

人工智能技术在软件相似性鉴定中的应用

余铭华¹
YU Minghua

摘要

针对电子数据相似性鉴定领域的技术需求,文章深入研究人工智能技术在软件相似性鉴定中的应用实践。首先阐述了人工智能技术应用于软件相似性鉴定的研究背景与现实意义;随后解析了机器学习、深度学习等核心人工智能技术在该领域的应用原理,并结合实际案例验证其应用效果,证实该技术能显著提升软件相似性鉴定的效率与准确性;最后探讨了人工智能技术应用过程中面临的数据隐私保护、算法解释性不足等关键挑战,并提出针对性解决策略及未来发展方向,为软件相似性鉴定领域高效、合理运用人工智能技术提供理论支撑与实践参考。

关键词

电子数据; 软件相似性; 人工智能; 司法鉴定

doi: 10.3969/j.issn.1672-9528.2026.04.028

0 引言

随着信息技术的发展,软件已经成为电子数据司法鉴定中重要的对象。在计算机软件著作权纠纷案件中,软件相似性鉴定结果往往是判定侵权行为是否成立的核心依据,根据近三年《最高人民法院知识产权法庭年度报告》的发布内容,计算机软件纠纷案件非常显著,在受理案件中一直稳居较高的比例。2022年最高人民法院新收2956件民事二审实体案件中包含计算机软件纠纷648件;2023年最高人民法院新收3222件民事二审实体案件中包含计算机软件纠纷704件;2024年新收民事二审实体案件1289件中包含计算机软件纠纷264件。

在司法实践中,电子数据相似性鉴定是明确软件侵权法律责任的重要依据,软件相似性鉴定技术规范^[1]和软件相似性检验技术方法^[2]标准为电子数据相似性鉴定工作提供了科学、规范的依据,确保了鉴定结果的准确性和公正性,在软件知识产权纠纷领域发挥着重要的作用。电子数据相似性鉴定通过比对源代码、目标代码、软件结构、功能模块等方面内容,为司法机关认定侵权提供重要的依据。

传统的电子数据相似性鉴定方法主要依靠鉴定人员进行重复性检验操作,但是随着软件系统的规模越来越大、复杂性越来越高,传统的电子数据相似性鉴定方法面临效率低、容易疲劳、主观误判等问题。近年来,人工智能技术在图像识别、自然语言处理等场景取得了显著成果,人工智能技术对基于规则任务、重复性工作具有天然的优势,其高效处理任务和自动化的特点为电子数据相似性鉴定带来了新的机遇^[3]。

1 电子数据相似性鉴定

1.1 概述及定义

在司法鉴定中,电子数据相似性鉴定是指遵循软件相似性鉴定标准,通过专业的技术手段和方法,对检材及样本的软件源代码、数据库、资源文件等按照对应比对的原则进行比对检验,来判断检材和样本在功能、代码结构、算法逻辑等方面是否存在相似性的过程。电子数据相似性鉴定通常用来处理涉及知识产权侵权的问题^[4]。

在软件产业中,软件开发者投入大量的时间、精力和资源进行软件研发,软件的知识产权是其核心利益所在。电子数据相似性鉴定能够准确判断软件是否侵权,为软件知识产权保护提供有力的证据支持,维护软件开发者的合法权益,防止其他企业通过抄袭、侵权等不正当手段获取竞争优势,对鼓励软件创新、推动软件产业的可持续发展具有重要意义。

一般来说,电子数据相似性鉴定中相似情形大致可以分为以下几类:

- (1) 软件代码内容完全相同。
- (2) 软件代码文件中的函数位置或者代码布局发生了变化,但是代码实质内容没有变化。
- (3) 除空格、注释之外的软件代码完全相同。
- (4) 软件代码文件中对函数名、变量名、常量名、类型名做了对应性替换,其他代码完全相同。
- (5) 软件代码中存在增加、修改、删除,或者少量语句顺序改变等情形,但仍然存在较多的相同代码内容。
- (6) 其他不合理的相似,如存在软件署名、开发者的姓名、单位、废程序段、独特的代码序列等相同。
- (7) 软件代码片段的逻辑功能相同,但是具体的编码实现方式有所区别,软件代码内容存在差异^[5-6]。

