

Hairpin 算法复现

摘要

随着 5G 网络的普及和用户体验要求的提高，交互式流媒体技术应运而生，要求数据能够低延迟双向传输以实现实时交互。然而，网络中的数据丢包、网络抖动和拥塞等问题，尤其是带宽瓶颈，仍然对交互式流媒体的性能构成挑战。传统的前向纠错算法虽然能够提高丢包恢复率，但往往需要额外的带宽资源，且在动态网络环境中效果不佳。为了突破基于前向纠错的算法在带宽损耗与截止日期错过率之间的瓶颈，本文提出了 Hairpin 算法，该算法通过联合优化数据包的重传与冗余机制，超越了传统方法的性能限制，并通过 Markov 决策过程模型优化了丢包恢复过程。然而，由于固定的 FEC 参数无法适应动态网络环境，本文进一步结合了 Markov 模型与贝叶斯理论，提出了一种新的方法来动态更新转移概率，从而提升算法在复杂网络环境中的适应能力。我们在 NS3 仿真平台上对该方法进行了验证，并与现有的 Hairpin 算法进行了对比。实验结果表明，尽管降低了丢包率，但我们的改进方法相较于原 Hairpin 算法的提升并不显著。

关键词：Markov 决策；截止日期错过率；带宽损耗率；贝叶斯

1 引言

流媒体技术自诞生以来，已广泛应用于视频点播、音乐播放、在线教育等多个领域。传统流媒体主要包括两类：一类是以 Netflix、YouTube 为代表的点播流媒体，这种形式下用户可以按需观看视频内容，但对实时性要求不高；另一类是实时流媒体，应用于直播平台、在线会议和远程医疗等场景，它要求视频数据能够以较低延迟传递到用户端。尽管传统流媒体已能够满足大多数用户的需求，但随着 5G 网络的普及和用户体验要求的提高，人们对流媒体的实时性、交互性和适应性提出了更高的要求，这促使了交互式流媒体的发展。

交互式流媒体是一种用户和内容双向互动的新型流媒体模式，与传统模式不同，它要求数据能够低延迟双向传输以实现实时交互。其典型应用包括多人在线游戏、AR/VR 体验、交互式直播和远程协作等场景。由于交互式流媒体需要实现更高的实时性和可靠性，它不仅对传输速度有严格要求，还需解决数据丢包、网络抖动和拥塞等问题。并且随着终端设备和网络环境的多样化，开始要求交互式流媒体需要具备更强的适应能力，以便其能够在不同条件下都能提供一致的用户体验。

为了解决交互式流媒体的高延迟和带宽瓶颈问题，研究者们提出了各种各样基于边缘计算的交互式视频流媒体架构，例如：基于网络的媒体处理 [1]。通过将计算和存储资源从云端下沉到靠近用户的边缘设备，边缘式流媒体能够有效减少传输延迟并提升服务的稳定性 [2]。边缘计算的分布式特性使得系统可以动态适配不同用户的需求，实现更高效的资源调度。典

型的边缘式流媒体系统中，数据处理和传输的任务被分配到边缘节点 [3]，这种设计能够降低主干网络的压力，并优化用户体验。

尽管边缘式交互式流媒体在性能上已有显著提升，但仍然存在若干问题亟待解决。网络传输中的数据丢包仍是影响用户体验的主要因素，传统的前向纠错算法（FEC）在动态网络环境中的表现不够理想，并且在实际应用中，FEC 算法需要在传输数据前添加冗余信息以提高丢包恢复率，这不可避免地会占用额外的带宽资源。在带宽资源有限的情况下，冗余信息的增加可能会进一步加剧网络拥堵，导致系统性能的下降。针对这类问题，许多研究者都提出了各种改进。例如，毛锦鑫等人提出了基于 GA-BP 的实时视频通信自适应前向纠错码 [4]，用于以更低的冗余率更高效地恢复丢失数据包；Franco 等人提出了一种基于经典神经网络和卷积神经网络的迭代码译码方法 [5]，用于提高编解码的性能；程胜等人提出了自适应比特率的 FEC 算法 [6]，通过对未来一段时间的网络丢失模式进行预测，结合实时视频流的模型共同计算最佳比特率 FEC 决策。然而，这些改进仅能在有限程度上提升 FEC 算法的性能。

