不同质量的数据对随机森林模型训练的精度和鲁棒性分析详见章节 5.2。

基于此，本节主要工作如下：

第一，介绍两种统计学上常用的相关系数概念——皮尔逊相关系数和回归相关系数，同时比较二者的优劣。

第二，将原始数据记为 Data0。用皮尔逊相关系数对特征进行两轮选择，产生两组不同的数据，分别记为 Data1，Data2。

第三，对 Data2 中的数据进行耦合性分析，剔除与其他特征耦合性较大的特征，再产生一组数据记为 Data3。

数据 Data0-Data3 将在章节 5.2 逐一被当作数据集来训练模型，并对训练后生成模型的精度和鲁棒性进行详细分析对比。

对于回归问题而言，统计学中的相关性系数概念可以非常准确地刻画特征与标签之间的相关性[8]。在工程上，经常使用到两类相关系数；即回归相关系数和皮尔逊相关系数。

对于回归相关系数而言，实际上是求取自变量与因变量的一个回归方程，在回归方程的基础上定义回归相关系数[31]。一般来说，对实际问题中的参数采用非

线性方程进行回归拟合。设自变量为 x，因变量真实值为 y，非线性回归方程采用二次多项式：

$$\hat{y}=a_0+a_1x+a_2x^2 \tag{2.38}$$

式中，a_0, a_1, a_2为待求参数,$\hat{y}$为 y 的估计值。

则对应式 2.38 的非线性回归相关系数可表示为：

$$R=\sqrt{1-\frac{\sum(y-\hat{y})^2}{\sum(y-\bar{y})^2}} \tag{2.39}$$

可以看到，这种回归方法尽管计算复杂度较低，但存在两方面的问题：第一，回归方程并不好选择；第二，回归方程中的未知参数需要使用额外的算法（比如最小二乘回归）才能确定。这两点就对实际操作带来了局限，特别是第一点，如果不能确定合适的回归方程，那么计算出的相关系数就没有太高的可信度，用这种方法筛选过后的特征也失去了价值。

皮尔逊相关系数用来表示两个变量之间的线性相关程度，由英国统计学家卡尔·皮尔逊在 19 世纪 80 年代提出，也通常被称作：相关系数。（下文统一称之为相关系数）其值介于[-1,1]之间：当两变量的相关系数为±1 时，表示二者具有严格线性性质；值为正数时表明二者具有正相关性，值为负数时表明二者具有负相关性。相关系数的求法简单，不受量纲影响，结果又十分有效，在实际工程中往往具有很重要的参考价值，相关系数的计算方法推导如下[31、33]：

设X，Y为两组待考察的随机变量；

若数学期望E[(X-E(X))(Y-E(Y))]存在，则：

X，Y之间的协方差可以表示为：

$$\mathrm{cov}(X,Y)=E\left[\left(X-E(X)\right)\left(Y-E(Y)\right)\right] \tag{2.40}$$

不难看出，协方差可以直观理解为两个变量与各自均值差的乘积的数学期望。在统计学中，一般用协方差表示两个随机变量之间总体误差。然而协方差是有量纲的量，为了避免量纲对相关性分析带来的影响，还需要将协方差除以两个变量各自标准差的乘积，即得到了两个变量的相关系数：

$$R=\frac{\mathrm{cov}(X,Y)}{\sqrt{D(X)D(Y)}} \tag{2.41}$$

带入数学期望和标准差的具体计算方法，将式 2.41 化简后得：

$$R=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{2.42}$$

式 2.42 中，x_i，y_i分别表示X，Y的采样值。$\bar{x}$，$\bar{y}$表示x_i，y_i采样值的平均值。式 2.42 即可用来直接计算各特征与标签变量钛含量的相关系数。

下面开始用相关系数进行特征选择。

首先对表 2.1 中的所有特征计算其与钛含量的之间的相关系数，所有参数的实际意义已在章节 2.1.3 做过详细分析，这里不再赘述。结果统计如表 2.2：

表 2.2 重要高炉参数与[Ti]的相关系数

Table2.2 correlation coefficients between blast furnace parameters and [Ti]

变量名	与[Ti]的相关系数	变量名	与[Ti]的相关系数
R2	-0.1169	M40	-0.0238
R3	-0.1161	CSR	0.1461
TiO_2	-0.094	CRI	0.2022
镁铝比	-0.1151	Ad	0.0785
风温	-0.1280	St	0.2182
实际风速	-0.1146	SiO_2	0.1087
鼓风动能	-0.0640	Si	0.8192
透气性指数	-0.1260	C	0.5814
富氧量	-0.1049	S	0.7083
喷吹速率	0.1100	喷吹煤 Vdaf	0.3156
煤气利用率	-0.1260	喷吹煤 Ad	0.0169
炉缸温度	0.2539	喷吹煤 St	0.0957
炉缸中心温度	0.0169	炉顶煤气 CO	-0.0502
炉腰温度	-0.1273	炉顶煤气H_2	0.033
炉身下二段温度	0.1915	炉顶煤气 CO_2	-0.1994
炉喉温度	0.1316	炉腹煤气量指数	0.0051

炉顶温度	0.1824	焦炭负荷	0.3369
铁水温度	-0.0305	焦比	0.1045
探尺差	0.2165	煤比	0.1100
M10	0.1410	燃料比	0.1138

统计其中 R>0.1 的特征，有：

R2、R3、镁铝比、风温、实际风速、透气性指数、富氧量、探尺差、喷吹速率、煤气利用率、炉顶温度、炉缸温度、炉腰温度、炉身下二段温度、炉喉温度、M10、CSR、CRI、St、SiO_2、Si、C、S、喷吹煤 Vdaf、炉顶煤气 CO_2、焦炭负荷、焦比、煤比、燃料比。共计 29 个特征，将它们放入 Data1 数据中。

统计其中 R>0.15 的特征，有：

CSI、St、喷吹煤 Vdaf、C、Si、S、探尺差、炉顶温度、炉身下二段温度、炉缸温度、炉顶煤气 CO_2、焦炭负荷。共计 12 个特征，将它们放入 Data2 数据中。

Data2 数据中的特征个数已经足够少而且与钛含量的相关系数都比较高。但是在 Data2 数据的特征之间仍然可能存在耦合，对模型的性能产生负面影响。因此，下面对 Data2 中各特征之间的耦合性进行分析。分别计算这些变量与其他变量的相关系数，绘制成相关系数矩阵如表 2.3 所示：

表 2.3 Data2 中特征间相关系数矩阵

Table 2.3 matrix of correlation coefficients between features in Data2

	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**	**F11**	**F12**
F1	1.00											
F2	-0.03	1.00										
F3	0.15	-0.16	1.00									
F4	0.26	-0.42	0.40	1.00								
F5	0.12	0.05	0.06	0.11	1.00							
F6	-0.20	0.40	-0.43	-0.69	-0.32	1.00						
F7	0.05	-0.24	0.20	0.15	0.13	-0.23	1.00					
F8	0.05	0.08	0.05	-0.08	0.29	-0.04	0.14	1.00				
F9	0.41	-0.20	0.18	0.10	0.15	-0.24	0.23	0.36	1.00			

F10	-0.29	0.16	-0.20	-0.19	-0.22	0.22	-0.19	0.13	-0.29	1.00		
F11	-0.25	0.02	-0.04	-0.03	-0.21	0.13	-0.20	-0.65	-0.51	0.10	1.00	
F12	0.53	-0.14	-0.20	-0.14	-0.34	0.20	-0.16	-0.25	-0.48	0.36	0.43	1.00

表中符号意义如下，F1 代表 CRI；F2 代表 St；F3 代表喷吹煤 Vdaf；F4 代表 C；F5 代表 Si；F6 代表 S；F7 代表探尺差；F8 代表炉顶温度；F9 代表炉身下二段温度；F10 代表炉缸温度；F11 代表炉顶煤气 CO_2；F12 代表焦碳负荷。

基于常识，如果某一变量跟其他变量的相关系数都较大，那么认为此变量与其他变量的耦合性较强。根据这一点，本文利用以下公式简单评价某一变量与其他变量之间的耦合性。

$$\partial_i = \frac{\sum_{j=1}^{n-1} |R_{ij}|}{n-1} \tag{2.43}$$

式中，∂_i表示随机变量i在待考察数据集中的耦合性指数，R_{ij}表示随机变量 i 与随机变量 j 之间的相关系数。n 表示待考察数据集中的特征总数。表 2.4 列出了 Data2 中所有变量的耦合性指数：

表 2.4 Data2 中特征之间的耦合性指数

Table2.4 The coupling index between features in Data2

变量名	耦合性指数∂	变量名	耦合性指数∂
CRI	0.2136	探尺差	0.1747
St	0.1732	炉顶温度	0.1935
喷吹煤 Vdaf	0.1873	炉身下二段温度	0.2866
C	0.2349	炉缸温度	0.2127
Si	0.1825	炉顶煤气 CO_2	0.2350
S	0.2820	焦碳负荷	0.2936

由表中数据可以发现：炉身下二段温度、焦炭负荷两个变量与 Ti 含量的相关性并不很高，而且与其他变量的耦合性指数最大。因此去除这两个特征，剩下的 10 个特征组成数据集 Data3。Data3 中包含的特征有：CSI、St、喷吹煤 Vdaf、C、Si、S、探尺差、炉顶温度、炉缸温度、炉顶煤气 CO_2。

2.3 本章小节

本章从高炉工艺入手，分析了数据中的重要高炉参数的实际意义。在此基础上对数据进行预处理，通过数据线性归一化、平均值法缺失值填充、3σ准则清洗数据和利用相关系数进行特征选择并做耦合性分析等步骤，最终得到了 Data0-Data3 四组品质不同的数据：其中，Data0 共有 301 条数据、40 个特征；Data1 共有 301 条数据、29 个特征；Data2 共有 301 条数据、12 个特征；Data3 共有 301 条数据、10 个特征。将在下面的章节利用这五组数据训练相同结构的随机森林模型以比较不同特点的数据对随机森林模型精度以及鲁棒性的影响。

3 基于随机森林的铁水钛含量预报模型

随机森林(Random Forest)是一种由决策树(Decision Tree)作为子学习机的一种集成学习类机器学习算法。该算法是由贝尔实验室的 Leo Breiman 和 Adele Cutler 在 1995 年综合改进 Ho 等人的研究思路，提出的一种集成学习算法。随机森林算法的提出大大拓宽了决策树算法的使用范围，同时也提高了集成学习类算法的精度上限。在今天，随机森林仍被认为是可以代表集成学习最高性能的一种算法。随机森林相比于传统的神经网络等学习算法而言，具备很多优点，比如出色的可解释性、不依赖权重训练，不容易陷入局部最优、超参数较少，便于调参等等。本章用经典的随机森林算法建立模型，实现基本的铁水钛含量预报任务，同时分析随机森林算法的超参数分别对学习性能的影响，为下一章的改进研究做准备。

3.1 随机森林算法概述

3.1.1 决策树算法简介

决策树(Decision Tree)是一种较为简单的机器学习算法，初始生成只有一个根节点的决策树桩；通过对输入特征属性“区域”进行划分，从而形成每一步的决策，得到不同的中间节点；直至划分到不可继续划分或者满足某种退出条件时，记录此时的中间节点成为叶子节点，一颗决策树的训练就完成了。用训练好的决策树进行测试时，该树将通过已划分好的属性区域将样本进行不断的向下分类，直到递交给叶子节点，将此叶子节点的输出作为决策树的输出结果。应该注意到的是，在一颗完整的决策树中叶子节点并不唯一[8]。图 3.1 是所有决策树学习机的通用算法流程。

算法 3.1：基础决策树学习算法

输入： 训练集 $\mathbf{D}=\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$

属性标签 $\mathbf{A}=\{a_1,a_2,\ldots,a_s\}$

最大层深 $\mathbf{d}$

属性划分准则 $\boldsymbol{\delta}$

节点组 $\mathbf{node}=\emptyset$

步骤：

1：生成根节点$\mathbf{node_0}$，设$\mathbf{D_{1,0}=D}$

2：**for i = 1:d ，do：**

3：　　根据划分准则**δ**从属性集 **A** 中选择最佳划分属性$\mathbf{a_*}$

4：　　**for** $\mathbf{a_*^v}$ **in** $\mathbf{a_*}$，**do：**($\mathbf{a_*^v}$表示$\mathbf{a_*}$属性的每个取值)

5：　　生成节点$\mathbf{node_v}$，将$\mathbf{D_i}$中取值为$\mathbf{a_*^v}$的样本设为$\mathbf{D_{i+1,v}}$作为$\mathbf{node_v}$的样本子集

6：　　**if** $\mathbf{D_{i+1,v}=\emptyset}$，**then：**

7：　　将$\mathbf{node_v}$标记为叶子节点

8：　　**else：**

9：　　将$\mathbf{node_v}$标记为中间节点

10：　　**end for**

11：**end for**

输出：节点组 **node**

图 3.1 基础决策树学习算法工作流程

Figure3.1 Workflow of basic decision tree learning algorithm

其中，针对全体特征属性区域的属性划分准则δ是形成每一步决策结果的标准。因此，根据划分准则的不同，决策树算法又可以细分成若干类算法。

（1） ID3 决策树

ID3 决策树生成算法(Iterative Dichotomiser 3)可以说是最经典的决策树算法之一。ID3 算法认为：在每一步的决策时，如果使得决策后的节点包含样本的“纯度”越大，则意味着该节点包含样本越有可能属于同一类别。因此，ID3 算法量化了样本纯度这个概念，在每一节点处分别尝试不同的划分方法，找到使划分后节点包含样本纯度最大的一个划分。

ID3 引入信息熵(Information Entropy)和信息增益(Information Gain)的概念来表示量化样本的纯度，并以此为依据进行选择，公式如下：

$$\begin{cases} Ent(D)=-\sum_{k=1}^{\gamma} p_k \log_2 p_k \\ Gain(D,a)=Ent(D)-\sum_{v=1}^{V}\frac{D_v}{D}Ent(D_v) \\ a_*=\underset{a\in A}{argmax}\, Gain(D,a) \end{cases} \tag{3.1}$$

式中，Ent(D)代表样本集D的信息熵，p_k代表 D 中第 k 类样本所占比例，Gain(D,a)代表 D 中属性a的信息增益，D_v代表属性a中取某值的样本个数，V代表属性a可不同值的个数。

（2） C4.5 决策树

观察式 3.1 可以发现：在某一属性a中，当其他条件不变时，随着V增大，信息增益也增大。这表明，ID3 算法在计算时没有考虑到各属性的可取值数目不同而对结果带来的影响。直观上上说，这将导致生成的决策树对可取值较多的属性有选择偏好，从而影响决策效果。C4.5 算法为了改进这一点，提出了增益率的概念，消去了部分信息增益中属性取值数目对结果造成的影响，增益率的表达形式为：

$$\mathrm{Gain_{ratio}(D,a)}=\frac{Gain(D,a)}{-\sum_{v=1}^{V}\frac{D_v}{D}log_2\frac{D_v}{D}} \tag{3.2}$$

可以看到，在其他条件不变的情况下，式 3.2 中分母的值同样随V增大而增大，这样就可以抵消掉一部分因属性可取值较多而造成的选择偏好。式 3.2 的分母又被称为属性的固有值。

（3） CART 决策树

CART 决策树(Classification And Regression Tree)是目前广泛用于各种分类和回归任务的一种决策树。对于分类问题，它将样本的纯度用基尼指数来表示。而对于回归问题，基尼指数可以表示为样本的残差平方和，即最小二乘方法，故应用于回归问题的 CART 树又被称为最小二乘树。样本集D中，属性a的基尼指数可以表示为：

$$\mathrm{Gini(D,a)}=\sum_{\mathrm{v=1}}^{\mathrm{V}}\frac{D_v}{D}(1-\sum_{\mathrm{k=1}}^{\gamma}p_k^2) \tag{3.3}$$

对于分类问题，CART 树将 Gini 指数最小的属性视作最佳划分属性，对于CART 树的回归形式将在下面介绍。

由于决策树的最早是用于解决分类问题的，所以要将决策树用于处理连续标签的回归任务，必须要对以上的划分机制加以修改。

要想将原本处理分类问题的决策树应用于回归问题，首先必须弄清楚回归与分类的不同之处；回归问题相比于分类问题有如下特殊点：第一，回归问题的本

质特点是标签连续，此时信息增益或者基尼指数中的 k 为无穷大，这就导致p_k的值无法统计；第二，一般来说，回归问题的输入特征大多是连续的，与第一点类似，此时V的值也为无穷大，这就导致D_v的值无法统计。由于以上两点，离散形式的信息增益或者信息率的计算方法都不能照搬到回归问题中。下面主要针对回归问题的两点特殊性给出最小二乘回归树的生成策略。

算法 3.2：最小二乘回归决策树生成算法

输入：训练集 $\mathbf{D}=\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$

回归树函数 $\mathbf{f(x)=0}$

步骤：

1：**while 退出条件ω不成立 ，do：**

2： **for j , s in R_i，do：**(a_*^v表示a_*属性的每个取值)

3： 选择最优分切变量 j 以及最优分切点 s，求解：

$$j,s=\underset{j,s\in R}{argmin}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2 \tag{3.4}$$

4： **end for**

5： 按照(j,s)分裂区域并计算对应区域的输出值$\hat{c}_m$：

$$\begin{cases} R_1(j,s)=\{x|x_j\le s\} \\ R_2(j,s)=\{x|x_j>s\} \\ \hat{c}_m=\dfrac{\sum_{x_i\in R_m(j,s)}y_i}{N_m},x\in R_m,m=1,2 \end{cases} \tag{3.5}$$

6：**end while**

输出：回归树函数 $f(x)=\sum_{m=1}^{M}\hat{c}_m I(x\in R_m)$

图 3.2 最小二乘回归决策树算法工作流程

Figure3.2 Workflow of least squares regression tree algorithm

在式 3.4 中，c_1、c_2分别代表区域$R_1(j,s)$和$R_2(j,s)$中所有标签的平均值，式 3.5 中，$\hat{c}_m$实际是对应区域中标签的平均值。通过图 3.2 可以看出，最小二乘回归树的工作机制使用了最小二乘损失函数，遍历所有可能的分切变量和分切点来对样本区域进行若干次划分，同时得到划分区域包含标签的平均值，作为此区域的输出值；在预测时，如果预测样本落到此区域，则将此区域的输出值作为回归树的预测值。

本文的随机森林就是以这种最小二乘回归树作为单个自学习加以集成构建的。

3.1.2 集成学习简介

集成学习(Ensemble Learning)在单个学习机的基础上，利用子学习机训练策略以及输出结合策略，将一定规模的子学习机集成起来完成学习任务以获得更好的性能[8]。集成学习的一般结构为：

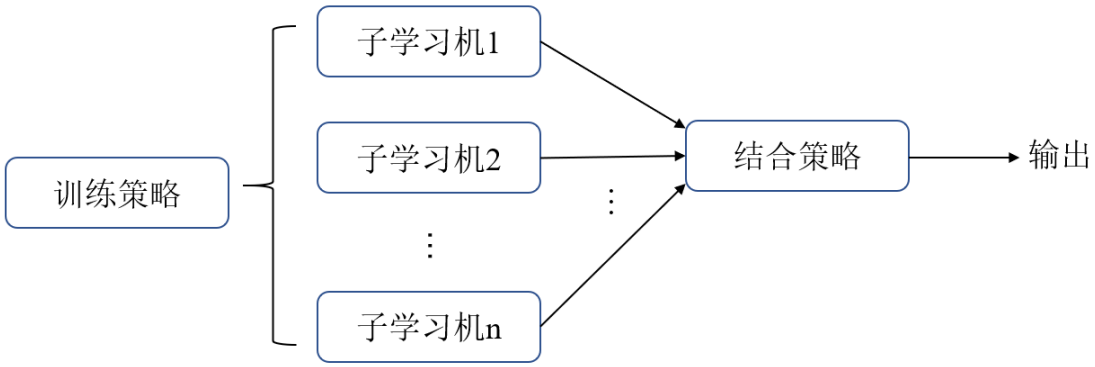

图 3.3 集成学习的结构框图

Figure3.3 Structure diagram of ensemble learing

已有大量的实验证明，在一次集成学习中所用到的子学习机的一些性质对最终的集成结果有着极大的影响。具体来说，单个子学习机的性能好坏会影响到最终的结果；在其他因素不变的前提下，单个子学习机学习性能与集成学习性能成正比。同时，子学习机之间的关联程度大小也会影响最终结果；在其他因素不变的前提下，一次集成学习中的子学习机之间相关联程度与集成学习性能成反比。也就是说，为了尽可能提高集成学习的性能，应该尽量提高单个子学习机的精度并且设法降低子学习机之间的相关联程度。通俗地说，集成学习所研究的目标就是让各子学习机达到“好而不同”。从这个角度而言，图 3.3 中的训练策略和输出结合策略本质上都是为了提高子学习机精度并降低子学习机之间的相关程度。

目前主流的集成学习根据训练策略不同可以分为两大类，即串行训练和并行训练。其中串行训练最典型的算法是 Boosting，而并行训练的代表算法是 Bagging。

在 Boosting 中，学习机串行地进行训练，通过设置每个训练样本的权重使每个学习机对不同的样本有所偏好，以着重训练前一个学习机判断错误的那部分样本；一般来说，在每一个子学习机的训练过程中，学习机的输出权重会随着每一轮的训练迭代不断更新。在测试时利用输出权重将各子学习机的结果进行综合。

由于 Boosting 特殊的串行训练机制，训练顺序靠后的学习机总可以针对靠前学习机的不足进行针对性地学习训练，这使得 Boosting 类算法对弱学习机有较强的兼容性，即学习能力很弱的一组子学习机也可以构建出学习能力相当强的集成学习。但其缺点在于当子学习机数目过多时，串行机制无法进行分布式训练，这会使总体的训练速度大幅降低；同时，这种特殊的训练机制也削弱了各学习机之间的独立性，使得子学习机之间的关联性增强，一定程度上也限制了最终的训练结果。

与 Boosting 相反，Bagging 中的学习机可以并行训练。为了让各学习机好而不同，每个子学习机所分配的样本应该不同，这需要使用合适的采样策略(Sampling)来分配各子学习机的样本。分配样本之后各学习机分别独立完成训练，最后再由一定的结合策略将各自学习结果结合输出即可。Bagging 的训练机制较为简单，复杂度更小，而且尽可能使各学习机之间的独立性增强；由于各子学习机的训练互不发生关系，这使 Bagging 对分布式训练的兼容性更好，可以大幅提高训练速度。不过，Bagging 的使用对子学习机的性能有一定要求，对学习能力太弱的学习机而言，很难像 Boosting 一样大幅度地提升总体的学习效果，而选用学习能力更强的学习机可能会造成更大的计算开销。

在结合策略方面，集成学习也可以采用很多种方法。针对分类问题，有简单投票法和加权投票法；对于回归问题，常用简单平均法和加权平均法。还可以使用一个额外的学习模型来学习各子学习机之间的最佳结合方案，代表性的方法是 Stacking。本文在针对随机森林进行改进时使用的方法实际上是一种加权平均策略，详见章节 4.2。