1. 中浦鉴云(上海)信息技术有限公司司法鉴定所 上海 200086

1.2 传统的软件相似性鉴定方法

传统的软件相似性鉴定方法主要有：

(1) 人工审查方法：人工审查方法完全依靠审核人员的自身理解、专业知识和判断来完成任务，鉴定人员通过检验工具对检材和样本软件的源代码、目标代码、电子文档等进行筛选比对和检验分析，以判断检材和样本的相似性。人工审查方法能够理解复杂结构、处理模糊语义，鉴定结果较为准确；但存在效率低、易疲劳等缺点，同时还容易出现不同鉴定人员的判断标准会存在差异，导致鉴定结果可能存在偏差的问题。

(2) 文本比对方法：文本比对方法的本质是将软件代码转换成文本内容来完成相似度比对，采用逐行的方式比较文本内容差异，比对过程中还会结合哈希算法、Beyond Compare、Diffuse 等比对工具进行分析判断。文本比对方法具有操作简单、易于实现等优点，能够快速发现明显的差异；但也存在一些缺点，如容易受代码编写风格、注释等影响，导致无法准确判断软件代码之间的相似性。

(3) 特征分析方法：特征分析方法是从检材和样本中提取具有代表性的特征，可提取的特征有数据结构、函数、变量等，通过比对提取特征的相似度来判断检材和样本之间的相似性。特征分析方法是能够在一定程度上克服文本比对的局限性，更加关注软件结构和逻辑；但是特征提取的过程较为复杂，需要依赖专门的工具及知识，而且不同的特征提取方法可能会得到不同的结果，对最终的鉴定结果会存在一定的影响。

传统的软件相似性鉴定方法在简单的软件相似性鉴定案件中能够发挥一定的作用，但当面临复杂度更高的软件相似性鉴定案件时，存在的效率低、准确性差等问题将凸显，难以满足实际应用的需求，亟需引入新的技术手段进行效率提升。

1.3 基于人工智能技术的软件相似性鉴定

传统的代码相似度鉴定方法主要基于文本匹配、语法树匹配等技术，方法简单易用，但难以适应大规模的软件及新的编程语言和代码风格，可扩展性较差，难以满足现实需求。随着互联网社会的快速发展，人工智能技术在软件开发与应用领域展现出重要价值，也为软件相似性鉴定提供了新的方法，解决了软件相似性鉴定过程中出现的问题。人工智能技术的应用主要有：

(1) 基于软件源代码的相似性比对：将软件源代码定义为特殊的“语言”，基于词法分析、句法分析、语义分析等自然语言处理技术，提取软件源代码中的关键特征信息，使用聚类算法对代码进行分类，查找相似的代码片段，并结合神经网络模型学习代码的上下文信息，判断代码之间的相似度。

(2) 基于软件功能的相似性比对：模拟用户行为对软件进行操作，获取软件的交互数据，利用强化学习来构建软件的功能行为模型，综合软件的功能行为结果，确定软件功

能相似程度。

(3) 基于软件界面的相似性比对：提取软件功能界面并转换为图像文件，将彩色图像进行灰度处理后，计算检材和样本图像的积分图，基于（normalized cross correlation, NCC）计算功能页面的相似度，并以可视化方式标记差异进行直观展示。

(4) 基于控制流图的相似性比对：通过分析源代码中程序的执行路径，构建程序的逻辑结构，利用图匹配算法计算控制流图的相似性^[7]，筛选出高相似度代码，并通过细粒度比对确认差异，判断软件代码的相似度。

2 人工智能技术应用

与传统的代码特征提取方法相比，基于深度学习的方法能够自动学习到更具代表性和判别力的特征，不需要人工手动设计和选择特征，大大提高了特征提取的效率和准确性，软件相似性检测可以概括为预处理与分析、代码特征转换及特征提取、相似性比对、结果分析与验证等阶段，基本流程概况如图 1 所示。

