

# 基于大数据的工程施工质量检测研究与应用

段连景 朱立会

潍坊智博建设工程质量检测有限公司

**摘要:** 随着工程规模和复杂度的不断增加, 传统的质量检测方法已经无法满足工程施工的要求。而大数据技术的出现, 为工程施工质量检测提供了新的思路 and 手段。本文对当前工程施工质量检测存在的问题进行了一定论述, 在此基础上, 进一步探讨了基于大数据的工程施工质量检测方法, 并分析了基于大数据的工程施工质量检测模型和检测系统设计 with 实现, 有助于推动大数据在工程施工质量检测中应用的不断深入, 进而为工程施工质量提供可靠保障。

**关键词:** 大数据; 工程施工; 质量检测

**【DOI】** 10. 12254/j. issn. 2096-6539. 2023. 07. 019

## 一、前言

随着社会和经济的不断发展, 各种基础设施建设、城市化进程不断推进, 对工程施工质量的要求也越来越高。传统的工程施工质量检测方法主要依靠人工检查和经验判断, 存在效率低、精度差、盲区多等问题, 不能满足现代化工程建设的要求。因此, 基于大数据的工程施工质量检测成了一种新的解决方案, 可以实现全面、精准、高效的施工质量监测, 为工程建设提供可靠的保障。

## 二、工程施工质量检测存在的问题

首先, 检测方法单一, 缺乏全面性。传统的工程施工质量检测方式采用人工检测, 只能对工程的部分环节进行检测, 难以全面地评估工程质量水平。例如, 传统的钢筋数量检测只能通过人工点验钢筋数量, 无法检测钢筋的粗细和长度等参数, 缺乏全面性和准确性。其次, 人工检测容易出现误差和主观因素。由于人员的技术水平和经验不同, 工程质量的评估结果容易受到主观因素的影响, 导致检测结果不够准确。例如, 在施工质量检测中, 检测人员可能会出于个人偏见或不同的检测标准对同一工程进行不同的评估, 从而影响检测结果的准确性。再次, 数据管理不便。传统的工程施工质量检测主要依靠人工记录和管理数据, 这种方式效率低下、数据不易管理。例如, 在传统的工程施工质量检测中, 检测数据需要手动记录, 且缺乏标准化的管理方式, 导致数据难以有效地分析和利用。最后, 检测周期长, 不能及时发现问题。传统的工程施工质量检测需要花费较长的时间和人力, 难以及时发现问题和解决问题, 对工程的后续施工和使用带来不利影响。例如, 在传统的工程施工质量检测中, 需要进行多次的人工检测和验收,

这种方式效率低下、周期长, 难以及时发现问题并采取相应的措施加以解决。

## 三、基于大数据的工程施工质量检测方法

### (一) 大数据在工程施工质量检测中的应用

随着大数据技术的快速发展和应用, 越来越多的领域开始尝试将大数据技术应用于实际工作中。在工程领域中, 工程施工质量的检测一直是一个重要的问题, 如何准确、高效地检测工程施工质量是工程管理人员面临的挑战。基于大数据的工程施工质量检测方法是一种新的技术手段, 可以大幅提高工程施工质量的检测效率和准确性。

在工程施工质量检测中, 大数据技术可以应用于以下方面:

首先, 通过传感器、监控设备等实时获取施工过程中的各项数据, 例如温度、湿度、振动等, 以及工程施工过程的监控数据。其次, 分析工程施工过程中采集到的数据, 检测施工中可能存在的问题和缺陷, 例如施工偏差、安全隐患等。最后, 借助大数据分析和机器学习技术, 对工程施工质量数据进行处理和分析, 挖掘关键因素和特征, 以提高工程施工质量的检测效率和准确性。

### (二) 基于大数据的工程施工质量检测框架

基于大数据的工程施工质量检测框架包括数据采集、数据处理和质量检测三个主要步骤。

首先是数据采集, 通过传感器、监控设备等实时获取施工过程中的各项数据。这些数据包括施工环境的温度、湿度、振动等数据, 以及施工过程中的监控数据。其次是数据处理, 对采集到的数据进行处理和分析, 通过大数据分析和机器学习等技术, 挖掘数据的关键特征和因素。例如, 可以分析温度和湿度的变化规律, 以及其对施工质量的影响。最后是质量检测, 根据数据处理的结果, 对工程施工质量进行检测。可以利用机器学习模型和算法, 对施工中可能存在的问题和缺陷进行自动化检测和判断。

### (三) 工程施工质量数据采集和处理

在工程施工过程中, 采集施工环境的温度、湿度、振动等数据, 以及施工过程中的监控数据, 对于工程施工质量的检测和评估具有重要作用。工程施工质量数据采集和处理主要包括以下内容:

首先是传感器的选择, 根据实际需求选择合适的传感器, 如温度传感器、湿度传感器、振动传感器等, 以

及监控设备，如摄像头等。其次是数据采集和传输，通过传感器和监控设备等实时采集数据，并将其传输到数据处理中心，实现数据的实时监控和管理。第三是数据预处理，对采集到的数据进行清洗、处理和筛选，去除不合格数据和异常数据，保证数据的准确性和可靠性。第四是大数据分析和处理，运用大数据分析和机器学习等技术，对数据进行分析和处理，挖掘数据的关键特征和因素，为施工质量的检测提供依据。最后是数据可视化，将处理后的数据以图表等形式进行可视化展示，方便管理人员进行分析和决策。