3.1.3 随机森林算法简介及基础随机森林模型搭建

前面已经介绍过，随机森林是一种特殊的集成学习。更具体地，可以算作一种特殊的 Bagging。根据上一节提到过的内容，集成学习研究的首要目标是如何设计训练策略和结合策略以提高单个子学习机的性能并降低子学习机之间的关联程度，随机森林在这一点上也不例外。为了做到这两点目标，随机森林在一般的结合策略上强调两点“随机”：

第一，每颗决策树的候选特征是随机的。假设训练集共用 f 个特征，而对于每颗决策树，并不是都可以在这 f 个特征集中随意选择特征用于划分、训练。而

是在每棵树的训练之前，必须先从 f 个特征中随机挑选 k 个特征作为候选特征子集，这棵树在训练时只能从该特征子集中选择最优划分属性，进行训练。这里 k 的值并不固定，可以取大于 1 小于 f 的任意整数；一般来说，随机森林的提出者 Breiman 建议 k 的取值为$\lceil \log_2 f \rceil$。应该注意到的是，特征子集容量 k 的大小会影响到每棵树之间的相关性。显然，k 值越大，则树之间的相关性越大，随机森林的学习能力将受到限制；而 k 值越小，尽管树之间的相关性越小，但取较小特征训练时可能影响单颗树的学习精度。因此 k 的合理取值是设计算法的一个关键。（详见章节 3.2）

第二，随机森林中每棵树的训练样本是随机的。一些研究随机森林方面的文献往往过度关注第一点，而忽略第二点。每棵树训练样本的随机性同样会影响树之间的相关性，因此需要选择一个合理的样本分配策略来为各决策树随机分配样本，在保证每颗树样本数量的同时也要最大程度上考虑抽取到的样本子集的互异性。与特征随机类似，样本随机的策略也可能影响到单颗树的学习能力以及树之间的关联程度两个方面；如果给每棵树分配的训练样本太少，则尽管树之间的关联程度会降低，但训练样本不足也将使树的学习性能下降，反之类似。

基于以上对随机森林的理解，下面介绍本文第三章所采用的随机森林算法，其流程框图如图 3.4 所示：

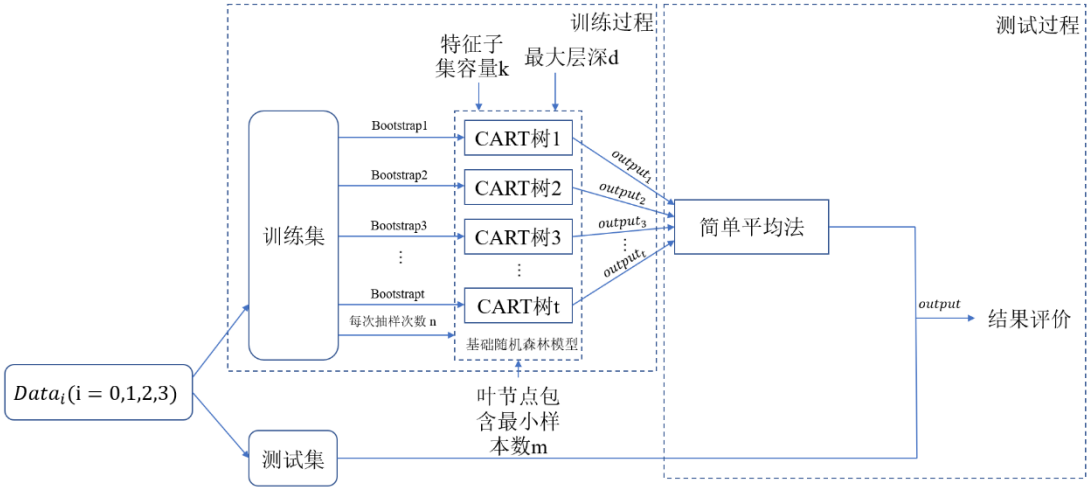

图 3.4 基础随机森林算法流程框图

Figure3.4 Flow diagram of basic random forest algorith

由图 3.4 描述，本章使用的随机森林模型使用 CART 回归树作为子学习机，选用最小二乘损失函数作为最优分切点划分准则。模型可以使用第二章中生成的

4 个数据集 Data0-Data3 进行训练和测试，需要手动根据数据集制作训练集和测试集，具体方法详见章节 5.1；同时，模型中共有 5 个待确定参数，分别是森林规模 t、每次抽取训练样本次数 n、特征子集容量 k、决策树最大层深 d 和叶节点包含最小样本个数 m。这些模型参数的意义以及可能对模型性能带来的影响将在章节 3.2 进行详细分析。

另外，图 3.4 中的简单随机抽样包含两个方面，即对特征集进行简单不放回抽样和对训练集进行简单有放回抽样，也称作自助采样法(Bootstrap Sampling)。对于所有子学习机的输出采用简单平均法进行结合，公式为：

$$\text{output} = \frac{\sum_{s=1}^{t} output_s}{t} \tag{3.6}$$

至此，可以直接用于训练的基础随机森林模型搭建完成，章节 3.2 中对随机森林模型参数的分析、章节 4.1 中对基础随机森林模型局限性的分析以及章节 5.2 中对基础随机森林模型预报结果的实验均基于图 3.4 所搭建的模型。

本节（章节 3.1）首先介绍了决策树和集成学习的算法原理和常用策略，再次基础上介绍经典的随机森林算法并依此搭建了基础随机森林模型（图 3.4），为后续章节的进一步分析基础模型、改进模型以及测试实验做好基础。

3.2 随机森林模型参数分析

3.2.1 影响随机森林模型性能的两方面因素及其表征方法

在机器学习中，模型参数分为两种：一种是普通参数，另一种是超参数。其中，普通参数需要模型通过迭代等机制自动收敛、学习而得；而超参数是为了使模型更好的收敛、学习参数而手动设定的一些参数。比如，在各种神经网络模型中，网络中每个神经元连接的权重就属于普通参数；而网络的层数、每一层的神经元个数、权重初值、学习率等参数就属于超参数。而在用于回归任务的随机森林模型中，每一颗决策树中每一次分裂的最优分切点、最优分切区间等可以认为是普通参数；而章节 3.1.3 中提到的 5 个参数显然属于超参数。众所周知，神经网络的调参是一项十分依靠经验，而且有时是十分具有挑战性的工作，因为特殊的结构设置可能导致其超参数非常多。而随机森林的优越性之一就在于不论森林的结构如何变化，超参数的个数总可以非常少。在有些不需要要求性能的情况下甚至只需要 1-2 个超参数即可完全控制模型的结构，这一点是传统的神经网络模

型所不具备的。

在章节 3.1.2 中论述到，约束集成学习性能的两个主要方面是各子学习机的学习能力和子学习机之间的相关联程度。因此，为了具体考察某一参数对模型性能是如何产生影响的，应该从这一参数对各决策树的学习能力以及各树间的关联程度来分别研究。以下的实验数据均为使用 Data2（301 条数据、12 个特征）对基础 RF 模型训练、测试得到的。

各决策树的学习能力是容易表征的，可以直接用各决策树输出的均方根误差表示其预测精度（均方根误差的概念详见章节 5.1）、用输出均方根误差的方差表示各决策树输出的稳定性。章节 3.2.2-3.2.6 中涉及到的单颗决策树学习能力的数据均按照此方法来统计。

而对于决策树之间相关性的表示方法，文献[44]中提到从样本的角度入手：对于一个数据集中的同一样本，如果两颗决策树都倾向于将它归结于同一个类别，则两颗决策树的相关性越强。如此，则可以统计在数据集中某两颗决策树将同一样本归结于相同类别的概率。如果概率越大，则说明两颗决策树间的相关性越强。然而这种方法对于回归问题而言并不准确，因为回归问题相当于有无数个输出类别，用这种方法显然会带来麻烦。因此，本文提出了一种从决策树输出角度入手研究决策树之间相关性的方法。

基于常识认为，如果两颗已训练好的决策树之间的关联程度很大，那么二者对同一个测试样本的响应应该非常相似，反之类似。那么对于不同的测试样本是否也是如此呢？换言之，这种相关性是否与测试样本的变化有关？

针对此疑问，先考虑一个简单的集合学问题：

如果集合A、B有一种映射关系f，即满足：

$$A=f(B) \tag{3.7}$$

且集合B中元素B_1、B_2、…、B_n分别与集合C_1、C_2、…、C_n有映射关系g_1、g_2、…、g_n。即分别满足：

$$\begin{cases} B_1=g_1(C_1) \\ B_2=g_2(C_2) \\ \cdots \\ B_n=g_n(C_n) \end{cases} \tag{3.8}$$

那么基于正常理解，A中的元素A_1、A_2、…、A_n可以表示为：

$$\begin{cases} A_1=f[g_1(C_1)] \\ A_2=f[g_2(C_2)] \\ \cdots \\ A_n=f[g_n(C_n)] \end{cases} \tag{3.9}$$

而此时，将式 3.8 带入式 3.9 中，得出A中的元素与B中元素的关系为：

$$\begin{cases} A_1=f(B_1) \\ A_2=f(B_2) \\ \cdots \\ A_n=f(B_n) \end{cases} \tag{3.10}$$

不难看到在这种情况下，随着集合C_1、C_2、…、C_n以及映射g_1、g_2、…、g_n的变化，A和B中的元素始终保持着固定的映射关系f。这就是映射关系的复合性质，不论中间映射g_i或中间集合C_1、C_2、…、C_n如何变化，A和B始终可以保持相同的映射关系。

现在，把以上的数学字母赋予实际意义：

g_1、g_2、…、g_n看成 n 次训练得出的不同单颗决策树模型；

C_1、C_2、…、C_n看成用于 n 次测试的 n 组不同数据集合；

把A和B看成随机森林中的两棵树对不同（n 组）测试集的响应输出集合；

f 则可以看成A和B两颗决策树输出集合之间的潜在函数关系。

此时再来回顾式 3.7-3.10，将有如下理解：对于随机森林中的任两颗决策树而言，在 n 次相同超参数的训练和 n 次不同测试集测试的过程中，这两颗决策树输出集之间的相关性不因测试集样本不同或训练不同而受到影响。（注意，尽管每次训练使用的超参数相同，但不同次训练对于每颗决策树而言仍然可以看成是不同的训练，因为每次训练给每棵树分配的样本、特征等不尽相同，形成的决策树模型也不会完全相同。）

基于以上认识，对于两颗决策树输出集之间的所隐含的函数关系的强弱，完全可以用相关系数来表征，对于这种方法本文只做出以上理论推导不对其做出具体实验进行验证。章节 3.2.2-章节 3.2.6 只通过实验具体分析不同参数取值对各决策树预报精度的影响。

3.2.2 森林规模对模型性能的影响

从直观上说，如果森林的规模增大，即森林中树的个数增加，那么不同的树选中相同样本及特征的概率就会增大；于是树之间的相关性也会增大，对集成学习造成负面的影响增加。另一方面，由于随机森林中树之间之间的训练彼此不发

生关联，这意味着树的增多并不会提高单颗树的学习能力。如此思考，取一种极端情况，当森林中只有一棵树时的训练效果应该最好，这样这显然不符合常理。这是因为忽略了一个简单的道理：单次训练产生的模型可能具有偶然性，不能代表真实的训练情况。因此才需要一个结合策略把多次训练的结果综合起来，这也是集成学习的价值所在。由此可见，在理想情况下，随着决策树个数的增加，模型整体的学习能力应该先增强后下降，当决策树个数取某一值时可以达到最佳学习效果。不少有关随机森林的文献提到森林规模的这点性质，文献[44]还对此做出了理论推导。但理论推导的分析方法对于分类问题会显得更加适合，而对于回归问题就会变得十分麻烦，因此本文只通过对比实验数据的方法对这种性质进行分析。

图 3.5 统计了森林规模与所有决策树单独进行样本预测均方根误差(RMSE)平均值的关系。

图 3.6 表示森林规模与所有决策树单独进行样本预测均方根误差(RMSE)的方差的关系。

图 3.7 表示森林规模与基础 RF 进行样本预测均方根误差(RMSE)平均值的关系。

在图 3.5、3.6、3.7 中，统计了决策树个数从 10-100 时的相关数据，其他参数保持不变：每棵树训练的特征个数为 3，其他超参数的值分别为可以取的最低值。

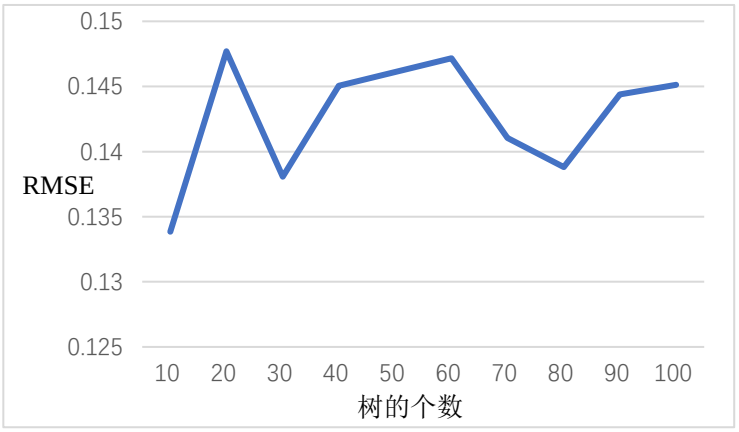

图 3.5 森林规模与所有决策树预测均方根误差平均值的关系

Figure 3.5 Relationship between forest size and the average predicted RMSE of all single decision trees

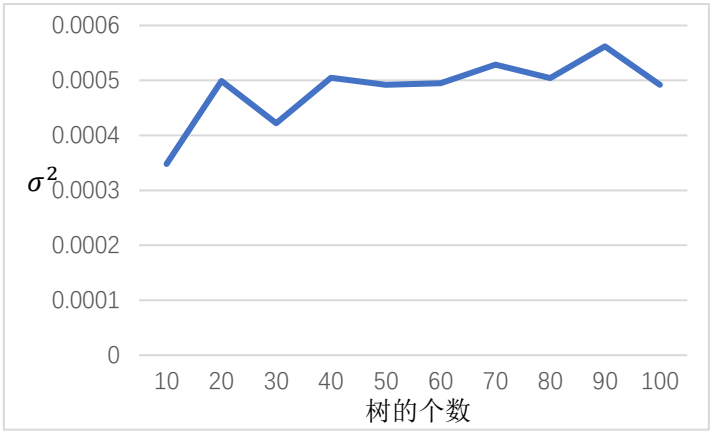

图 3.6 森林规模与所有决策树预测均方根误差的方差的关系

Figure 3.6 Relationship between forest size and the predicted RMSE variance of all single decision trees

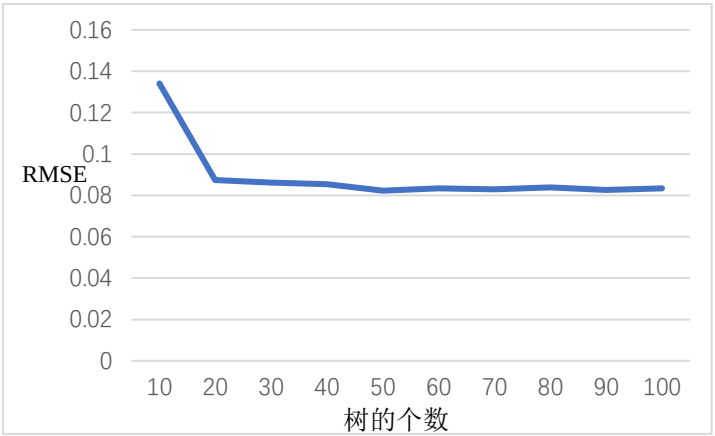

图 3.7 森林规模与基础 RF 预测均方根误差平均值的关系

Figure 3.7 Relationship between forest size and the average predicted RMSE of basic RF

由以上数据，当森林中树的个数增多时，单颗树的决策能力总体呈上升趋势；但方差增大说明随机森林的预测稳定性总体上呈下降趋势。而对于基础 RF 的预测能力：当决策树个数在 10-20 时，预测能力快速提升；决策树个数在 20-50 时，预测能力有缓慢提升；而决策树个数在 50 以上时，预测能力基本不变。

综上分析，使随机森林达到最佳效果的决策树个数应该在 20-50 之间。

3.2.3 采样次数对模型性能的影响

采样次数同样会对随机森林的学习能力造成影响。在章节 3.1.3 提到，基础随机森林模型在为每棵树分配样本时采用简单无放回随机抽样的办法。如果抽样次数增大，那么就有越多的样本可能被抽到当作训练集参与训练。这会造成两方

面的影响：第一，每棵树的训练样本增多会使每棵树的学习能力更强，对整体集成学习起到积极作用。这一点对于绝大多数对样本敏感的生成式模型而言都是如此，随机森林也是生成式学习模型的一种。第二，每棵树的训练样本增多意味着更多的样本被选择，使得树之间选择到重复样本的风险增大，导致树之间的相似性更强，对集成学习带来负面影响。两方面同时作用，导致很难通过理论方法分析出对于特定模型而言最有利的采样次数，只能通过实验方法进行分析。

图 3.8 统计了采样次数与所有决策树单独进行样本预测均方根误差(RMSE)平均值的关系。

图 3.9 表示采样次数与所有决策树单独进行样本预测均方根误差(RMSE)的方差的关系。

图 3.10 表示采样次数与基础 RF 进行样本预测均方根误差(RMSE)平均值的关系。

在图 3.8、3.9、3.10 中，统计了采样次数相对于测试集容量（50 个样本）的倍数（以下简称采样次数倍数）从 1-5.5 时的相关数据，其他参数保持不变：每棵树训练的特征个数为 3，其他超参数的值分别为可以取的最低值。

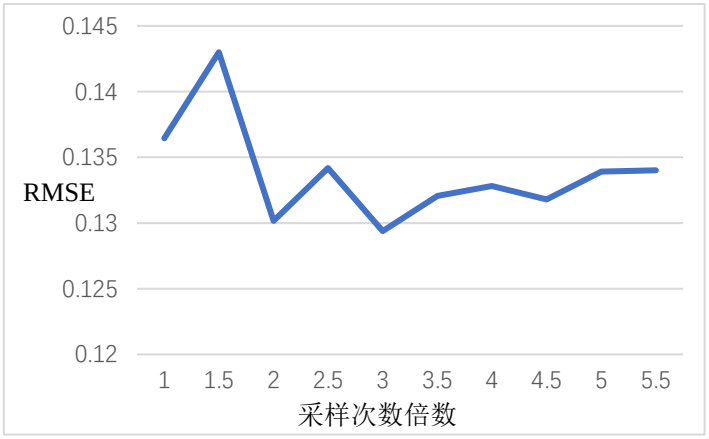

图 3.8 采样次数倍数与所有决策树预测均方根误差平均值的关系
Figure 3.8 Relationship between sampling times and the average predicted RMSE of all single decision trees

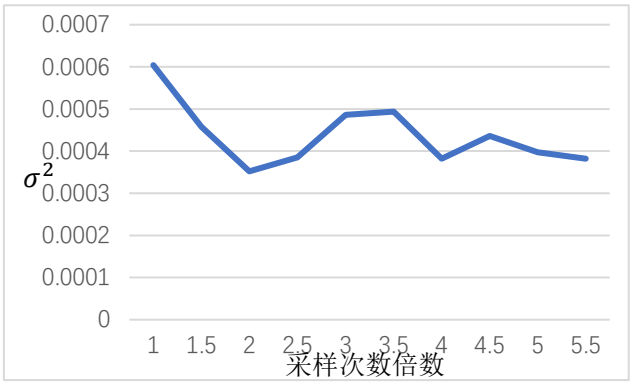

图 3.9 采样次数倍数所有决策树预测均方根误差的方差的关系

Figure 3.9 Relationship between sampling times and the predicted RMSE variance of all single decision trees

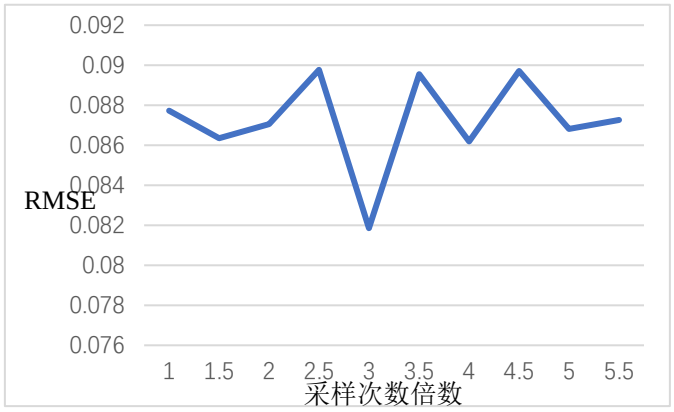

图 3.10 采样次数倍数与基础 RF 预测均方根误差平均值的关系

Figure 3.10 Relationship between sampling times and the average predicted RMSE of basic RF

由以上数据，当采样次数增加时，单颗决策树的决策能力和森林中所有决策树的预测稳定性总体上呈先上升后下降的趋势。而对于基础 RF 的预测能力：当采样次数是测试集容量的 2.5-3.5 倍时的决策能力最强。

综上分析，使随机森林达到最佳效果的采样次数倍数应该在测试集容量的 2.5-3.5 倍之间。

3.2.4 特征子集容量对模型性能的影响

与采样次数类似，特征子集容量作为随机森林算法最重要的超参数之一，同样对于模型的性能有两方面影响。当特征子集容量增大时：一方面，每颗决策树将会用到更多的特征进行训练，如果这些特征是与输出具有较高相关性的优质特征，那么这会大大提高模型的学习能力。这一点在很多学习模型的文献中常用提及；另一方面，被用于训练的特征增多，导致树之间选择到相同特征的概率会大大提高，提高了树之间的相似性，对模型的学习能力产生负面影响。应该注意到

的是，一般来说，数据特征的数量是远小于数据样本数量的。那么当指定选择数量增大时，选择到相同特征的风险会远大于选择到相同样本的风险。因此，特征子集容量不应过大，这也可以说明为什么 Breiman 建议随机森的使用者运用对数函数来选择特征子集容量。下面用数据分析特征子集容量变化对模型性能带来的影响。