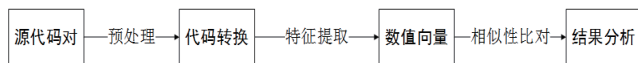


图 1 软件相似性检测流程

人工智能技术可以充分应用在整个检验流程中，对于使用者而言，可以在各阶段实现相关功能辅助软件相似性检测。

2.1 预处理与分析

利用人工智能技术设计功能，实现对检材预处理与分析。主要设计实现功能：

(1) 对检材文件分类并汇总，如整理出文件类型及数量，整体分析检材文件特性。

(2) 提取有效代码数据，如处理代码文件中的注释、空格等与相似性比较不相关的信息，减少相似性度量过程中的噪声干扰。

(3) 第三方开源代码过滤，如将已知第三方开源库的代码创建代码指纹，然后从代码文件中过滤掉第三方开源库的代码，提高代码文件中关键代码的权重。

2.2 代码转换及特征提取

对代码的相似度度量主要涉及三个方面，即代码结构、代码语法，以及代码语义，在度量过程中需要利用人工智能技术将源代码进行特征提取，并将提取的代码特征转化成中间形式表示结构，如代码单元 token、抽象语法树 AST 或程序依赖图 PDG 等结构，随后将表示结构的特征转化为对应的数值特征向量，以便完成相似性度量^[8]。

2.3 相似性比对

使用相似性度量算法计算代码间数值特征向量的相似度，给出相似性度量结果。常用的方法有余弦相似度、Jaccard 相似度、编辑距离、TF-IDF、Words2Vec、Doc2Vec、

基于深度学习的预训练语言模型（如 BERT、GPT 等）等。

2.4 结果分析与验证

对相似性度量结果进行整理，同时利用关联字段将比对结果与原始文件进行关联，将原始文件与度量结果进行可视化展示，方便鉴定人员进行深入分析和评估验证，确保鉴定结果准确和可靠。

3 比对实践

Python 已经成为最受欢迎的程序设计语言之一，提供了一系列优秀的库，被广泛地应用于人工智能领域，使得人工智能和机器学习的开发和实现更加高效、便捷。本文选择 TF-IDF、自然语言向量、Siamese Network 深度学习 3 个模型进行实践^[9-10]。

3.1 基于 TF-IDF 模型检测的实现

TF-IDF 是一种统计方法，用来综合衡量一个单词对于一个文件集或一个语料库中的其中一份文件的重要性，广泛用于自然语言处理和信息检索领域的各种任务，TF-IDF 模型的主要思想是一个词在当前文档中出现次数越多（TF 高）且同时在其他文档中出现的次数越少（IDF 高），则该词对当前文档的区分能力越强，越能够代表当前文章。

实现原理是首先对代码文件进行分词操作，形成分词列表并建立词典，基于词典建立语料库，再用语料库训练 TF-IDF 模型，获取特征数，得出代码文件对应的向量，最后利用余弦定理计算向量的夹角余弦值得到相似度结果。结合鉴定过程该实现概述如下：

- （1）从库 sklearn 中导入 TfidfVectorizer、cosine_similarity 工具。
- （2）加载待比对代码文件。
- （3）利用 TfidfVectorizer 工具创建 TF-IDF 向量器。
- （4）计算 TF-IDF 矩阵。
- （5）利用 cosine_similarity 工具计算余弦相似度，得到相似度结果。

3.2 基于自然语言向量模型处理的实现

通过对代码注释、用户手册等文件进行自然语言处理，使用文本分类算法对功能描述进行分类，提取关键词和关键词句。使用词袋模型或者词向量模型来表示代码注释、用户手册等文件，然后计算文本之间的相似度，同时结合分析函数名和变量名的语义信息，综合评估软件的相似性。常见的词向量模型有 Word2Vec、GloVe 和 Doc2Vec 等。具体实现过程概述如下：

- （1）从库 transformers、gensim 中导入 BertTokenizer、BertModel、Doc2Vec 等工具。
- （2）加载待比对代码文件。
- （3）利用 BertModel 类加载预训练模型，完成模型基本的操作处理。

