

BP神经网络与XGBoost预测隧道开挖地表沉降

贺子健^{1*}, 曾兴¹

(1.湖南科技大学土木工程学院, 湖南省湘潭市, 411201; * 通讯作者, 1030150785@qq.com)

摘要: 本章探讨了利用机器学习算法预测隧道开挖引起的地表沉降问题。首先详细介绍了BP (Back Propagation 反向传播) 神经网络的原理、数据选取方法 (包括地层物性参数的层厚标记法、施工参数和监测数据)、参数归一化以及算法框架构建。随后, 通过BP神经网络对地表沉降进行预测, 并使用平均绝对误差 (MAE)、均方根误差 (RMSE) 和决定系数 (R^2) 进行性能评估, 结果显示BP神经网络的预测效果不佳, 与实际情况偏差较大。接着, 介绍了XGBoost算法的原理, 并采用相同的数据集和评估指标对其进行地表沉降预测。对比分析表明, XGBoost算法在预测精度和泛化能力上均显著优于BP神经网络, 其预测结果与数值模拟值高度吻合, 最大预测误差远小于BP神经网络。研究结果证实XGBoost算法更适用于浅埋暗挖隧道掘进引起的地表沉降预测。

关键词: 地表沉降; 机器学习; BP神经网络; XGBoost; 参数归一化; 性能评估

引言

随着隧道工程复杂性的增加和对施工安全要求的提高, 传统经验公式在预测地表沉降方面有时存在局限性。机器学习算法因其强大的非线性拟合能力, 为地表沉降预测提供了新的途径。本章旨在研究并比较两种常用的机器学习算法——BP神经网络和XGBoost——在隧道开挖地表沉降预测中的应用效果, 以为其他工程实践提供更准确的预测工具。

1 BP神经网络预测隧道开挖地表沉降

1.1 BP神经网络原理

人工神经网络模仿了生物神经系统的行为特性[1], 旨在处理多种非线性难题。依据其构造及连接模式的差异, 这类网络可被归类为前馈、相互连接以及自组织三种类型。在前馈型网络中, 数据流从输入端逐层传递至输出端, 每一层的输出成为下一层的输入基础。BP (反向传播) 算法作为一种典型的前馈网络实现方式, 通过逆向调整权重与偏置值以减小预测结果与实际观测之间的差距, 直至满足既定误差标准为止。

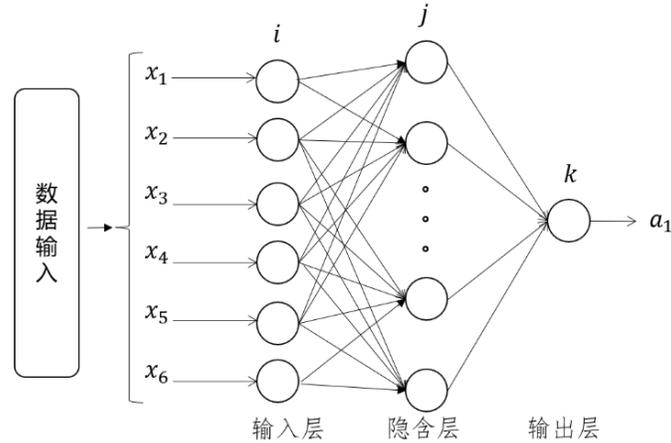


图1 前馈神经网络示意图

如图1所示，这是一个单层网格的前馈神经网络，在这种网络结构中，信息流动是单向的，从输入层开始，通过隐藏层（如果有的话），最后到达输出层。网络中不包含任何反馈或循环，这意味着每层的输出仅作为下一层的输入，直至产生最终输出。这样的设计使得网络能够专注于从输入到输出的直接映射，适用于处理一系列不同的预测和分类任务。

然而，前馈神经网络表达能力有限：单层网络由于只有一个隐藏层（或甚至没有隐藏层），其能够学习的特征和表达能力相对较弱，难以处理复杂的非线性问题。泛化能力有限：对于复杂的数据集和任务，单层网络的泛化能力不足，容易出现欠拟合，即模型无法在未见过的数据上表现良好。