图 3.11 统计了特征子集容量与所有决策树单独进行样本预测均方根误差(RMSE)平均值的关系。

图 3.12 表示特征子集容量与所有决策树单独进行样本预测均方根误差(RMSE)的方差的关系。

图 3.13 表示特征子集容量与基础 RF 进行样本预测均方根误差(RMSE)平均值的关系。

在图 3.11、3.12、3.13 中，统计了特征子集容量从 1-10 时的相关数据，其他参数保持不变，均为可取到的最小值。

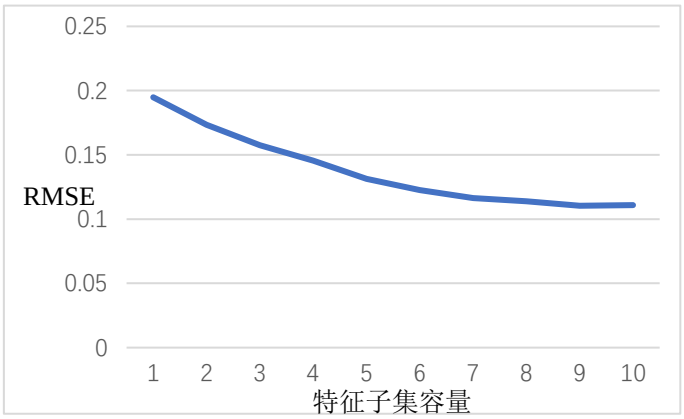

图 3.11 特征子集容量与所有决策树预测均方根误差平均值的关系

Figure 3.11 Relationship between characteristic subset capacity and the average predicted RMSE of all single decision trees

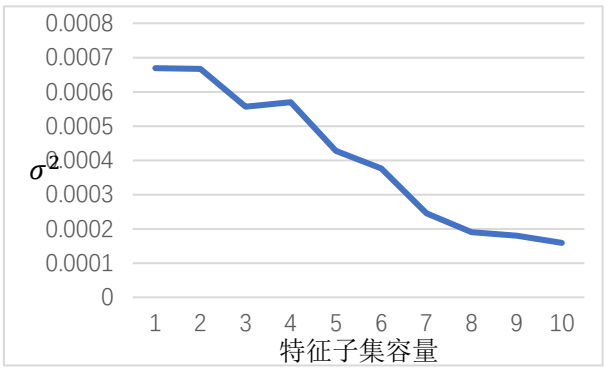

图 3.12 特征子集容量与所有决策树预测均方根误差的方差的关系

Figure 3.12 Relationship between characteristic subset capacity and the predicted RMSE variance of all single decision trees

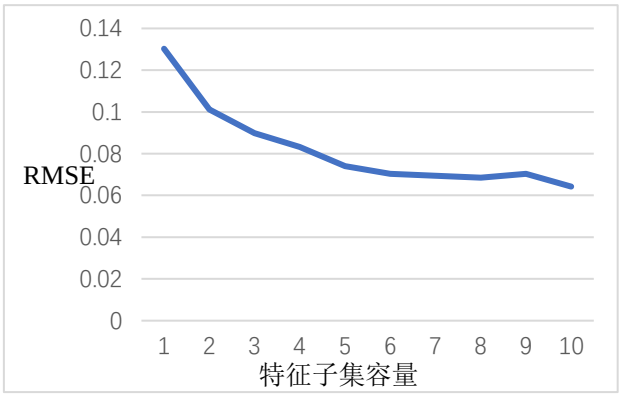

图 3.13 特征子集容量与基础 RF 预测均方根误差平均值的关系

Figure 3.13Relationship between characteristic subset capacity and the average predicted RMSE of basic RF

由以上数据，当特征子集容量增加时，单颗决策树的决策能力和森林中所有决策树的预测稳定性呈现出明显的上升趋势。与单颗决策树的实验结果类似，随着特征子集容量的增加整体 RF 的预测能提高的速度先快后慢，当特征子集容量达到 6 以后（训练集特征总数的一半），预测能力的提升不明显。

据此推测，使随机森林达到最佳效果的特征子集容量应该在数据总特征数的一半左右。

3.2.5 最大层深对模型性能的影响

最大层深指每棵树在分裂时的分裂层数上限。由于最小二乘树的节点分裂机制（详见章节 3.1.3），每一个中间节点都会向下分裂出两个节点。因此，一颗层深为d的最小二乘树最多可能的节点数为2^d个。实际上，最大层深可以看成时一个决策树的退出条件：当算法会统计决策树当前层数，当层数大于d时，将当前所有的节点设置为叶子节点并退出训练。由于最大层深不涉及决策树之间的训练

机制，因此这一超参数只与单颗决策树的学习能力有关。然而其对单颗决策树学习能力的影响并不容易研究。因为这种影响同样也是双方面的！首先，层深的增加可以使决策树对样本的利用更加充分，对决策树的精度有所提升；然而，对训练集的预测精度并不一定代表学习能力就更强，学习能力的重要体现在于泛化能力(Generalization Ability)，即模型对陌生的测试集的预测精度。如果层深过大，容易过度学习训练样本，反而在测试集上表现不佳，这就是机器学习中常被研究的一个及其重要的问题：过拟合(Overfitting)。因此，层深的选择同样很难根据理论推导得出，下面通过实验分析最大层深对模型性能的影响。

图 3.14 统计最大层深与所有决策树单独进行样本预测均方根误差(RMSE)平均值的关系。

图 3.15 表示最大层深与所有决策树单独进行样本预测均方根误差(RMSE)的方差的关系。

图 3.16 表示最大层深与基础 RF 进行样本预测均方根误差(RMSE)平均值的关系。

在图 3.14、3.15、3.16 中，统计了最大层深从 10-19 时的相关数据，其他参数保持不变：每棵树训练的特征个数为 3，其他超参数的值分别为最低值。

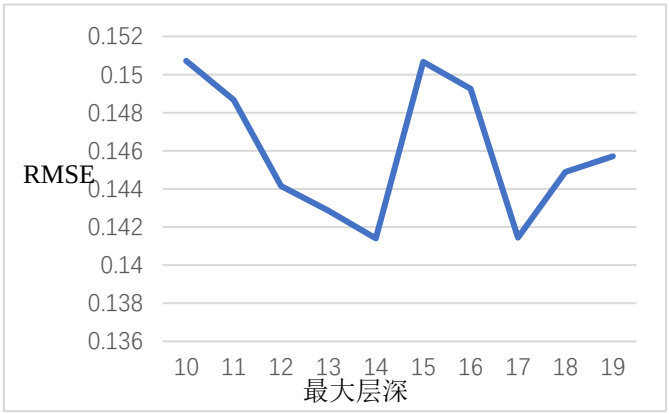

图 3.14 最大层深与所有决策树预测均方根误差平均值的关系

Figure 3.14 Relationship between maximum layer depth and the average predicted RMSE of all single decision trees

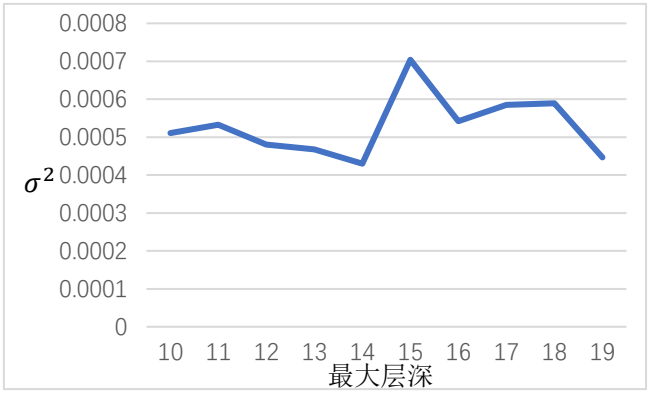

图 3.15 最大层深与所有决策树预测均方根误差的方差的关系

Figure 3.15 Relationship between maximum layer depth and the predicted RMSE variance of all single decision trees

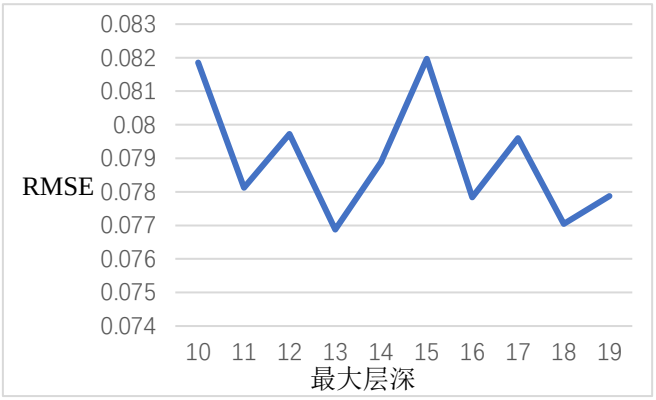

图 3.16 最大层深与基础 RF 预测均方根误差平均值的关系

Figure 3.16 Relationship between maximum layer depth and the average predicted RMSE of basic RF

由以上数据，当最大层深增加时，单颗决策树的决策能力和森林中所有决策树的预测稳定性表现出不确定性，没有明显的上升或下降趋势。与单颗决策树的实验结果类似，整体 RF 的预测能力随最大层深变化依旧不确定。

综上分析，最大层深与单颗决策树预测能力及稳定性关系不大，与整体 RF 的预测能力关系不大。

3.2.6 叶子节点包含最少样本数对模型性能的影响

与最大层深类似，叶节点包含最少样本数也是决策树分裂的中止条件之一。与层深参数只限制分裂的最后一层不同，该超参数可以限制决策树的任一个节点的分裂。通过限制每个节点的最少样本数来控制该节点是否设定为叶子节点：如果某一节点在分裂后包含的样本数小于该参数设定的值，则立即将此节点设置为叶子节点，停止继续分裂。这种机制的好处是显而易见的，如果每一个节点都向

下分裂到不能再分裂为止（即节点只包含一个样本）而不做任何限制的话，则此时学习到的模型明显是对训练集的一个过学习。这种对每一个节点是否向下分裂的控制机制与限制层深的作用类似，都可以起到防止模型出现过拟合，从而提高泛化能力。由于叶子节点包含最小样本数会影响到决策树中所有节点的分裂情况，因此这个超参数对模型的限制作用远大于最大层深，因此也是十分难调整的超参数。

图 3.17 统计了最少叶节点样本数与所有决策树单独进行样本预测均方根误差(RMSE)平均值的关系。

图 3.18 表示最少叶节点样本数与所有决策树单独进行样本预测均方根误差(RMSE)的方差的关系。

图 3.19 表示最少叶节点样本数与基础 RF 进行样本预测均方根误差(RMSE)平均值的关系。

在图 3.17、3.18、3.19 中，统计了最少叶节点样本数从 5-50 时的相关数据，其他参数保持不变：每棵树训练的特征个数为 3，其他超参数的值分别为可以取的最低值。

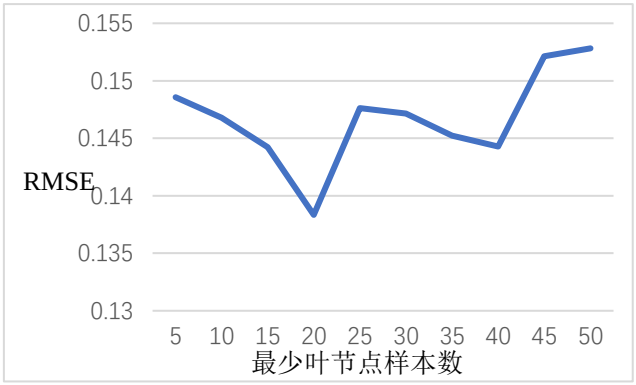

图 3.17 最少叶节点样本数与所有决策树预测均方根误差平均值的关系

Figure 3.17 Relationship between minimum leaf samples and the average predicted RMSE of all single decision trees

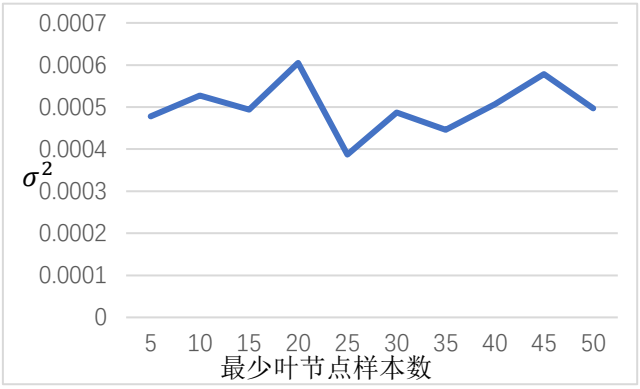

图 3.18 最少叶节点样本数与所有决策树预测均方根误差的方差的关系

Figure 3.18 Relationship between minimum leaf samples and the predicted RMSE variance of all single decision trees

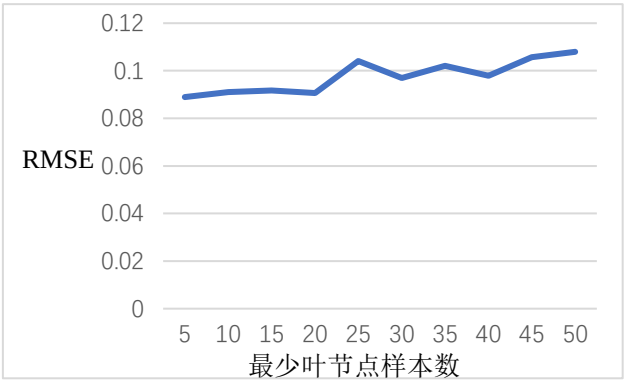

图 3.19 最少叶节点样本数与基础 RF 预测均方根误差平均值的关系

Figure 3.19 Relationship between minimum leaf samples and the average predicted RMSE of basic RF

由以上数据，当最少叶节点样本数增加时，单颗决策树的决策能力和森林中所有决策树的预测稳定性总体上呈先上升后下降的趋势；当最少叶节点样本数在 20-25 之间时单颗决策树的预测能力及稳定性达到最佳。而对于基础 RF 的预测能力：当最少叶节点样本数处于 5-20 之间时，整体 RF 预测能力变化不明显，当最少叶节点样本数达到 25 以上时，整体 RF 预测能力下降明显。

综上分析，使随机森林达到最佳效果的最少叶节点样本数应该在 15-25 之间。

3.3 本章小结

本章的主要工作主要分为两方面：

（1） 从算法原理层面介绍了随机森林算法的相关内容，具体包括四种不同的决策树算法：ID3、C4.5、CART 分类树以及 CART 回归树（使用最小二乘策略）；两类集成学习策略：串行训练集成（Boosting）以及并行训练集成（Bagging）；

最后介绍了限制随机森林模型性能的两方面因素：单颗决策树学习能力以及树间的相关联程度。

在分析算法原理的基础上，搭建了可以直接用作训练的基础随机森林模型（如图 3.4），为后续章节的实验分析提供模型。

（2） 在搭建的基础随机森林模型的结构上具体分析了 5 个模型超参数（森林规模、采样次数、特征子集容量、最大层深、最少叶子节点包含样本数）的意义并通过数据分析说明它们对模型性能产生的影响；通过实验分析缩小了超参数可选择的区间范围。需要说明的是，以上针对 5 个超参数的实验中，所有实验数据均经过在相同条件下重复训练、测试 6 次取平均值处理，因此数据具有一定的普遍适应性。通过分析实验数据，得到如下结论：

为了尽可能接近随机森林的最佳预测效果：森林规模应在 20-50 之间；采样次数倍数应该在测试集容量的 2.5-3.5 倍之间；特征子集容量应该在数据总特征数的一半左右；最少叶节点样本数应该在 15-25 之间；而最大层深与单颗决策树预测能力及稳定性关系不大，与整体 RF 的预测能力关系不大。

4 随机森林算法预报模型的改进研究

通过第三章的论述，目前已经完成了基础随机森林模型的搭建（下文称之为基础 RF），并初步分析了各超参数对模型的影响情况。然而仔细分析图 3.4 中的模型结构，难免会发现其中存在着一些不足和漏洞。本章 4.1 节首先直观地分析基础模型结构的一些缺陷，进而在章节 4.2、章节 4.3 中将针对模型存在的局限提出并设计具体的改进方案。

4.1 基础 RF 预报模型算法局限性分析

通过在编程时的一些各模块简单试验以及对基础模型结构的分析，本节主要从以下几个方面论述基础 RF 模型的局限性：采样环节存在的漏洞、简单平均结合策略可能的不足以及超参数的选择。希望通过数学推导结合常识判断来解释原有模型中可能存在的问题。

（1） 采样环节存在的漏洞：

基础 RF 模型中对训练样本采用有放回的简单随机抽样法即经典的 Bootstrap 方法，这种方法有很多显而易见的好处：除了在编写程序时极易实现，复杂度非常低以外，最重要的优点在于对于某一样本来说，有放回地简单抽样可以使其在抽样过程中被抽到的概率始终不变，即某一样本被抽到的概率与当前抽样次数无关，其概率为：

$$p=\frac{1}{n} \tag{4.1}$$

式中，p为训练集中某样本的被采概率，n为训练集容量。

而更复杂的无放回抽样就无法保证这一点，其被采概率为:

$$p=\frac{1}{n-(s-1)}\ (s=1,2\ldots n) \tag{4.2}$$

式中，s为当前采样的次数。

可以看到，其被采概率会随着当前采样次数的增加而增减。直观上讲，进行无放回采样时某些样本会更容易被抽到，而有些样本则更不容易被抽到。这就改变了样本原有的分布情况(Distribution)，导致训练集中出现了模型偏好权重(Preference Weight)。在有些情况下，偏好权重的出现有利于模型的学习，比如说

在分类问题中正样本与负样本数量往往相差很大，甚至十倍以上。这种情况的处理方法之一是将负样本设置一定倍数的偏好权重以平衡正负样本的比重。但是在本文中的任务中，如果不可预知地、随意地给一些样本添加偏好权重对模型的学习毫无好处，甚至会有坏处：模型将会对这些被采概率稍大的样本产生偏好，这样的预测值就会向这些偏好样本附近靠拢，影响预测的准确度。

那么，如果我们采用 Bootstrap 就会避免偏好样本的出现吗？不幸的是，通过局部模块的实验，（原理十分简单，这里不具体列出实验细节）发现到的是 Bootstrap 仍然会导致偏好样本的出现。原因其实很简单，因为在进行简单有放回随机采样时，尽管保证了每次采样时每个样本的被采概率相同，但这同时也意味着已经被采到过的样本仍然可能被重复采集。于是采样后的训练集中就会出现相当数量的重复样本，而且每个样本重复的次数大多不同。实际上这相当于另一种形式的样本加权，与无放回采样类似，我们无法控制具体偏好权重和被加权的具体样本。如此，经典 Bootstrap 方法这样的简单有放回的随机采样显然不能满足实际的训练需要，应该对原始的 Bootstrap 稍作改进。

（2） 简单平均结合策略可能存在的不足：

基础 RF 中采用了如式 3.6 的简单平均结合策略来综合各个决策树的输出以得到整个森林的预测结果。这种做法对于回归问题而言是十分容易实现的，复杂度很低，而且对于各决策树学习能力相近的情况是十分有帮助的。然而，如果考虑以下情况：在森林中有一颗或几颗决策树学习能力或者泛化能力较其他决策树相差很多，甚至完全在输出随机数，类似“瞎猜”。那么此时仍然采用平均法进行结合，那么就可以把这棵树看成一段噪声信号，如果不做任何处理的话一定会对原本的信号造成影响，可能导致原本信号的方差增大，精度下降等等一些负面效果。

但跟信号处理领域或图像领域对噪声的处理算法不同，这里的“噪声”非常容易剔除，不需要复杂的滤波卷积或变换操作，只需设法将“噪声”树的权重降低，正常树的权重提高即可。其实从本质上讲这和滤波、卷积的原理是类似的。只不过信号处理领域的噪声辨识是一个较为困难的问题，这就使卷积核或者滤波器函数的选取较为麻烦。然而在本文的情况下，辨识到“噪声”树是容易的；在此基础上，选取所谓的“卷积核函数”也就容易了。

借鉴信号或图像领域中处理噪声的相关方法，这里我们可以将式 3.6 进行改进：

设森林中所有决策树的输出构成行向量：

$$\vec{O}=(output_1, output_2,...,output_t) \tag{4.3}$$

假设式 4.3 中有 i 个噪声分量，则应该构建一个权重列向量与之相乘达到将 i 个噪声分量消掉的目的：

$$\vec{W}=(w_1, w_2,...,w_t)^T \tag{4.4}$$

式 4.4 所列出的列向量在这里起到了滤波器的作用，那么森林的输出此时可以写成：

$$\text{output}=\vec{O}\times\vec{W} \tag{4.5}$$

注意到，其实简单平均法的中的$\vec{W}=(1/t, 1/t,..., 1/t)^T$实际上是一个所有权重分量都相同的特殊情况。除此以外，设计权重列向量时应该注意，为了保持滤波前后的线性性质不变，在回归问题中权重列向量中所有分量的和应该等于 1，这也是跟分类问题中的权重设计不同的地方之一。

（3） 超参数的选择：

超参数的选择过程也是机器学习、深度学习研究者常说的“调参”问题。一定程度上说，调参的好坏直接决定了模型训练的性能。如果直接采取手动调参，需要研究者对算法结构、内部运行机制关系以及参数间的相互作用十分了解，而且可能需要一些耗时的训练试验才能最终确定一组相对较优的参数。可见对很多经典的学习模型而言，调参都算作一项十分耗费时间又具有挑战性的工作。实际上，在很早就有人提出使用一些优化方法对神经网络的参数进行调整。而且随着智能优化算法概念的提出，使优化算法的寻优能力及效率远超过经典的优化方法。这为机器学习模型超参数的调整提供了一种方向。

因此，在使用智能优化算法来调整超参数时，需要重视的问题就变成优化算法的选择、优化策略的改进以及输入参数的规范化。本文选用了一种经典的智能算法——粒子群算法(Particle Swarm Optimization)作为超参数的优化算法。之所

以选用这种算法是因为：第一，PSO 算法的收敛代数较 GA 等算法更快；第二，PSO 不同于 GA，其可以直接使用实数编码对问题进行求解，不需要考虑编码问题对结果造成的影响；第三，算法数据结构较为简单，相对易于实现。有关 PSO 算法的具体介绍详见章节 4.3。

4.2 改进自助采样的树间加权随机森林模型算法设计

为了章节 4.1 中（1）和（2）的问题，本节提出了一种改进自助采样树间加权随机森林模型。(Trees Weighted Bootstrap Random Forest，下文简称 TWB-RF)

先来回顾上节（1）中的问题：在基础 RF 中，使用简单有放回随机抽样会使得样本出现意料之外的偏好权重且无法控制，使用这样的训练集训练每棵树明显会对结果造成偏差。

TWB-RF 中采用了改进的自助采样法(Advanced Bootstrap Sampling，下文简称 Bootstrap)来对训练集进行采样。其原理非常容易理解，在一般自助采样的基础上添加一个判断即可：即每次采样之后都对采到的样本进行检查，如果本次采样得到的样本与之前某次的相同，则不在训练集中添加之。如此经过一次完整的采样后，采样得到的训练集中就没有任何重复的样本了，自然就没有偏好权重了；除此以外，由于这种方法本质上仍然是有放回的采样，因此每个样本被采到的概率始终相同，不发生变化，在理想情况下可以保持采样前后的样本分布。这样为每颗决策树分配到的样本集才最适合用作训练，不会因采样环节的问题而产生额外偏差。