（4）利用 BertTokenizer 类分析提取代码文件中的关键词。

（5）使用 Doc2Vec 将提取后的内容生成唯一向量表示。

（6）计算从代码文件获取的向量表示之间的相似度，得到相似度度量结果。

3.3 基于 Siamese Network 深度学习技术的实现

Siamese Network 是一种特殊的神经网络结构，由两个或多个相同的子网络组成，这些子网络共享相同的权重和参数，通常被用于解决基于相似度比较的任务，其基本思想是将输入数据同时输入到两个完全相同的神经网络中，通过学习输入数据在这两个网络中的表示，来计算出两个输入样本之间的相似度。其工作原理可以概括如下：

（1）提取输入样本特征：对于两个输入样本，Siamese Network 分别通过共享相同权重和参数的子网络来提取输入样本的特征，得到两个表示向量。

（2）相似度比对：在特征空间中，网络使用一种度量方法（如欧氏距离、曼哈顿距离、余弦相似度等）来计算两个特征表示向量之间的相似度得分。

（3）优化损失函数：通过一个损失函数（如对比损失函数），模型学习到调节网络参数，以便更准确地判断样本间的相似度。

3.4 关键代码实现

采用 Python 实现的测试数据比对结果如图 2 所示，具体关键代码如下：

```
vectorizer = TfidfVectorizer(tokenizer=None, stop_words = None)

tfidf_matrix = vectorizer.fit_transform(documents)

similarity_matrix = cosine_similarity(tfidf_matrix, tfidf_matrix)

similarity_df = pd.DataFrame(
    similarity_matrix,
    index=[f"代码 {i+1}" for i in range(len(documents))],
    columns=[f"代码 {i+1}" for i in range(len(documents))])

print("相似度矩阵:")

print(similarity_df.round(4))

print("\n代码相似结果:")

for i in range(len(documents)):
    similarities = similarity_matrix[i].copy()
    similarities[i] = -1 # 设置自身相似度为 -1
    most_similar_idx = similarities.argmax()
    max_similarity = similarities[most_similar_idx]
    print(f"代码 {i+1} 最相似的是代码 {most_similar_idx+1}, 相似度为 {max_similarity:.4f}")
```

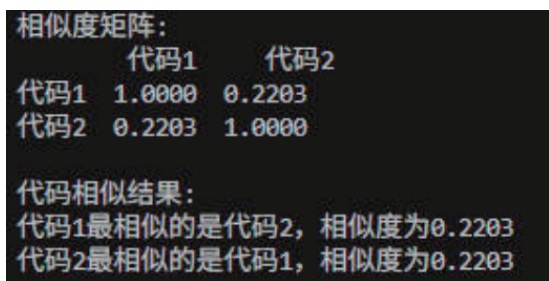


图 2 软件相似性比对结果

4 面临挑战与应对策略

新技术时代, 软件类产品如: 操作系统软件、办公软件、开发软件、行业应用软件、游戏软件等产生了井喷爆发式的增长, 软件产品的复杂性和多样性也不断增加, 相应的关于软件侵权类纠纷也日益增多。现代软件开发涉及多种编程语言、开发框架和开发工具, 并且越来越依赖开源软件和第三方库, 这使得软件代码复用变得非常普遍, 增加了侵权行为的隐蔽性, 给软件相似性司法鉴定增加了一定的难度。软件的快速迭代和多平台部署也对相似性鉴定提出了更高的技术要求, 需要更精准、更高效的方法来识别软件之间的相似性。人工智能技术通过深度数据挖掘与模式识别能力提升了鉴定结果的科学性和可靠性, 有效地完成司法鉴定, 同时也面临诸多挑战。

4.1 数据安全性与隐私

在相似性鉴定过程中, 需要收集大量的软件代码和相关数据, 这些数据可能包含企业的核心商业秘密、用户隐私信息等。人工智能技术的应用通常需要将数据上传至云端进行处理, 这就增加了数据泄露的风险。如果数据在传输、存储或处理过程中被窃取或篡改, 将会给企业和用户带来严重的损失。