#### 四、基于大数据的工程施工质量检测模型

##### （一）设计思路

基于大数据的工程施工质量检测模型的设计思路主要是从采集到的大量数据中提取关键特征，建立合适的机器学习模型，并利用算法进行优化，实现对施工质量的检测和判断。工程施工质量检测模型的设计思路主要包括以下几个方面：

首先是数据预处理，在进行数据分析和建立模型之前，需要对采集到的数据进行预处理，包括数据清洗、数据归一化、数据平滑等步骤。这些步骤可以使得数据更加准确、可靠，避免模型出现过拟合等问题。其次是特征提取，通过特征提取，将原始的大量数据转化为更加有效的特征。特征提取需要考虑到施工质量检测的具体目标，例如可选取的特征包括振动强度、温度变化、湿度变化等。特征提取的目的是为了减少数据的维度，降低模型复杂度，提高模型的鲁棒性。第三是建立模型，根据提取到的特征，建立合适的机器学习模型，例如基于支持向量机（SVM）、随机森林（Random Forest）等模型进行建模。模型的建立需要考虑到数据分布的特点，以及模型的复杂度和泛化能力等因素。第四是模型优化，模型优化主要包括参数调优和模型选择两个方面。在参数调优中，可以采用网格搜索、交叉验证等方法，寻找最优的模型参数；在模型选择中，需要权衡模型的精度、泛化能力、复杂度等因素，选择最适合的模型。最后是模型评估，通过对建立的模型进行评估，可以了解模型的预测能力和精度。评估方法包括交叉验证、ROC曲线、混淆矩阵等。评估结果可用于改进模型，提高施工质量检测的精度和可靠性。

##### （二）优化方法

工程施工质量检测模型的优化方法主要包括以下几个方面：

首先是数据增强，数据增强是通过对原始数据进行变换、扩展等方式增加数据量和多样性，从而提高模型的鲁棒性和泛化能力。例如，在施工现场中，可以通过增加传感器的数量和种类，增加监控设备的分布范围，扩大数据采集的覆盖范围，从而获取更多的施工质量数

据。此外，还可以利用数据增强技术，对原始数据进行旋转、翻转、缩放等变换，增加数据的多样性和覆盖范围。其次是特征选择，特征选择是从提取到的大量特征中筛选出对施工质量影响最大的关键特征，减少模型的复杂度和计算量，提高模型的训练和预测效率。特征选择可以采用过滤、包裹、嵌入等不同的方法，从而确定最优特征子集。在选择特征时，需要考虑特征之间的相关性、重要性以及对模型的贡献程度。再次是模型集成，模型集成是通过将多个模型的预测结果进行整合，提高模型的预测精度和鲁棒性。常见的模型集成方法包括投票、加权平均、堆叠等。通过模型集成，可以弥补单一模型的缺陷，从而提高模型的性能和效果。最后是在线学习，在线学习是利用一些自适应算法和模型，动态地更新模型参数和结构，以适应施工过程中不断变化的环境和条件，提高模型的实时性和准确性。在工程施工质量检测中，通过在线学习，可以实时地更新模型，捕捉施工过程中的变化和异常情况，从而及时发现和解决问题。常见的在线学习算法包括随机梯度下降、增量学习、在线支持向量机等。

#### 五、基于大数据的工程施工质量检测系统设计与实现

##### （一）系统总体设计

基于大数据的工程施工质量检测系统总体设计包括三个主要模块：数据采集与处理模块、工程施工质量检测模块和用户界面模块。

（1）数据采集与处理模块的具体实现包括多个方面，例如数据采集设备的选择和配置、数据传输协议的设计、数据清洗和去重算法的实现等。对于不同类型的数据采集设备，需要选择合适的通信协议和接口，并对数据进行预处理和格式转换，以保证数据的质量和可用性。数据清洗和去重算法主要是针对采集到的原始数据进行预处理和优化，例如去除异常值和噪声数据、填充缺失值等操作。

（2）工程施工质量检测模块的具体实现包括多个方面，例如模型选择和训练、模型集成、特征选择和优化等。模型选择和训练是指选择合适的机器学习模型和算法，并利用采集到的数据进行模型训练。模型集成是指将多个模型的预测结果进行整合，提高模型的准确性和鲁棒性。特征选择和优化是指从采集到的大量特征中，选择对施工质量影响最大的关键特征，并对模型进行优化和调整，以提高模型的性能和泛化能力。

（3）用户界面模块的具体实现包括多个方面，例如界面设计和交互设计、数据可视化、报告生成等。界面设计和交互设计是指设计直观、易用的用户界面和交互方式，提高用户体验和操作效率。数据可视化是指将采集到的数据进行图表、曲线等形式的展示，以便于用