应该注意到的是，在进行一轮 Bootstrap 采样后，原样本集中总有一部分样本始终没被采集到，下面做一个简单的计算：

设 Bootstrap 的采样次数为n，则原样本集中某一样本始终没有被采到的概率显然为：

$$p=\left(1-\frac{1}{n}\right)^{n} \tag{4.6}$$

当采样次数n很大时，可对式 4.6 取如下的极限：

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{n} \tag{4.7}$$

根据高等数学知识，式 4.7 的解为e^{-1}，约等于 0.368。由此可以看出，在理

想情况下，不论采样次数多大，每轮采样总会有大约 36.8%的原始样本没有被采集到。（这里讨论这个问题是为了给即将介绍的树间加权策略提供支持）而在实际实验中，这个理想情况不容易达到；通过简单的单个模块试验，当采样次数值在训练集容量的 2.5 倍左右时，基本上每次采样后的样本集容量约为 36%。这是由于使用的产生随机数算法并不是产生绝对数学意义上的随机数，而是一种伪随机数，导致实际实验结果很难达到理论推导的效果。

在解决了采样环节的漏洞后，再来回顾上节（2）提到的问题：如果基础 RF 模型在训练后产生若干个“噪声”决策树，而此时仍然采用简单平均策略来综合决策树输出的话，将使训练后的模型精度下降且不稳定。在上节中也简单推导了一个简易滤波器的表达形式如式 4.4、4.5。

针对此问题，在 TWB-RF 中采用树间加权策略来构建式 4.4 中的权重列向量。思考式 4.4 中的向量可以看出，构造权重向量的关键在于如何得到可以表征各决策树学习能力的各权重分量。因此，各决策树对应的权重分量也应该作为模型的一部分进行训练；既然要进行训练，就需要有训练集，那么训练各树权重的训练集应该是什么呢？根据本节上文中的内容，在 Bootstrap 采样后，每棵树除了得到一部分训练样本之外，还剩余一部分样本没有使用，正好在这里可以使用这部分没有被 Bootstrap 采到的样本作为这颗树的测试集，在这棵树训练结束后不马上进行下一颗树的训练，而是先用这部分测试集评价这棵树的泛化能力，据此得出该树的权重。

TWB-RF 使用每颗决策树在测试集上的均方根误差(Root Mean Square Error)作为评价该树泛化能力的指标，均方根误差的具体计算方法详见章节 5.1。根据章节 4.1（2）所述，在回归问题中，为了保证输出结果的线性性质不变，所有树的权重的和必须为 1。而均方根误差是和模型的泛化能力成反比的，即模型泛化能力越强，均方根误差越低，反之类似。因此，TWB-RF 针对回归问题的这个特点设计了如下的权重计算函数：

$$w_i=\frac{1-\frac{R_i}{\sum_{s=1}^{t}R_s}}{t-1}=\frac{(\sum_{s=1}^{t}R_s)-R_i}{(t-1)\sum_{s=1}^{t}R_s} \tag{4.8}$$

式中，w_i表示第i颗树的权重，R_i表示第i颗树在对应测试集上的均方根误差，t表示森林中树的个数。通过简单推导不难发现，式 4.8 中的w_i与R_i成反比，且

$\sum_{i=1}^{t} w_i$=1，因而满足以上对权重计算的两点要求。

在每棵树训练完成后立即按 4.8 计算每棵树的权重，组成式 4.5 所示的权重向量，作为训练结果的一部分。在预测时用每棵树的输出向量与权重向量相乘，即可得到森林的输出结果：

$$\text{output}=\vec{O}\times\vec{W}=\sum_{i=1}^{t} w_i \times output_i \tag{4.9}$$

TWB-RF 模型的算法流程框图如图 4.1 所示：

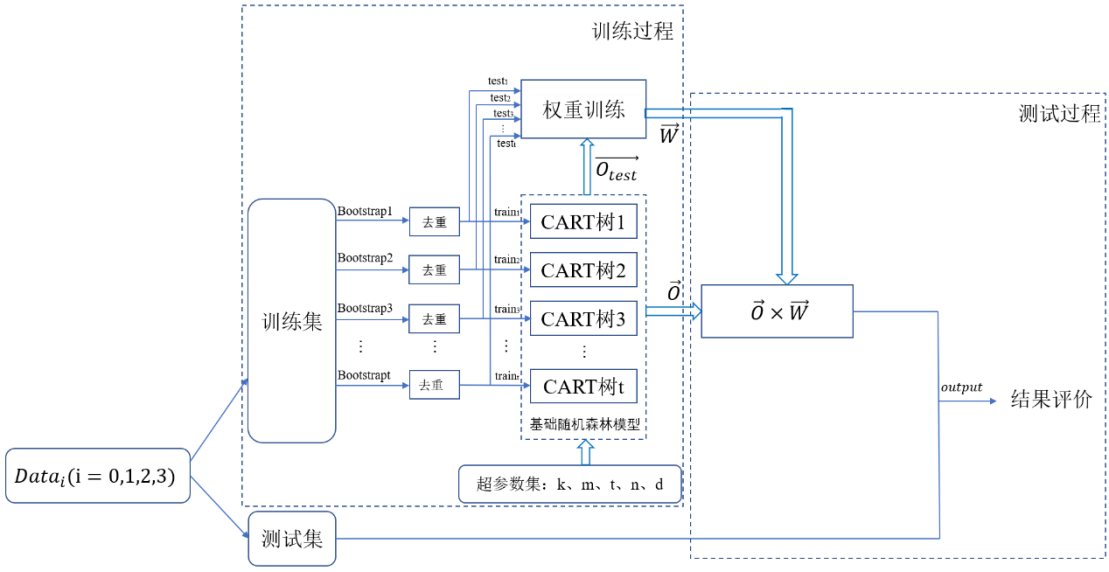

图 4.1 TWB-RF 模型算法流程框图

Figure4.1 Flow diagram of TWB-RF model

图中，超参数 k、m、t、n、d 分别代表特征子集容量、叶节点包含最少样本数、森林规模、采样次数、最大层深；权重训练模块公式如式 4.8 列出；train_i、test_i 分别代表去重之后每棵树的训练样本和测试样本；$\overrightarrow{O_{test}}$为各决策树对于各自测试样本的响应。

至此，可以 TWB-RF 模型已搭建完成，章节 5.3.1 中将对 TWB-RF 模型预报的结果进行对比实验。

4.3　基于改进粒子群算法的随机森林模型结构设计

4.3.1　基础粒子群算法简介

智能优化算法是在传统优化的基础上，受自然界某些现象的启发下而发明发展起来的。这一类算法不主张完全使用传统优化方法通过推导求解损失函数在特

定约束下的解析方程来寻找待优化参数（比如最小二乘优化、极大似然估计、拉格朗日乘数法等等），因而这类算法的使用不需要很深的数学功底，而且实用性较强，所有的约束条件和目标函数几乎都可以使用。而在传统优化算法中，在面对复杂的约束条件或目标函数，往往无法推导损失函数的解析表达式，导致方法无法使用。智能优化算法又可分为很多种，比如，简单搜索迭代：贪婪搜索、邻域搜索、禁忌搜索等；模拟物理过程：模拟退火算法等；仿生算法：遗传算法、神经元网络等；群体智能算法：粒子群算法、蚁群算法、鱼群算法等。

粒子群优化算法(Particle Swarm Optimization，下文简称 PSO)是群体智能优化算法的一种。是由美国社会心理学家 James Kennedy 和电气工程师 Russell Eberhart 在 1995 年在 Frank Heppner 建立的鸟群飞行计算机仿真模型的基础上共同提出的。算法通过在计算机中模拟自然界中鸟群集体觅食现象提炼出的数学律建立仿生模型用来完成一类在一定规模的解空间中中寻找一个最优解的典型优化任务。在 PSO 中，自然中的鸟个体被抽象成“粒子”(Particle)，每一个粒子都是针对最优化问题的一个潜在解(Solution)；最优化问题的整个解空间被模拟成“粒子群”(Particle Swarm)的搜索域(Search Space)；自然界中鸟群食物的分布规律被抽象成“适应度函数”(Fitness Function)；而每只鸟离“最优食物”(Best Solution)的距离被抽象成粒子的“适应度”(Fitness Value)；除此之外，每个粒子还被设定位置向量(Position Vector)和速度向量(Velocity Vector)，随着求解过程不断迭代。PSO 的迭代求解过程如下：

首先将种群初始化为一群随机分布的粒子。在每一轮迭代中，每一个粒子都按照两个当前最优值为标尺，按照一定的规律更新自己的位置：一个是个体当前找到的最优解，称为“个体最优解”；另一个是种群当前找到的最优解，称为“种群最优解”。最终，随着若干轮次的迭代输出算法的结果。图 4.2 为基础 PSO 的算法工作流程。

从图中可以看出，PSO 算法的工作流程较为简单，仅仅需要描述好个体的运动行为，那么整个群体就可以在搜索域中找到一组优秀的解。这也充分体现出了群体智能优化的思想：尽管个体的智能是简单笨拙的，但若干个这样的群体所体现出的智能不仅仅是简单的加和关系来计算的。已有大量的实验表明，PSO 算法的收敛速度较快，同时适合用于机器学习算法调参等任务。但 PSO 也有不足之

处，在于其收敛速度过快，容易导致算法“早熟”，使算法后续寻优速度过慢，容易陷入“局部最优”(Local Optimum)，这也是最优化领域中最受关注的问题之

算法 4.1：基础 PSO 算法

输入： s 维待优化参数集 $\mathbf{P=\{\theta_1,\theta_2,\theta_3,\ldots,\theta_s\}}$；

最优化问题的适应度函数 $\mathbf{\Omega(P)}$；PSO 参数 $\mathbf{w,r_1,r_2}$

步骤：

1：初始化粒子群： 设定种群规模 **N**；设定社群规模 **n**；

每个粒子的初始位置向量 $\mathbf{x_1^s,x_2^s,x_3^s,\ldots,x_N^s}$；

每个粒子的初速度向量 $\mathbf{v_1^s,v_2^s,v_3^s,\ldots,v_N^s}$；

个体与全局最优解 $\mathbf{Pbest^s=0;Gbest^s=0}$；

2：**while** 退出条件不成立，**do**：

3： 计算每个粒子的适应度：$\mathbf{f_i^s=\Omega(x_i^s),\ i=1,\ldots,N}$

4： **for** $\boldsymbol{i}$ **=1: n，do:**

5： **if** $\mathbf{f_i^s<f(Pbest^s)}$ **，do：**

6： $\mathbf{Pbest^s=x_i^s}$

7： **if** $\mathbf{f_i^s<f(Gbest^s)}$ **，do：**

8： $\mathbf{Gbest^s=x_i^s}$

9： **end for**

10： **for i =1:N，do:**（更新每个粒子的位置向量和速度向量）

11： $$v_i^s=w\times v_i^s+r_1\times\left(Pbest^s-f_i^s\right)+r_2\times\left(Gbest^s-f_i^s\right) \quad (4.10)$$

12： $$x_i^s=x_i^s+v_i^s \quad (4.11)$$

13： **end for**

14：**end while**

输出： 全局最优解 $\mathbf{Gbest^s=\{\theta_1,\theta_2,\theta_3,\ldots,\theta_s\}}$

图 4.2 基础 PSO 算法工作流程图

Figure 4.2 Workflow of basic PSO algorithm

图 4.2 中，个体位置x_1^s均为 s 维的向量，每个分量代表粒子在该参数分量上的值。需要输入的 PSO 参数中，w为个体的惯性因子（又称惯性权重），用来表示个体维持自身位置和速度意愿的强弱；r_1为个体的记忆因子，用来表示个体对

自己历史经历的记忆能力；r_2为个体的社群因子，用来表示个体向其所处社群学习的能力。这三个参数都是介于 0，1 之间的数。

本文将使用以图 4.2 为基本框架的 PSO 算法对 TWB-RF 模型中的 5 个待确定超参数进行优化（下文简称 PSOTWB-RF）。下面简要分析基础 PSO 中存在的局限，并介绍几种常见的 PSO 算法改进策略。需要说明的是，在图 4.3-图 4.13 中，用种群中所有粒子规范化前在某一维度上的平均移速来表示整个种群此时在该维度的移速；用种群中所有粒子当前代的均方根误差损失中位数来表示种群当前位置离最佳位置的距离，即为优化目标。

4.3.2 线性递减的惯性权重策略

由式4.10可以看到，惯性权重w起到控制粒子速度大小的作用，如果w越大，则粒子移动速度越快，算法将更快达到收敛，然而过快的收敛易引起算法的早熟，进而陷入局部最优。针对这一问题，Yuhui Shi[54]在 1998 年提出了一种具有线性递减惯性权重的改进粒子群算法。通过让w在迭代过程中逐渐递减，粒子群在前若干代可以保持较快的收敛速度，迅速收敛到目标附近；而在后若干代速度减慢，可以在目标附近“仔细”寻找，避免了因粒子速度过大而出现在目标附近大幅振荡的现象。下式刻画了惯性权重线性递减的数学性质：

$$w=w_{min}+\left(1-\frac{t}{t_{set}}\right)\times(w_{max}-w_{min}) \tag{4.12}$$

式中，w_{max}、w_{min}分别代表设定的最大权重和最小权重；t代表当前代数；t_{set}代表设定的惯性权重开始线性递减的代数；在本文中，$t_{set}=0.75t_{max}$，t_{max}为设定的最大代数。如此可见，在迭代过程中w将随着代数在最大最小权重之间线性均匀地减少。

图 4.3 描绘了线性递减的惯性权重和恒定惯性权重在算法算法迭代过程中的具体变化情况。

图 4.4-图 4.7 分别描绘了使用 2 种惯性权重策略时，粒子移速在 4 个不同分量上随代数增长的下降情况

图 4.8 描绘了使用 2 种惯性权重策略时，随机森林预报的均方根误差损失随代数的下降情况。

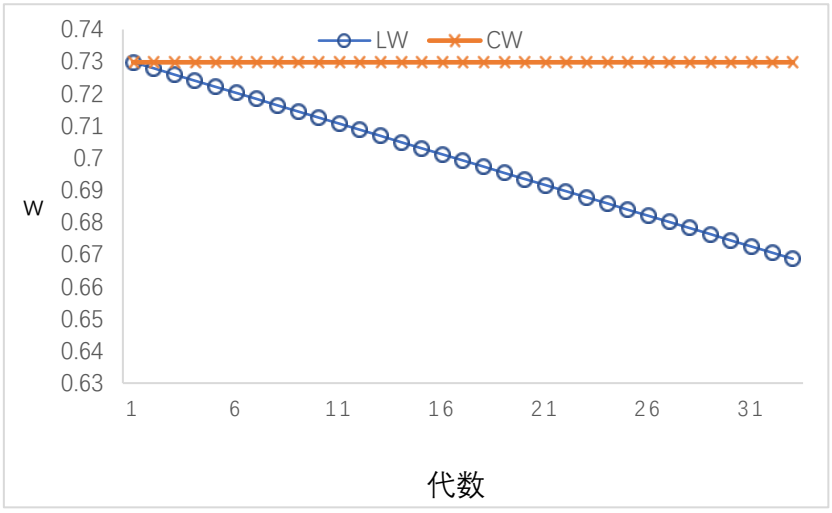

图 4.3 两种惯性权重策略的权重变化情况

Figure 4.3 Weight changes of two inertial weight strategies

LW-Linear Weight 线性递减惯性权重；CW-Constant Weight 恒定惯性权重

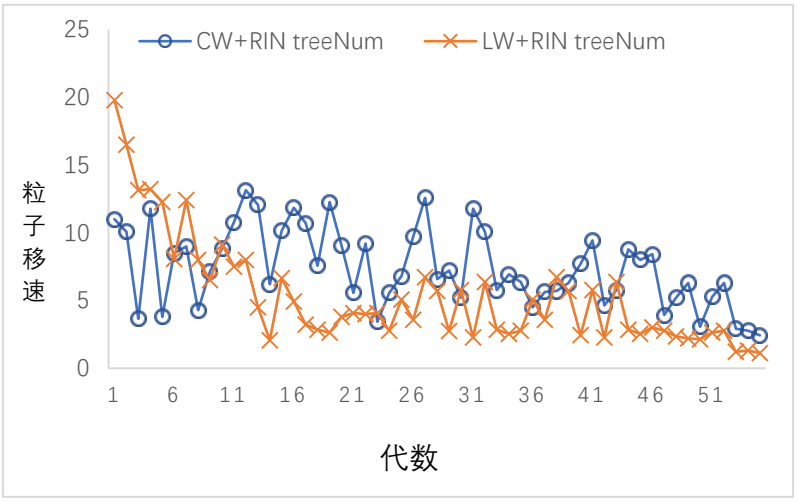

图 4.4 两种惯性权重策略在森林规模分量上的移速随代数变化关系

Figure 4.4 The velocity of two kinds of inertia weight strategies on the forest scale component varies with the algebra

CW+RIN-Constant Weight+Ring Topology Neighborhood 恒定惯性权重+环形邻近拓扑学邻域；LW+RIN-Linear Weight+Ring Topology Neighborhood 线性递减惯性权重+环形邻近拓扑学邻域

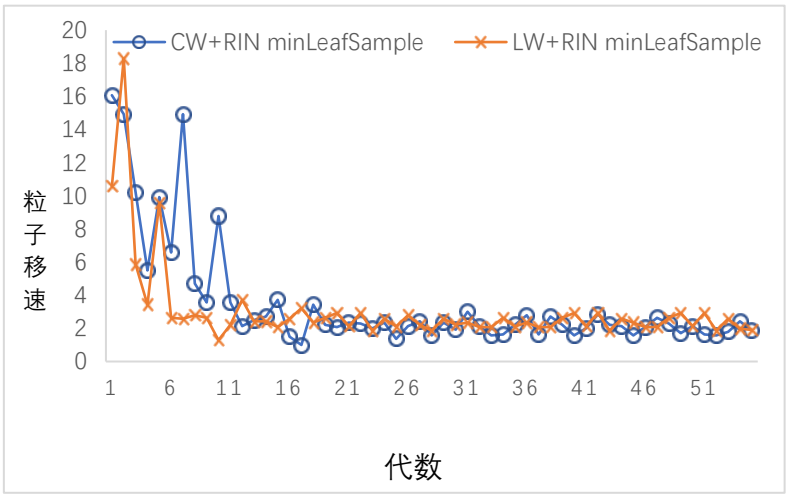

图 4.5 两种惯性权重策略在最少叶子节点样本个数分量上的移速随代数变化关系

Figure 4.5 The velocity of two kinds of inertia weight strategies on the minimum sample of leaf samples component varies with the algebra

CW+RIN-Constant Weight+Ring Topology Neighborhood 恒定惯性权重+环形邻近拓扑学邻域；LW+RIN-Linear Weight+Ring Topology Neighborhood 线性递减惯性权重+环形邻近拓扑学邻域

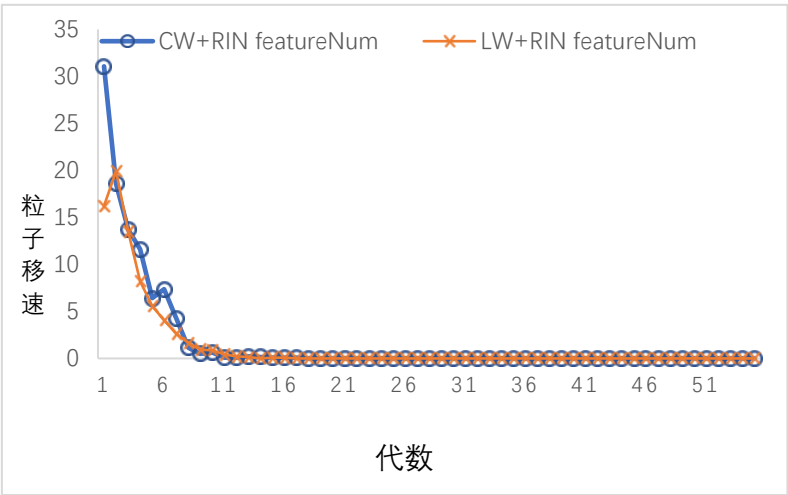

图 4.6 两种惯性权重策略在特征子集容量分量上的移速随代数变化关系

Figure 4.6 The velocity of two kinds of inertia weight strategies on the characteristic subset capacity component varies with the algebra

CW+RIN-Constant Weight+Ring Topology Neighborhood 恒定惯性权重+环形邻近拓扑学邻域；LW+RIN-Linear Weight+Ring Topology Neighborhood 线性递减惯性权重+环形邻近拓扑学邻域

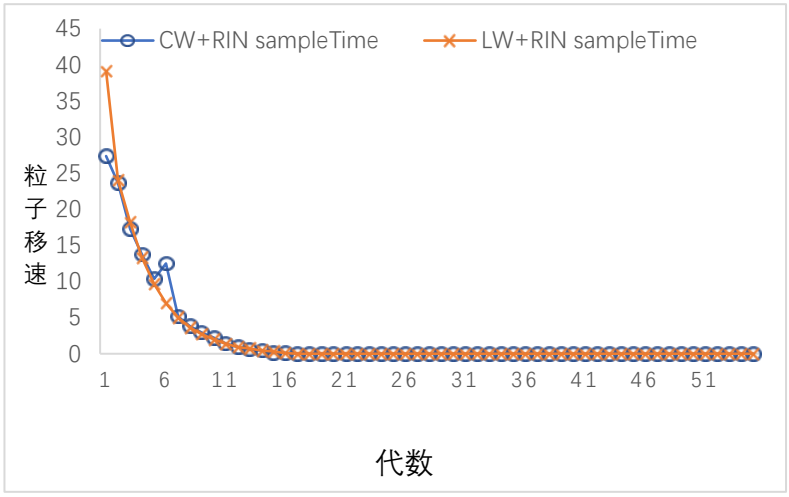

图 4.7 两种惯性权重策略在采样次数分量上的移速随代数变化关系

Figure 4.7 The velocity of two kinds of inertia weight strategies on the sample times component varies with the algebra

CW+RIN-Constant Weight+Ring Topology Neighborhood 恒定惯性权重+环形邻近拓扑学邻域；LW+RIN-Linear Weight+Ring Topology Neighborhood 线性递减惯性权重+环形邻近拓扑学邻域

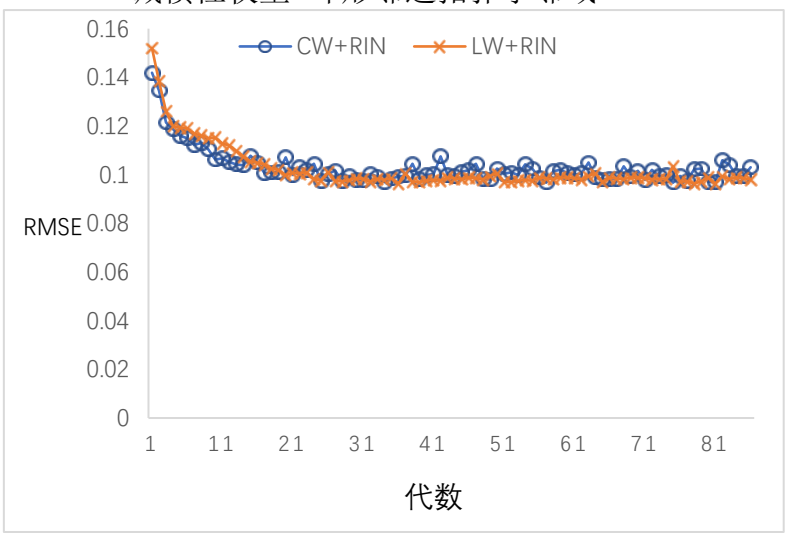

图 4.8 两种惯性权重策略下均方根误差损失随代数的下降关系

Figure 4.8 The decreasing relation of root mean square error loss with algebra under two inertial weight strategies

