用于日志文件中模式定量可视化的基于字典的压缩算法扩展

I.Cherepanov[1],J.G. Joewono[1] ,A. Kuijper[1,2],J. Kohlhammer[1,2]

[1]Fraunhofer IGD，德国[2]达姆施塔特工业大学，
德国

arXiv:2304.04631v1 [cs.IR] 10 Apr 2023

摘要

如今，许多服务会大量且持续地产生格式各异的日志文件。这些日志非常重要，因为其中包含有关应用程序活动的信息，而通过分析行为并维护系统的安全性和稳定性，这些信息对于改进工作而言不可或缺。为了减小日志文件的庞大体积，以压缩形式存储日志文件是一种常见做法。压缩算法会识别日志文件中的频繁模式，以删除冗余信息。本研究提出了一种检测文本数据中频繁模式的方法，这些模式可以在文件压缩过程中以较低的资源消耗被同时记录下来。通过基于所获取模式的频率、长度和根前缀等属性的指标，可以对日志文件进行可视化，并探索其中提取出的模式。这使分析人员能够更高效地获得相关洞见，从而减少对日志数据进行人工、耗时检查的需求。所实现的基于字典的压缩算法的扩展具有识别任意格式日志文件中模式的优势，并消除了手动执行日志文件准备和预处理的需求。

1 引言

日志文件是计算机生成的文本文件，由运行中的应用程序在特定事件发生时生成，并包含有关系统相应状态的必要信息。日志记录是开发和监控运行中系统的重要组成部分。日志记录涉及多种任务，例如错误分析、检测系统弱点、意外事件和异常 [1, 2,3]。此外，日志还提供了分析系统中实际内容分布的可能性。了解数据所代表的含义可能会有多种用途。分析人员可以利用数据分布知识，判断数据是否偏离标准行为，或是否存在可识别的趋势 [4, 5]。由于服务会生成海量日志记录，人工分析并不可行。因此，需要采用自动化方法。许多已提出的方法都依赖针对特定日志类型的预处理步骤，涵盖从数据准备到解析，以从日志中提取内容 [6]。此外，提取的数据可以结构化呈现，也可用于分类等其他用途 [3]。

由于日志量会随着时间增加，应用压缩方法是一种常见做法。数据压缩有多种不同的方法 [7]。对于日志，有必要应用一无损压缩，从而使解码能够可靠地恢复原始日志文件。其中一种是所谓的基于字典的压缩，它通过码字 [8]替换变长子串。

在本文中，我们将一种可视化方法与一种无损字典式压缩技术的扩展相结合，该扩展除了压缩日志文件外，还能够记录模式的统计信息。由此生成的包含所获取模式的字典，可以根据我们创建的指标对模式进行筛选。这些指标基于模式的属性，例如频率、根前缀和长度。该可视化方法展示了如何借助色标对原始日志文件数据进行可视化映射，以表示所选指标的定量属性。这样，分析人员便能更好地检查数据，因为日志中的频繁模式和不频繁模式会被直接突出显示。此外，我们的方法无需进行数据预处理，即可应用于任何类型的文本数据格式。鉴于现有系统中应用的多样性，这一点非常有用。

