https://doi.org/10.7840/kics.2025.50.4.616

통계 추정 기반 ABR 알고리즘의 딥러닝 기반 성능 향상

문 이 빈◆, 안 동 혁°

Deep Learning-Based Performance Improvement of Statistical Estimation-Based ABR Algorithm

Ie-bin Moon◆, Donghyeok An°

요 약

최근 넷플릭스 등과 같은 OTT 플랫폼의 수요가 상승하고, 비디오 스트리밍 시장이 커짐에 따라, 스트리밍 서비스의 핵심 알고리즘인 ABR(Adaptive Bitrate) 알고리즘의 성능 향상 및 사용자 경험 품질(QoE, Quality of Experience)의 향상 연구가 더욱 중요해지고 있다. 종래의 ABR 알고리즘 중 모델 제어 예측 기반 알고리즘인 MPC(Model Predictive Control)와 Robust MPC ABR 알고리즘의 네트워크 대역폭 예측 알고리즘은 통계적인 추정 방식을 사용하여 대역폭의 변동성이 매우 큰 경우 예측의 오류로 인해 최적에 비해 성능이 저하될 수 있다. 이에 본 연구에서는 LSTM, Transformer 기반 시계열 예측 모델을 이용하여 네트워크 대역폭을 예측하고, MPC에 적용하여 개선점을 제안한다. ABR 알고리즘 시뮬레이션 프레임워크를 이용하여 QoE 지표로 측정한 성능을 기존 MPC ABR 알고리즘과 정량적으로 비교한 결과, LSTM, Transformer 모델에서 모두 높은 성능을 보였으며, 기존 대비 성능을 각각 8.71%, 8.91% 개선하였다.

Key Words : ABR Algorithm, Deep Learning, LSTM, MPC, Quality of Experience, Transformer

ABSTRACT

With the increasing demand for OTT platforms such as Netflix and the growth of the video streaming market, enhancing the performance of Adaptive Bitrate (ABR) algorithms, a key component of streaming services, and improving the Quality of Experience (QoE) for users have become more critical. The network bandwidth prediction algorithms in conventional Model Predictive Control (MPC) and Robust MPC-based ABR algorithms rely on statistical estimation methods, which can lead to performance degradation due to prediction errors, especially in scenarios with highly fluctuating bandwidth. In this study, we propose improvements by employing LSTM (Long Short-Term Memory) and Transformer-based time series prediction models for network bandwidth forecasting and applying them to the MPC algorithm. By quantitatively comparing the ABR performance measured through QoE metrics using an ABR algorithm simulation framework, we observed that both LSTM and Transformer models achieved superior performance, improving the conventional MPC-based ABR algorithm by 8.71% and 8.91%, respectively.

※ 이 논문은 2024년도 국립창원대학교 학생주도 창의연구프로젝트 지원사업연구비에 의하여 연구되었음.
※ 이 논문은 2025년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임(P0017006, 2025년 산업혁신인재성장지원사업).
◆ First Author : Changwon National Univercity, Department of Computer Engineering, muniebin@gmail.com, 학생회원
° Corresponding Author : Changwon National Univercity, Department of Computer Engineering, donghyeokan@changwon.ac.kr, 종신회원
논문번호 : 202411-276-C-RE, Received November 8, 2024; Revised January 24, 2025; Accepted February 17, 2025

Ⅰ. 서 론

최근 넷플릭스 등과 같은 OTT 플랫폼의 수요가 증가하고, 비디오 스트리밍 시장이 커지고 있으며, statistica 전세계 수익 전망에 따르면, 그림 1의 그래프와 같이 2027년까지 지속적인 성장을 예상하고 있다[1]. ABR(Adaptive Bitrate) 알고리즘은 적응형 스트리밍 서비스에서 사용되는 핵심 기술이다. ABR 알고리즘은 주로 클라이언트에서 동작하는데, 실시간 동적으로 변하는 네트워크 환경에서 재생 버퍼 레벨 및 다 운로드 가능한 비디오 화질 등을 고려하여 사용자에게 최적의 체감품질 QoE(Quality of Experience)를 제공하는 데 그 목적이 있다.

적응형 스트리밍 서비스에서 QoE는 비디오 화질, 비디오 화질 변화, 리버퍼링 시간으로 정의한다[2,3]. Xiaoki Yin 등의 연구에서는 정의된 QoE를 기반으로 비트레이트를 선택하는 모델 예측 제어 기반의 ABR 알고리즘인 MPC(Model Predictive Control)를 제안하였다[2]. MPC ABR 알고리즘은 향후 네트워크 처리량을 예측하고, 이를 바탕으로 최적의 비트레이트를 선정하여 사용자의 QoE를 극대화한다. MPC가 최적의 성능을 나타내기 위해서는 동적으로 변화하는 네트워크 전송처리량의 오차를 줄여야 한다. 기본적으로 MPC는 네트워크 전송처리량 오차를 줄이기 위해서 과거 네트워크 전송처리량의 이동평균(moving average)과 같은 통계적 방식을 이용한다.

Robust MPC는 추정된 과거 네트워크 전송처리량과 실제 전송처리량과의 오차를 반영하여 네트워크 전송처리량을 추정하는 방식을 이용하였다. MPC와 같은 QoE 기반의 ABR 알고리즘은 변동성이 큰 무선 네트워크 등과 같은 환경에서는 처리량을 정확히 추정하기 어려운 단점이 있다. 이에 본 연구에서는 LSTM(Long Short-Term Memory)층을 활용한 딥러닝 기반 시계열 예측 모델을 통해, 처리량을 예측하는 대역폭 추정 기법을 제안한다. 제안한 네트워크 처리량 추정 기법을 Robust MPC에 적용하여 시뮬레이션 실험을 통해 QoE 지표를 바탕으로 ABR 알고리즘의 성능을 정량적으로 비교하였으며, LSTM과 Transformer 모델에서 기존 대비 성능을 각각 약 8.71%, 8.91% 향상시켰다.

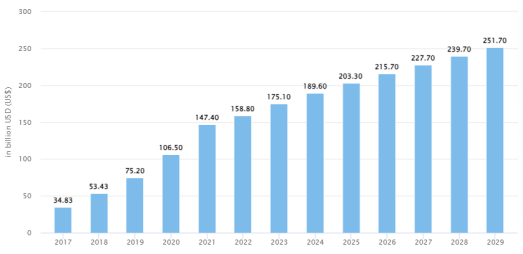

그림 1. statistica 전세계 OTT 시장 광고수익 전망
Fig. 1. Market-Insights-OTT-Video Advertising Revenue-Worldwide

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에서 성능 개선을 수행한 ABR 알고리즘인 MPC와 본 연구와 관련된 연구 동향을 기술한다. 3장에서는 실험에 사용한 데이터 및 딥러닝 모델, 시뮬레이션 환경을 설명하고, 기존 Robust MPC 알고리즘에 적용한 딥러닝 기반 네트워크 대역폭 예측 알고리즘과 적용 방법을 설명한다. 4장에서는 대역폭 예측 모델을 검증하고 제안한 기법을 시뮬레이션에서 측정한 QoE 기반 성능 평가를 진행한 결과를 서술한다. 마지막으로 5장에서는 본 논문의 결론 및 향후 연구에 대해서 서술한다.

Ⅱ. 배경 지식 및 관련 연구

이 장에서는 MPC ABR 알고리즘 및 LSTM에 대한 배경 지식과 ABR 알고리즘 및 딥러닝 기반 시계열 예측 관련 연구 동향을 설명한다.