为突破基于 FEC 改进算法在带宽损耗与截止日期错过率之间的性能瓶颈，孟子立等人提出了 Hairpin 算法 [7]。该算法通过联合优化数据包的重传和冗余机制，超越了传统方法的性能限制。初始的传输包数量较多，但对时间要求相对宽松，而重传包虽然数量较少，但由于已经消耗了部分时间，因此具有更严格的时间限制。Hairpin 通过在传输层中同时考虑这两类数据包的不同需求，将丢包恢复问题建模为一个 Markov 决策过程。在该模型中，数据包的传输与丢失过程被编码为节点状态及其状态间的转移关系，每个节点表示剩余数据包数量和传输机会。算法通过递归计算各节点的最优冗余率，并结合丢包率计算该转移过程的转移概率，最终选择转移概率最大的那条路径，从而优化截止时间错过率和带宽成本。同时，Hairpin 通过枚举可能的块大小，进一步优化块大小与冗余率的组合，从而在特定网络条件下确定最优的 FEC 参数。

但由于其最优的 FEC 参数是固定的，在动态网络环境中的适应性仍然有限。因此，基于 Hairpin 算法，我提出了一种将 Markov 模型与 Bayes 理论耦合的新方法，模型在初始时是最优 FEC 参数状态，当服务端的数据收集器检测到包丢失时，根据上一次传输的丢包率、帧大小等信息定位到上一次传输的 FEC 参数，增加该层数据块的丢包一定数量的先验概率，然后似然估计丢失一定数量的数据包组合概率，最后求得边际似然，再根据边际似然求解出后验概率，就是该节点在这一 FEC 参数下转移到下一节点的质量评估（QoE）的一部分。在这篇文章中，我的主要贡献如下：

- 提出了基于贝叶斯的改进 Hairpin 算法，通过动态更新转移概率以提升对复杂网络环境的适应能力。
- 在 NS3 仿真平台上验证了该方法的性能，并与现有 Hairpin 算法进行了对比。
- 实验结果显示，尽管降低了丢包率，但我的方法对 Hairpin 算法并没有明显地提升。

2 相关工作

2.1 Markov 决策

Markov 决策过程作为一种决策理论，广泛应用于解决各种需要决策的动态问题。该模型由五个基本元素构成：状态集、动作集、转移概率、奖励函数以及折扣因子。每个决策时刻，

系统处于某个状态，决策者根据当前状态选择一个动作，随后系统转移到另一个状态，并根据该状态和采取的动作获得一个奖励。

Markov 决策过程的关键特性在于每个节点当前状态的转移只依赖于当前状态和选择的动作，和历史状态无关。通过这个特性，Markov 决策过程为复杂的决策问题提供了一种系统化的求解框架。在强化学习中，Markov 决策过程为代理提供了一个训练框架，代理通过探索环境中的状态、动作及其后果，最终学会如何做出最优决策。

近年来，随着深度学习技术的发展，基于深度神经网络的强化学习方法结合 Markov 决策过程框架，取得了在诸如游戏、机器人控制、自动驾驶等领域的显著成果。例如，徐瑾等人提出了基于非齐次马尔科夫链的延迟预测模型 [8]，并引入高斯核密度估计的矩阵恢复方法，用于处理像是列车延迟这类的大规模预测问题；Padi 等人利用多级隐藏 Markov 模型研究 1 类糖尿病等 [9]。