2 相关工作

与我们的方法相关的工作分为三类。首先，我们讨论处理日志文件并从中提取知识的方法。然后，我们介绍相关的压缩方法，最后讨论日志文件中模式的可视化技术。

对于复杂系统而言，生成日志文件 [9] 至关重要。这些日志文件用于系统运行期间的调试。日志文件记录系统在运行时发生的事件、内部状态和数据内容。分析人员可以通过检查日志更好地了解系统状态，并在发生错误时诊断系统、寻找改进系统的着力点。因此，从日志中进行知识提取对决策过程 [10] 尤为重要。许多方法基于日志解析方法，例如使用正则表达式来提取信息 [11]。这些方法需要投入相当大的精力来手动创建和维护表达式规则。使用非结构化日志的应用种类不断动态增长，使这一过程面临挑战。因此，许多工作致力于通过使用数据挖掘技术进行自动化日志解析，以实现日志分析 [12, 13]。这些方法的目标是将非结构化日志数据转换为结构化数据。这些方法从日志数据中学习模式，并自动创建通用事件模板。此类工具包括 Drain [14], 、LogMine [15] 和 MoLFI [16]。在本研究中，我们同样致力于避免手动对日志文件进行任何预处理准备。此外，我们还关注对一种现有的基于字典的压缩技术进行扩展，并将其应用于日志文件。

压缩方法领域广泛且多样 [7]。有损方法通过减少原始文件中的信息来实现压缩，适用于图像和音频文件。相比之下，文本文件即使仅删除其中最小的一段，也可能变得毫无价值。因此，这类数据只能使用无损压缩方法进行压缩 [17]。文本数据的一种压缩方法基于所谓的基于字典的方法。该方法将符号字符串替换为编码，并使用字典对每个字符串进行编码。字典存储符号字符串及其对应的编码。本文重点介绍一种广受欢迎的无损基于字典的方法，称为 LZW（Lempel-Ziv 和 Welch）压缩 [8]。LZW 压缩以迭代方式进行：每个阶段都会将输入字符聚合成一个序列，直到下一个字符使该序列成为字典中尚无对应编码的序列。随后，将不含该字符的序列编码添加到输出中，并将聚合后形成的新序列编码添加到字典中。此方法可以轻松扩展，添加另一个字典来统计所有发现的模式和子模式，我们采用这种方法进行定量可视化。

我们的目标是满足用户发现频繁和较少频繁模式的任务 [18]。我们通过在日志文件中使用不同色图进行直观可视化来实现这一目标 [19, 第 10] 章。用不同颜色突出显示结构中的元素，可以让模式的发现成为前注意过程，从而显著提升搜索效率 [20, 21]。借鉴这一思路，本文通过对文本片段进行颜色高亮，向用户强调相关信息。同时，我们利用压缩功能的优势，无需进行数据预处理。名为 gzthermal2 的程序也采用了类似思路 [22]。它旨在可视化 GZIP 压缩。该程序使用 GZIP 算法压缩数据，然后生成图像表示，展示数据的压缩方式。不同的颜色高亮展示了在 GZIP 文件格式中用于存储数据段的字节数差异。与此相反，我们呈现的是压缩文件中发现的不同模式的数量。

3 方法

在本节中，我们重点介绍所提出的LZW算法扩展，并更详细地描述其编码部分。随后，我们介绍基于频率、长度和根前缀创建的指标，这些指标允许用户对字典中的数据进行排序，并据此对日志文件进行定量着色。最后，我们描述如何在原始日志文件上可视化定量属性。我们将该方法开发为一个基于Web的原型，并据此以示例日志文件片段的形式生成和呈现结果。

3.1 LZW算法的扩展

```python
def compress(uncompressed: str):
    dictionary = {}
    count = {}
    # Build the dictionary.
    dict_size = 256
    for i in range(dict_size):
        dw = chr(i)
        dictionary[dw] = i
        count[dw] = 0
    w = ''
    result = []
    for c in uncompressed:
        wc = w + c
        if wc in dictionary:
            w = wc
        else:
            result.append(dictionary[w])
            for i in range(len(w)):
                count[w[:len(w) - i]] += 1
            dictionary[wc] = dict_size
            count[wc] = 1
            dict_size += 1
            w = c
    if w:
        result.append(dictionary[w])
    return result, dictionary, count
```