2.1 MPC 알고리즘

비디오 스트리밍 서비스에서 QoE는 그림 2의 식과 같이 나타낸다[2,3].

$$QoE_1^K = \frac{1}{K}\sum_{k=1}^{K} q(R_k) - \frac{\lambda}{K-1}\sum_{k=1}^{K-1}|q(R_{k+1}) - q(R_k)| - \frac{\mu}{K}\sum_{k=1}^{K}\left(\frac{d_k(R_k)}{C_k} - B_k\right)_+ \quad (1)$$

여기서, $k, q(\bullet), B_k, C_k, R_k,$ 는 각각 전체 청크 수, 비디오 화질 함수, 버퍼레벨, 전송처리량, 비트레이트를 의미한다. $q(R_k)$는 n개 화질 조합 내 요소의 비트레이트(kbps) 값을 의미한다. 조합 내 총합이 높을수록, 사용자는 더 높은 화질로 시청하므로, $q(R_k)$값은 QoE에 비례한다.

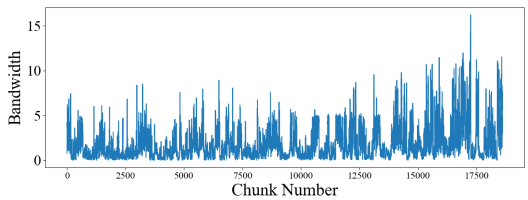

그림 2. 자동차 시나리오의 LTE 대역폭 분포
Fig. 2. LTE bandwidth distribution of car scenario

$|q(R_k+1)-q(R_k)|$는 화질 변동량(kbps)을 의미하며 조합 내 총합이 클수록, QoE는 낮아진다. λ는 화질 변동 가중치, μ는 리버퍼링 시간 $(\frac{d_k(R_k)}{C_k}-B_k)_+$는 청크를 내려받는 데 걸릴 시간과 버퍼레벨의 차로, 이는 리버퍼링에 걸릴 시간을 의미한다. 이 값은 리버퍼링이 발생하는, 가용 버퍼보다 청크 다운로드 시간이 큰 경우에만 계산된다.

MPC는 알고리즘 1과 같이, 먼저 플레이어가 시작 단계에 있는지, 즉 초기 버퍼링을 수행 중인지 확인한다. 이 단계에서 사전에 구현된, 네트워크의 처리량(throughput)을 예측하는 ThroughputPred() 함수의 알고리즘을 바탕으로, 앞으로의 네트워크 성능을 예측한다. 이 과정을 통해 플레이어는 최적의 버퍼링 시간을 결정하고, 일정 시간이 지나면 재생을 시작한다. 재생이 시작된 이후에도 플레이어는 지속적으로 처리량을 예측하여 다음 청크의 비트레이트를 결정한다. 이 과정에서 MPC는 이전에 비트레이트 R_{k-1}, 현재 버퍼레벨 B_k, 예측된 처리량 $\overline{C}_{[k,k+N]}$을 종합적으로 고려하여 QoE를 최대화할 수 있는 비트레이트 R_k를 선택한다. 플레이어는 결정된 비트레이트로 해당 청크를 다운로드하고, 다운로드가 완료되면 다음 청크를 처리하는 과정을 반복한다.

여기서 N은 예측하는 청크의 수를 의미한다. 가령 N값이 5인 경우, MPC는 QoE를 최대화하는 R_k, R_{k+1}, R_{k+2}, R_{k+3}, R_{k+4}를 예측하여 비트레이트가 R_k인 청크를 서버에 요청하게 된다. 이 때, $\overline{C}_{[k,k+N]}=\overline{C_k}$ 로 가정하여 예측 처리량 $\overline{C_k}$를 이동평균으로 계산하는데, MPC와 Robust MPC가 $\overline{C_k}$를 계산하는 방식이 다르다[4].

Algorithm 1 Video adaptation workflow using MPC

```
Initialize
for k = 1 to K do
    if player is in startup phase then
        Ĉ_[t_k,t_{k+N}] = ThroughputPred(C_[t_1,t_k])
        [R_k, T_s] = f^st_mpc(R_{k-1}, B_k, Ĉ_[t_k,t_{k+N}])
        Start playback after T_s seconds
    else if playback has started then
        Ĉ_[t_k,t_{k+N}] = ThroughputPred(C_[t_1,t_k])
        R_k = f_mpc(R_{k-1}, B_k, Ĉ_[t_k,t_{k+N}])
    end if
    Download chunk k with bitrate R_k, wait till finished
end for
```

알고리즘 1. MPC 최적화 알고리즘
Algorithm. 1. Video adaptation workflow using MPC

$$\overline{C_k}=\begin{cases}\frac{1}{N}\sum_{i=k-N+1}^{k-1}C_i & ,MPC\\ \frac{1}{N(1+e_{\max})}\sum_{i=k-N+1}^{k-1}C_i & ,Robust\ MPC\end{cases} \quad (2)$$

여기서 최대 오차 $e_{\max}$는 식 (3)과 같다.

$$e_{\max}=\max\frac{|\overline{C_l}-C_l|}{C_l}, l=k-N+1,...,k-1 \quad (3)$$

2.2 LSTM

LSTM은 RNN의 장기 의존성 문제를 해결하기 위해 개발된 구조이다(Hochreiter & Schmidhuber, 1997)[5]. LSTM은 "셀 상태"라는 개념을 도입하여, 중요한 정보를 장기적으로 유지하고 불필요한 정보를 망각하게 함으로써, 시퀀스의 길이에 관계없이 효율적으로 학습할 수 있다. LSTM은 입력 게이트, 출력 게이트, 망각 게이트라는 세 가지 주요 게이트 구조를 통해 정보의 흐름을 제어하며, 이는 RNN에 비해 더 정교하고 정확한 예측을 가능하게 한다. LSTM은 다양한 시계열 예측 문제에서 뛰어난 성능을 발휘하며, 특히 긴 시퀀스에서의 예측 정확도를 크게 향상시킨다.

2.3 Transformer

Transformer는 RNN 또는 LSTM과 같은 순차적 구조를 사용하지 않고, 완전히 Attention 메커니즘에 기반하여 동작모델이다(Vaswani et al., 2017)[6]. Transformer는 Attention 메커니즘은 입력 데이터의 각 요소가 다른 모든 요소와의 연관성을 학습하도록 하는 방법으로, 딥러닝 모델이 중요한 정보에 "집중"할 수 있게 한다. 또한, Transformer는 모든 입력 위치 간의 관계를 동시에 고려할 수 있는 병렬 처리 구조를 채택하여, 순차적 처리로 인한 병목 문제를 해결하였다. 특히 Self-Attention 메커니즘은 입력 데이터 내 각 요소의 중요도를 동적으로 학습하여, 장기 의존성 문제를 효과적으로 처리한다. Transformer는 인코더와 디코더로 구성된 구조를 가지며, 각 모듈은 다중 헤드 어텐션(Multi-Head Attention)과 피드포워드 신경망(Feed-Forward Neural Network)으로 이루어져 있다. 인코더는 입력 시퀀스의 특징을 추출하고, 디코더는 이를 바탕으로 출력 시퀀스를 생성한다. 또한, 포지셔널 인코딩(Positional Encoding)을 통해 입력 데이터의 순서를 반영하여 순차적 특성을 보완한다. Transformer는 기계 번역, 언어 모델링 등 자연어 처리 분야에서 뛰어난 성능을 발휘하며, 그 이후 시계열 데이터 예측으로도 적용 범위를 확장하였다. 특히 병렬 처리의 이점을 활용

하여 기존의 RNN 및 LSTM 기반 모델에 비해 학습 속도와 예측 정확도를 크게 향상시키며, 딥러닝 모델의 새로운 표준으로 자리 잡았다.