2.2 贝叶斯算法

贝叶斯算法基于贝叶斯定理，是一种统计推断的工具，广泛应用于统计学、机器学习、人工智能等领域，并且它在不确定性推理和决策支持中起着很大的作用。贝叶斯定理的核心思想是根据新的观测数据更新我们对一个假设或事件的估计，它为我们提供了一种从先验分布出发，通过观察证据逐步调整假设的概率分布的方式。贝叶斯定理的数学表达式为：

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

其中 $P(A|B)$ 是后验概率，表示在观察到数据 B 后，事件 A 发生的概率； $P(B|A)$ 是似然概率，表示在假设 A 为真的情况下，观测到数据 B 的概率； $P(A)$ 是先验概率，表示事件 A 发生的初步概率；而 $P(B)$ 是数据 B 的边际概率。

贝叶斯方法的特性是它能够结合先验信息与新的观测数据进行更新。先验分布通常是在观测到数据之前已有关于某个事件或现象的分布情况，贝叶斯推断通过不断地更新这一估计，使得我们的模型能够随着数据的增加变得更加准确。贝叶斯方法强调更新的过程，适合用于动态环境中，在新信息到来时自动调整推断结果。

在实际应用中，贝叶斯方法的优势在于其能够量化不确定性。通过后验概率分布，贝叶斯方法不仅告诉我们某一事件发生的概率，还能够量化我们对该事件发生的概率推断。相比传统的频率学派方法，贝叶斯方法能够提供更为细致的信念更新过程，特别是在数据稀缺或不完全的情况下，其表现通常更为稳健。

贝叶斯方法在许多领域都得到了广泛的应用。在深度学习领域，Subbiyan 等人提出了贝叶斯深度卷积神经网络，通过贝叶斯反向传播来管理不确定性，从而预测空气污染引起的信号衰减 [10]。在机器学习领域，Mavhemwa 等人提出了加权朴素贝叶斯多用户分类方法，通过对加权和未加权数据集进行分类，并采用交叉验证评估性能，从而在自适应认证中实现了多类用户的风险分类 [11]。

3 本文方法

3.1 本文方法概述

Hairpin 算法将冗余率的决策编码为 Markov 决策过程，每一个节点由其截止日期错过率和带宽损耗率构成。在整体构建过程中，其转移概率是根据当前网络数据收集器所分析的丢包率和带宽损耗率所计算的 QoE，根据 QoE 的大小对每一层传输的数据的决策进行选择。在每次丢包后，系统会根据丢失数据包的位置，更新该节点在某一条路径上的 QoE，从而更新出新的选择。

Hairpin 算法会通过评估每个块大小的 QoE 表现，最大化整体传输质量，从而决定不同层在不同丢包率情况下的块大小。不同的块大小会影响传输层次的数量、每层的 QoE 值以及冗余带来的影响，因此选择最佳的块大小可以在各层之间找到平衡，从而提升最终的传输体验。系统架构图如图 1 所示。