4.2 数据质量

人工智能技术的性能依赖于大量高质量的数据。在软件相似性鉴定中, 获取全面、准确的软件样本数据存在一定困难, 而且数据中可能存在噪声、错误标注等问题, 影响模型的训练和鉴定结果的准确性。

4.3 算法可解释性

深度学习等人工智能算法的内部工作机制复杂, 通常被视为“黑盒”模型, 其决策过程和结果难以解释^[11]。

4.4 应对策略

面对人工智能技术的发展带来的数据安全性与隐私、数据质量、算法可解释性等诸多挑战和干扰问题, 这就需要我们打破传统司法鉴定手段与新方法的壁垒, 建立完善的数据收集和预处理机制, 采用加密技术对传输和存储的数据进行加密处理, 使用规范的数据收集和标注流程提升数据标注的准确性, 研究和应用可解释的人工智能算法, 采用可解释技术、人机协同验证等方式进行技术补充完善, 明确人工智能技术在司法鉴定中的角色和定位, 推动人工智能技术产品化、产品市场化。

5 总结与展望

在软件产品著作权侵权、专利侵权、侵犯商业秘密类等诉讼中, 判断侵权产品与权利产品是否构成实质性相似是认定侵权行为的关键环节。软件相似性鉴定通过比源代码、目标代码、软件结构、功能模块等方面内容, 可以对两款软件的相似程度进行数值量化, 为司法机关认定侵权提供重要的依据^[12]。

将人工智能技术应用到软件相似性鉴定过程中, 通常会涉及数据预处理、特征提取、模型训练与预测、结果分析与验证等多个阶段。在软件相似性鉴定过程中充分利用人工智能技术, 能够明显提高鉴定效率, 结合源代码、软件功能、程序运行机制等多角度的比对分析方法, 实现了更精准的软件相似性鉴定, 为司法鉴定提供了更加科学、更加可靠的技术支撑。注入的新动能能够推动司法鉴定形成高效、自动化、智能辅助的应用新局面。

参考文献:

- [1] 中华人民共和国司法部. 软件相似性鉴定技术规范: SF/T 0158-2023 [S]. 北京: 中华人民共和国司法部, 2023.
- [2] 公安部信息安全标准化技术委员会. 软件相似性检验技术方法: GA/T 1175-2014[S]. 北京: 中国标准出版社, 2014.
- [3] 张颖. 人工智能浪潮下司法鉴定的机遇与挑战 [J]. 中国司法鉴定, 2024(5):11-17.
- [4] 牟萍. 软件著作权鉴定疑难问题与保护技巧探讨 [J]. 中国版权, 2023(4):7-12.
- [5] 陈思思. 基于深度学习的代码相似性检测算法研究 [D]. 武汉: 华中科技大学, 2021.
- [6] 展佳俊, 赵逢禹, 艾均. 基于多特征值的源代码相似性检测技术 [J]. 计算机技术与发展, 2021,31(1):103-109.
- [7] 梁瑶, 谢春丽, 王文捷. 基于图嵌入的代码相似性度量 [J]. 计算机科学, 2022,49(S2):801-806.
- [8] 李玫, 高庆, 马森, 等. 面向代码相似性检测的相似哈希改进方法 [J]. 软件学报, 2021,32(7):2242-2259.
- [9] 张丹, 罗平. 代码相似性检测方法 with 工具综述 [J]. 计算机科学, 2020,47(3):5-10.
- [10] 李奕霖, 周艳平. 基于孪生网络和字词向量结合的文本相似度匹配 [J]. 计算机系统应用, 2022,31(10):295-302.
- [11] 邢颖. 基于可解释性人工智能的软件工程技术方法综述 [J]. 计算机科学, 2023,50(5):3-11.
- [12] 杨晨. 电子数据司法鉴定中的软件鉴定方法研究 [J]. 楚天法治, 2023(24):191-193.

【作者简介】

余铭华 (1985—), 男, 江西黎川人, 本科, 高级工程师, 研究方向: 电子数据司法鉴定工作及相关研究。

(收稿日期: 2026-02-26 修回日期: 2026-04-14)