2.4 ABR 알고리즘 연구 동향

ABR(Adpative BitRate) 알고리즘은 네트워크 처리량 등을 기반으로 사용자 경험 품질(QoE, Quality of Experience)을 최대화하기 위해서 비디오 세그먼트 품질을 지속해서 동적으로 조정하는 역할을 한다. QoE의 저감 요소로는 재생 시작 지연시간, 재생 버퍼에 렌더링 할 컨텐츠가 없을 경우 발생하는 리버퍼링, 평균 비디오 품질 변화가 있으며, 화질이 높을수록 QoE는 증가한다. 평균 화질을 높이거나, 리버퍼링 및 품질 변화를 줄이는 방법의 QoE를 높이는 연구가 활발히 진행중이며, QoE는 콘텐츠 제공 업체의 수익과 직결되기 때문에 매우 중요하다. 한편, 최근에는 Pensieve[3]를 비롯하여 강화학습 기반 ABR 알고리즘 성능 향상 연구가 활발히 이루어지고 있다. Ye et al.의 연구[14]에서는 인간 시각 시스템(Human Visual System, HVS)의 시각적 민감도를 고려한 ABR 알고리즘을 제안하였다. 기존 알고리즘들이 비디오 콘텐츠의 품질 왜곡에 대한 사용자 인지 차이를 무시하는 문제점을 보완하기 위해, 시각적 민감도 모델을 개발하고 강화학습을 기반으로 비트레이트 결정을 수행하였다. 이 알고리즘은 네트워크 상태, 버퍼 점유율, 시각적 민감도를 통합적으로 고려하였다. Zhang et al.의 연구[15]에서는 가변 비트레이트(Variable Bitrate, VBR) 인코딩이 실시간 비디오 통신 시스템에서 비트레이트 변동성을 크게 증가시킨다는 점을 지적하고, 이를 해결하기 위해 Anableps라는 새로운 ABR 모델을 제안하였다. 이 모델은 송신자 측의 비디오 콘텐츠 정보를 활용하여 향후 비트레이트 범위를 예측하고, 수신자 측 네트워크 상태와 결합하여 강화학습 기반으로 비트레이트 결정을 수행한다. Liu et al.의 연구[16]에서는 기존 코덱 기반의 ABR 알고리즘이 비트레이트 결정을 수정할 수 없는 한계를 극복하기 위해, 신경 표현(Neural Representation)을 활용한 새로운 프레임워크인 EVAN을 제안하였다.

본 연구는 기존 연구들과 여러 차별점이 존재한다. 먼저, 기존 연구들은 ABR 알고리즘의 비트레이트 결정 과정의 최적화를 집중한다. 하지만 본 연구에서는 비트레이트 결정 과정이 아니라 대역폭 예측 정확도 향상을 목표로 한다. 두 번째로 기존 연구들에서는 강화학습, CBPN(Context-Based Predictive Network), NeRV(Neural Representations for Videos)를 사용하기 위해 ABR 알고리즘 구조를 변경한다. 하지만 본 연구에서는 기존의 MPC 모델 구조를 최대한 유지한다. 마지막으로 Zhang et al.의 연구[15]에서는 송신 측 정보를 기반으로 비트레이트를 조정하지만, 본 연구에서는 기존 MPC와 동일하게 수신측 정보를 활용한다. 최근 관련 연구와 본 연구의 주요 차이점을 표 1로 정리하였다. 기존 연구인 VS-ABR, Anableps, EVAN은 각각 강화학습(DRL), 규칙 기반 VBR, SAC 강화학습 및 NeRV 프레임워크를 사용하며, 3G 또는 유선(Wired) 환경에서 실험되었다. 반면, 본 연구는 LTE 및 5G를 포함한 최신 네트워크 환경에서 성능을 평가하였으며, LSTM 및 Transformer 기반의 딥러닝 접근법을 활용하여 네트워크 대역폭 예측 정확도를 향상시켰다. 기존 연구들은 대부분 MPC 구조와 호환되지 않으며, 새로운 아키텍처나 독립적인 모델을 필요로 한다. 그러나 본 연구는 기존 MPC 기반 ABR 알고리즘의 구조를 유지하면서 최신 네트워크 환경에서도 높은 성능을 보장할 수 있는 차별화된 접근법을 제시한다.

표 1 .관련 연구와 본 연구의 비교
Table 1. Comparison of Existing Studies and Our Approach

Approach	Methodology	Network Trace	MPC Compatibility
Ours	Deep Learning	LTE, 5G	O
VS-ABR	Reinforcement Learning	3G	X
Anableps	Rule-based	Wired, 3G, LTE	X
EVAN	Soft Actor-Critic (SAC) NeRV	3G	X

2.5 딥러닝 기반 시계열 예측 연구 동향

시계열 예측 모델은 의료, 금융 등 다양한 분야에서 활용되고 있으며, 딥러닝은 전통적인 이동평균, ARIMA 등과 같은 통계적 방법론보다 높은 성능을 보이고 있다. ‘Mohammed Badawy’ 등의 연구에서는 질병 진행 예측 및 환자 결과 예측과 같은 작업에서 ARIMA와 같은 통계적 방법 대비 LSTM을 활용한 딥러닝 기법의 높은 성능을 보여주었으며[7], ‘Jorge Guijarro-Ordonez’ 등의 연구에서는 금융 분야의 통계적 차익거래 작업에서 딥러닝 모델이 금융 시장의 복잡한 관계를 더 잘 포착하여 금융 자산 간의 관계 예측에 더 높은 성능을 보여주었다[8]. ’K. S. Nithin’ 등의 연구에서는 버스 승객 수요 예측에서 통계 추정 기반의

SARIMA 모델과 딥러닝 기반의 LSTM 모델을 비교하였으며, 예측 주기가 짧을수록 데이터가 복잡해짐에 따라, 통계적 방식 대비 딥러닝 모델이 더 나은 성능을 나타내었다[9]. 한편, Transformer는 현재 시계열 예측 분야에서 State-of-the-Art 모델로 자리 잡고 있으며, 다양한 연구를 통해 그 우수성이 입증되고 있다. 특히, 'Wu et al.'이 제안한 Informer 모델은 Transformer의 계산 복잡도를 줄이면서도 긴 시계열 데이터의 예측 성능을 크게 향상시킨 사례로 주목받았다. Informer는 Sparse Self-Attention 메커니즘과 같은 최적화 기법을 통해 기존 Transformer 대비 메모리 효율성을 개선하였으며, 대규모 시계열 데이터셋에서도 탁월한 성능을 보였다[17]. 또한, 'Oreshkin et al.'의 Temporal Fusion Transformer(TFT)는 시계열 예측의 해석 가능성과 성능을 동시에 고려한 모델로, 에너지 수요 예측과 같은 실시간 의사결정이 중요한 분야에서 높은 성능을 입증하였다[18]. 'Zerveas et al.'의 연구에서는 금융, 기후, 의료 등 다양한 응용 분야에서 다변량 시계열 데이터 분석에 Transformer를 적용하여, 금융 시장 데이터와 같이 복잡한 상관관계를 가진 데이터에서도 기존 대비 높은 성능을 보였다[19].