在1986年，Rumelhart和McClelland提出了BP神经网络（反向传播神经网络，BPNN，Back Propagation Neural Network）[2]，其核心特点是通过误差的反向传播来修正权值和阈值，直到预测值与实测值之间的误差小于预先设定的标准。BP（反向传播）算法相对于单层前馈神经网络而言，引入了多层结构和反向传播的学习机制，极大地扩展了神经网络的应用范围和解决问题的能力。

如图2所示，BP神经网络算法中的基本处理单元是人工神经元，它是一种多输入、单输出的非线性处理单元。人工神经元的数学关系可用下面式（1）与式（2）进行描述：

$$I = \sum_{j=1}^n \omega_j x_j - \theta \quad (1)$$

$$y = f(I) \quad (2)$$

$x_j(j=1,2,3,\dots,n)$ 表示来自其他神经元的输入信号； θ 表示阈值；权函数 ω_{ij} 表示连接的强度； $f(x)$ 称为激活或作用函数，通过激活函数处理后得到的数值； y 即为下一层的输入值。在隐藏层中，使用正切 S 型传递函数 tansi ，如下图3所示，而在输出层中使用线性传递函数 purelin 。

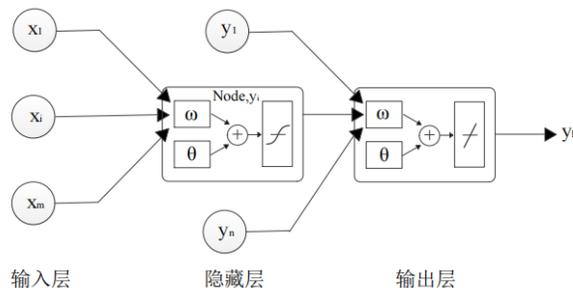


图2 BP神经网络示意图

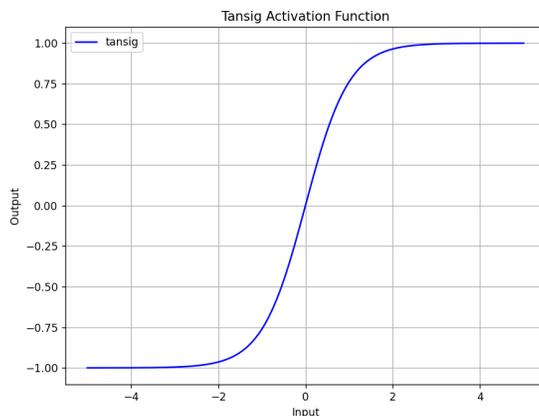


图 3 传递函数 tansi 示意图

BP 算法的主旨在于通过梯度下降法在假设的权重空间中进行搜索，目的是寻找能够最优拟合训练样本的权重配置。换句话说，它通过计算损失函数的梯度，并以此梯度的反方向调整权重和偏置，以期损失函数能达到其最低点。

具体来讲，反向传播算法通过计算损失函数对每一层权重和偏置的偏导数—即梯度—来更新权重和偏置。这一过程持续进行，直到达到损失函数的最小化或满足预设的迭代轮次。通过这种方式，算法能够找到神经网络参数的最优解。然后通过多次迭代反复执行前向传播、计算损失和反向传播的过程，逐渐调整网络参数，使得损失函数逐渐减小，直至达到收敛或设定的停止条件。BP 神经网络通过不断地调整权重和偏置，使得模型能够学习到输入与输出之间的复杂映射关系，从而实现回归模型的建立。

1.2 数据选取

(1) 地层物性参数

在考虑地质因素的机器学习算法中，必须对地质参数进行量化处理。地层参数主要有天然密度，黏聚力，内摩擦角，泊松比，压缩模量这五个参数。岩土层的物理力学性质、土层厚度以及土层的位置都会对隧道开挖造成的地表沉降产生影响。然而，目前文献中对土层参数的处理方法相对简单。通常的做法是直接输入土层的地层物性参数。尽管这些方法能够区分不同的岩土层，但存在以下问题：确认方法较为主观，物理意义不明确，并且考虑的因素不够全面。