CW+RIN-Constant Weight+Ring Topology Neighborhood 恒定惯性权重+环形邻近拓扑学邻域；LW+RIN-Linear Weight+Ring Topology Neighborhood 线性递减惯性权重+环形邻近拓扑学邻域

综合分析以上曲线可以看出，当采取恒定惯性权重时，粒子的平均移速比线性递减惯性权重时的粒子平均移速略大，容易造成曲线波动。这一点在代数较小时并不明显；而当代数较大，算法基本达到收敛时，平均移速和损失中位数的波

动明显更强烈，这可能导致算法不稳定。

4.3.3 种群邻域交流策略

4.3.1 中介绍到，r_2作为个体的社群因子，表示个体向其所处社群学习的能力。意味着每一个粒子都直接和社群(Neighborhood)交换信息，社群定义的不同将直接影响$Gbest^s$值。那么每个粒子所处的社群应该如何定义呢？根据通常的理解，可以注意到有三种定义方式：（1）每个粒子都与种群中其他所有的粒子构成一个社群；（2）每个粒子只与其紧邻的粒子构成社群；（3）每个粒子随机地与种群中其他任意位置的若干粒子构成种群。

为了更好地表示并实现这三种社群交流策略，先介绍本文使用的交流矩阵定义：交流矩阵C_i表示第 i 个粒子与种群中其他粒子的交流关系，即组成社群情况。C_i中的第 i 行第 j 列上的值表示第 i 个粒子与种群中第 j 个粒子是否存在交流关系。若其值为 1，表示二者进行交流，在同一社群中；若其值为 0，则表示二者不进行交流，不在同一社群中。

（1） 全局邻域(Global Neighborhood)：

全局邻域的策略十分简单，种群中所有的个体共同组成一个社群，种群即社群，共同分享信息。在每次迭代中$Gbest^s$值为种群所有个体的最佳值。式 4.13 表示当种群数目为 5 时，第 1 个粒子的交流矩阵C_1。可见第一个粒子与种群中另外四个粒子都发生交流关系，构成同一社群。第 2、3、4、5 个粒子的情况类似，就不一一列出其交流矩阵了。

$$C_1=\begin{pmatrix}1&1&1&1&1\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\end{pmatrix} \tag{4.13}$$

（2） 环形邻近拓扑学邻域(Ring Topology Neighborhood)：

在这种策略下，认为种群中所有粒子依次排列成一个首位相接的环形，且只与自己和两侧紧临的粒子发生交流。式 4.14、4.15、4.16 分别表示当种群数目为 5 时，第 1、3、5 个粒子的交流矩阵C_1、C_3、C_5。通过对比这三个矩阵可以说明这种策略的交流机制。

$$C_1=\begin{pmatrix}1&1&0&0&1\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\end{pmatrix} \tag{4.14}$$

$$C_3=\begin{pmatrix}0&0&0&0&0\\0&0&0&0&0\\0&1&1&1&0\\0&0&0&0&0\\0&0&0&0&0\end{pmatrix} \tag{4.15}$$

$$C_5=\begin{pmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\1&0&0&1&1\end{pmatrix} \tag{4.16}$$

（3） 随机拓扑学邻域(Random Topology Neighborhood)：

在这种策略下，认为种群中所有粒子的排列在空间上不确定，完全随机地与自己和其他一定数量的粒子发生交流。式 4.17、4.18、4.19 分别表示当种群数目为 5，社群规模为 3 时，第 1、3、5 个粒子的交流矩阵C_1、C_3、C_5。

$$C_1=\begin{pmatrix}1&0&1&0&1\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\end{pmatrix} \tag{4.17}$$

$$C_3=\begin{pmatrix}0&0&0&0&0\\0&0&0&0&0\\0&1&1&0&1\\0&0&0&0&0\\0&0&0&0&0\end{pmatrix} \tag{4.18}$$

$$C_5=\begin{pmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&1&1&1&0\end{pmatrix} \tag{4.19}$$

尽管这种策略下某一粒子不与特定的粒子交换信息，但这也是三种策略中最接近自然鸟群飞行过程中的交流现象。

图 4.9-图 3.12 描绘了在线性递减惯性权重下，各邻域策略在 4 个维度中粒子移速随代数下降的关系。

图 4.13 描绘了在线性递减惯性权重下，使用各邻域策略时随机森林预报的

的均方根误差损失随代数的下降关系。

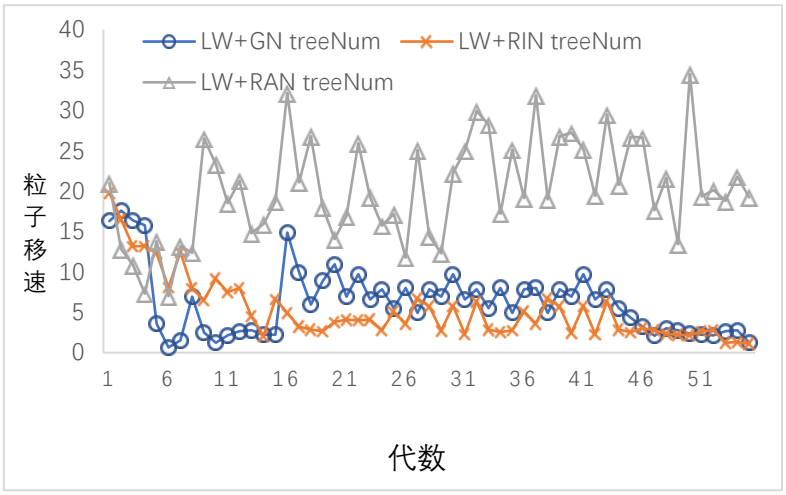

图 4.9 各邻域策略在森林规模维度中的粒子移速随代数下降关系

Figure 4.9 The particle velocity of each neighborhood strategy in the dimension of forest size decreases algebraically

LW+GN- Linear Weight+Global Neighborhood 线性递减惯性权重+全局邻域；LW+RIN-Linear Weight+Ring Topology Neighborhood 线性递减惯性权重+环形邻近拓扑邻域；LW+RIN-Linear Weight+Random Topology Neighborhood 线性递减惯性权重+随机拓扑邻域

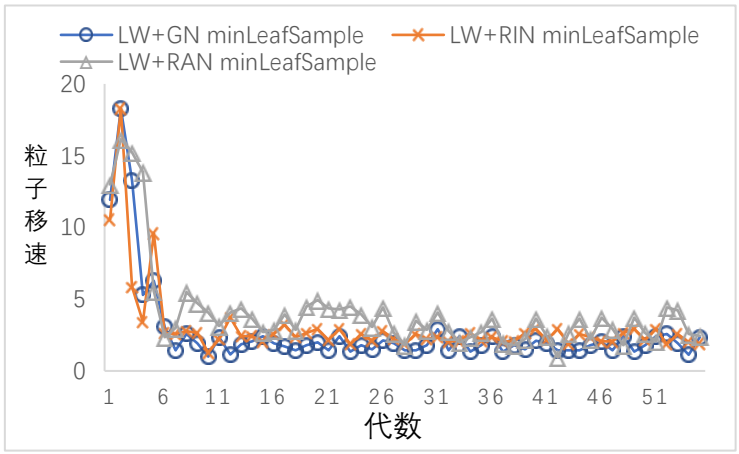

图 4.10 各邻域策略在最少叶子节点样本个数维度中的粒子移速随代数下降关系

Figure 4.10 The particle velocity of each neighborhood strategy in the dimension of the minimum sample of leaf samples decreases algebraically

LW+GN- Linear Weight+Global Neighborhood 线性递减惯性权重+全局邻域；LW+RIN-Linear Weight+Ring Topology Neighborhood 线性递减惯性权重+环形邻近拓扑邻域；LW+RIN-Linear Weight+Random Topology Neighborhood 线性递减惯性权重+随机拓扑邻域

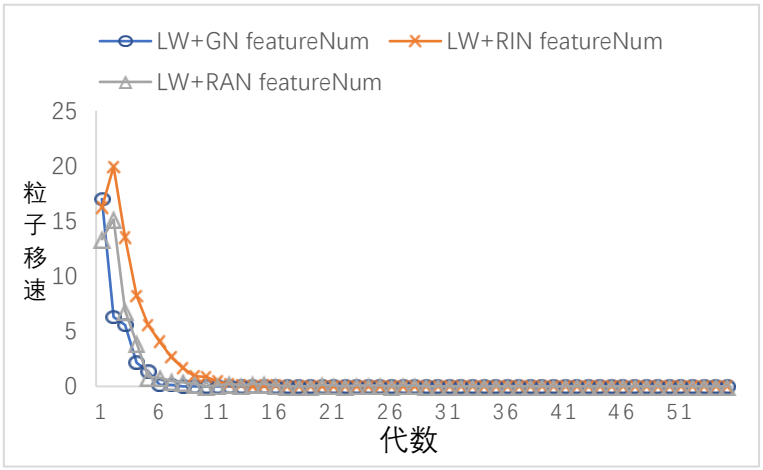

图 4.11 各邻域策略在特征子集容量维度中的粒子移速随代数下降关系

Figure 4.11 The particle velocity of each neighborhood strategy in the dimension of characteristic subset capacity decreases algebraically

LW+GN- Linear Weight+Global Neighborhood 线性递减惯性权重+全局邻域；LW+RIN-Linear Weight+Ring Topology Neighborhood 线性递减惯性权重+环形邻近拓扑邻域；LW+RIN-Linear Weight+Random Topology Neighborhood 线性递减惯性权重+随机拓扑邻域

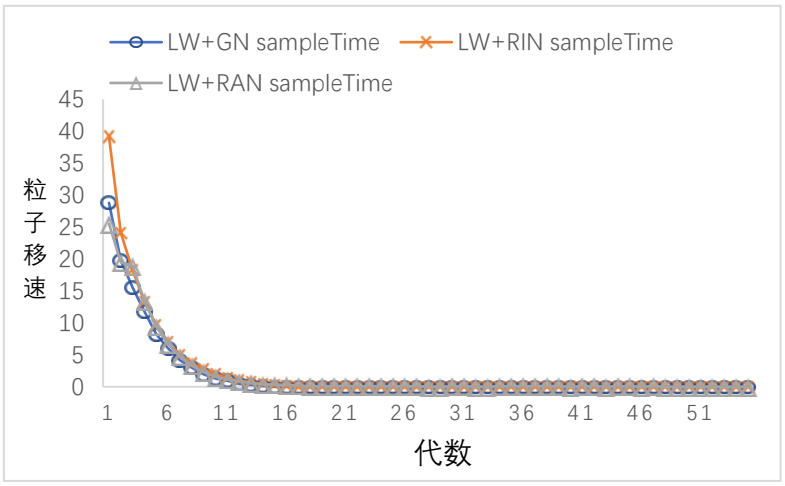

图 4.12 各邻域策略在采样次数维度中的粒子移速随代数下降关系

Figure 4.12 The particle velocity of each neighborhood strategy in the dimension of sample times decreases algebraically

LW+GN- Linear Weight+Global Neighborhood 线性递减惯性权重+全局邻域；LW+RIN-Linear Weight+Ring Topology Neighborhood 线性递减惯性权重+环形邻近拓扑邻域；LW+RIN-Linear Weight+Random Topology Neighborhood 线性递减惯性权重+随机拓扑邻域

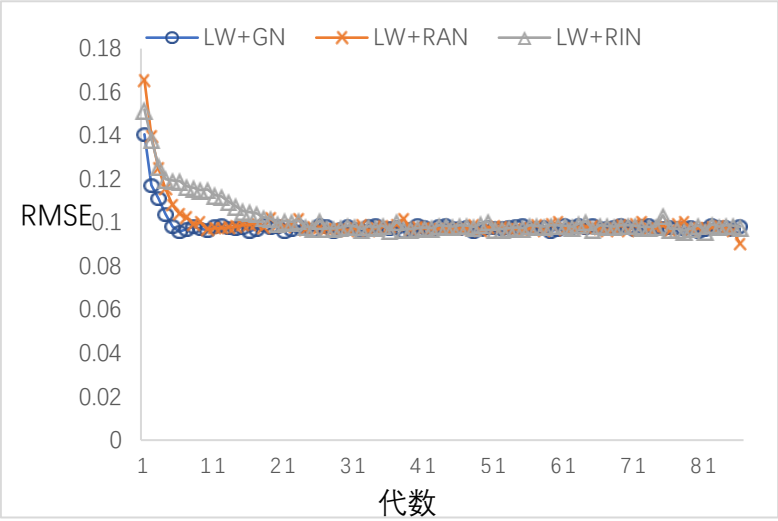

图 4.13 不同策略下均方根误差损失随代数的下降关系

Figure 4.13 Decreasing relation of root mean square error loss with algebra under different strategies

LW+GN- Linear Weight+Global Neighborhood 线性递减惯性权重+全局邻域；LW+RIN-Linear Weight+Ring Topology Neighborhood 线性递减惯性权重+环形邻近拓扑邻域；LW+RIN-Linear Weight+Random Topology Neighborhood 线性递减惯性权重+随机拓扑邻域

由图 4.13 可以直观看到，随机拓扑邻域收敛较慢（24-26 代收敛），且收敛后种群损失中位数略有波动，这是因为在算法接近收敛后随机拓扑邻域下的粒子在森林规模和最少叶节点样本数两个分量上的移速波动剧烈且移速一直较大，从图 4.9 和图 4.10 中可以看到这一点。而全局邻域的收敛速度（8-10 代收敛）要略微快于环形邻域（11-13 代收敛），这很容易在直观上理解，由于全局邻域下信息交流比环形邻域时更为充分，各粒子都向着整个种群历史上找到的最好解方向前进，因而更容易达到收敛，各维度上粒子移速下降地更快。但过多的信息交流同时会导致算法收敛后的稳定性下降；因此环形拓扑邻域在收敛后的稳定性最强。

另外，图 4.4、图 4.5、图 4.9 和图 4.10 描绘了在森林规模和最少叶节点样本数两个分量上的种群粒子移速下降。不难发现，相较于其他分量上的速度，这两个分量上的速度变化更加随机，没有呈现更为规则的平滑下降曲线。这可以从优化收敛的角度证明，随机森林算法的性能受森林规模和最少叶节点样本数的影响并不大，而受特征子集容量和采样次数的影响相对较大一些。

而对于最终收敛时的损失大小，三种策略结果相近，没有明显优劣，种群均方根误差中位数都在 0.1 以下，优化能力可以达到预期效果。

4.3.4 随机森林的优化参数规范化与算法实现

前文提到，PSO 算法的一个重要优点在于对实数编码有很好的支持性，使用者不必像遗传算法那样将过多的精力用于考虑不同问题的编码方式上。但实数编码并不意味着完全不用对实际参数做任何处理就可以直接输入优化算法。在运行 PSO 算法之前，仍然需要对输入参数进行规范化操作。这样做到目的在于：考虑到不同维参数的取值范围各不相同，如果不将它们的取值范围统一，就会导致粒子群在搜索域各维度间的运动速度不同，这样的直观后果就是会使算法对各参数有了偏好，造成结果陷入局部最优。

参数规范化的最简单方法就是线性规范化，类似章节 2.2.1 数据归一化处理中线性归一化的处理方法：将各参数线性钳位到一个相同的取值区间中，这样既可以保证规范化后的参数取值区间相同，又可以确保变换前后参数数据的线性性质不变。对于随机森林中 5 个优化参数的规范化公式如下：

$$\theta^{*}=\begin{cases}\left[\dfrac{\theta_{max}-\theta_{min}}{100}\times\theta+\theta_{min}\right], \theta \text{ 为整数取值参数} \\ \dfrac{\theta_{max}-\theta_{min}}{100}\times\theta+\theta_{min}, \theta \text{ 为实数取值参数}\end{cases} \tag{4.20}$$

式中，θ为待规范化参数；θ^{*}为规范化后的参数；θ_{max}、θ_{min}分别为该参数的最大、最小取值。按式 4.20 对所有输入参数钳位处理后，所有的参数将被线性钳位到区间[0,100]。此区间即为粒子在所有维度的统一运动范围，使得粒子在各维度中的速度范围均相同。

介绍完参数规范化之后，下面参照图 4.2 的 PSO 算法流程图说明本文对 PSO 算法的具体实现细节。章节 4.3.2、章节 4.3.3 和章节 5.4.2 中的所有实验均使用下面的处理方法。

首先，输入待优化参数集P为随机森林的 4 个超参数：森林规模、最少叶节点样本数、特征子集容量和采样次数。由上一章最后的实验分析结果（详见章节 3.2.5），在正常范围内，最大层深对随机森林的最终精度影响可忽略不计；因此 PSO 的输入参数维度可以降低一维。

适应度函数Ω(P)为特定参数集随机森林算法按特定训练方法重复训练六次取平均结果的均方根误差（具体训练方法以及均方根误差的概念详见章节 5.1）。

记忆因子r_1、社群因子r_2均取 1.496，恒定不变；惯性权重 w 最大值取 0.7298，最小值取 0.3；可按照线性递减或恒定不变两种策略赋值权重。

社群规模n设定为常数 5；种群规模N按维度大小推算最适合的值，公式如下：

$$N=2\sqrt{s}+10 \tag{4.21}$$

式中s为参数集维度。

4.4 本章小结

本章首先从采样环节、结合策略以及超参数选择三个方面分析了在第三章建立的基础随机森林模型的局限性。针对基础 RF 在前两个方面的局限设计了 TWB-RF 模型，并使用 PSO 算法来对随机森林模型调参。将在章节 5.4 中运用这两种改进的算法 TWB-RF 和 PSOTWB-RF 进行实验，分析二者与基础 RF 相比的优越性；另外，本章还复现了几种常见 PSO 算法改进策略（线性权重递减策略、三种邻域策略）并对这些策略分别进行了逐代粒子速度下降情况、损失下降情况这两个层面的实验分析，实验结果如下：

线性递减的惯性权重较恒定惯性权重而言，种群速度变化更为平滑，算法达到收敛后的波动更小，有利于提高优化算法的稳定性。

采用随机拓扑邻域策略时，种群的收敛速度较慢，粒子移速波动更剧烈；采用全局邻域策略时，种群的收敛速度最快，但收敛后稳定性一般，略有波动；采用环形拓扑邻域策略时，种群的收敛速度较快，且收敛后稳定性较好。

表 4.1 不同策略的优劣性比较

Table 4.1 Comparison of advantages and disadvantages of different strategies

策略/指标	收敛代数	稳定性	收敛后的种群损失
全局邻域(GN)	8-10	中	0.95-1.0
环形拓扑邻域(RIN)	11-13	强	0.95-1.0
随机拓扑邻域(RAN)	24-26	弱	0.95-1.0

在算法收敛过程中，从粒子在不同维度的移速变化情况来看：森林规模和最少叶节点样本数两个分量的粒子移速波动剧烈且没有较为理想平滑的下降曲线，可以印证这两个超参数与随机森林均方根误差之间的相关性并不高；与之相反，特征子集容量和采样次数与随机森林均方误差之间的相关性较高。

5 实验与验证

本章主要介绍几种在机器学习领域常见的模型训练、测试方法以及模型评价指标，同时简单介绍本文所有章节实验所依赖的平台及其搭建过程；在此基础上针对第二、三、四章中得出的数据集和模型进行实验比较和结果验证。

5.1 学习模型的训练与评价

在机器学习中，有很多常见的模型训练与评价方法；本节介绍三种常见的模型训练方法：留出法、重采样法、交叉验证法和三种常用的模型评估指标：平均误差、均方根误差、平均相对误差。

（1） 模型的训练方法：

（a） 留出法：

留出法是最简单的也是最实用的一种模型训练方法。它将整个数据集通过随机抽取得到一定比例的训练集和测试集，一般来说训练集占数据集的比例为 70%左右，也可根据需求自由调整大小。划分后，使用训练集对准备好的模型进行训练，训练完毕后用测试集对模型按照一定方法测试或评价。

（b） 重采样法：

重采样法又称 Bootstrap 法，在本文章节 4.2 中运用这种采样方法对训练集进行再次采样，为每颗决策树分配训练样本。这种方法除了可以用于为集成学习的子学习机提供训练样本外，也可以用作将数据集划分成训练集和测试集。与留出法类似，Bootstrap 使用无放回随机抽样对数据集进行抽取，最后去掉重复样本。与留出法相比，Bootstrap 的最大优点在于每次训练和测试使用的样本都不完全相同，这可以最大程度的避免因特殊样本对训练或测试结果造成的影响。非常适合用于需要对模型进行多次反复训练的场合。比如本文章节 4.3 中使用 PSO 优化算法对学习模型超参数进行调整，其中必然涉及大量重复训练，若此时仍采用留出法可能使实验结果不具有普遍性。

（c） 交叉验证法：

交叉验证(Cross Validation)是一种经常使用的模型训练测试方法。其实现过程比前两种方法稍微复杂，但交叉验证通过使用不同的样本进行重复训练可以很好地避免模型出现过拟合。在交叉验证方法中，根据使用者期望对模型训练训练

的次数，交叉验证可以具体描述为：2 次 2 折交叉验证；3 次 3 折交叉验证…k 次 k 折交叉验证。其中折数(fold)代表将数据集划分成的份数，分成几折就意味着需要进行几次训练。以 k 次 k 折交叉验证为例：首先将数据集均匀划分成 k 份，第一次训练使用第一份样本作为测试集，其他(k-1)份作为训练集，对模型进行训练和测试；第二次训练使用第二份样本作为测试集，其他样本则作为训练集，以此类推。直到进行过 k 次训练测试过程。这种做法虽然可以为模型训练带来好处；但劣势同样显而易见，当数据集规模过大时，会为处理器带来额外的算量，因此这种方法在实际训练模型时并没有前两种方法常用。图 5.1 描绘了 k 次 k 折交叉验证的数据集划分过程：