清单 1： 扩展版 LZW 编码部分的 Python 代码版本

编码的前几行初始化了两个字典。其中一个存储所有模式及其对应的在线a

编码（第 2 行），另一个则具有记录模式出现次数的寄存器功能（第 3 行）。LZW 方法首先将字典初始化为字母表中所有可能的符号（第 6–8 行）。我们将其转换为 ASCII，否则就必须使用所有可能的 Unicode 符号初始化字典，这会增加开销。由于日志条目通常使用拉丁字母表书写，ASCII 字符的范围足以构成初始字典。此外，还会更新一个字典，用于记录所有实际出现和新出现的模式及子模式的频率，并使用相同的初始模式（根前缀）（第 9 行）。编码方法在每次迭代中将一个符号累加到初始字符串 w（第 13 行），并检查所得的 wc 是否存在于字典中（第 12 行）。如果wc在字典中，则将其赋值给 w（第 15 行）。经过一定步骤后，添加下一个符号c会导致在字典中搜索 wc 失败（第 16 行）。此时，指向字符串 w 的字典指针会被存储到编码后的结果数组中（第 17 行），而字符串 wc 会作为新的字典条目，并使用下一个可用编码存储（第 20 行）。随后，编码加一，以用于下一个未知字符串（第 22 行）；由于该模式是首次发现的，因此会在负责统计模式的字典中将其首次出现次数赋值为 1（第 21 行）。在同一步骤中，我们遍历字符串w 的所有子模式，并增加其对应的出现次数（第 18–19 行）。最后，将字符串 w 初始化为符号c（第 23 行）。输入流结束后，我们得到压缩字符串、包含模式及其后续解码所需编码的字典，以及记录模式出现次数的字典（第 26 行）。

字典不受前缀符号数量的限制，算法会从文件开头开始存储所有字典条目。实践中，字典通常会在变得过大后重置，以便释放系统使用的内存，并减少将输入流中的下一个字节与字典进行匹配时的开销[8]。然而，这项工作的目标是尽可能多地找出模式，因此在处理过程中不应重置字典。其理念是，整个日志文件应完整归档并检查其中的模式，不作任何拆分。

编码后所得输入的解码方式与原始LZW算法 [8] 相同。解码器遍历压缩数据，并使用代码字典将编码还原为原始文本文件。为了后续可视化，我们还将附加定义的定量属性添加到每个模式中，这些属性将在第 3.2 节中进行描述。这些属性决定了所呈现日志可视化中的模式，如图3所示。用户可以设置前缀长度，该长度决定根前缀的长度并设置其频率。此属性可以变化，并会在用户更改它时进行调整。具体做法是搜索具有所选长度的根前缀。通过这种方式，可以获得该根前缀的频率，并将其作为模式属性分配，以便后续进行可视化。

生成的修改增加了一些开销，因为需要遍历每个缓冲区的长度（变量 w）。不过，每当有新模式存入字典时，缓冲区 w 就会重置，使缓冲区长度恢复为 1（第 23 行）。由于缓冲区长度趋近于 1，因此可以认为整个算法的运行时间复杂度为 $O(n)$。测试并测量算法的平均运行时间也进一步印证了这一点。测试针对最坏情况，即输入由重复的同一字符组成，共计 N 次，以使缓冲区重置次数最少，并尽可能多地对各个子模式进行计数。结果如表 1 所示。平均时间是通过在 5 秒间隔内运行算法 20 次计算得出的。作为更直观的比较，长度为 191,726 字节的常规文本样本数据平均需要 115.11 毫秒进行处理（每个 N耗时 0.600 纳秒）。进一步分析统计了计数器被访问和递增的次数，并确认计数器恰好被访问了 N 次。总运行时间约为 $T(2N)$。

表1： 比较处理由 N 字节的相同重复字符组成的输入流的平均时间。

N (字节)	平均时间（ms）	每 N所需时间（ns）
10.000	4.61	0.461
20.000	10.17	0.509
30.000	14.80	0.493
40.000	20.83	0.521
50.000	26.63	0.533
80.000	48.55	0.607
160.000	104.04	0.650
320.000	227.40	0.711
640.000	524.06	0.819