MPC 기반 ABR 알고리즘에서 필요한 대역폭 예측 과정에서는 예측 주기가 비디오 청크 단위로 매우 짧으며, 화질 선택 과정에 있어 가용 버퍼레벨과 같이 고려해야 할 요소가 많다. 또한 기존 MPC는 이동평균 방식의 통계적 방법을 처리량 추정 작업에 사용하는데, 이는 딥러닝을 활용하면, 통계적인 방식이 포착하지 못하는 비선형적 특징을 파악하고, 보다 높은 성능을 기대할 수 있다. 이에 본 논문에서는 LSTM 및 Transformer 기반의 시계열 예측 모델을 각각 활용하여 대역폭을 예측을 수행하였다.

Ⅲ. 제안 방안

해당 장에서는 실험에 사용한 데이터 및 딥러닝 모델, 시뮬레이션 환경을 설명하고, 2.1절에서 설명한 기존 Robust MPC 알고리즘에 적용한 딥러닝 기반 네트워크 대역폭 예측 알고리즘과 적용 방법을 설명한다.

3.1 네트워크 대역폭 학습 데이터셋

Xiaoki Yin 등의 연구에서는 FCC broadband 데이터 및 3G/HSDPA 데이터를 활용하거나 인위적으로 합성한 데이터를 실험에 사용하였다. 해당 연구에서 사용한 비디오는 h.264/MPEG-4 AVC 코덱으로 압축되었으며, 350kbps, 600kbps, 1000kbps, 3000kbps의 화질 단계까지 설정되어 있고, 이는 Youtube 비디오에서 240p, 360p, 480p, 720p, 1080p에 각각 해당되는 수치이다. 본 연구에서는 최근 높아진 비디오 스트리밍 환경을 반영하기 위해, Cork 대학의 Mobile and Internet Systems Laboratory에서 수집한 4G,5G 무선 네트워크 트레이스를 가공하여 실험에 사용하였다[10,20,21]. 4G 데이터셋의 경우 도시 및 교외를 자동차로 이동하였을 때, 시내를 도보로 이동하였을 때, 실내에서 정지하였을 때 각각 측정된 3가지 이동 패턴 시나리오에 대하여 초당 네트워크 대역폭 트레이스를 가공하였다. 5G 데이터셋의 경우, 두 가지 이동 패턴(정지 환경과 자동차 이동)과 두 가지 애플리케이션 패턴(비디오 스트리밍과 파일 다운로드)을 기반으로 측정된 데이터셋으로, 자동차 이동 및 파일 다운로드 시나리오에서 측정된 트레이스에 대하여 같은 방법으로 가공하였다. 표 2과 같이, 해당 데이터셋들은 다운링크 대역폭, 업링크 대역폭 뿐만 아니라, RSRP(Reference of neiboring cell), RSSI (Received signal strength indicator), 사용자 이동 속도, 신호 대 잡음비(Signal-to-noise ratio)), RSRQ (Reference signal receive quality) 채널 품질 지표 (Channel quality indicator) 등을 함께 제공하고 있으며, 다운링크 대역폭을 선별적으로 본 실험에 활용하였다. kbit/s로 측정된 1초 간격의 다운링크 대역폭 샘플을 표 3와 같이 1초 간격의 Mbit/sec 대역폭 트레이스로 전처리하였다.

표 2. 대역폭 데이터셋 전처리 전 (앞 5행, 주요 제공 지표)
Table 2. Bandwidth dataset before preprocessing (first 5 rows, Key network metrics)

Timestamp	Speed	RSRP	RSRQ	SNR	CQI	RSSI	Downlink Bitrate	Uplink Bitrate
2018.01.17_10.01.04	0	-96	-12	9	9	-79	62238	1027
2018.01.17_10.01.05	0	-92	-12	9	11	-75	49869	846
2018.01.17_10.01.06	0	-92	-12	9	9	-80	49567	867
2018.01.17_10.01.08	0	-93	-9	10	10	-76	57642	989
2018.01.17_10.01.09	0	-93	-9	10	10	-76	57295	1016

표 3. 대역폭 데이터셋 전처리 후 (앞 5행)
Table 3. Bandwidth dataset before preprocessing (first 5 rows, Key network metrics)

Timestamp	Downlink Bitrate
0	63.338
1	49.869
2	49.567
3	57.642
4	57.295

또한 비디오의 경우, 메사추세츠 대학에서 제공한 6000kbps까지 총 10종의 화질로 인코딩된 10분 길이의 Big Buck Bunny 비디오 데이터를 실험에 사용하였으며 3초 길이의 198개 청크로 구성되었다[11]. MPC 알고리즘을 시뮬레이션 하기 위해 필요한 트레이스는 리버퍼링에 소요되는 시간을 고려하여, 6분 50초씩 분할 후 사용하였으며, 시뮬레이션 중 측정되는 청크 단위의 대역폭 데이터를 표 4와 같이 수집하였다. 이 중 70%를 딥러닝 대역폭 예측 모델의 학습에 사용하였으며, 학습에 사용하지 않은 30%를 딥러닝 대역폭 예측 모델의 예측 성능 평가와 ABR 성능 비교 실험에 각각 사용하였다. 그림 2, 3, 4, 5는 각 이동 패턴 시나리오별 데이터셋으로부터 표 3와 같이 전처리된 트레이스로 시뮬레이션을 수행하며 수집된 대역폭 데이터의 분포를 보여준다. MPC 시뮬레이션 중 선정된 청크의 크기 / 청크 다운로드 소요 시간으로 산정된 매 청크 단위의 대역폭이며, 단위는 KByte/sec이다. 시뮬레이션에서 청크 단위로 수집한 선정된 청크 크기 평균, 최댓값, 최솟값, 표준편차를 각각 표 5로 정리하였다. 이동하는 속도가 빠른 버스 이동 패턴에서 보다 높은 표준편차를 보이며, 최댓값, 최솟값 간의 차도 크게 나타난다. 5G 데이터셋의 경우 잦은 핸드오프(hand-off)로 인한 높은 변동성을 보인다.

표 4. 표 3의 트레이스로 시뮬레이션하여 수집한 대역폭 (청크 번호, 누적 시간, 청크 크기, 다운로드에 걸린 시간, 대역폭, 앞 5행)
Table 4. Bandwidth collected through simulation using the trace in Table 3 (chunk number, timestamp, chunk size, download time, bandwidth)

Chunk index	Time Stamp (sec)	Chunk Size (byte)	Download Time (sec)	Bandwidth
1	0.2793	1180512	0.2793	4.225
2	1.188	4908	0.909	5.399
3	2.240	5718960	1.051	5.438
4	4.767	17426160	2.527	6.895
5	6.459	12744936	1.692	7.530

표 5. 학습 데이터셋 세부 측정값
Table 5. Detailded Metrics of Training Dataset

Scenario	LTE			5G
	Car	Pedestrian	Static	Car
Average	1.86	1.41	0.68	5.05
Maximum	16.22	8.01	5.09	46.56
Minimum	0.01	0.01	0.03	0.01
Standard Deviation	1.80	1.31	1.02	7.82
Total Number of Data	18,612	7,524	3,762	8,514