户更加直观地了解施工质量情况。报告生成是指根据检测结果，自动生成相应的报告和建议，以供用户参考和决策。

### （二）数据库设计

系统数据库是基于大数据的工程施工质量检测系统的重要组成部分，用于存储采集到的施工环境数据、监控数据和施工质量数据等。数据库的设计应当满足数据的实时性、完整性、可扩展性和查询效率等要求，以保证系统的性能和可靠性。

首先，数据库类型的选择，关系型数据库和NoSQL数据库各有优劣，需要根据具体需求和场景来选择。关系型数据库适合于事务性系统和数据结构稳定的应用，如MySQL、Oracle等；NoSQL数据库适合于处理大规模的非结构化数据和数据处理速度要求高的应用，如Hadoop、HBase等。在大数据工程施工质量检测系统中，可以综合考虑使用关系型数据库和NoSQL数据库的优点，如MySQL和HBase结合使用。其次，数据表的设计，数据库中的数据表应当根据数据类型和业务需求进行设计。在大数据工程施工质量检测系统中，可以设计施工环境数据表、监控数据表、施工质量数据表等数据表，以存储不同类型的数据。具体的数据表字段设计应当考虑到数据的实时性、完整性、可扩展性和查询效率等因素，以确保数据的质量和可靠性。例如，施工环境数据表可以包含温度、湿度、振动等字段，监控数据表可以包含摄像头、传感器等字段，施工质量数据表可以包含质量评估结果、质量问题类型、解决方案等字段。再次，数据库索引的设计，数据库索引是数据库中用于加速数据查询和提高查询效率的一种数据结构，可以提高系统的性能和响应速度。在大数据工程施工质量检测系统中，可以根据具体的查询需求和数据表字段设计情况，设计适当的数据库索引，如基于时间的索引、基于地理位置的索引等。需要注意的是，过多的索引会增加数据库的存储和维护成本，同时也会降低数据库的写入性能。最后，数据库的备份和恢复，数据库的备份和恢复是保证系统数据可靠性和安全性的重要措施。在大数据工程施工质量检测系统中，可以采用自动备份和定期备份等方式，以确保数据的可靠性和完整性。同时，也需要注意备份和恢复操作的时效性和有效性，以保证系统的实时性和准确性。

### （三）功能设计

基于大数据的工程施工质量检测系统总体设计包括三个主要模块：数据采集与处理模块、工程施工质量检测模块和用户界面模块。

#### （1）数据采集与处理模块：

该模块主要用于实时采集施工现场环境数据和监控数据，并对这些数据进行预处理。数据采集可以通过传

感器、监控设备等实现，例如温度计、湿度计、振动传感器等。采集的数据可能存在异常值和噪声，因此需要进行清洗、去重、转换等预处理操作，以确保数据的准确性和可靠性。同时，该模块采用分布式架构，可以支持多种数据采集设备，并实现数据的实时采集和处理。

#### （2）工程施工质量检测模块：

该模块主要基于机器学习模型和算法对采集到的数据进行分析 and 建模，实现施工质量的自动化检测和判断。在模型设计方面，可以选择不同的机器学习算法和模型，例如决策树、支持向量机、神经网络等。通过对采集的数据进行处理和分析，挖掘数据的关键特征和因素，建立相应的机器学习模型，实现对施工质量的预测和评估。同时，为了提高模型的准确性和泛化能力，需要进行模型的训练和优化。

#### （3）用户界面模块：

该模块主要是整个系统的前端展示部分，支持用户登录、数据可视化、报告生成等功能。该模块采用Web技术，支持跨平台、跨设备的访问。用户可以通过界面实现数据的可视化展示，例如图表、曲线等形式的展示，以便于用户更加直观地了解施工质量情况，并进行相应的决策和调整。同时，用户也可以通过该模块生成相应的报告，对施工质量进行总结和评估。

## 六、结语

本研究基于大数据的工程施工质量检测方法及其在实际工程中的应用，旨在提高工程施工过程中的质量监控和管理水平，减少质量问题的发生，提高工程施工的效率和安全性。未来，基于大数据的工程施工质量检测技术将得到更广泛的应用，可以与传统的工程方法相结合，提高工程质量和效率，促进工程行业的可持续发展。

## 参考文献

- [1] 赵峰, 周斌, 温晓凯, 冉弥. 第三方工程质量检测信息平台设计与开发[J]. 现代隧道技术, 2022, 59(S1): 1037-1042.
  - [2] 张善华, 郭春伶. 公路工程质量检测监督信息化发展研究[J]. 中国建设信息化, 2022, (10): 62-64.
  - [3] 刘有为. 建筑工程质量检测行业规范性评价及其改进研究[D]. 重庆大学, 2021.
  - [4] 郑留欢. 土木工程施工质量检测信息化研究[D]. 安徽理工大学, 2020.
  - [5] 徐琰. 新科技促进工程质量检测与监管的发展[J]. 科技风, 2020, (23): 104.
  - [6] 陈杰. 建设工程质量检测信息化管理系统的应用[J]. 住宅与房地产, 2018, (28): 120.
- 作者简介：段连景（1991.08-），男，汉，山东曹县人，本科，中级工程师，研究方向：建筑材料。