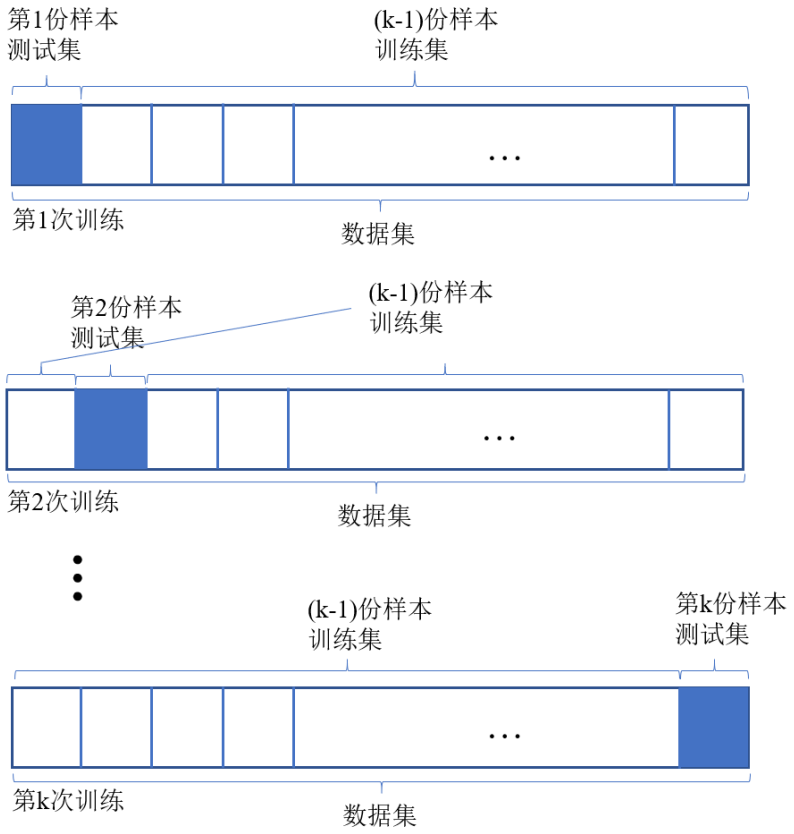

图 5.1 交叉验证训练过程

Figure5.1 The training process of cross validation

（2） 模型的评价指标：

（a） 平均误差：

平均误差是一种计算简单，结果直观明了的评价指标，它直接统计模型在测试集上的所有预测结果与实际结果的差值的绝对值，而后对所有绝对差值取平均。

由于平均误差在所有预测点处均为线性，因此平均误差的大小对所有预测点的预测结果没有偏好。平均误差的计算公式如下：

$$E=\frac{\sum_{i=1}^{n}(\hat{y}_i-y_i)}{n} \tag{5.1}$$

式中，n为测试集样本总数，$\hat{y}_i$为模型对第i个样本的预测值，y_i为第i个样本的实际值，E即为模型在测试集上的平均误差。

（b） 均方根误差：

均方根误差又称标准误差是统计学中常用的一种误差评估指标，表示所有观测值与真实值差值的平方和与观测次数的比值的平方根。与平均误差不同，均方根误差大小对一组观测值中特别大或者特别小的误差具有明显偏好，因此十分适合用于衡量一组预测的精密程度。均方根误差的计算公式如下：

$$R=\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i-y_i)^2}{n}} \tag{5.2}$$

式中，n为测试集样本总数，$\hat{y}_i$为模型对第i个样本的预测值，y_i为第i个样本的实际值，R为模型在测试集上的均方根误差。

（c） 平均相对误差：

平均相对误差对测试误差的衡量方式与前两者不同，平均误差或者均方根误差可以理解成对样本的绝对误差衡量，而相对误差是指观测误差占真实值的大小。测试集的平均相对误差可以表示为：

$$E=\frac{\sum_{i=1}^{n}\frac{(\hat{y}_i-y_i)}{y_i}}{n} \tag{5.3}$$

下面说明本文中的所有实验使用到的模型训练方法和评价指标：

章节 3.2.2-3.2.6 中对影响随机森林性能的参数探究中所有实验、曲线图均采用留出法进行训练，涉及到的评价指标均采用均方根误差。

章节 4.2.2 和章节 4.3.3 中对 PSO 算法惯性权重和邻域策略进行的若干对比探究中所有实验、曲线图均采用 Bootstrap 方法进行训练，涉及到的评价指标采用均方根误差。

章节 5.3 和章节 5.4 中对基础 RF、TWB-RF 以及 PSOTWB-RF 的所有结果验证实验均用留出法进行训练，将数据集分割为 250 个样本点的训练集和 51 个

样本点的数据集；并采用均方根误差和相对误差进行评价。

5.2 实验平台的搭建

本文的实验可以分为两类：数据处理类和模型算法类。根据这两类实验的不同特点需要在不同的平台上进行。为了更方便准确地进行实验，本文主要使用了两个实验平台：Windows10 系统-Visual Studio 2017-Windows 控制台工程-Release-x86 平台；Linux(Ubantu 19.02)系统-GNU 编译器-GDB 调试器-Vscode 编辑器。为了使用者更容易复现与本文有关的相关代码，现简单介绍本文平台的搭建与使用。此外，与本文引用的代码来自：

https://github.com/handspeaker/RandomForests （随机森林）

https://github.com/kkentzo/pso （PSO）

（1） Windows10 系统-Visual Studio 2017-Windows 控制台工程-Release-x86 平台：

Windows 10 是由美国微软公司开发的应用于计算机和平板电脑的操作系统，于 2015 年 7 月 29 日发布正式版。Windows 10 操作系统在易用性和安全性方面有了极大的提升，除了针对云服务、智能移动设备、自然人机交互等新技术进行融合外，还对固态硬盘、生物识别、高分辨率屏幕等硬件进行了优化完善与支持。截至 2020 年 4 月 16 日，Windows 10 正式版已更新至十一月更新 10.0.18363 版本，预览版已更新至 2020 更新 10.0.19608 版本[55]。

Visual Studio 2017 是微软于 2017 年 3 月 8 日正式推出的新版本，是一款十分经典且性能强大的集成开发环境(Integrated Development Environment)。其内建工具整合了.NET Core、Azure 应用程序、微服务（Microservices）、Docker 容器等所有内容[56]。

在 Windows10 系统下的 Visual Studio 中，开发或者编辑变得十分容易，大多情况下不需要考虑底层的代码编译或者调试过程。但有些类库（比如本文 PSO 实现代码需要安装调用 gsl 库）往往很难在 Windows 系统上安装，这也对开发造成了一定障碍。本文在 Visual Studio 2017 中新建 Windows 控制台工程项目并选择 Release-x86 平台就可以开始对本文部分代码进行编辑、调试、运行等基本工作。

在本文中，章节 2.1.3 高炉参数意义分析中的生成、处理特征散点分布图（图

2.4-图 2.26）；章节 2.2 中数据的归一化处理、缺失值填充以及特征选择；章节 3.2.2-章节 3.2.6 中研究随机森林 5 个超参数对模型性能相关分析的代码部分均在此平台下完成。

（2） Linux(Ubantu 19.02)系统-GNU 编译器-GDB 调试器-Visual Studio Code 编辑器：

Linux，全称 GNU/Linux，是一套免费使用和自由传播的类 UNIX 操作系统，其内核由林纳斯·本纳第克特·托瓦兹于 1991 年第一次释出，它主要受到 Minix 和 Unix 思想的启发，是一个基于 POSIX 和 Unix 的多用户、多任务、支持多线程和多 CPU 的操作系统。它能运行主要的 Unix 工具软件、应用程序和网络协议。它支持 32 位和 64 位硬件。Linux 继承了 Unix 以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。Linux 有上百种不同的发行版，如基于社区开发的 debian、archlinux、Ubantu、Manjaro 等，和基于商业开发的 Red Hat Enterprise Linux、SUSE、oracle linux 等[57]。

Ubuntu 是一个以桌面应用为主的 Linux 操作系统，基于 Debian 发行版和 Gnome 桌面环境（从 11.04 版本起改用为 Unity 图形引擎）。Ubantu 是一款十分经典的 Linux 开源系统，Ubantu 的出现让更多开发者可以简单快捷的安装使用 Linux。因此 Ubuntu 拥有庞大的社区力量，用户可方便地从社区获得帮助[58]。

GNU 是一个自由的操作系统，其内容软件完全以 GPL 方式发布。是 GNU 计划的主要目标，名称来自 GNU's Not Unix!的递归缩写，因为 GNU 的设计类似 Unix，但它不包含具著作权的 Unix 代码。GNU 的创始人理查德·马修·斯托曼将 GNU 视为“达成社会目的技术方法”[59]。

GCC 是以 GPL 许可证所发行的自由软件，也是 GNU 计划的关键部分。GCC 的初衷是为 GNU 操作系统专门编写一款编译器，现已被大多数类 Unix 操作系统（如 Linux、BSD、Mac OS X 等）采纳为标准的编译器，甚至在微软的 Windows 上也可以使用 GCC。GCC 支持多种计算机体系结构芯片，如 x86、ARM、MIPS 等，并已被移植到其他多种硬件平台 。

GCC 原名为 GNU C 语言编译器（GNU C Compiler），只能处理 C 语言。但其很快扩展，变得可处理 C++，后来又扩展为能够支持更多编程语言，如 Fortran、Pascal、Objective -C、Java、Ada、Go 以及各类处理器架构上的汇编语言等，所

以改名 GNU 编译器套件（GNU Compiler Collection）[60]。本文在 Linux 系统下的开发主要使用 GNU 编译器。

与在 Windows 系统下的 Visual Studio 中开发不同，使用 Linux 开发必须首先自行搭建一套开发环境。本文选用 GNU 编译器-GDB 调试器-Visual Studio Code 编辑器。这套开发环境对 C/C++开发来说十分容易搭建，界面友好，且易于编辑、调试、运行等。安装方法相对容易，本文不展开说明。

在本文中，章节 4.3.2、章节 4.3.3 中所有针对 PSO 算法性能的实验；章节 5.3、章节 5.4 中对基础 RF、两种改进 RF 的实验验证均在该平台中完成。

此外，使用 GNU-GDB-Visual Studio Code 环境开发 C/C++代码时必须自行完成编译命令。完成编译命令主要有四种方法：直接命令行输入、编写 JSON 文件、编写 Makefile 脚本和使用 Cmake 工具。不论哪种方法都需要开发者清楚针对代码的正确编译命令。本文代码在此环境下用到的编译命令主要为：

基础 RF 代码&TWB-RF 代码编译命令：

g++ -g Sample.cpp Node.cpp Tree.cpp RandomForest.cpp ReadTxtFile.cpp main.cpp -o main.out

PSOTWB-RF 代码编译命令：

gcc -g -Wall -I /home/zkcc/program/gsl2.4/include/(gsl 库 include 文件路径) -L /home/zkcc/program/gsl2.4/lib(gsl 库 lib 文件路径) pso.c Sample.cpp Node.cpp Tree.cpp RandomForest.cpp ReadTxtFile.cpp main.cpp -lgsl -lgslcblas -lm -lstdc++ -o main.out

5.3 基础随机森林算法预报结果验证

在本文第二章的最后总共生成了 4 组不同品质的数据集，Data0-Data3。其中，Data0 为原始数据，共 301 条数据、61 个特征；Data1 在保证没有特别糟糕的特征的情况下，侧重将特征数量最大化，共 301 条数据、29 个特征；Data2 侧重选择相关性更高的特征而不考虑特征数量，共 301 条数据、12 个特征；Data3 在 Data2 的基础上剔除了 2 个与其他特征耦合性较高的特征数据，共 301 条数据、10 个特征。

本节首先利用基础 RF 模型对这 4 组数据集进行横向比较，使用其中效果最好的一组数据集作为基础 RF 的预报的结果，同时为后面的实验提供最佳数据集。

需要说明的是，为了控制变量，本节和章节 5.4.1 中的所有训练实验随机森林超参数均一致：森林规模为 20，最大深度为 10，最少叶节点样本数为 10，特征子集容量为 2，采样次数倍数为 1.2。

图 5.2、图 5.3 分别为基础 RF 采用 Data0 数据对指定测试集的跟踪情况以及相对误差。

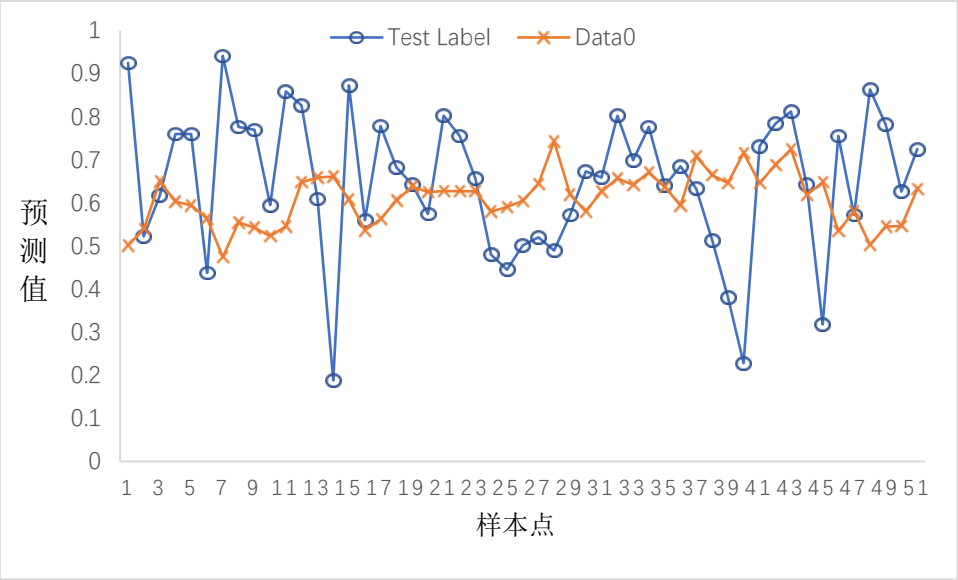

图 5.2 采用 Data0 数据集训练基础 RF 时的测试集样本跟踪情况

Figure 5.2 Sample tracking of the test set when Data0 was used to train the basic RF

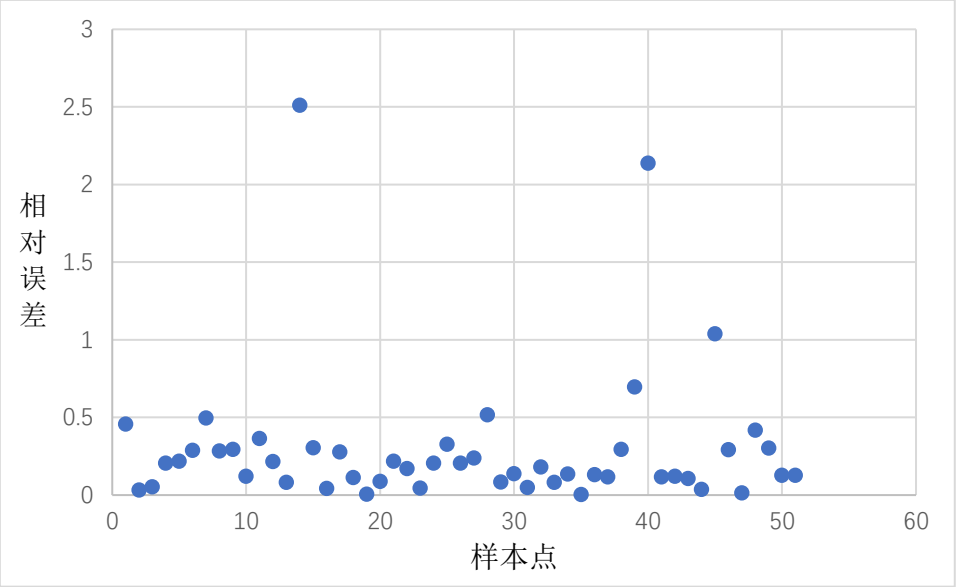

图 5.3 采用 Data0 数据集训练基础 RF 时的预测样本相对误差

Figure 5.3 Relative errors of predicted samples when Data0 was used to train the basic RF

由以上两幅图可以看出，尽管使用原始数据集 Data0 训练时大部分样本点的相对误差并不大，但对真实数据的变化趋势几乎不能进行正确地跟踪，甚至对有些样本的预测趋势完全相反。可见，对于随机森林模型而言，如果数据集中的低相关性特征过多，则会对模型造成干扰，其影响主要体现在对真实数据变化趋势的预测方面。尽管大部分样本点的相对误差并不大，但如果预测完全不能反映数据变化趋势，那么这样的训练出的模型是不能直接使用的。

图 5.4、图 5.5 分别为基础 RF 采用 Data1 数据对指定测试集的跟踪情况以及相对误差。

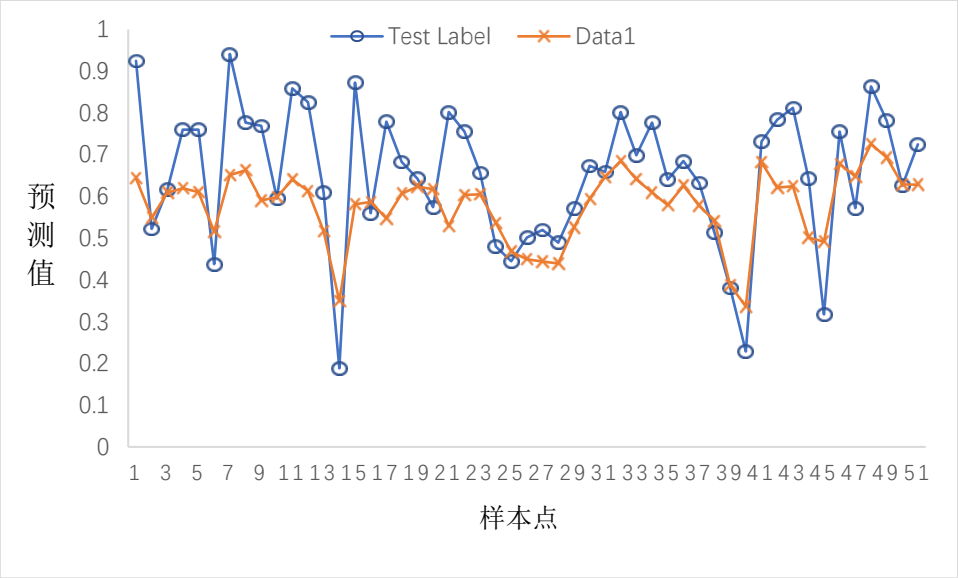

图 5.4 采用 Data1 数据集训练基础 RF 时的测试集样本跟踪情况

Figure 5.4 Sample tracking of the test set when Data1 was used to train the basic RF

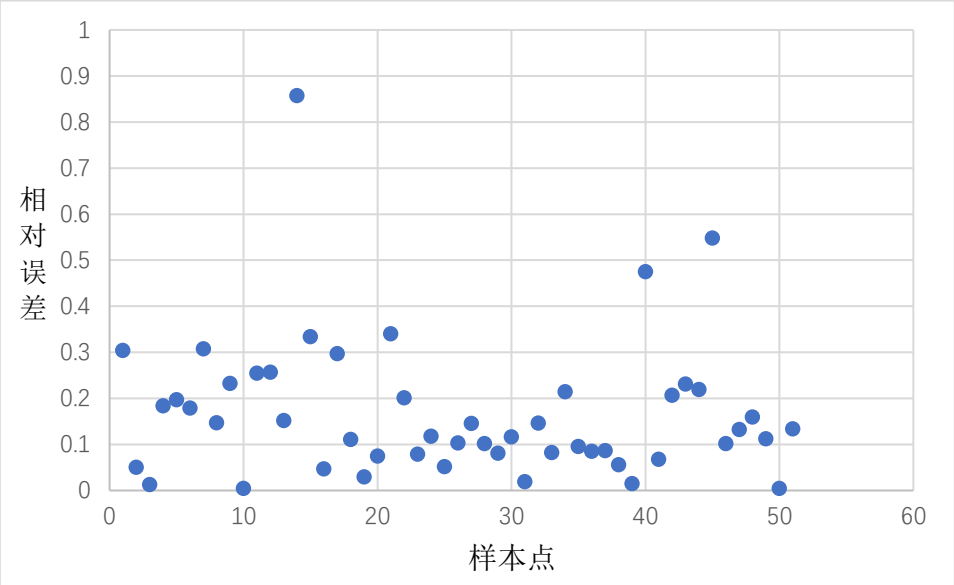

图 5.5 采用 Data1 数据集训练基础 RF 时的预测样本相对误差

Figure 5.5 Relative errors of predicted samples when Data1 was used to train the basic RF

由以上两图可见，在剔除掉一些相关性极低的特征数据后，模型的跟踪能力比之前有了明显的改观。可以说明，对随机森林而言，个数再多的低相关性特征也不会对模型预测结果带来任何好处，反而会大幅削弱模型的跟踪能力。

Data1 尽管去掉了相关性极低的一些特征，但为了使数据集特征个数尽可能

多，一些特征的相关性仍然不高。通过上图同时可以看出，虽然跟踪趋势基本正确，但预测精度仍然差强人意，特征数量的保持并没有对模型的预测精度带来改善。

图 5.6、图 5.7 分别为基础 RF 采用 Data2 数据对指定测试集的跟踪情况以及相对误差。

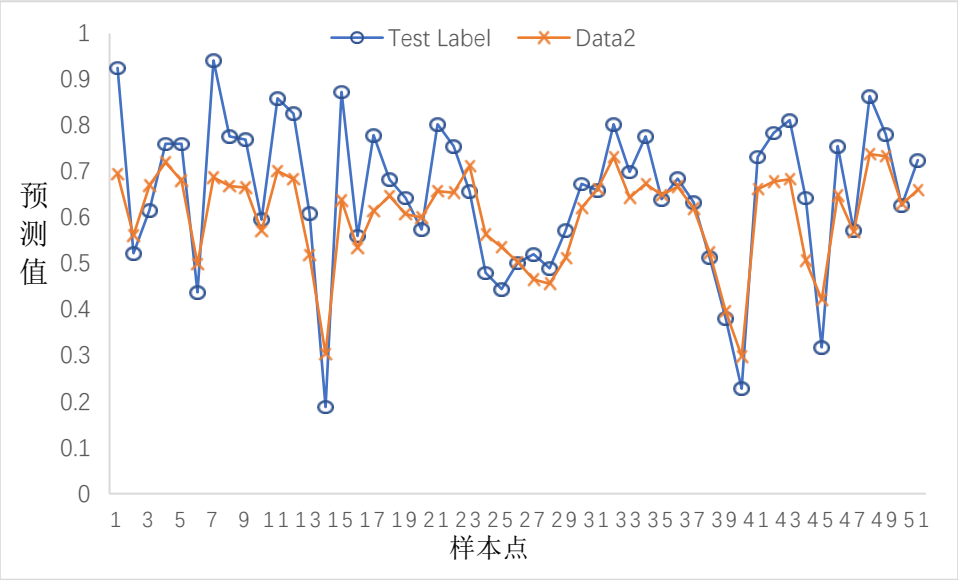

图 5.6 采用 Data2 数据集训练基础 RF 时的测试集样本跟踪情况

Figure 5.6 Sample tracking of the test set when Data2 was used to train the basic RF