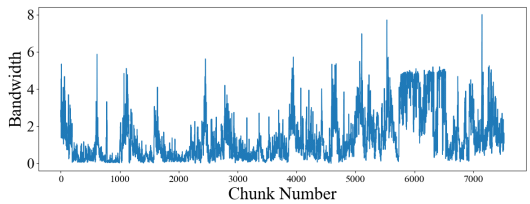

그림 3. 도보 시나리오의 LTE 대역폭 분포
Fig. 3. LTE bandwidth distribution of pedestrian scenario

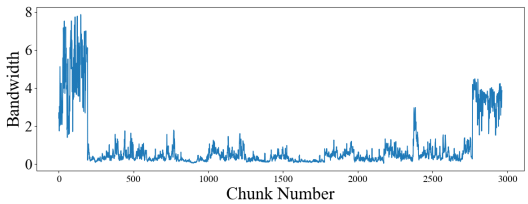

그림 4. 정지 시나리오의 LTE 대역폭 분포
Fig. 4. LTE bandwidth distribution of static scenario

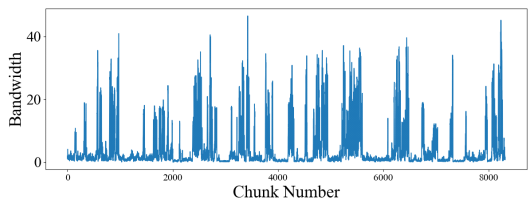

그림 5. 자동차 시나리오의 5G 대역폭 분포
Fig. 5. 5G bandwidth distribution of car scenario

3.2 네트워크 대역폭 예측 모델

본 논문에서 실험에 사용한 딥러닝 기반 네트워크 대역폭 예측 모델을 표 6과 같이 Tensorflow 2.13.0 및 Pytorch 2.4.1 프레임워크와 Python 3.8로 구축하였으며, 운영체제는 Windows 10을 사용하였고, CPU는 i7-13700K 3.40GHz을 사용하였다. 실험 플랫폼에 사

표 6. 실험 환경
Table 6. Experiment Environment

System Specifications	Configuration Details
CPU	i7-13700K 3.40GHz
RAM	64GB
GPU	RTX 4060 Ti 39.9GB
OS	Windows 10
IDE/ Language	PyCharm / Python 3.8
Library	Tensorflow 2.13.0 Keras 2.13.1 Pytorch 2.4.1

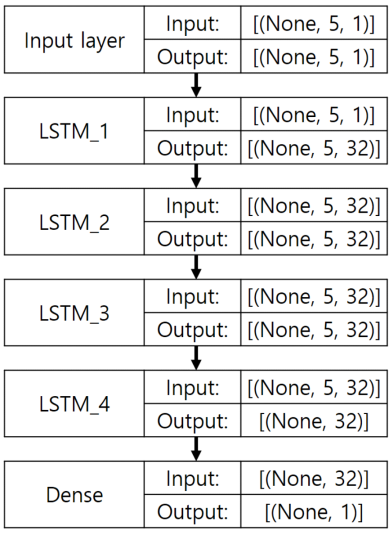

그림 6. LSTM 모델 입출력 구조
Fig. 6. LSTM model input-output architecture

용된 GPU는 RTX 4060 Ti이다. 그림 6와 표 6은 LSTM 기반 네트워크 대역폭 예측 모델을 구성하는 중간 계층구조를 설명한다. 표 7의 Input Size는 입력층의 노드 수, Hidden Size는 은닉층의 노드 수, Layer는 모델에서 사용된 은닉층의 층 수, Output Size는 출력층의 노드 수를 각각 의미한다. 본 연구에서 사용된 LSTM 기반 대역폭 예측 모델은 총 4개의 LSTM 레이어와 1개의 출력층으로 구성되며, 입력 데이터는 일정 길이의 연속된 시계열 값을 포함하는 다차원 형태로 주어진다. 첫 번째 LSTM 레이어는 입력 데이터를 고차원 특징 공간으로 변환하며, 이후의 LSTM 레이어들은 시계열 데이터의 시간적 관계를 학습하여 점진적으로 중요한 패턴을 추출한다. 마지막 LSTM 레이어에서는 전체 시퀀스 중 가장 중요한 정보가 반영된 상태를 출력하며,

표 7. LSTM 기반 모델 아키텍처
Table 7. LSTM based Model Architecture

Input Size	1
Hidden Size	32
Layer	4
Output Size	1

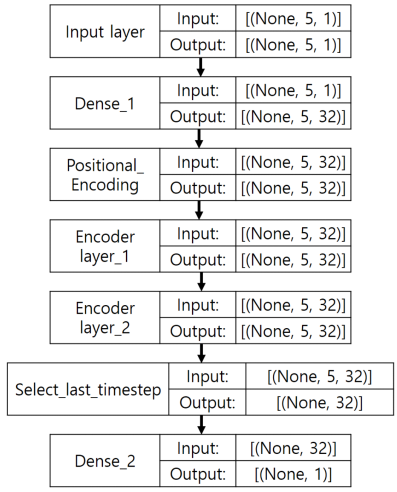

그림 7. Transformer Encoder 모델 입출력 구조
Fig. 7. Transformer Encoder model input-output architecture

표 8. Transformer 모델 아키텍처
Table 8. Transformer Model Architecture

Embedding Dimesion	32
Number of Heads	2
Layers	2
Feedforward Dimension	64
Dropout Rate	0.1

이를 활용하여 최종 예측을 수행한다. 최종 출력층은 LSTM의 결과를 하나의 예측값으로 변환하는 역할을 한다. 그림 7과 표 8는 Transformer 기반 시계열 예측 모델의 구조를 설명한다. Embedding Dimension은 입력 데이터를 고차원 공간으로 변환하는 과정에서 생성되는 벡터의 차원을 의미하며, Number of Heads는 Multi-head Attention에서 사용하는 병렬 어텐션 연산의 개수를 의미한다. Layers는 Transformer 인코더의 계층 수를 의미한다. Feedforward Dimension은 Transformer Layer 내의 피드포워드 네트워크의 중간 차원을 의미한다. Dropout Rate는 학습 과정에서 일부 뉴런을 해당 확률로 비활성화하여 과적합을 방지하는 데 사용된다. 본 연구에서 사용된 Transformer 기반 시

표 9. 모델 하이퍼파라미터
Table 9. Hyperparameter for model

Learning Rate	0.01
Batch Size	12
Sequence Length	5
Epoch	100

계열 예측 모델의 입력은 일정 길이의 연속된 시계열 데이터이며, 이를 고차원 특징 공간으로 변환한 후, 시계열의 순서 정보를 보완하기 위해 위치 인코딩을 추가한다. 변환된 입력은 다중 층의 Transformer 인코더를 통해 처리되며, 두 개의 인코더 층에서는 시계열 내 각 시점 간의 관계를 학습하여 중요한 패턴을 추출한다. 모델의 최종 출력은 전체 시퀀스 중 마지막 시점의 특징을 선택하여 예측층으로 전달되며, 이를 활용하여 미래 대역폭 값을 추정한다.

모델별 하이퍼파라미터의 설정값은 표 9과 같으며, Learning Rate는 학습률로, 모델이 학습할 때 가중치를 조정하는 속도를 나타낸다. Batch Size는 한 번의 학습 단계에서 모델이 처리하는 데이터 샘플의 수를 의미한다. Sequence Length는 모델이 한 번에 처리하는 시퀀스 데이터의 길이를 나타낸다. 이 길이를 실험 프레임워크 내 구현된 기존 MPC에서 대역폭 예측 시 고려하는 이동평균 범위 크기와 동일한 5로 설정하였다. Epoch는 전체 데이터셋이 모델에 한 번 완전히 학습된 횟수를 의미한다.