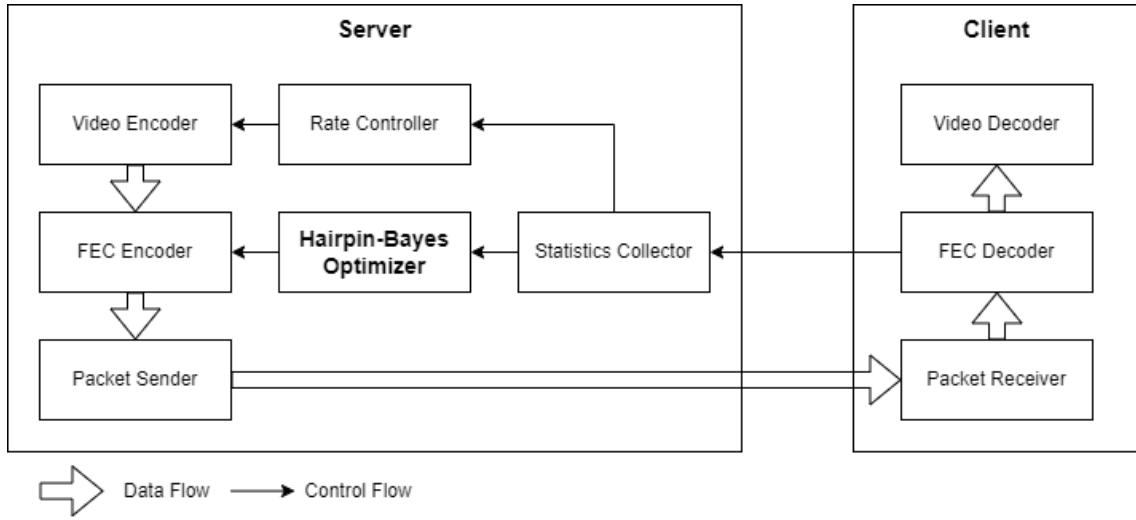


图 1. 系统架构图

3.2 Markov 和贝叶斯的结合

在网络传输中，由于丢包现象的存在，接收端有可能无法获取完整的数据包。为了提高传输成功率，可以引入冗余包，通过在发送端添加额外的冗余信息，接收端可以在一定程度上纠正丢失的数据包。然而，增加冗余包会消耗带宽资源，因此需要合理设计冗余包数量，使得传输的成功率最大化。

假设传输数据包总数为 N ， n_0 表示原始数据包数， f 是冗余包数。 n_1 表示发送后剩余数据包数量，符合 $0 \leq n_1 \leq n_0 + f$ ，已知丢包率 P_{loss} 是当前丢包率。目标是求解每种冗余配置 (n_0, n_1, f) 下传输失败的概率 P_{fail} ，并结合丢失 i 个数据包对应的先验概率，从而求解出后验概率。

可以将发生丢包事件建模为一个二项分布问题，假设每个丢包事件是独立且同概率发生， N 个总包中任意 i 个包丢失的概率为：

$$P(i) = \binom{N}{i} \cdot P_{loss}^i \cdot (1 - P_{loss})^{(N-i)} \quad (2)$$

其中 $\binom{N}{i} = \frac{N!}{i!(N-i)!}$ 是组合数，表示从 N 个包中选择 i 个丢包的可能性。 P_{loss}^i 是 i 个包丢失的概率。 $(1 - P_{loss})^{(N-i)}$ 是其余包成功接收的概率。

为进一步优化冗余包设计，系统引入了先验概率 $P_{prior(i)}(f)$ 和后验概率 $P_{posterior(i)}(f)$ 。先验概率假设服务端在选择冗余包数量时具有均匀分布 $P_{prior(i)}(f) = \frac{1}{f+1}$, $f \in [0, f_{max}]$ 根据贝叶斯定理，后验概率的计算公式为：

$$P_{posterior(i)}(f) = \frac{P_{likelihood(i)}(f) \cdot P_{prior(i)}(f)}{\sum_{f=0}^{f_{max}} P_{likelihood(i)}(f) \cdot P_{prior(i)}(f)} \quad (3)$$

其中，似然函数 $P_{likelihood(i)}(f)$ 表示给定冗余配置下的失败概率，即 $P(i)$ 。

系统状态的转移概率可以表示为 $P_{trans}(n_0, n_1, f)$ ，即当前需要发送数据包数、发送后剩余数据包数和冗余包数。转移概率受网络丢包率和冗余包设计的影响。因此，当在给定冗余包数量 f 下从状态 (n_0, t_0) 转移到 (n_1, t_1) 的概率定义如下：

$$P_{trans}(n_0, n_1, f) = \sum_{i=\max(f+1, n_1)}^{n_1+f} P_{posterior(i)} \cdot \binom{n_0+f}{i} \cdot P_{loss}^i \cdot (1 - P_{loss})^{(n_0+f-i)} \quad (4)$$