3.2 指标

通过统计所有模式，可以将其以表格形式展示，并可根据图2所示的以下指标筛选模式：

- **模式** 及其 **频率** 表示所有模式

Pattern	Length ↑	Frequency	Frequency * Length	Prefix Class	Prefix Count
19|2519	7	3	21	19|25	5
19||| 1	7	2	14	19|||	4
1:9498_	7	28	196	1:949	49
1:9498|	7	19	133	1:949	49
1|206.1	7	146	1022	1|206	148
1|45271	7	4	28	1|452	6

图 2: 字典中所获取模式的表格概览。除模式外，还显示了其他指标，例如长度、频率、频率乘以长度、前缀以及前缀的频率。

及其频率。因此，可以按频率或字母顺序排序。

- 模式**长度** 可用作排序选项。
- **频率∗长度**。频率乘以其长度可以突显出高频且较长的模式。尽管许多其他模式中也存在重复的元信息，这一指标仍然很有用，因为它能够突出更具意义的模式。
- **前缀类别**及其**计数**。前缀类别是一个属性，用于确定根前缀的长度，所有更长的模式都由该根前缀派生。用户可以限制前缀的长度，以显示有多少个该长度的模式共享相同前缀。在图 2 中，前缀长度为 5。根前缀的频率至少等于对应模式的频率；如果有许多模式以相同根前缀开头，则根前缀的频率会更高。所有短于设定长度的模式都将被视为稀有模式，并在着色时赋值为 0。

3.3 频繁模式的可视化

用户的任务是在日志文件中发现频繁和较少出现的模式。频率是根据多种提供的指标（第 3.2 节）计算得出的定量值。这些值通过各种色图表示。色图有多种选择，各有优缺点，因此很难选出合适的一种 [23]。因此，向用户提供多种色图是合适的。发散色图可以让我们看出数值是否接近极值，以及它们接近哪一端的极值，如图 3a 所示；这种色图适合区分非常频繁和较少出现的模式。这样，在呈现整个日志文件时，用户可以通过颜色对比轻松定位频繁模式。在缩小范围搜索日志文件时，定量值之间的差异可能不大，因此需要突出这些差异。这时，类似彩虹的色图可能更合适，因为与发散色彩方案不同，它能以不同颜色表示相近的定量值，如图 3b、3c 和 3d 所示。

收集到的模式以表格形式列在我们的系统中，包含以下属性：模式本身、模式的长度、频率、其频率与长度相乘所得的值，以及另外两列，分别显示与该模式共享前缀的不同模式数量和前缀计数本身，如图 2 所示。用户可以根据每个属性对表格进行排序。模式可以按字母顺序排序，其余内容则按数值排序。分析人员需要检查的相关日志可以解码为原始文件，并根据所选指标（第3.2 节）以彩色形式显示在包含所获取模式及其频率的表格视图旁边。这样可以更高效地识别相关发现，无需手动检查全部日志数据。图2 展示了同一日志文件的片段，其中选择了不同指标进行着色。表格与原始日志文件的彩色可视化结果可以相互补充。由于表格可以排序，并能直接指出频繁模式，而彩色可视视图则强调这些模式所在的位置，因此分析人员可以检查其旁边的信息。

4 使用场景

为了验证所提出的方法，我们针对一个真实场景开展了案例研究。由于保存日志文件对于改进系统至关重要，许多服务提供商会按时间顺序以压缩形式保存日志。例如，我们熟悉一家名为欧洲 IP 网络协调中心（RIPE NCC）的服务提供商。RIPE NCC 的职能是作为区域互联网资源注册管理机构，提供全球互联网资源及相关服务。它为互联网社区提供服务，其中包括公开访问所收集的边界网关协议（BGP）更新日志档案。这些更新日志按时间顺序以归档形式存储，采用 gzip 这一不同的方法进行压缩，并且日志以 MRT 格式写入。基于这些日志，有许多可视化方法能够帮助领域专家进行故障排查 [24]。这些方法使用所提供的解析器从日志中提取信息。通过与 RIPE 的一位专家交流，我们验证了这一想法：将压缩档案与有助于理解模式的表示形式结合起来提供给用户。