3.3 MPC ABR 알고리즘 시뮬레이션

시뮬레이션 실험 환경의 경우, ABR 알고리즘 실험 프레임워크를 활용하였으며[3]. 클라이언트와 서버 사이에 80ms의 RTT(Round Trip Time)를 적용하였다. 미래 시점의 5개 청크에 대해 MPC 최적화 알고리즘을 수행하였다.

3.4 딥러닝 기반 네트워크 대역폭 추정 알고리즘

먼저, 기존에 처리량(대역폭) 추정을 위해 사용된 알고리즘인 조화평균을 이용한 이동평균 로직을 삭제하였다. 조화평균은 식 (4)에서와 같이 각 값의 역수를 평균내어 그 결과의 역수를 구하는 방식으로 계산되는데, MPC 이전 연구에서는 이러한 조화평균을 이용한 방식이 청크별 추정치에서 발생할 수 있는 이상치에 강인하다고 설명한다[12].

$$\text{Harmornic mean} = \frac{n}{\sum_{i=1}^{n}\frac{1}{x_i}} \tag{4}$$

반면, 본 논문에서 제안하는 기법에서는 기존의 조화평균 이동평균 방식을 사용하지 않고, 대역폭 데이터를 학습한 딥러닝 모델이 예측한 대역폭 값을 바탕으로 다음 번 청크에 소요될 시간을 예측한다. 직전에 측정된 5개의 대역폭 값을 딥러닝 기반 예측 모델의 입력으로 각각 사용하였으며. 5개 값을 측정하지 못한 첫 5개 청크 구간에서는 기존 MPC 알고리즘과 동일하게 조화평균 기반의 이동평균으로 최적화를 수행하였다. 식 (5)는 본 논문이 제안하는 네트워크 대역폭 추정 알고리즘을 설명한다.

$$\overline{C_k} = \begin{cases} \dfrac{n}{\sum_{i=1}^{n}\frac{1}{R_{k-i}}} & (k \leq 5) \\ DLmodel.predict(C_{k-5}, C_{k-4}, C_{k-3}, C_{k-2}, C_{k-1}) & (k > 5) \end{cases} \tag{5}$$

Ⅳ. 성능 평가

해당 장에서는 본 논문이 제안하는 네트워크 대역폭 예측 모델의 정확도를 측정 및 평가하고, 기존 방법과 딥러닝 모델의 예측값을 활용한 제안 방법의 ABR 시뮬레이션에서 QoE를 비교 및 평가한다.

4.1 네트워크 대역폭 예측 모델 평가

3.3장에서 설명한 Cork 대학의 네트워크 트레이스를 기반으로 MPC 알고리즘을 수행하여 네 가지 이동 패턴에 대해 각각 18,612개/7,524개/3,970개/8514개의 대역폭 데이터를 수집하였다. 학습 데이터와 시험 데이터를 각각 7:3의 비율로 할당하였으며, 성능 지표로는 RMSE(Root Mean Square Error)와 MAE(Mean Absolute Error)를 각각 사용하였다.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{7}$$

식 (6), (7)에서 y_i는 실제값, $\hat{y}_i$는 예측값, n은 데이터의 총 개수를 의미한다. RMSE는 각 데이터의 예측 오차를 제곱하여 평균을 구한 후, 그 결과에 제곱근을 취한 값

이다. 이는 오차가 클수록 그 영향력이 커지기 때문에, 큰 오차에 민감하게 반응하는 지표이며 예측 모델이 극단적인 오차를 얼마나 자주 발생시키는지 평가하는 데 유용하다. MAE는 오차의 절댓값을 사용하여 평균을 구한 값이다. 이는 RMSE와 달리, 큰 오차에 민감하지 않다는 특징을 가지며, 이를 통해 전반적인 예측 오차의 크기를 직관적으로 파악하고자 하였다.

4.2 시나리오별 딥러닝 모델 예측 성능

그림 8~10과 그림 11~13은 각각 자동차로 이동한 시나리오, 보행 시나리오, 정지 실내 시나리오에서 LSTM과 Transformer 기반 네트워크 대역폭 예측 모델의 예측 정확도를 평가한 결과를 나타낸다. 그림 14, 15는 5G 대역폭을 포함하는 자동차 시나리오에서 각 모델의 예측 정확도 평가 결과이다. 파란색 그래프는 실제 대역폭이며, 주황색 그래프는 예측값을 의미한다. 각 모델의 예측 정확도 지표를 표 10과 같이 정리하였다.

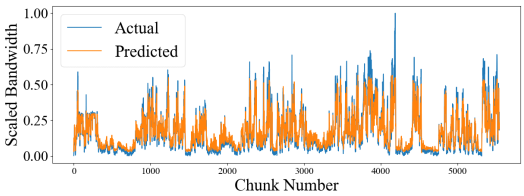

그림 8. 자동차 시나리오에서 LSTM 모델의 네트워크 LTE 대역폭 예측
Fig. 8. LTE bandwidth prediction of LSTM model in car scenario

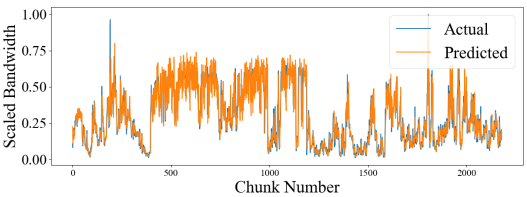

그림 9. 도보 시나리오에서 LSTM 모델의 LTE 대역폭 예측
Fig. 9. LTE bandwidth prediction of LSTM model in pedestrian scenario

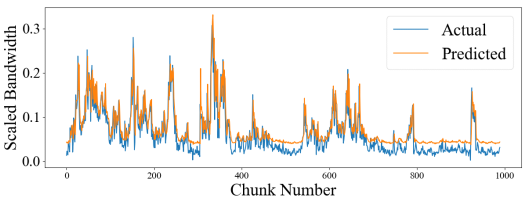

그림 10. 정지 시나리오에서 LSTM 모델의 LTE 대역폭 예측
Fig. 10. LTE bandwidth prediction of LSTM model in static scenario

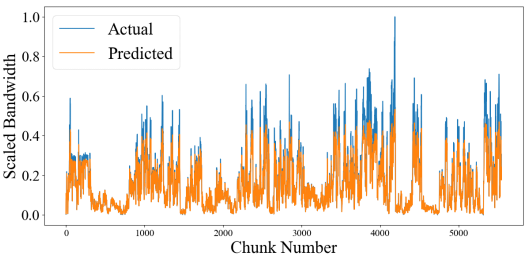

그림 11. 자동차 시나리오에서 Transformer 모델의 LTE 대역폭 예측
Fig. 11. LTE bandwidth prediction of Transformer model in car scenario

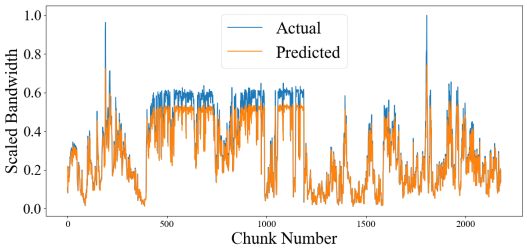

그림 12. 도보 시나리오에서 Transformer 모델의 LTE 대역폭 예측
Fig. 12. LTE bandwidth prediction of Transformer model in pedestrian scenario