由图 5.6，5.7 可以看出，再次剔除掉相关性稍低一些的特征数据后，尽管特征数量已经非常少了，但预测精度对比图 5.4 来说反而有提高；而且对于数据走向趋势而言，只有两个样本点出现错误（样本点 26、27）可以算是较为理想的预测。但观察图 5.7 发现，有三个样本点的相对误差非常大（样本点 14、40、45），再重复做了一组相同实验后发现仍然有此问题。猜测有可能是因为特征间存在的较强耦合关系制约着预测精度，导致使用 Data2 数据集训练时相对误差指标不理想。

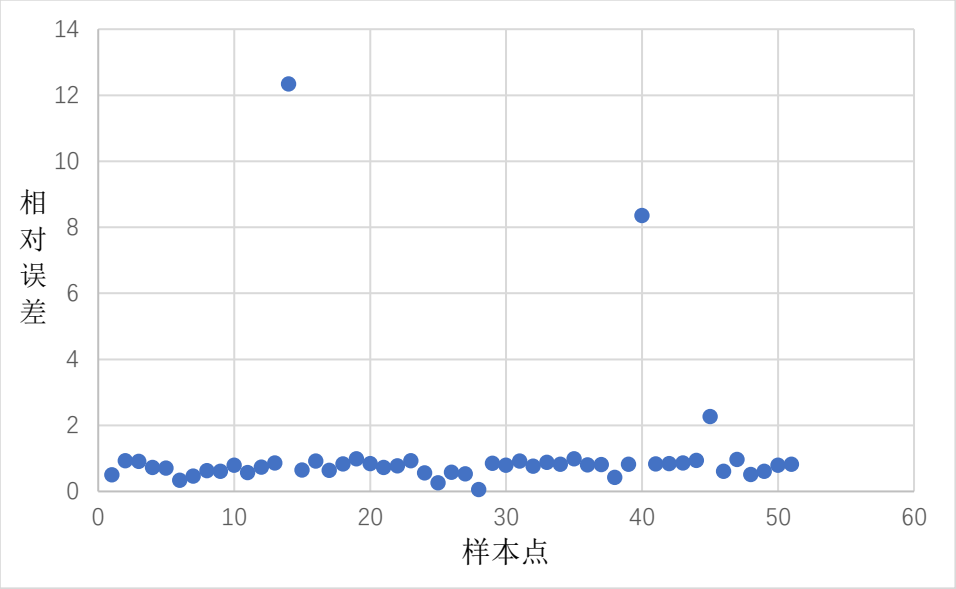

图 5.7 采用 Data2 数据集训练基础 RF 时的预测样本相对误差

Figure 5.7 Relative errors of predicted samples when Data2 was used to train the basic RF

图 5.8、图 5.9 分别为基础 RF 采用 Data3 数据对指定测试集的跟踪情况以及相对误差。

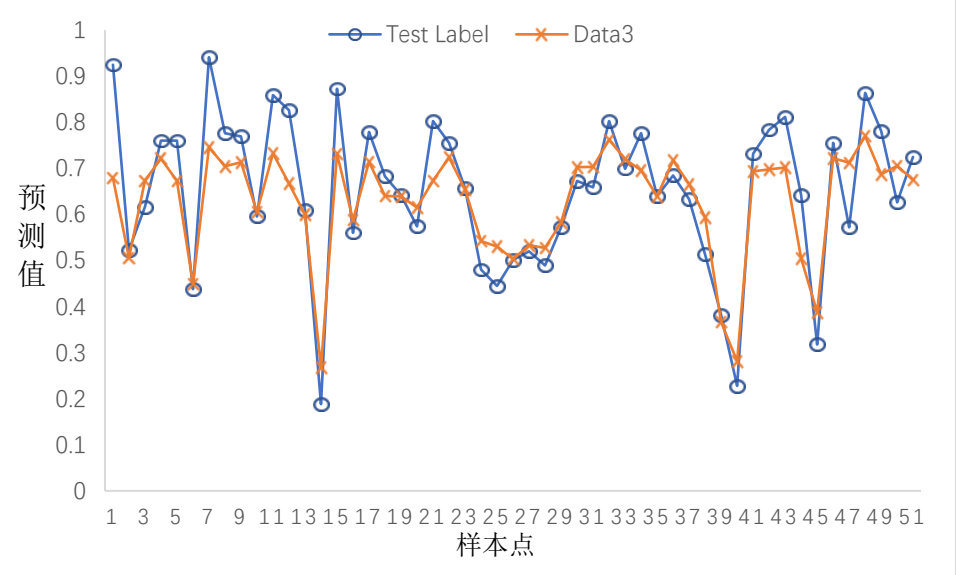

图 5.8 采用 Data3 数据集训练基础 RF 时的测试集样本跟踪情况

Figure 5.8 Sample tracking of the test set when Data3 was used to train the basic RF

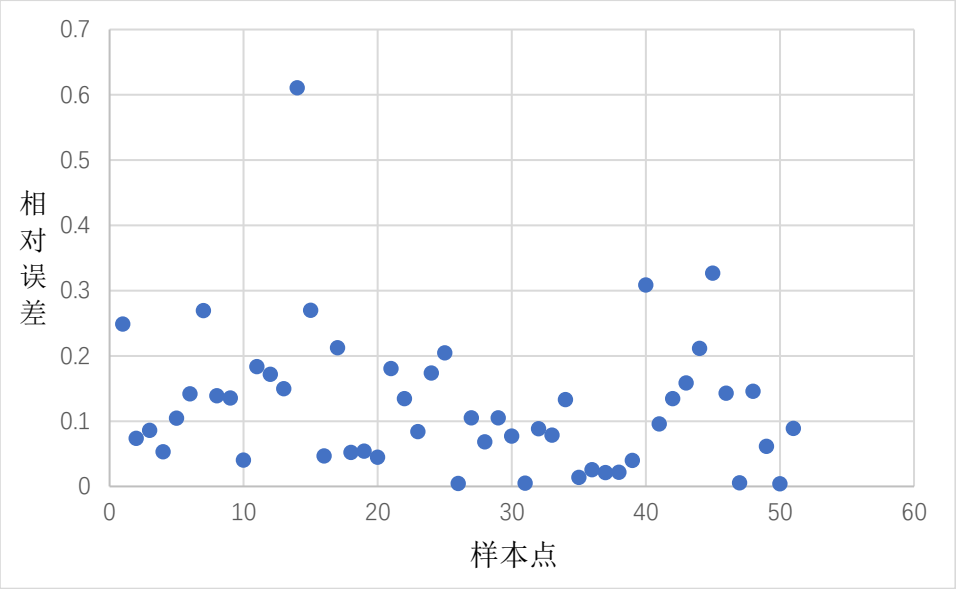

图 5.9 采用 Data3 数据集训练基础 RF 时的预测样本相对误差

Figure 5.9 Relative errors of predicted samples when Data3 was used to train the basic RF

由以上两图可以看出，使用 Data3 训练模型的跟踪效果与 Data2 不相上下，同样只有 3 个样本点变化趋势预测错误（样本点 4、50、51），但预测精度较使用 Data2 作为数据集时有明显提高。50.9%的样本点相对误差在 10%以内，同时也没有相对误差超过 100%的样本点。可见去除耦合性较大的特征主要对模型的预测精度有较大的积极影响。

综合章节 5.3 的实验分析，可以得出结论：使用 Data3 作为数据集时的训练效果最佳，对原始数据剔除低相关性数据并按照章节 2.2.4 中的方法去除耦合特征后的数据十分适合随机森林模型的训练；训练后的模型兼具较理想的跟踪能力，也有一定的准确性。因此使用图 5.8、图 5.9 所示的数据作为基础 RF 的实验结果，与后面章节的改进模型进行对比。后面章节的所有模型均在 Data3 数据集上完成训练和测试。

5.4 改进随机森林算法的预报结果对比

本节主要验证论文第四章中对基础 RF 做出的两种改进算法的预报性能。通过比较改进算法的跟踪能力、测试集上的均方误差以及各预测点相对误差来分析对比其与基础 RF 模型的优劣。

5.4.1 TWB-RF 模型的预报性能实验

图 5.10、图 5.11、图 5.12 分别统计了 TWB-RF 的样本跟踪能力、测试集上的均方误差以及各样本点的相对误差。

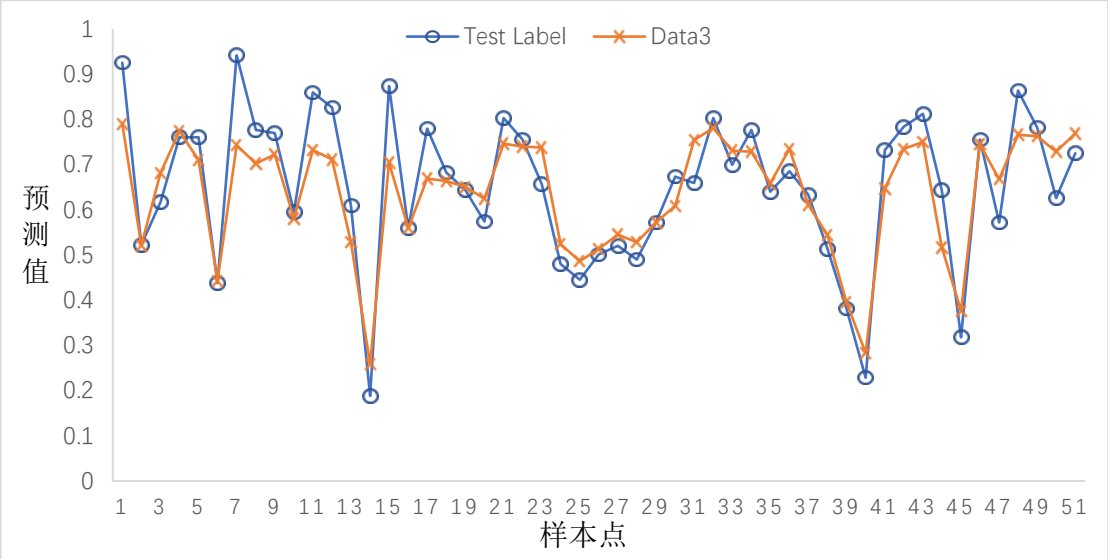

图 5.10 TWB-RF 使用 Data3 数据集训练的预测样本跟踪情况

Figure 5.10 Sample tracking of the test set when Data3 was used to train TWB-RF

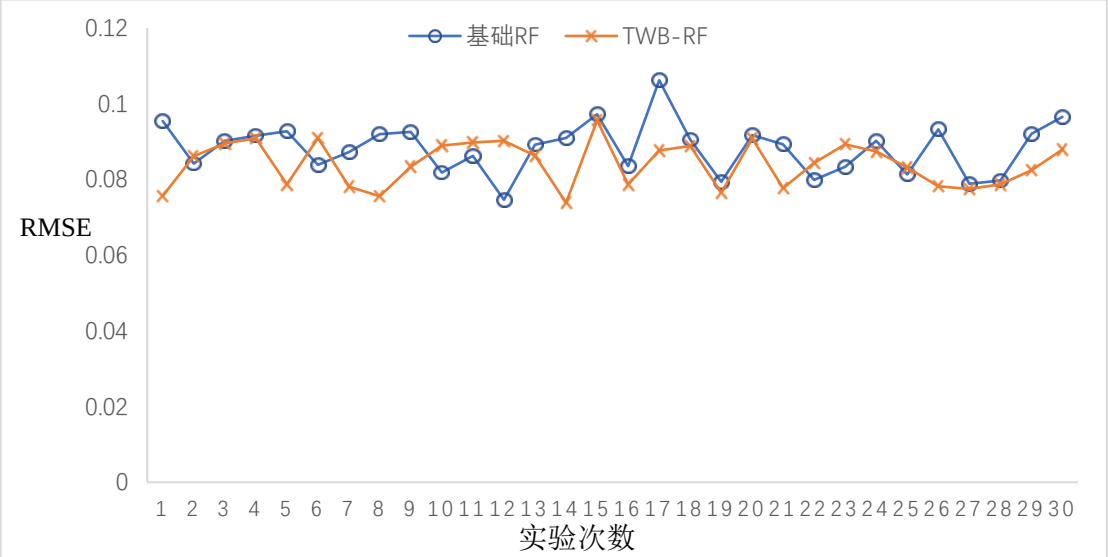

图 5.11 基础 RF 与 TWB-RF 使用 Data3 数据集训练时在测试集上的均方根误差对比

Figure 5.11 Comparison of root mean square error on the test set between the base RF and TWB-RF training using Data3

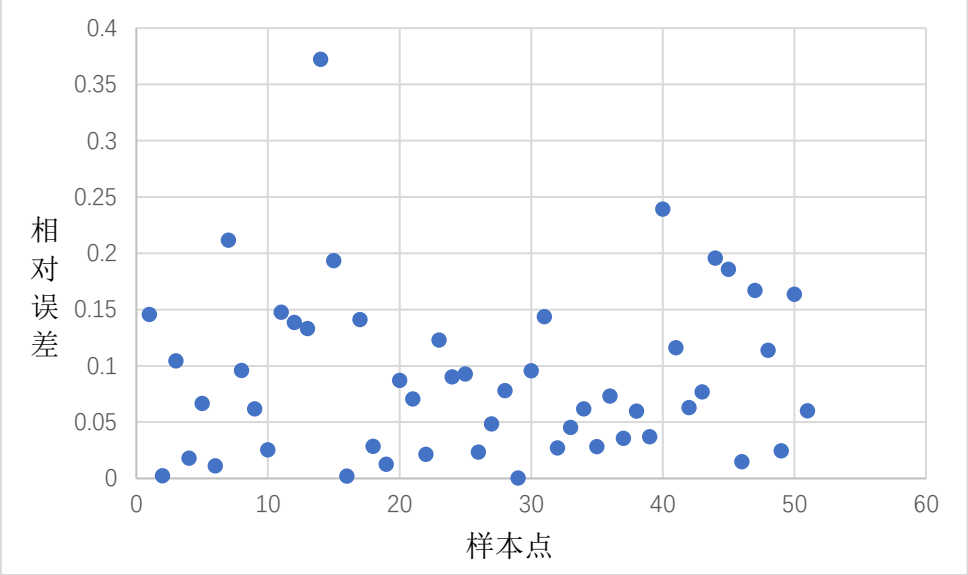

图 5.12 采用 Data3 数据集训练 TWB-RF 时的预测样本相对误差

Figure 5.12 Relative errors of predicted samples when Data3 was used to train TWB-RF

通过对比图 5.10 与图 5.8，可以看到 TWB-RF 的样本跟踪性能略优于基础 RF，只有一个样本点预测趋势错误（样本点 4）。

通过观察图 5.11，TWB-RF 在 30 次测试集均方误差实验中结果略优于基础 RF：基础 RF 的 30 次平均 RMSE 为 0.08818，而 TWB-RF 的平均 RMSE 为 0.08409；基础 RF 的 30 次 RMSE 方差为4.5922×10^{-5}，而 TWB-RF 的 30 次 RMSE 方差为 3.5920×10^{-5}。

通过观察图 5.12，TWB-RF 有 64.8%的预测点相对误差在 10%以内，这个结果要优于基础 RF 的 50.9%。说明 TWB-RF 相比于基础 RF 有更精确的预测能力。

可见在同样参数下 TWB-RF 的相比于基础 RF 具有更好的跟踪能力，预测结果更加精确、稳定。

5.4.2 PSOTWB-RF 模型的预报性能实验

本节实验首先验证在数据集 Data3 中，PSO 算法的收敛情况；此后用 PSO 寻找出的一组最优超参数训练 TWB-RF 模型，以此作为 PSOTWB-RF 模型的预报结果。

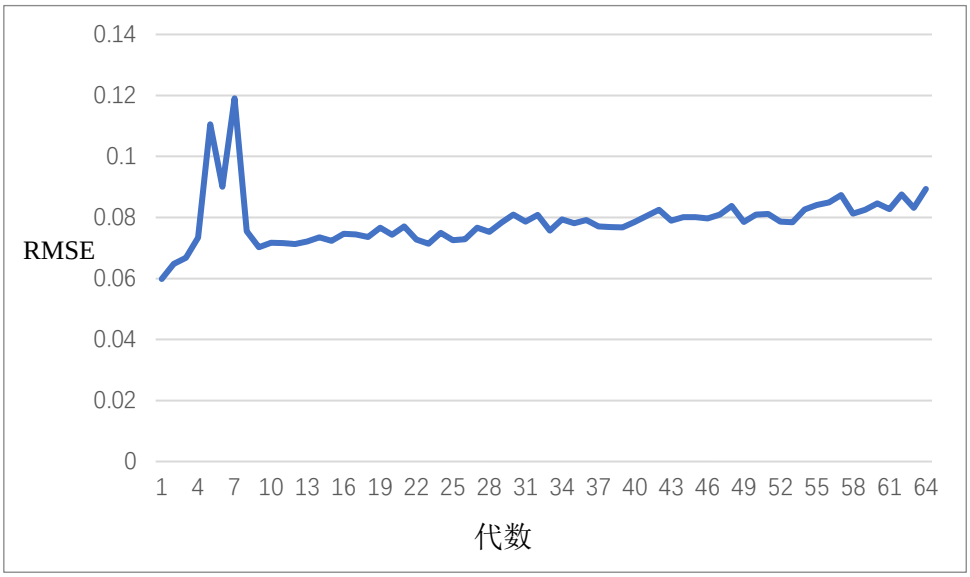

图 5.13 PSOTWB-RF 的逐代损失曲线

Figure 5.13 Generation loss curve of PSOTWB-RF

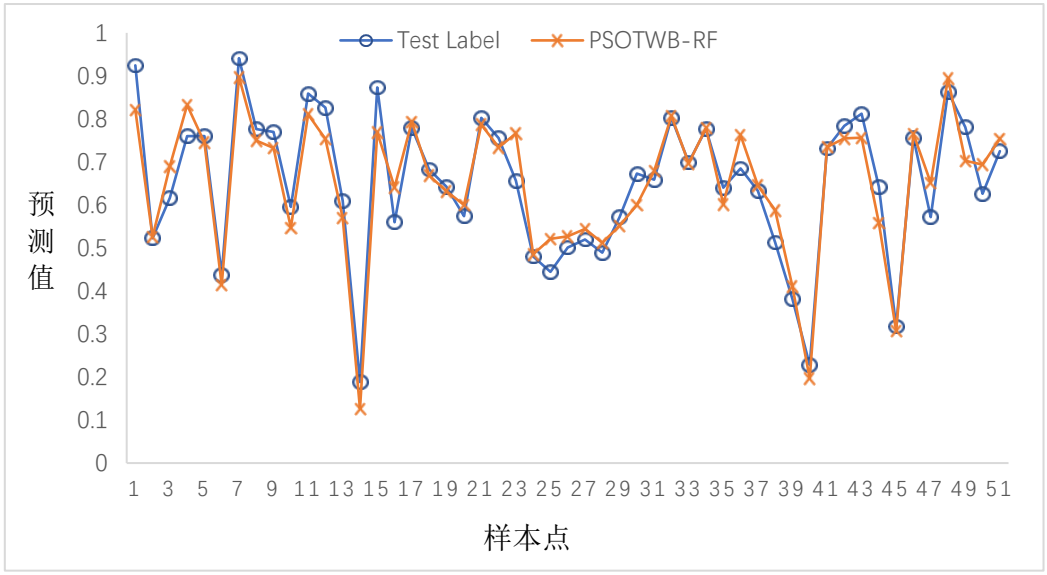

图 5.14 PSOTWB-RF 使用 Data3 数据集训练的预测样本跟踪情况

Figure 5.14 Sample tracking of the test set when Data3 was used to train PSOTWB-RF

图 5.13 显示了 PSO 算法中种群中粒子中位数损失随代数变化的曲线。可以看到，粒子在第 1 代就取得了很好的效果，所以在此之后种群几乎不向更优方向前进。经过相同条件下多次实验仍然会出现这种现象，可能是由于 Data3 数据集十分适合用于训练，初始种群已经有粒子找到了较为优秀的解；而第四章的实验中使用 Data2 数据集训练始终没有出现这种现象，这也可以说明 Data3 相比于 Data2 剔除掉两个高耦合特征操作是十分必要的。

通过 PSO 找到的最佳参数是：森林规模为 20，最少叶节点样本数为 10，特征子集容量为 10，采样次数倍数为 1.33，最大层深不代入优化，仍取 10（其原因在上文已分析过）。图 5.14 显示了使用这组超参数时的 TWB-RF 的样本点跟踪能力，只有样本点 24 跟踪趋势出现错误，可以基本准确地跟踪测试样本。

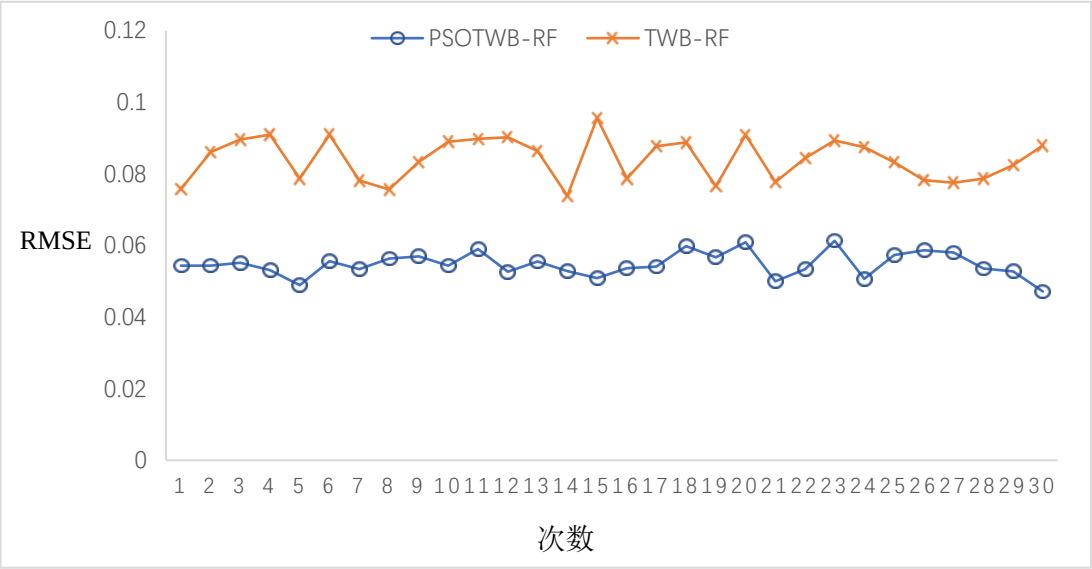

图 5.15 TWB-RF 与 PSOTWB-RF 使用 Data3 数据集训练时在测试集上的均方根误差对比

Figure 5.15 Comparison of root mean square error on the test set between TWB-RF and PSOTWB-RF training using Data3