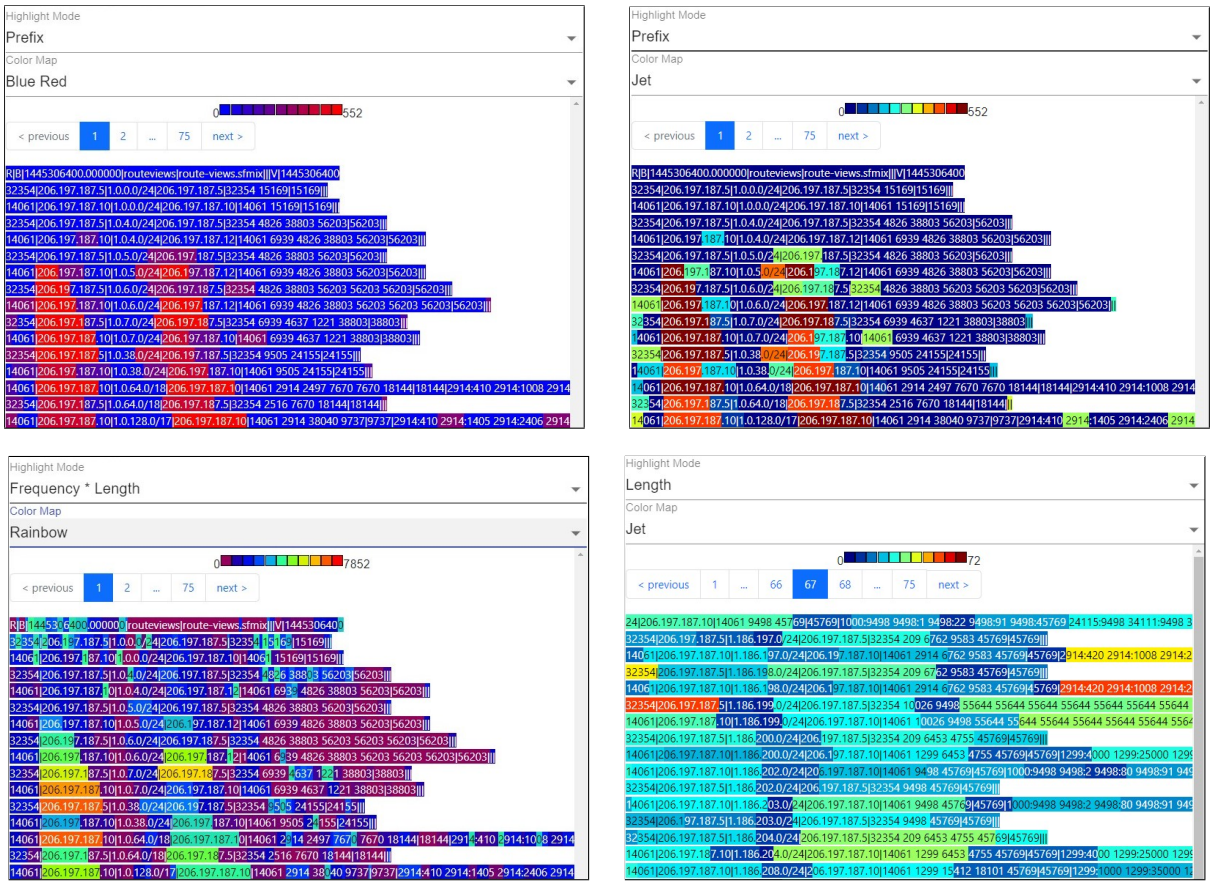


图 3： 日志文件着色片段的表示。图（a）表示基于长度为 5 的根前缀频率，并使用顺序色图进行的着色。相比之下，图（b）使用 jet 色图表示相同片段，在此特定示例中，这使相似的定量属性更加明显。（c）展示了基于频率*长度指标进行着色的相同片段。（d）表示同一日志文件中的另一片段，其着色使用的是基于模式长度的指标。