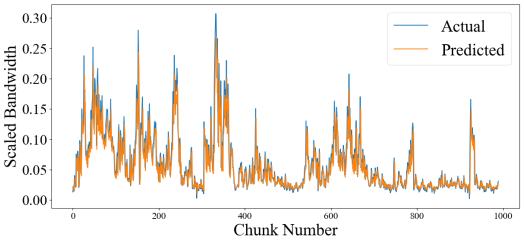

그림 13. 정지 시나리오에서 Transformer 모델의 LTE 대역폭 예측
Fig. 13. LTE bandwidth prediction of Transformer model in static scenario

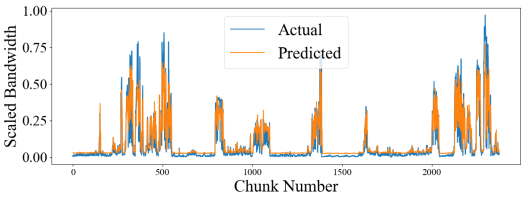

그림 14. 자동차 시나리오에서 LSTM 모델의 5G 대역폭 예측
Fig. 14. 5G bandwidth prediction of LSTM model in car scenario

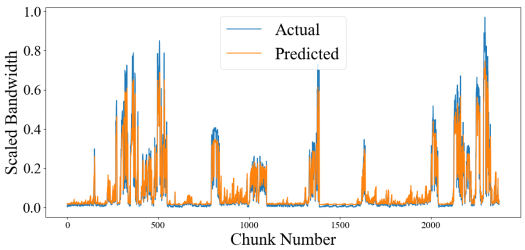

그림 15. 자동차 시나리오에서 Transformer 모델의 5G 대역폭 예측
Fig. 15. 5G bandwidth prediction of transformer model in car scenario

표 10. 시나리오 및 예측 모델 별 예측 정확도
Table 10. Prediction accuracy by scenario and prediction model

Scenario	LSTM		Transformer	
	MAE	RMSE	MAE	RMSE
Car	0.04	0.07	0.04	0.06
Pedestrian	0.02	0.03	0.05	0.07
Static	0.022	0.029	0.01	0.02
5G	0.05	0.08	0.03	0.08

4.3 딥러닝 활용 MPC의 ABR 성능 평가

제안하는 딥러닝 기반 MPC 알고리즘(MPC+)의 ABR 성능을 평가하기 위해서 학습에 사용하지 않은 30%의 트레이스로부터 자동차, 도보, 실내 시나리오 각각 28회, 11회, 5회분의 트레이스 파일을 생성하여 시뮬레이션 결과를 비교하였다. 식 (1)에서 QoE 관련 파라미터를 표 11과 같이 정리하였다. 그림 16~19는 (1)의 QoE 식과 표 11의 파라미터를 이용하여 계산한 시나리오별 QoE 비교 수행 결과를 각각 나타낸다. 파라미터는 MPC 알고리즘과 기존 연구에서 사용한 디폴트 값을 활용하였다[2,13]. 그림 20은 네 가지 시나리오별 전체 평균 QoE를 비교한 결과를 나타낸다. 기존 Robust MPC의 QoE를 파란색 막대로, MPC+(LSTM)의 QoE

표 11. QoE 계산을 위한 파라미터
Table 11. Parameters for QoE Calculation

parameter	value	unit
K	198	
$q(R_k)$	R_k	Mbps
λ	1	
μ	4.3	
$\left(\frac{d_k(R_k)}{C_k} - B_k\right)$		sec

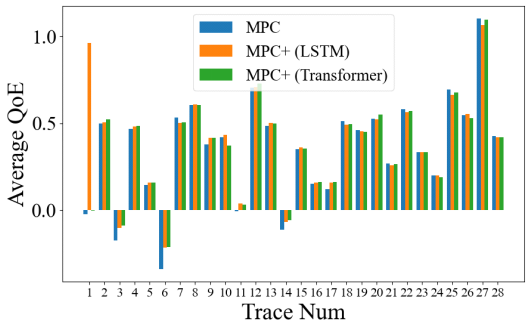

그림 16. 자동차 LTE 시나리오에서 QoE 비교
Fig. 16. QoE comparison in car LTE scenario

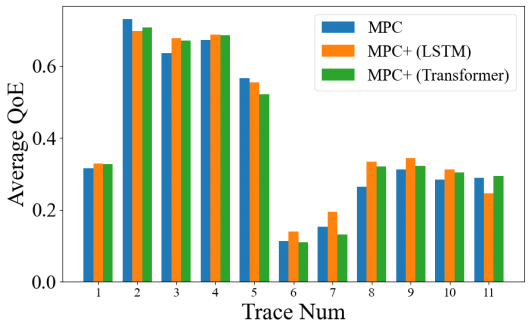

그림 17. 도보 LTE 시나리오에서 QoE 비교
Fig. 17. QoE comparison in pedestrian LTE scenario

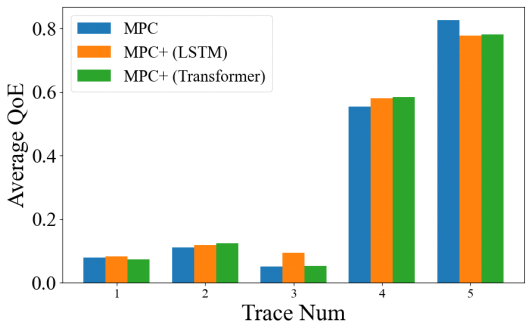

그림 18. 정지 LTE 시나리오에서 QoE 비교
Fig. 18. QoE comparison in static LTE scenario

를 주황색 막대로, MPC+(Transformer)의 QoE를 초록색 막대로 각각 나타내었다. LSTM 모델의 경우, 자동차로 이동한 LTE 트레이스 시나리오에서 실험한 28회 시뮬레이션 중 18회 시뮬레이션에서 QoE가 증가하고 전체 평균 13.0% 증가하였다. Transformer 모델의 경우, 자동차로 이동한 LTE 트레이스 시나리오에서 실험한 28회 시뮬레이션 중 16회 시뮬레이션에서 QoE가 증가하고 전체 평균 3.62% 증가하였다. 3번, 6번, 14번의 경우, QoE가 세 방법 모두에서 낮게 계산되었는데,

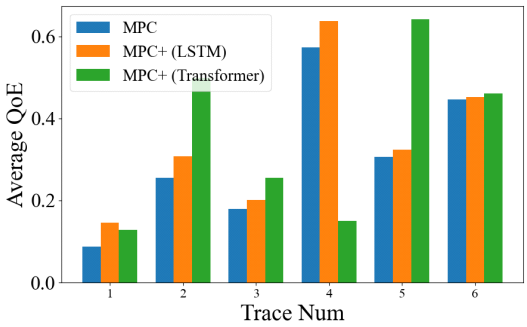

그림 19. 자동차 5G 시나리오에서 QoE 비교
Fig. 19. QoE comparison in car 5G scenario

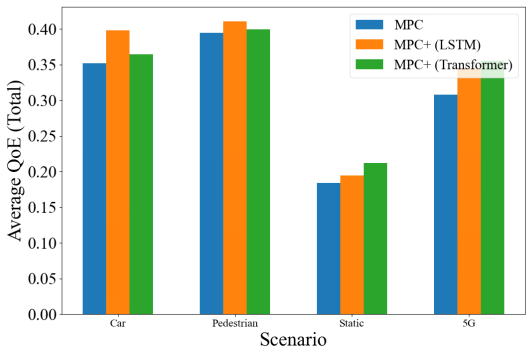

그림 20. 각 시나리오 별 평균 QoE 비교
Fig. 20. Comparison of average QoE for each scenario

무선 네트워크 접속이 불가능한 구간의 트레이스로 실험한 경우로, 두 기법 모두 현저히 낮은 QoE가 측정되었다. 표 12과 같이, 각 QoE 측정 요소에서, 화질은 비슷하게 나타나면서도 리버퍼링이 유 의미하게 감소하는 것으로 나타났다. 화질 변동의 경우 LSTM 모델을 활용한 MPC+에서 10.36% 증가하였다.