本文将每层土的地层参数按照层厚进行修正，即添加一个层厚标记，用式 (3) 表示

$$\lambda_i = \frac{h_i}{H} \quad (3)$$

式中：

- λ_i ——第*i*层土的层厚标记
- h_i ——第*i*层土的层厚；
- H ——土体的总层深厚

本方法不需要获取额外的地层物性参数，层厚标记 λ_i 将直接与土厚成正比，能够更好的反应地质参数的物理意义。至于层深顺序，即土体从上至下进行标号。

(2) 施工参数

本文将施工参数分为隧道埋深，开挖进尺。这些都与隧道最终沉降密切相关。

(3) 监测数据

断面监测的拱顶下沉，水平收敛等参数，这与最后的地表沉降有紧密的联系。

1.3 参数归一化

为减少计算时间，优化计算过程，如下式（4）所示，将数据库中数据进行最小最大归一化（Min-Max Normalization）：将原始数据线性地映射到[0,1]的范围内。归一化后的数据计算方式如下：

$$x_{nowm} = \frac{x-x_{min}}{x_{max}-x_{min}}(x_{max}-x_{min}) + x_{min} \quad (4)$$

其中， x 是原始数据， x_{max} 和 x_{min} 分别是数据的最大值和最小值。 x^* 和1(原文为“和则是归一化后的最大最小值”，应指归一化后范围为[0,1])则是归一化后的值。

1.4 BP神经网络算法框架构建

建立数据库是机器学习模型构建的第一步，它承载了大量的数据，包括输入和输出参数。为了验证模型的性能和泛化能力，通常将数据库分为训练集和测试集。在这个过程中，常见的做法是将数据库中的90%数据作为训练集，而剩下的10%作为测试集，以确保模型能够充分学习和泛化。

在训练集和测试集中，包含了各种类型的输入输出参数，它们涵盖了模型需要学习和预测的各个方面。这些参数的数值范围和平均值对于模型的性能和预测结果至关重要。因此，在构建数据库时，需要确保输入参数和输出参数的范围合理，并且考虑到数据的平均值，以便模型能够更准确地学习和预测。训练集和测试集数据范围见表1、表2：

表1 训练集数据范围

变量（缩写）	参数类型（缩写）	最小值	最大值	平均值	单位
几何参数	埋深（M）	6	6	6	m
	开挖距离	0	80	40	m
监测参数	拱顶下沉（GD）	0.74	3.95	0.35	mm
	水平收敛左（SPZ）	3.46	51	0.4	mm
	水平收敛右（SPY）	3.8	24.2	0.76	mm
地质参数	天然密度（ ρ ）	1600	2200	2000	kg/m ³
	内摩擦角（ φ ）	2	5	3	°
	粘聚力（c）	18	45	28	kpa
输出	最大沉降（S）	-11.75	0	-9	mm

表2 测试集数据范围

变量（缩写）	参数类型（缩写）	最小值	最大值	平均值	单位
几何参数	埋深（M）	6	6	6	m
	开挖距离	0	78	39	m
监测参数	拱顶下沉（GD）	1.02	3.55	0.45	mm
	水平收敛左（SPZ）	3.46	51	0.4	mm
	水平收敛右（SPY）	3.8	24.2	0.76	mm
地质参数	天然密度（ ρ ）	1600	2200	2000	kg/m ³
	内摩擦角（ φ ）	2	5	3	°
	粘聚力（c）	18	45	28	kpa
输出	最大沉降（S）	-11.75	0	-9	mm

1.5 性能评估指标

为了评估不同机器学习算法在预测由盾构掘进引起的地表沉降方面的性能，Tseranidis 等人在2016年的研究中总结了八种误差评价指标[3]。在本项研究中，虽然本文是浅埋暗挖法施工，但性能评估的指标仍旧可

以参考，本文选择了平均绝对误差（MAE）、均方根误差（RMSE）和决定系数（ R^2 ）这三个指标来衡量模型的性能。三个指标的计算方法如式（5），式（6），式（7）所示。

$$MAE = \left(\frac{1}{n}\right) \sum_i^n |r_i - p_i| \tag{5}$$

$$RMSE = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (r_i - p_i)^2} \tag{6}$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (r_i - \bar{r})^2}{\sum_{i=1}^n (r_i - \bar{r})^2} \tag{7}$$