当日志格式因新增应用程序而随时间变化时，我们的方法同样具有优势。我们的方法采用一种可应用于任何文本数据类型的压缩方法，无需进行数据预处理。因此，它适合作为分析人员使用的通用接口。对于包含已计数模式和子模式的字典，表格概览非常有帮助（图 2）。表格概览不仅允许分析人员按字母顺序对模式进行排序，还可以按其频率、长度或其他标准进行排序（第 3.2 节）。这样，分析人员便能看到哪些模式较为常见，哪些不太常见。这使分析人员能够更快地了解日志文件的内容，例如查看哪些 IP 地址经常出现在日志中。此外，还可以跟踪所提取字典的数据分布，从而直接检测日志事件中的任何偏差。第二部分是分析人员要检查的完整日志。借助我们的方法，模式会被突出显示。这使分析人员能够更高效地发现日志文件中的相关信息，减少手动、耗时检查的需要

日志数据。附加指标允许用户以不同方式衡量模式的属性。可以使用不同的色图来优化模式定量属性之间的可感知差异。

5 讨论与未来工作

我们的方法避免了对日志数据进行预处理或解析，并且可以直接应用于任何文本数据，不受语法格式的限制。然而，由于日志文件中的元信息频繁出现，大量元信息很可能会被检测为高频模式。这些冗余元信息中有很大一部分无法反映日志数据的内容，因此并不具有信息价值。为了获得更有用的模式，需要进行预过滤以移除冗余信息。然而，预过滤过程与每种日志文件格式都密切相关，并且需要具备相应领域的知识。我们只需移除日志中位于每行开头的所有重复元信息，就能获得明显更简洁的模式。这包括

元信息，例如许多日志共享的时间戳。不过，仍然可以浏览排名较低的模式。这就是引入一种新指标的原因：将长度与频率相乘，从而尽管存在元信息本身，也能提升更有意义的模式。因此，我们在 BGP 日志 [25] 中获得了具有语义意义的模式，例如 IP 网段和 AS 路由。此外，该方法还应用于多个不同的文本语料库，以测试其识别文学作品中语义模式的性能。本实验使用的文本语料库来自 Project Gutenberg 提供的书籍 [26]。结果表明，用某种语言写成的书籍具有与最常见停用词一致的相同频繁模式 [27]。

本文描述了我们所采用方法的第一个版本，我们将使用该方法与领域专家进行讨论，旨在相较于他们当前的流程实现明确的附加价值。未来，我们有兴趣探索其他压缩方法，以了解这些方法是否也能够扩展出模式提取和计数的类似功能。特别是，那些专门针对日志文件进行调整的方法同样十分有趣，同时还可以比较它们在模式提取方面的性能 [28, 29]。我们将把我们的方法集成并扩展到某一领域的现有工具中，并结合用于领域数据的其他可视化技术；同时，我们也有兴趣与领域专家开展评估，以便通过改进进一步聚焦于他们的需求。此外，由于日志文件规模庞大，我们希望提供一种带有可缩放概览的细节层次可视化，以便在文本细节之间切换。还可以对这些模式进行自动分析，根据数据的异质性选择合适的色彩映射。此外，所得模式还可以通过分类等进一步方法进行处理，类似于词袋等方法 [30]。用户还可以进一步参与其中，自行以交互方式创建类别。