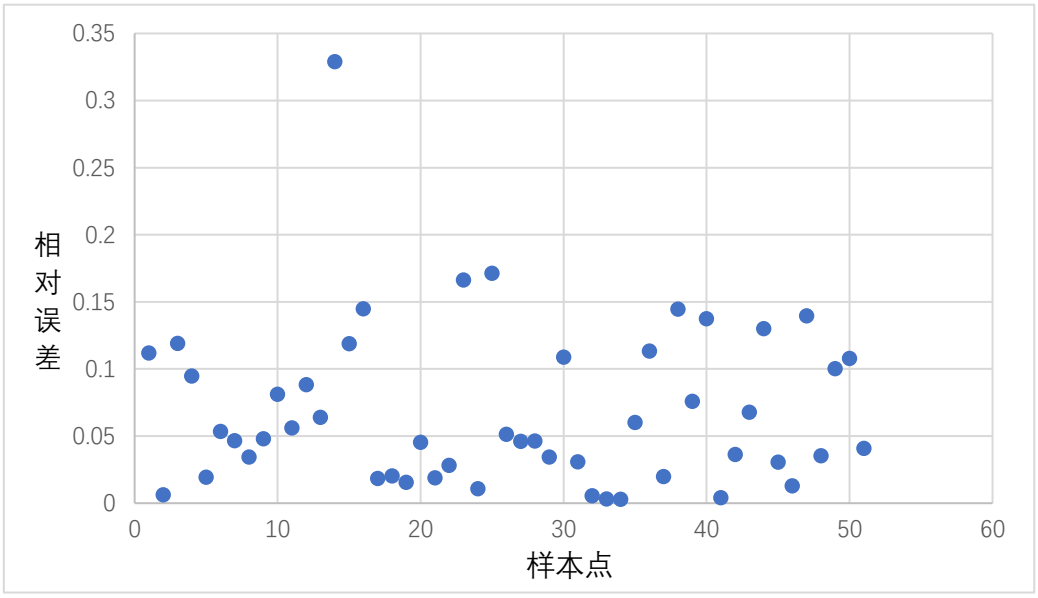

图 5.16 采用 Data3 数据集训练 PSOTWB-RF 时的预测样本相对误差

Figure 5.16 Relative errors of predicted samples when Data3 was used to train

PSOTWB-RF

图 5.15 为 TWB-RF 与 PSOTWB-RF 的 30 次均方误差对比实验。TWB-RF 的 30 次平均 RMSE 为 0.08409，而 PSOTWB-RF 的平均 RMSE 为 0.05471；TWB-RF 的 30 次 RMSE 方差为 3.5920×10^{-5}，而 PSOTWB-RF 的 30 次 RMSE 方差为1.1623×10^{-5}。可以看出，经过优化算法寻优后的 PSOTWB-RF 预测精度和稳定性都远高于没有经过参数寻优的 TWB-RF。

图 5.16 为 PSOTWB-RF 在测试集样本点上的相对误差分布图，PSOTWB-RF 有 72.6%的预测点相对误差在 10%以内；50.9%的预测点相对误差在 5%以内。这个结果要优于 TWB-RF 的 64.8%（10%误差带）和 35.3%（5%误差带）。说明 PSOTWB-RF 的预测精度对比 TWB-RF 而言有了明显的提升。

5.5 本章小结

本章先用基础 RF 模型测试 Data0-Data3 数据集的质量。最终得出结论，Data3 数据集在相同条件下训练的模型具有最高的跟踪能力、精度和稳定性，证明本文第二章中的一系列数据处理方法对提升数据训练效能十分有效。

本章主要从三个方面（跟踪能力、预测精度、预测稳定性）、四个指标（跟踪趋势错误样本点个数、30 次平均 RMSE 值、30 次 RMSE 方差、10%误差带准确率）考察三个模型的性能优劣。统计最终实验结果如下表列出。

表 5.1 本文模型的评价指标统计

Table 5.1 evaluation index statistics of the model in this paper

方面	**跟踪能力**	**预测精度**		**预测稳定性**
指标	趋势错误样本点占比(%)	30 次平均 RMSE 值	10%误差带准确率(%)	30 次 RMSE 方差(10^{-5})
基础 RF	5.8	0.08818	50.9	4.5922
TWB-RF	1.9	0.08409	64.8	3.5920
PSOTWB-RF	1.9	0.05471	72.6	1.1623

从表中可以看出，本文对基础 RF 的两种改进算法在三个评价维度都有十分比较大的提升。证明本文第四章中对模型局限性的分析较为准确，提出的两种改进方案对模型性能提升起到积极作用。

6 结论与展望

6.1 结论

由于高炉内发生的化学反应具有相当的复杂性和黑盒性，在直接对其机理分析十分困难的前提下，本文利用了一种基于改进随机森林的数据驱动模型算法对高炉机理进行解释，用以预测高炉铁水中的 Ti 含量值。本文完整地处理了数据、建立模型并进行测试，最终通过实验证实这些数据和模型具有相对好的效果。本文的主要工作以及对应结论如下：

（1）本文在详细分析高炉工艺和炉内反应机理的基础上对原始数据进行参数分析并预处理，经过归一化处理、缺失值填充、数据清洗、特征选择以及剔除耦合操作后，形成 4 组不同品质的数据 Data0-Data3。经过对 4 组数据的训练实验，得出图 5.2-图 5.9 的结果，可以明显看出 Data3 在测试中表现最佳，可以证明本文使用的特征筛选方法以及本文提出的剔除耦合方法具有非常好的效果。

（2） 本文在研究经典随机森林模型相关理论的基础上搭建了基础 RF 模型并进行实验分析了可能影响随机森林性能两方面因素对应的 5 个超参数（森林规模、最大层深、最少叶节点样本数、特征子集容量以及采样次数）。最终得出结论：特征子集容量、采样次数的变化对随机森林性能有较大影响；在标准范围内森林规模、最少叶节点样本数对随机森林性能影响不大；而在标准范围内最大层深几乎不影响随机森林的性能。

（3） 本文基于基础 RF 模型，分析了基础 RF 模型的三种局限（采样环节、结合策略、超参数选择），针对这些局限提出了改进策略并据此搭建了 TWB-RF 模型和 PSOTWB-RF 模型。经过相同条件下的测试验证，TWB-RF 和 PSOTWB-RF 的预测效果要优于基础 RF 模型，具体数据见表 5.1。最终本文改进的最佳模型的测试指标数据为：预测趋势错误样本占比为 1.9%，30 次平均 RMSE 值为 0.05471，30 次 RMSE 方差为1.1623×10^{-5}，10%误差带准确率为 72.6%。如果考虑到本文数据具有很大的波动性（测试集归一化后方差为 0.0245，级差倍数为 7.47），为预测带来更大的难度，本文的模型性能对比其他论文来说也十分不错。

（4） 本文在构建 PSOTWB-RF 模型的同时，还研究了几种常见的 PSO 算法改进策略及其应用效果。结论如下：线性递减的惯性权重较恒定惯性权重而言，

种群速度变化更为平滑，算法达到收敛后的波动更小，有利于提高优化算法的稳定性；使用环形拓扑邻域策略相比于全局邻域策略和随机拓扑策略而言，PSO 的收敛速度适中，稳定性更强，更适合用于寻优工作，具体实验数据见表 4.1。

6.2 展望

尽管本文的研究内容比较多，但在研究深度的层面上仍显得十分单薄，应用层面上仍需投入大量工作。

第一，本文的工作内容中还有很多可以进行更深入的研究。比如在数据预处理部分，所用到的 3σ 准则没有检查出任何一条噪声数据，这可能是因为本文所使用数据本身质量较好，也可能因为所使用的方法不够完善，本文因为时间受限，没有对这个问题深入挖掘，没有尝试使用诸如本文提到的聚类算法或者其他一些方法对数据进行清洗。另外，本文对耦合数据的处理方法还有待深入研究，本文只做到了实别高耦合特征，并将其简单剔除，而没有研究是否可以通过类似现代控制理论中的解耦合方法解除或者削弱特征的耦合程度，这样既可以保证特征数量，同时也降低了特征间的耦合度；在学习模型部分，缺少随机森林模型与其他经典模型（比如 SVM，神经网络等）的对比研究；在优化算法部分，只采用了 PSO 一种优化算法对参数进行调整，没有与其他算法进行对比分析。此外，没有研究多目标优化方法对学习模型性能的改良效果；在随机森林模型部分，没有研究剪枝策略对模型有何影响；在随机森林超参数研究方面，由于时间原因没有对各参数与森林中各决策树间的耦合性进行实验分析，只停留在理论阶段。另外，本文对超参数的研究方法仍存在问题，导致在章节 3.3 中得出的有关最佳超参数范围的结论与第 4、5 章 PSO 寻优得到的最佳超参数有较大偏差，这个现象也值得深入挖掘。

第二，本文的研究只停留在理论阶段，而缺少在实际高炉中的现场测试。同时，本文并没有为这套预测模型针对高炉现场设计开发可以应用的软件系统。接下来的研究可以调查学习高炉现场需要的具体工艺流程，为理论模型设计一套软件系统，进行现场数据调试。

参考文献

[1] 崔玉平. 周济：智能制造是“中国制造 2025”主攻方向[N]. 中国工业报, 2015-08-03(B03).

[2] 张福明. 中国高炉炼铁技术装备发展成就与展望[J]. 钢铁, 2019, 54(11):1-8.

[3] 李军朋. 高炉冶炼过程的铁水硅含量分析及其建模研究[D]. 燕山大学, 2015: 37-42.

[4] 刘壮壮, 吴巍, 郭贤利, 孟华栋, 赵军普. SiO_2-TiO_2-CaO-MgO-FeO-MnO 渣系与脱钛过程铁水中 Ti-Si 平衡热力学模型[J]. 钢铁, 2013, 48(06):34-39+49.

[5] 马世文, 任盛怡, 曹长修. 单变量 ARIMA 模型的铁水钛含量预报[J]. 重庆工学院学报(自然科学版), 2009, 23(03):43-47.

[6] 雷家柳, 薛正良. 帘线钢生产过程中钛含量的影响因素及控制[J]. 江苏大学学报(自然科学版), 2015, 36(06):728-732.

[7] 范和华, 何波, 吴艺鹏. 铁水中钛含量及其还原率影响因素分析[J]. 山东冶金, 2018, 40(01):36-38.

[8] 周志华. 机器学习[M]. 北京:清华大学出版社, 2016: 178-190.

[9] 杨志昌. 基于模糊理论的高炉炉温[Si]预测模型的研究[D]. 浙江大学, 2009: 12-24.

[10] 陈树文. 高炉专家系统在太钢高炉的应用[J]. 山西冶金, 2019, 42(06):117-119+144.

[11] 孙冠群. 基于 FCM 多支持向量机的高炉冶炼硅含量预测模型[J]. 中国战略新兴产业, 2017(28):141-149.

[12] 吴金花. 高炉冶炼过程分析及其铁水硅含量预测模型研究[D]. 燕山大学, 2016: 36-62.

[13] 黄陈林. 基于粒子群-极限学习机的高炉铁水硅含量预测研究[D]. 安徽工业大学, 2019: 17-54.

[14] 刘忻梅, 石琳. 考虑时滞因素的 RBF 神经网络模型在高炉铁水硅预报中的应用[J]. 内蒙古大学学报(自然科学版), 2012, 43(02):188-191.

[15] 蒋朝辉, 董梦林, 桂卫华, 阳春华, 谢永芳. 基于 Bootstrap 的高炉铁水硅含量二维预报[J]. 自动化学报, 2016, 42(05):715-723.

[16] Breiman L. Random forests[J]. Machine learning, 2001, 45(1):5-32

[17] 王奕森,夏树涛. 集成学习之随机森林算法综述[J]. 信息通信技术, 2018, 12(01):49-55.

[18] 方匡南, 吴见彬, 朱建平, 谢邦昌. 随机森林方法研究综述[J]. 统计与信息论坛, 2011, 26(03):32-38.

[19] Ganaie MA, Tanveer M, Suganthan PN. Oblique Decision Tree Ensemble via Twin Bounded SVM[J]. Expert Systems With Applications,2020,pp. 143-159.

[20] Gamze A, Medine Y, Kaan S. Determination of the costs of falls in the older people according to the decision tree model. [J]. Archives of gerontology and geriatrics, 2019, pp. 87-95.

[21] Cuadrado J, Gómez D, Laria JC, Rodríguez-Cuadrado S. Merged Tree-CAT: A fast method for building precise computerized adaptive tests based on decision trees[J]. Expert Systems With Applications, 2020, pp. 143-150.

[22] Meng XF, Zhang P, Xu Y, Xie H. Construction of decision tree based on C4.5 algorithm for online voltage stability assessment[J]. International Journal of Electrical Power and Energy Systems, 2020, pp. 118-126.

[23] Yong H, Karmaka K, Ron B, Shitanshu K, Matthew F, John Y. Identifying smoker subgroups with high versus low smoking cessation attempt probability: A decision tree analysis approach. [J]. Addictive behaviors, 2019, pp. 103-111.

[24] Gohari M, Eydi AM. Modelling of shaft unbalance: Modelling a multi discs rotor using K-Nearest Neighbor and Decision Tree Algorithms[J]. Measurement, 2020, pp. 151-158.

[25] 喻爱国, 张新义, 韩淑范, 车玉满. 低钛铁水生产实践[J]. 鞍钢技术, 2007(02):28-31.

[26] 梁振华, 何少松, 赵华森, 聂志水, 常鹏飞, 刘佳. 高炉冶炼低钛铁水的生产实践[J]. 河北冶金, 2019(01):25-27.

[27] 马富涛. 高炉 Rist 操作线模型的研究与开发[C]. 全国冶金自动化信息网、

《冶金自动化》杂志社. 全国冶金自动化信息网 2013 年会论文集. 全国冶金自动化信息网、《冶金自动化》杂志社:《冶金自动化》杂志社, 2013:625-630.

[28] 周传典. 鞍钢炼铁技术的形成与发展[M]. 北京：冶金工业出版社，1999: 123-137.

[29] 李胜杰, 赵恒山, 王瑞玲. 一种预测和控制高炉铁水[Ti]的方法及应用[J]. 河南冶金, 2017, 25(03):4-6+42.

[30] 文光远, 鄢毓璋, 周培土, 周永成, 廖代华, 王戈. 攀钢高炉铁水的性质[J]. 钢铁钒钛, 1996(03):24-29.

[31] 温继勇. 高炉主要操作参数与铁水含硅量滞后关系分析[J]. 甘肃科技, 2014, 30(20):47-50+15.

[32] 刘忻梅, 石琳. 考虑时滞因素的 RBF 神经网络模型在高炉铁水硅预报中的应用[J]. 内蒙古大学学报(自然科学版), 2012, 43(02):188-191.

[33] 闫冲. 基于量子遗传神经网络的铁水温度预报研究[D]. 东北大学, 2014: 23-32.

[34] 祁鹏. 基于偏最小二乘的高炉铁水硅含量预测研究[D]. 内蒙古科技大学, 2010: 24-59.

[35] 徐循进. 智能控制在高炉炉温预测中应用研究[D]. 合肥工业大学, 2006: 1-59.

[36] 于涛, 李江鹏, 李明昕, 石琳. 基于分类回归树的高炉铁水硅含量预测模型[J]. 内蒙古大学学报(自然科学版), 2015, 46(05):548-552.

[37] 罗世华, 陈坤. 基于偏态深度分类的高炉硅含量及波动预测[J/OL]. 控制与决策:1-7[2020-02-26].

[38] 庄田. 基于 Elman-Adaboost 模型的高炉铁水硅含量回归与分类预测研究[D]. 浙江大学, 2018:26-32.

[39] 张俊玉, 胡家豪, 黄嵩. CART 决策树方法在煤电厂节能降耗中的应用[J/OL]. 控制与决策:1-8[2020-02-26].

[40] 廖明生, 江利明, 林珲, 杨立民. 基于 CART 集成学习的城市不透水层百分比遥感估算[J]. 武汉大学学报(信息科学版), 2007(12):1099-1102+1106.

[41] Wang L. Construction of decision analysis system based on improved decision

tree pruning algorithm and rough set classification theory [C]. Institute of Management Science and Industrial Engineering. Proceedings of 2019 9th International Conference on Education and Social Science (ICESS 2019). Institute of Management Science and Industrial Engineering: 计算机科学与电子技术国际学会(Computer Science and Electronic Technology International Society), 2019:1463-1468.

[42] 许允之. 基于随机森林算法的徐州雾霾回归预测模型[C]. 《环境工程》编委会、工业建筑杂志社有限公司. 《环境工程》2019 年全国学术年会论文集. 《环境工程》编委会、工业建筑杂志社有限公司:《环境工程》编辑部, 2019:175-179+185.

[43] 闫云凤. 基于决策森林的回归模型方法研究及应用[D]. 浙江大学, 2019: 12-32.

[44] 李贞贵. 随机森林改进的若干研究[D]. 厦门大学, 2013: 45- 56.

[45] 周天宁, 明冬萍, 赵睿. 参数优化随机森林算法的土地覆盖分类[J]. 测绘科学, 2017, 42(02):88-94.

[46] 温博文, 董文瀚, 解武杰, 马骏. 基于改进网格搜索算法的随机森林参数优化[J]. 计算机工程与应用, 2018, 54(10):154-157.

[47] 谢诗雨, 李君豪, 王劲峰, 熊双菊, 唐阳. 基于粒子群优化加权随机森林的非侵入式负荷辨识[J]. 电器与能效管理技术, 2019(09):22-26+44.

[48] 马骊. 随机森林算法的优化改进研究[D]. 暨南大学, 2016: 32-44.

[49] 马晓君, 董碧滢, 王常欣. 一种基于PSO优化加权随机森林算法的上市公司信用评级模型设计[J]. 数量经济技术经济研究, 2019, 36(12):165-182.

[50] 王杰, 程学新, 彭金柱. 一种基于粒子群算法优化的加权随机森林模型[J]. 郑州大学学报(理学版), 2018, 50(01):72-76.

[51] 杜鹤桂. 高炉冶炼钒钛磁铁矿原理[M]. 北京: 科学出版社, 1996.5: 142-153.

[52] StarHai. 标准化和归一化对机器学习经典模型的影响. https://www.cnblogs.com/csushl/p/9966397.html

[53] 蒋朝辉, 尹菊萍, 桂卫华, 阳春华. 基于复合差分进化算法与极限学习机的高炉铁水硅含量预报[J]. 控制理论与应用, 2016, 33(08):1089-1095.

[54] Shi Y, Eberhart RC. A modified particle swarm optimizer [J]. Proceedings of the IEEE International Conference on Evolutionary Computation . 1998, 47(6): 1123-1131.

[55] 李志鹏．精解 Windows 10[M]. 北京：人民邮电出版社，2015.9: 2-3.

[56] Visual Studio 2017 发行说明. Visual Studio 官方网站[引用日期 2017-03-10]

[57] 刘振洪、吴敏凤．Linux 操作系统实用教程[M]．天津：天津科学技术出版社，2016: 12-13.

[58] 杜焱，廉哲，李耸．Ubuntu Linux 操作系统实用教程[M]．北京：人民邮电出版社，2017: 13-15.

[59] 陈肖. Linux:自由的操作系统[J]. 微电脑世界, 2004(18): 184-185.

[60] 周立功. 嵌入式 Linux 开发教程[M]. 2016: 180-181.

致谢

时光荏苒，转眼间大学四年即将接近尾声。随着各项事情一件件落幕，在敲下这一段文字之时，离毕业又近了一些。四年时间可能不长，但对我而言，在此刻，有太多的记忆需要唤醒，太多的人需要感谢。

首先我想感谢我的父母、家人；我深知，没有你们我不可能完成本科的学业。你们给予了我太多，每当我遇到任何困难，我可以毫不犹豫地望向你们、家的方向。你们是我最坚强的后盾，如果世界上有我可以百分之百信任的人，那一定就是你们。特别的，大二暑假在做大创项目，我错过了与奶奶最后一段相处的时光，我清楚记得，哪怕最后一次见到意识清醒、躺在病床上的奶奶，也不忘对我叮嘱再三，不要熬夜、不要累到自己、注意身体…现在回想那段时光，心中也只能充满愧疚和感恩。

其次，我想感谢本科期间悉心教导我的老师们；感谢在课堂上为我们奉献一堂堂精彩课堂的东大各个学院的老师们以及为 16 级学生辛勤工作的辅导员宋晓燕老师、李世鹏老师等。感谢大创项目时的刘金海老师对我们项目的细致指导、提供帮助。感谢大三智能优化算法选修课的王大志老师为我单独指导，耐心地为我讲解算法。特别感谢智能工业数据解析与优化实验室的王显鹏老师，大三时进入实验室，面对一个陌生的环境、全新的知识，王老师慷慨地为我提供尽可能多的帮助；从电脑设备到书籍，从为我提供宝贵的锻炼机会到循循善诱，一次次为我讲解学院研究生招生的形式、未来我们专业的发展方向等等。一年半的实验室时光，一历历一幕幕，在老师的慷慨帮助下，我相对于之前的自己获得了太多的成长，学习到了很多新知识。在我考完研后，王老师作为我的毕业设计导师，对我的论文提供了莫大的支持和帮助。从一开始为我推荐文献，开会讲解项目内容；到后来开题时为我把握项目方向，关注我毕设的每一步进展；在我撰写论文时，帮助我修改论文提纲、论文细节，每当我对论文有任何问题，询问老师时，王老师都能第一时间为我提供帮助，解答我的疑惑。前前后后，没有王老师的帮助与指导，我怎能完成我的论文。与王老师一年半多时间的相处，我看到王老师时时刻刻都为他的学生着想，不论是做本科毕设的同学，还是在读硕士生、博士生，

甚至像我这样的“插班生”，每一位同学都能得到老师悉心的指导、周到的关怀。在这里我想真诚地感谢王老师长时间以来对我的帮助，对我的指导。

另外，我想感谢我本科阶段的同学、朋友们。包括 308 寝室的朋友：程铭、段鹏硕、柳琦、杨胜雄、于泽；218 寝室的朋友：洪宇望、卢启星、宋治民、杨胜雄、叶兆晖；和我一起完成大创项目的朋友：代琪源、刘旭、刘资；在智能工业实验室为我提供帮助的师兄师姐们：程浩、吴强、王赞、张云佳等；全体 1607 班同学、全体 1604 班同学；特别感谢大学的挚友：刘资、秦浩宇、叶兆晖等，四年的上下铺杨胜雄，考研时一起奋战的研友洪宇望、秦浩宇、宋雪、叶兆晖等，祝你们前程似锦、研路顺利。完成本文对应代码部分时提供给我帮助的宋琪先生和 Kentzoglanakis 先生。还有所有大学期间为我提供过帮助的、一起度过四年时光的朋友们，谢谢你们为我带来了一段难忘的大学校园经历。

最后，感谢这四年我遇到的每一个人和每一段经历，你们共同成就了现在的我，祝愿所有人都找到自己的前路，共赴愿景中的未来。对我而言，求学之路还未结束，这次的离别也只是一个新的开始，我希望自己可以把感恩和经历永存心底，踏上研路。

我尽可能缓慢地敲下这段文字，想着慢慢回味每一片记忆，尽管家中毕业显得有些遗憾，但或许不完美也是一种常态吧。