平均绝对误差（MAE）衡量了预测值与实际值之间的平均误差大小，它提供了一个直观的误差量度。均方根误差（RMSE）则考虑了预测误差的平方，因此对较大的误差给予了更高的权重，这有助于我们了解模型预测的波动性。决定系数（ R^2 ）则衡量了模型预测值与实际值之间的相关程度，其值越接近 1，表示模型的预测能力越强。

通过这三个指标的综合评估，我们可以BP神经网络算法在地表沉降预测任务上的性能进行有效的分析。

1.6 性能评估指标与预测沉降效果分析

通过BP神经网络模型的训练，5折交叉验证以及寻找最优超参数后，得出了3个性能评估指标。

测试集上的性能：

均方误差（MSE）：2.4143e-06，这个值相对较小，表明模型在测试集上的预测精度较高。

决定系数（ R^2 ）：0.6993，这个值表明模型能够解释测试数据中约 69.94%的变异性，但还有进一步优化的空间。

训练集上的性能：

均方误差（MSE）：9.94316e-06，这个值也相对较小，表明模型在训练集上的预测精度较高。

决定系数（ R^2 ）：0.4957，这个值低于测试集上的 R^2 值，表明模型在训练集上的解释能力较弱。

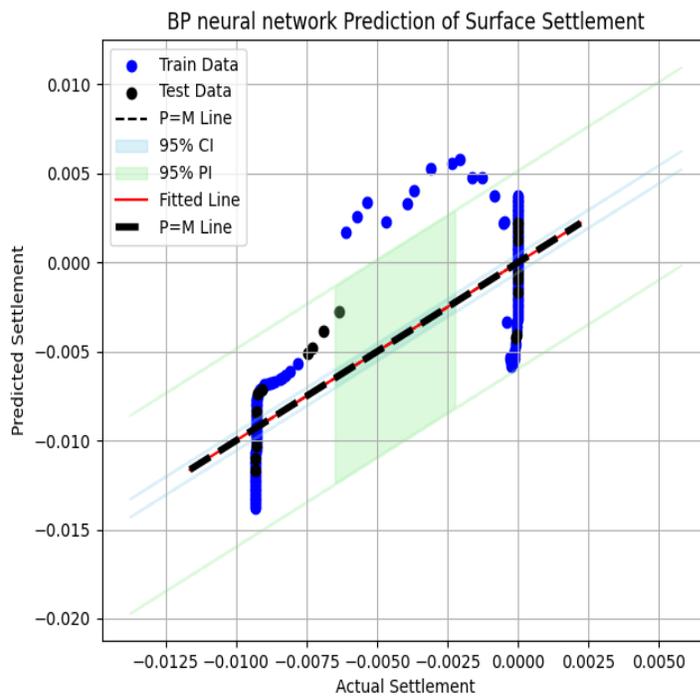


图 4 地表沉降的 BP 神经网络预测分析图

如图4所示,训练数据(Train Data)与测试数据(Test Data)大部分都不在完美预测线和95%置信区间和预测区间里,这说明了模型在训练过程中的学习情况不理想,而模型在未见过的数据上的泛化能力也比较一般。

误差分析:通过观察95% CI和95% PI,我们可以评估模型预测的可靠性。较窄的区间表明模型预测较为稳定,而较宽的区间则表明预测的不确定性较高。而图中的两个区间都比较宽,而且区间之间还有断层,这说明了模型预测的稳定性也较差。

虽然P=M Line(完美预测线)和拟合线(Fitted Line)几乎完全重合,这显示了模型预测的趋势与理想情况下的预测,极其相似,预测分析相当优秀。但由于95%置信区间的点太多,预测的不确定性太高了。