6 结论

本文介绍了一种可视化方法，并将其与基于字典的压缩算法扩展相结合。该方法能够在压缩过程中统计模式，其优点是无需对数据进行预处理。无论文本文件采用何种语法格式，都可以直接应用此方法。我们基于模式的频率，为模式的定量属性设计了一种可视化方案，并通过示例日志文件展示了结果。此外，我们还展示了如何以表格形式向用户呈现字典，并将其置于彩色日志文件旁边。用户可以据此根据特定属性对模式进行排序。该方法使用户能够快速直观地突出显示已识别出的频繁模式，并展示其各项定量属性。

致谢

本研究工作由德国联邦教育与研究部和黑森州高等教育、研究与艺术部共同资助，作为对国家应用网络安全研究中心 ATHENE 的联合支持。

参考文献

[1] Lei Zeng 等。“计算机操作系统日志记录与安全问题：综述”。载于：安全与通信网络 9.17（2016），第 4804–4821 页。[2] Shilin He 等。“面向可靠性工程的自动化日志分析综述”。载于：*ACM* 计算综述（*CSUR*） 54.6（2021），第 1–37 页。[3] Hudan Studiawan、Ferdous Sohel 和 Christian Payne。“操作系统日志取证调查综述”。载于：数字调查 29（2019），第 1–20 页。ISSN：1742-2876。DOI：https://doi.org/10.1016/j.diin.2019.02.005。网址：
h
t
t
p
s://
w
ww.sciencedirect.com/science/article/pii/S1742287618303980。[4] Ahmed Oussous 等。“大数据技术：综述”。载于：《沙特国王大学学报：计算机与信息科学》 30.4（2018），第 431–448 页。ISSN：1319-1578。DOI：
https://doi.org/10.1016/j.jksuci.2017.06.001。网址：
h
t
t
p
s://
w
ww.sciencedirect.com/science/article/pii/S1319157817300034。[5] Shikha Agrawal 和 Jitendra Agrawal。“使用数据挖掘技术进行异常检测的综述”。载于：*Procedia* 计算机科学 60（2015）。知识型与智能信息及工程系统第 19 届年度会议，KES-2015，新加坡，2015 年 9 月会议论文集，第 708–713 页。ISSN：1877-0509。DOI：https://doi.org/10.1016/j.procs.2015.08.220。网址：
h
t
t
p
s://
w
ww.sciencedirect.com/science/article/pii/S1877050915023479。[6] Pinjia He 等。“日志解析及其在日志挖掘中的应用评估研究”。载于：*2016* 年第 *46* 届 *IEEE/IFIP* 国际可靠系统与网络会议（*DSN*）。2016 年，第 654–661 页。DOI：10.1109/DSN.2016.66。

[7]Uthayakumar Jayasankar、Vengattaraman Thirumal 和 Dhavachelvan Ponnurangam。“数据压缩技术综述：从数据质量、编码方案、数据类型和应用的角度出发”。载于：《沙特国王大学学报：计算机与信息科学》 33.2（2021），第 119–140 页。ISSN：1319-1578。DOI：https://doi.org/10.1016/j.jksuci.2018.05.006。