由于在丢包发生后，需要增加该事件发生的先验概率，这会导致 $P_{trans}(n_0, n_1, f)$ 增加，因此在最后的 QoE 计算中，采取取负的方式。

3.3 数据块大小决策构建

在网络视频传输的场景中，由于丢包、延迟、带宽限制等因素的影响，传输质量体验往往会受到显著干扰。为了解决这个问题，我们需要通过调整数据的区块大小来优化传输，从而提高用户的 QoE。一个总数据包由基础数据包和冗余数据包组成。为了建模这一问题，我们需要考虑每个不同数量的总数据包数的传输延时以及丢包对 QoE 的影响。系统的优化目标是根据给定的网络条件，找到每个网络参数组合下的最优区块大小 BlockSize，以最大化 QoE。

$$\max_x QoE(x) = \sum_{i=1}^N QoE_i(x) \quad (5)$$

其中 x 为块大小。 N 为传输层的数量。 $QoE_i(x)$ 表示第 i 层的传输质量体验。

为此，在每一层中我们都使用网格搜索的方式，遍历所有可能的参数组合，计算每组参数下的 QoE，并选取 QoE 最大的区块大小。在指定参数时，利用公式 (6) 需要确认时间存在多少次机会能够进行传输。如果机会大于 0，就可以根据截止日期、RTT、剩余机会和块间延迟，该层计算出最大冗余的块数为多少，即公式 (7)。再去依次遍历所有可能的冗余块数，计算不同冗余块数时，冗余数据包的大小和原始数据包的大小，即公式 (8) 和公式 (9)，进而求得 QoE。依此类推，结合之前 Markov 链中所求得的 QoE，找出最大 QoE。

$$BaseLayer = \left\lfloor \frac{DDL - FrameSize \cdot R_{disp}}{RTT} \right\rfloor \quad (6)$$

$$MaxAdditionalBlock = \left\lfloor \frac{DDL - (BaseLayer + 1) \cdot RTT}{R_{disp}} \right\rfloor \quad (7)$$

$$AddLayerBlockCnt = \left\lfloor \frac{DDL - (BaseLayer + 1) \cdot RTT}{BlockSize \cdot R_{disp}} \right\rfloor \quad (8)$$

$$\text{BaseLayerBlockCnt} = \left\lceil \frac{\text{FrameSize}}{\text{BlockSize}} \right\rceil - \text{AddLayerBlockCnt} \quad (9)$$

其中 DDL 为截至日期错过率, FrameSize 表示帧大小, R_{disp} 表示数据包间的传输延迟, RTT 表示网络往返时间。

4 复现细节

4.1 与已有开源代码对比

Hairpin 算法目前已在 GitHub 开源, 在本次实验中, 我修改了其开源代码中生成 Markov 表的 `fec_w_rtx.py` 文件, 使用该方法生成初始 Markov 表, 并测试在修改先验概率后能够有效修改缓存的 `pkl` 文件中的转移概率。在实验仿真模拟中, 我修改了其使用 NS-3 仿真模拟所采用的文件, 确保在检测到历史数据包丢包后, 能够定位到其使用的 FEC 参数所处文件的位置, 并有效修改在二进制文件中的参数。

4.2 实验环境搭建

1. 前往 <https://github.com/hkust-spark/hairpin>。在 linux 环境下下载 C++, python 和 hairpin 的代码。
2. 使用 `sudo apt install build-essential libboost-all-dev` 命令, 安装对应依赖。
3. 使用 `./setup-env.sh` 命令, 安装 NS-3 环境。
4. 由于使用 NS-3.33 及以下版本且使用 C++14 以及上的版本, 需要输入 `LDFLAGS="-lboost_filesystem -lboost_system" ./waf configure -cxx-standard=-std=c++17` 命令配置。
5. 若要复现实验需要在 `ns-allinone-3.33/ns-3.33` 目录下输入 `python run_ns3.py -conf wifi-hairpinone.conf` 命令。
6. 但是由于需要配置文件 `wifi-hairpinone.conf`, 因此需要在 `ns-allinone-3.33/ns-3.33` 目录下输入 `./generate_conf_hairpinone.sh` 命令, 但它需要根据该目录下的 `traces` 文件夹中带有 `wifi` 名称的追踪数据, 因此, 还需要将 `sparkrtc` 中 `sample.tr` 移动到这里, 并且修改为一个带有 `wifi` 名称的文件。
7. 同时, 还会存在找不到 `rtc-test.cc` 的报错, 因此还需要将 `sparkrtc` 中 `rtc-test.cc` 转移到 `ns-allinone-3.33/ns-3.33/scratch` 目录下。
8. 在完成上述内容后, 实验环境就已经成功搭建并能够开始仿真模拟。