将训练出的模型保存下来,将GKD+086断面的数据输入模型,得到BP神经网络预测的地表沉降。

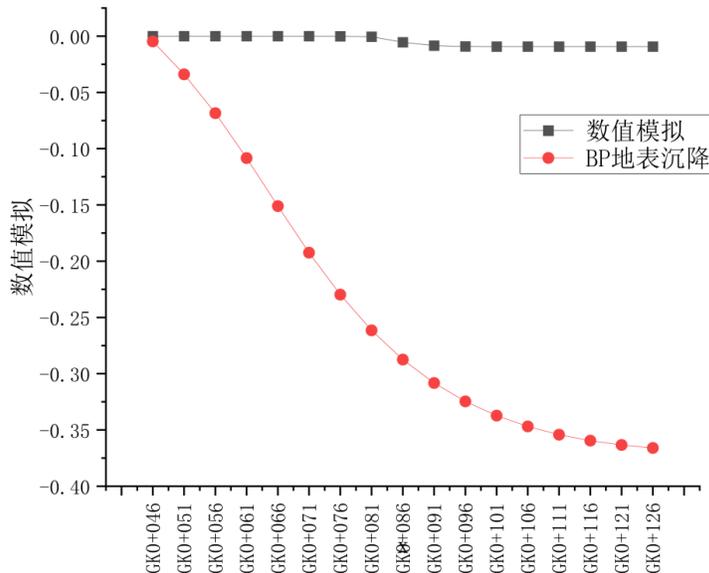


图5 GKD+086 BP神经网络的地表沉降预测图

如图5所示, BP神经网络对本工程的拟合比较差,不具有预测价值。

2 XGBoost 算法预测隧道开挖地表沉降

2.1 XGBoost 算法原理

XGBoost算法[4],即eXtreme Gradient Boosting,是Boosting算法家族中的一种。Boosting算法的核心思想是将多个弱学习器(也称为弱分类器)串行集成在一起,形成一个强学习器,通过迭代地训练每个弱学习器,使其关注于前一轮训练中被错误分类的样本,从而不断提升模型的预测性能。针对一个训练集,XGBoost首先使CART树[5]训练得到一个模型,这样针对每个样本都会产生一个偏差值;然后将样本偏差值作为新的训练集,继续使用CART树训练得到一个新模型;以此重复,直至达到某个退出条件为止。示意图如下图6所示

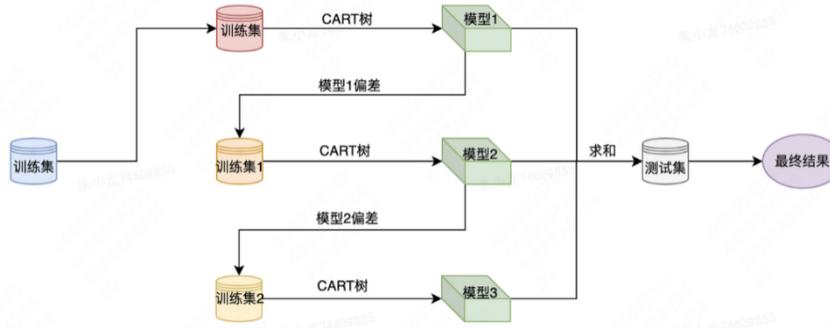


图 6 XGBoost 算法原理示意图

具体来说，XGBoost 通过梯度提升（Gradient Boosting）的方式训练决策树模型。在每一轮迭代中，XGBoost 先计算当前模型的梯度损失，然后使用决策树模型去拟合这个损失的负梯度。这样，XGBoost 能够有效地降低损失函数，并逐步提升模型的预测能力。

XGBoost 是一个加法模型，如式（8）所示，在每一步迭代中只优化当前步中的子模型。在第 m 步中：

$$F_m(x_i) = F_{m-1}(x_i) + f_m(x_i) \tag{8}$$

$f_m(x_i)$ 为当前步的子模型， $\hat{y}_i^{(m-1)}$ 为训练完已经固定了的前个子模型。

式（9）为目标函数，由经验风险+结构风险（正则项）组成：

$$\begin{aligned} Obj &= \sum_{i=1}^N l[F_m(x_i), y_i] + \sum_{j=1}^m n(f_j) \\ &= \sum_{i=1}^N l_i [F_{m-1}(x_i) + f_m(x_i), y_i] + \sum_{j=1}^m n(f_j) \end{aligned} \tag{9}$$

如式（10）所示，XGBoost 运用泰勒公式的二阶展开来近似表达损失函数：

$$f(x_0 + \Delta x) \approx f(x_0) + f'(x_0)\Delta x + \frac{f''(x_0)}{2} (\Delta x)^2 \tag{10}$$