[8] Welch.《高性能数据压缩技术》。载于：《计算机》17.6（1984），第8–19页。doi：10.1109/MC.1984.1659158。[9] Reza Sherafat Kazemzadeh 和 Hans-Arno Jacobsen。《可靠且高可用的分布式发布/订阅服务》。载于：*2009*年*IEEE*可靠分布式系统国际研讨会。IEEE，2009年，第41–50页。[10] Youcef Djenouri、Asma Belhadi 和 Philippe Fournier-Viger。《从事件日志中提取有用知识：一种频繁项集挖掘方法》。载于：《知识型系统》139（2018），第132–148页。[11] Wei Xu 等。《通过挖掘控制台日志检测大规模系统问题》。载于：*ACM SIGOPS*第*22*届操作系统原理研讨会论文集。2009年，第117–132页。[12] Jie ming Zhu 等。《自动化日志解析的工具与基准测试》。载于：*2019*年*IEEE/ACM*第*41*届软件工程国际会议：软件工程实践（*ICSE-SEIP*）。IEEE，2019年，第121–130页。[13] Sara Alspaugh 等。《分析日志分析：用户日志挖掘的实证研究》。载于：第*28*届大型系统安装管理会议（*LISA14*）。2014年，第62–77页。[14] Pinjia He 等。《Drain：一种采用固定深度树的在线日志解析方法》。载于：*2017*年*IEEE*国际*Web*服务会议（*ICWS*）。IEEE，2017年，第33–40页。[15] Hossein Hamooni 等。《LogMine：用于日志分析的快速模式识别》。载于：第*25*届*ACM*信息与知识管理国际会议论文集。2016年，第1573–1582页。[16] Salma Messaoudi 等。《一种基于搜索的日志消息格式精确识别方法》。载于：第*26*届程序理解会议论文集。2018年，第167–177页。[17] Khalid Sayood。《数据压缩导论（第四版）》。第4版。美国加利福尼亚州旧金山：Morgan Kaufmann出版社，2012年。ISBN：0124157963。

[18] Matthew Brehmer 和 Tamara Munzner。《抽象可视化任务的多层分类》。载于：*IEEE* 可视化与计算机图形学汇刊 19.12（2013），第 2376–2385 页。doi：10.1109/TVCG.2013.124。[19] T. Munzner。可视化分析与设计。AK Peters 可视化丛书。CRC Press，2015。isbn：9781498759717。url：https：/ /books . google.de/books?id=NfkYCwAAQBAJ。[20] Jen-Her Wu 和 Yufei Yuan。《改善搜索与阅读表现：高亮显示和文本颜色编码的影响》。载于：信息与管理 40.7（2003），第 617–637 页。issn：0378-7206。doi：https://doi.org/10.1016/ S0378-7206(02)00091-5。url：https://www.sciencedirect.com/science/article/pii/S0378720602000915。[21] Anne Treisman。《视觉处理中的特征与物体》。载于：《科学美国人》255.5（1986），114B–125。[22] Anton Khlynovskiy Simon Willison。$gzthermal$。gzthermal 的 Google 首页：https : / / gzthermal.vercel.app/?url=https%3A%2F%2Fwww.google.com%2F。2017。url：https://github.com/simonw/gzthermal-web。[23] Kenneth D Moreland。科学可视化中的发散色图（扩展版）。技术报告。Sandia 国家实验室（SNL-NM），新墨西哥州阿尔伯克基（美国），2009。[24] Justin Raynor 等。《BGP 可视化工具的最新进展：将可视化技术映射至网络攻击类型》。载于：*IEEE* 可视化与计算机图形学汇刊 29.1（2023），第 1059–1069 页。doi：10.1109/TVCG.2022.3209412。[25] RIPE。*RIPE* 网络协调中心。https://www.ripe.net/。[26] Martin Gerlach 和 Francesc Font-Clos。《用于自然语言与定量语言学统计分析的标准化古腾堡计划语料库》。载于：熵 22.1（2020），第 126 页。[27] W John Wilbur 和 Karl Sirotkin。《停用词的自动识别》。载于：信息科学杂志 18.1（1992），第 45–55 页。[28] Bo Feng、Chentao Wu 和 Jie Li。《MLC：一种用于云备份系统的高效多层日志压缩方法》。载于：*2016 IEEE Trustcom/BigDataSE/ISPA*。IEEE。2016，第 1358–1365 页。

[29] Hao Lin 等。《Cowic：一种用于日志流分析的按列独立压缩方法》。载于：*2015* 年第 *15* 届 *IEEE/ACM* 集群、云与网格计算国际研讨会。IEEE。2015，第 21–30 页。[30] Dongyang Yan 等。《用于文本分类的基于网络的词袋模型》。载于：*IEEE Access* 8（2020），第 82641–82652 页。doi: 10.1109/ ACCESS.2020.2991074。