4.3 创新点

我的研究主要创新在于提出了一种基于贝叶斯理论的 Hairpin 算法改进方法, 目的是通过引入动态更新转移概率的机制, 提升算法在复杂网络环境中的适应能力。我将数据包的丢失恢复建模为一个结合 Markov 模型和贝叶斯理论的联合优化问题。相比 Hairpin 算法, 该方

法不仅在初始状态下优化了 FEC 参数，还通过利用服务端数据收集器实时检测的丢包率、帧大小等特征信息，根据先验概率和边际似然对丢包模式进行动态调整，从而为后续传输的节点状态转移提供动态的质量评估。

5 实验结果分析

NS-3 仿真实验分别模拟了不同偏好情况下算法的 pareto 前沿，通过该仿真实验，我对比了原 Hairpin 算法和基于贝叶斯的 Hairpin 算法的仿真结果，实验对比如图 2 所示。

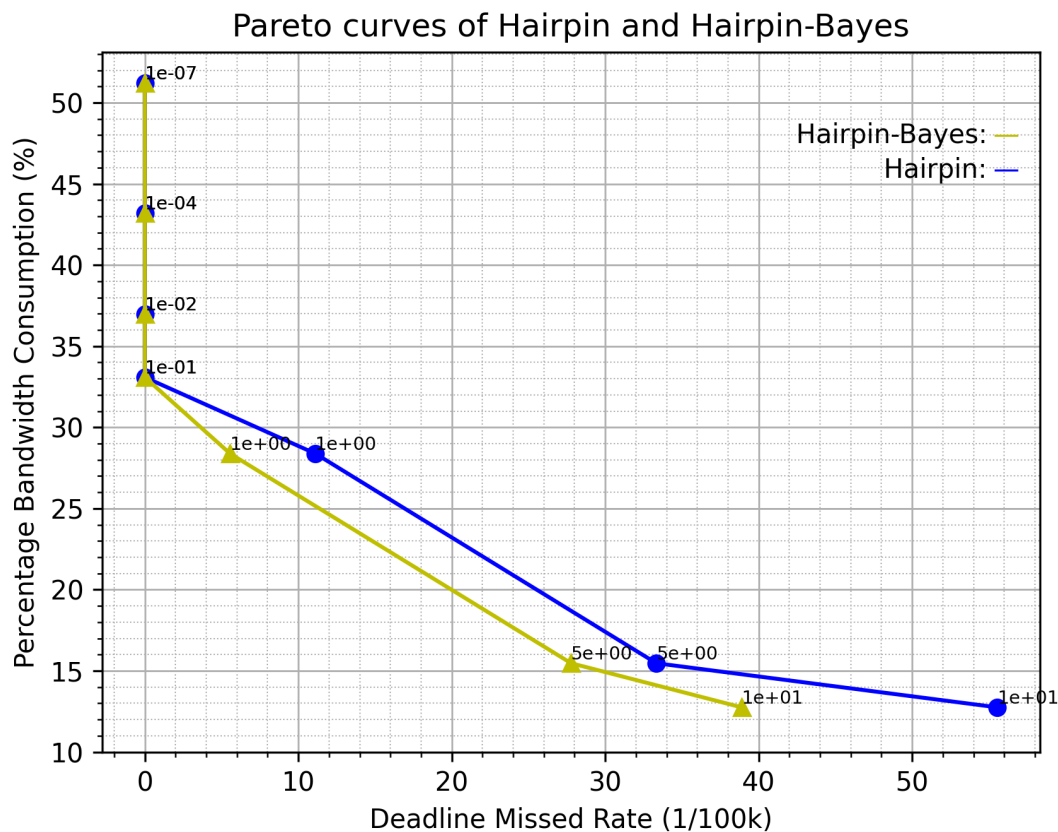


图 2. 实验对比

实验对比图表明，在相同 λ 参数的情况下，虽然丢包率有所下降，但实际带宽损耗率也高了一些。因此，在发生丢包事件后，由于增加了某一节点在某一丢包情况下的先验概率，该条路径的 QoE 存在一定下降，因此可能会更换为一个冗余率更高的路径，从而使得丢包率降低，带宽损耗率提升。

6 总结与展望

当前仿真的数据量较大，但没有出现特别明显的丢包问题，在大量丢包发生的情况下，该方案是虽然是动态的，但其实也是一个增加带宽消耗换取截止日期错过率减小的方法。并且有时出现的正常网络波动导致的丢包事件，也会降低该路径的 QoE，但这不是一个好的解决方案，需要进一步优化。我的方案都是在每一层结束后进行更新的，在实时方面并不能起到

一个很好的作用，只能将其应用于下一组数据包的传输，尽管我测试了更新路径的延迟都大于一个 20ms 的 RTT，但也许使用更多缓存策略能够有效减少这种延迟，这在对时间非常敏感的网络传输上，显得非常重要。

参考文献

- [1] Yiling Xu, Jiaoyang Yin, Qi Yang, and Le Yang. Media production using cloud and edge computing: Recent progress and nbmp-based implementation. *IEEE Transactions on Broadcasting*, 68(2):545–558, 2022.
- [2] Yung-Yao Chen, Yu-Hsiu Lin, Yu-Chen Hu, Chih-Hsien Hsia, Yi-An Lian, and Sin-Ye Jhong. Distributed real-time object detection based on edge-cloud collaboration for smart video surveillance applications. *IEEE Access*, 10:93745–93759, 2022.
- [3] Tao Jiang. Digital media application technology of mobile terminals based on edge computing and virtual reality. *Mobile Information Systems*, 2021(1):3940693, 2021.