式（10）中，将 $\hat{y}_i^{(m-1)}$ 用 x_0 代替，将 $f_m(x_i)$ 视作 Δx ，当成关于 $f_m(x_i)$ 的函数

$$Obj = \sum_{i=1}^N \left\{ L[F_{m-1}(x_i), y_i] + \frac{\partial L}{\partial F_{m-1}(x_i)} f_m(x_i) + \frac{1}{2} \frac{\partial^2 L}{\partial^2 F_{m-1}(x_i)} f_m^2(x_i) \right\} + \sum_{j=1}^m \Omega(f_j) \tag{11}$$

由于模型是迭代计算的，式（11）中只有 $f_m(x_i)$ 是变量。目标函数可转化为式（12）：

$$Obj = \sum_{i=1}^N \left[g_i f_m(x_i) + \frac{1}{2} h_i f_m^2(x_i) \right] + \Omega(f_m) \tag{12}$$

再通过 CART 决策树 $f_m(x) = w_{q(x)}$

x 为某一样本，这里的 $q(x)$ 代表了该样本在哪个叶子节点上， w_q 代表叶子节点取值。

当一棵树的结构被确定下来后，树中各个节点所包含的数据样本 x_i 、 y_i 、 g_i 、 h_i 也随之确定。在这种情况下，每个叶节点所应输出的回归值是为了最小化给定的目标函数。如式（13）所示，根据二次函数求极值的原理，这个输出值应当选取能使目标函数达到最小值的那个点：

$$w_j^* = - \frac{\sum_{i \in I(j)} g_i}{\sum_{i \in I(j)} h_i + \lambda} \tag{13}$$

之后再通过找到叶子的最优切点，通过贪心和近似算法来确定。对于较小的数据量，使用贪心算法。在树的深度为零的起点，对每个叶节点进行特征遍历。针对每个特征，根据其值将该节点的训练数据按升序排序，并通过线性搜索确定最优分裂点，记录下相应的分裂增益。选择增益最大的特征作为分裂依据，以此划分节点，形成左右两个新的叶节点，每个新节点都与相应的数据集关联。这个过程递归执行，直到满足特定的停止条件。

递归执行以上每一步，直到满足设定的特定停止条件为止。但是每次分裂，枚举所有特征可能的分割方案，从效率上看是比较低的。所以对于较大的数据量，使用近似算法。近似算法是为了应对贪婪算法在处理大规模数据时的计算复杂度过高的问题而提出的。该算法通过考察每个特征的分位点来减少计算复杂度。

具体而言，该算法首先根据特征的分布情况提出候选划分点，然后将连续型特征映射到这些候选点所确定的区间中。接着，算法聚合统计信息，找到所有区间的最佳分裂点。

2.2 数据选取

直接使用表 1 与 2 的测试集与训练集数据。

2.3 性能评估指标与预测沉降效果分析

通过模型的训练，交叉验证以及寻找最优超参数后，得出了 3 个性能评估指标。

测试集的 MSE 为 $2.04\text{e-}07$ ，这是一个非常小的数值，表明模型在测试集上的预测非常接近真实值。

测试集的 R^2 为 0.9886，接近 1，表明模型能够解释测试数据中 99% 的变异性，这是一个非常好的结果。

测试集的 MAE 为 0.0002，也非常接近 0。

训练集的 MSE 为 $7.77060\text{e-}09$ ，同样非常小，表明模型在训练集上也有很好的拟合。

训练集的 R^2 为 0.99960，几乎接近 1，表明模型几乎能够完美解释训练数据中的所有变异性。

这些结果表明，该模型在训练集和测试集上都有很好的性能，具有很高的预测准确性和解释能力。然而，需要注意的是，尽管训练集上的 R^2 非常高，但过度拟合的风险仍然存在，尤其是当训练集上的 R^2 接近 1 而测试集上的 R^2 相对较低时。不过在一定情况下，测试集上的 R^2 也非常高，这表明模型可能具有很好的泛化能力。

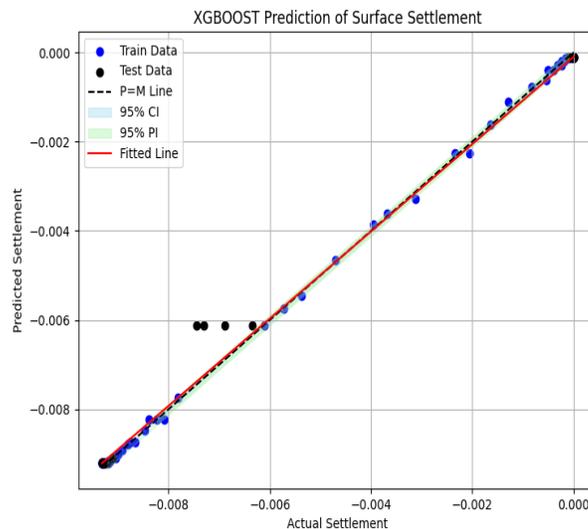


图 7 地表沉降的 XGBoost 预测分析图

如图 7 所示，训练数据 (Train Data) 与测试数据 (Test Data) 大部分都在完美预测线和 95% 置信区间和预测区间里，测试集有几个点不在，这说明了模型在训练过程中的学习情况非常优秀，而模型在未见过的数据上的泛化能力稍为逊色一些，但也非常优秀。

P=M Line (完美预测线) 和拟合线 (Fitted Line) 几乎完全重合。这显示了模型预测的趋势与理想情况下的预测，极其相似，预测分析相当优秀。

误差分析：通过观察 95% CI 和 95% PI，我们可以评估模型预测的可靠性。较窄的区间表明模型预测较为稳定，而较宽的区间则表明预测的不确定性较高。而图中的两个区间都比较窄，这说明了模型预测的稳定性也较高。

在交叉验证寻找最优参数时，将学习率的数据提取如图 8 所示，可以看出平均测试分数在学习率为 0.1 和 0.2 时更接近 0，也就是性能更优秀。当学习率太小的时候，决策树容易过拟合，所以在本工程中，学习率设定为 0.2，模型性能最优。

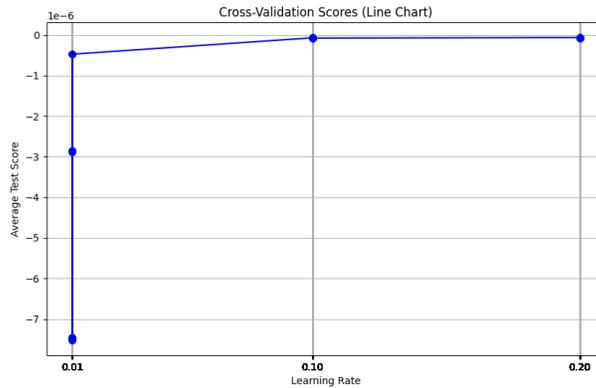


图 8 交叉验证分数与学习率的关系图

将训练出的模型保存下来，将 GKD+086 断面的数据输入模型，得到 XGBoost 预测的开挖到每一个断面里程时 GKD+086 的最大地表沉降。

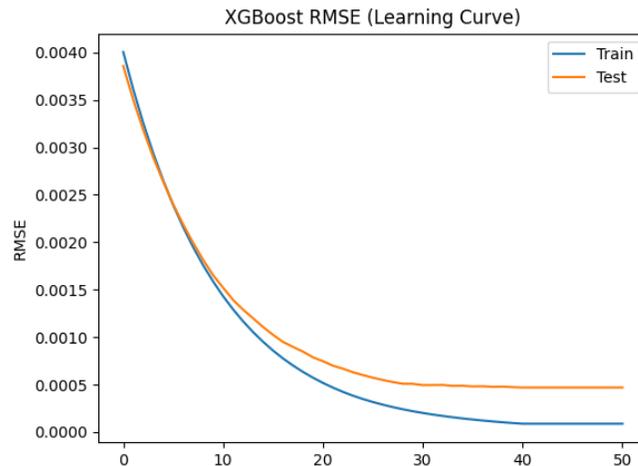


图 9 迭代损失率图

图 9 是 XGBoost 迭代损失率图，可得，初期快速提升：训练集和测试集的 RMSE 在最初的 boosting 轮数中（大约从 0 到 10）迅速下降。这表明模型正在快速学习数据中的主要模式。

收敛：两条曲线都趋于平缓，并收敛到一个较低的 RMSE 值。训练误差继续略微下降，而测试误差趋于平稳。这种收敛表明模型已经有效地学习了数据中的潜在关系，并且不会随着 boosting 轮数的增加而显著改善。