- [4] Ali Ekşim and Hasan Yetik. Performance improvement of ambe 3600 bps vocoder with adaptive forward error correction. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2018.
- [5] Marek Blok and Bartosz Czaplewski. A novel iterative decoding for iterated codes using classical and convolutional neural networks. In *International Conference on Computational Science*, pages 231–238. Springer, 2024.
- [6] Sheng Cheng, Han Hu, and Xingong Zhang. Abrf: Adaptive bitrate-fec joint control for real-time video streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9):5212–5226, 2023.
- [7] Zili Meng, Xiao Kong, Jing Chen, Bo Wang, Mingwei Xu, Rui Han, Honghao Liu, Venkat Arun, Hongxin Hu, and Xue Wei. Hairpin: Rethinking packet loss recovery in edge-based interactive video streaming. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pages 907–926, 2024.
- [8] Jin Xu, Weiqi Wang, Zheming Gao, Haochen Luo, and Qian Wu. A novel markov model for near-term railway delay prediction. *Computers & Industrial Engineering*, 181:109302, 2023.
- [9] Tirupathi Rao Padi and Surnam Narendra. Multilevel hidden markov models for studying type-1 diabetes. *JP Journal of Biostatistics*, 24(1):161–176, 2024.
- [10] Prakash Subbiyan and Sangeetha Krishnamurthy. Prediction of signal attenuation due to air pollutants using bayes-dcnn. *Environmental Research Communications*, 6(12):125001, 2024.

- [11] Prudence M Mavhemwa, Marco Zennaro, Philibert Nsengiyumva, and Frederic Nzanywayingoma. Weighted naïve bayes multi-user classification for adaptive authentication. *Journal of Physics Communications*, 8(10):105005, 2024.