训练集和测试集之间的差距小：训练集和测试集的 RMSE 值非常接近，特别是在大约 20 轮 boosting 之后。这是一个非常好的迹象。它表明模型很好地泛化到未见过的数据（测试集），并且没有过拟合训练数据。过拟合的表现是存在较大差距，即训练误差远低于测试误差。

最佳轮数：该图表明，对于这个特定的模型和数据集，大约 40-50 轮 boosting 就足够了。超过这个点再增加轮数并不会导致训练或测试性能的显著提高，甚至可能略微增加过拟合的风险（尽管在这种情况下，过拟合很小）。

所以在该数据集中 XGBoost 模型表现非常出色。它学习速度快，能很好地泛化到未见过的数据，并实现了非常低的错误率。没有过拟合的迹象。最佳 boosting 轮数似乎在 40-50 左右。

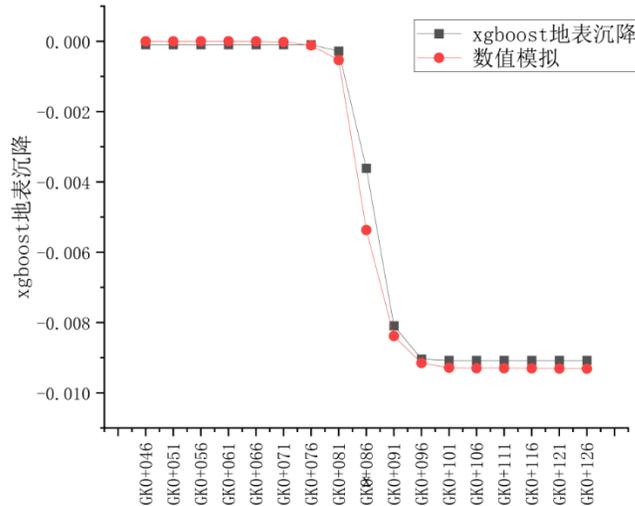


图 10 GKD+086 XGBoost 的地表沉降预测图

如图 10 所示，GKD+086 XGBoost 的地表沉降预测非常准确，无论是数值还是趋势都与数值模拟的结果拟合的很好。

3 结论

1.建立了地层物性参数，施工参数，监测数据等为自变量，地表沉降为因变量的两种机器算法回归预测模型。并通过对比得出了 XGBoost 算法更优的结论。

2.测试集和预测集的预测结果说明了 BP 神经网络预测的结果波动性较大，与数值模拟值差别很大，且趋势不同。测点上失真严重。

3.BP 神经网络算法的最大预测误差达到 350mm，而 XGBoost 的最大预测误差为 2mm。综上所述，XGBoost 算法更适于预测浅埋暗挖掘进引起的地表沉降问题。

4.在训练数据集上，XGBOOST 模型的预测结果与实际观测值高度吻合；而在测试数据集上，该模型生成的平均绝对误差（MAE）和均方根误差（RMSE）分别为 0.00024331689843658062 和 7.770601335660143e-09，测试数据点绝大多数都紧密分布在理想预测线（P=M 线）的附近。这一现象表明模型在测试集上也展现出了良好的预测性能。

5.在本工程中，XGBoost 的学习率设置的太小时，平均测试分数也比较小，学习率设置为 0.2 为交叉验证后的最优参数。

参考文献

- [1] MCCULLOCH W S, PITTS W. A logical calculus of the ideas immanent in nervous activity [J]. Bulletin of Mathematical Biology, 1990, 52(1): 99-115.
- [2] RUMELHART D E, HINTON G E, WILLIAMS R J J N. Learning representations by back-propagating errors [J]. 1986, 323(6088): 533-6.
- [3] TSERANIDIS S, BROWN N C, MUELLER C T. Data-driven approximation algorithms for rapid performance evaluation and optimization of civil structures [J]. Automation in Construction, 2016, 72: 279-93.
- [4] CHEN T, GUESTRIN C, POTNASICO K D, MINING D. XGBoost: A Scalable Tree Boosting System [J]. 2016.
- [5] LOHWY J, WIRD M, DISCOVERY K. Classification and regression trees [J]. 2011, 1(1): 14-23