도보로 이동한 시나리오에서 실험한 LSTM MPC+의 11회 시뮬레이션 중 8회 시뮬레이션에서 QoE가 증가하고 전체 평균 4.11% 증가하였다. Transformer 기반 모델은 11회 시뮬레이션 중 7회 시뮬레이션에서 QoE가 증가하고 전체 평균 1.31% 증가하였다. 표 13와 같이, 각 QoE 측정 요소에서 화질은 비슷하게 나면서도 리버퍼링이 유의미하게 감소하는 것으로 나타났다. 화질 변동의 경우 LSTM 모델을 활용한 MPC+에서 5.85% 낮게 나타났다.

실내에서 정지한 시나리오에서 실험한 LSTM 기반 모델과 Transformer 기반 모델 모두 5회 시뮬레이션 중 4회 시뮬레이션에서 QoE가 증가하고 각각 전체 평균 5.86%, 15.44% 증가하였다. 표 14와 같이, 각 QoE 측정 요소에서 화질은 비슷하게 나타나면서도 리버퍼

표 12. 자동차 시나리오 - 세부 성능 측정값
Table 12. Car Scenario - Detailed Performance Metrics

QoE element	MPC	MPC+ (LSTM)	MPC+ (Transformer)
Average Bitrate	2409.08 Kbps	2498.87 Kbps	2391.36 Kbps
Average Bitrate Change	258.19 Kbps	284.94 Kbps	282.69 Kbps
Rebuffering time	19.26 sec	11.84 sec	14.40 sec

표 13. 도보 시나리오 - 세부 성능 측정값
Table 13. Pedestrian Scenario - Detailed Performance Metrics

QoE element	MPC	MPC+ (LSTM)	MPC+ (Transformer)
Average Bitrate	2360.21 Kbps	2334.95 Kbps	2339.04 Kbps
Average Bitrate Change	273.81 Kbps	257.77 Kbps	287.56 Kbps
Rebuffering time	7.31 sec	3.17 sec	4.52 sec

표 14. 정지 시나리오 - 세부 성능 측정값
Table 14. Static Scenario - Detailed Performance Metrics

QoE element	MPC	MPC+ (LSTM)	MPC+ (Transformer)
Average Bitrate	1472.27 Kbps	1472.23 Kbps	1445.23 Kbps
Average Bitrate Change	142.69 Kbps	114.52 Kbps	140.78 Kbps
Rebuffering time	21.77 sec	20.59 sec	14.11 sec

표 15. 5G 시나리오 - 세부 성능 측정값
Table 15. 5G Scenario - Detailed Performance Metrics

QoE element	MPC	MPC+ (LSTM)	MPC+ (Transformer)
Average Bitrate	1798.26 Kbps	2245.46 Kbps	2226.59 Kbps
Average Bitrate Change	192.45 Kbps	269.66 Kbps	234.49 Kbps
Rebuffering time	32.20 sec	15.54 sec	41.35 sec

링이 각각 5.42%, 35.18% 감소하였다.

자동차 주행중 측정된 5G 시나리오에서 표 15와 같이, LSTM 모델은 6회 시뮬레이션 중 6회 시뮬레이션에서 QoE가 증가하고 전체 평균 11.88% 증가하였다. Trasformer 기반 모델은 6회 시뮬레이션 중 5회 시뮬레이션에서 QoE가 증가하고 전체 평균 15.32% 증가하였다. LSTM에서 비트레이트는 약 24.86% 증가, 화질 변동량은 약 40.1% 감소, 리버퍼링 시간은 약 51% 감소하였으며, Transformer 모델에서 비트레이트는 약 23.81% 증가, 화질 변동량은 약 21.8% 감소, 리버퍼링 시간은 약 28% 각각 증가하였다.

V. 결 론

본 논문에서는 최신 ABR 알고리즘 중 MPC 알고리즘의 대역폭 추정 알고리즘에 주목하여 LSTM, Transformer 기반 모델을 활용한 개선점을 제안하였다. ABR 알고리즘 실험 프레임워크를 활용하여 시뮬레이션한 결과 제안한 MPC+ 알고리즘이 평균 화질(Mbps), 리버퍼링 시간(초), 화질 변동량 평균(Mbps)으로 계산된 QoE 지표에서 3가지 서로 다른 이동 패턴 및 5G 시나리오에서 측정된 네트워크 트레이스로부터 모두 향상된 결과를 보였으며, LSTM은 평균 8.71%, Transformer는 평균 8.92% 향상되었다. 이는 변동성이 높은 무선 환경에서 기존의 이동평균 방식보다 딥러닝을 활용한 추정이 화질 최적화에 있어 이점이 있다고 판단된다. 제안한 기법의 실질적인 성과를 평가하기 위해서 향후 연구를 통해 DASH(Dynamic Adaptive Streaming over HTTP) 기반의 실시간 테스트베드에서 성능을 측정할 계획이다.

References

[1] OTT Video Advertising - Worldwide(n.d.). Retrieved November 04, 2024, from https://www.statista.com/outlook/amo/media/tv-video/ott-video/ott-video-advertising/worldwide

[2] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A control-theoretic approach for dynamic adaptive video streaming over http,” *ACM SIGCOMM Comput. Commun. Rev.*, pp. 325-338, 2015. (https://doi.org/10.1145/2785956.2787486)

[3] H. Mao, R. Netravali, and M. Alizadeh, “Neural adaptive video streaming with pensieve,” in *Proc. Conf. ACM Special Interest Group on Data Commun. (SIGCOMM ’17),* pp. 197-210, Association for Computing Machinery, New York, NY, USA, 2017. (https://doi.org/10.1145/3098822.3098843)

[4] J.-W. Lee, E. Hyun, S. Kim, and J.-Y. Jung, “A performance improvement method of adaptive bitrate algorithm in video streaming system,” in *Proc. Conf. The Korean Inst. Broadcast and Media Eng.*, Seoul, 2022.

[5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, 1997. (https://doi.org/10.1162/neco.1997.9.8.1735)

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. 31st Int. Conf. Neural Inf. Proc. Syst.* (NIPS’17), pp. 6000-6010, Curran Associates Inc., Red Hook, NY, USA, 2017.

[7] M. Badawy, N. Ramadan, and H. A. Hefny, “A survey on deep learning techniques for predictive analytics in healthcare,” *SN Conput. Sci.*, vol. 5, p. 860, 2024. (https://doi.org/10.1007/s42979-024-03188-3)

[8] J. Guijarro-Ordonez, M. Pelger, and G. Zanotti, “Deep learning statistical arbitrage,” Mar. 2019. (https://doi.org/10.2139/ssrn.3862004)

[9] K. S. Nithin and R. H. Mulangi, “Development and comparison of deep learning and statistical models to predict bus passenger flow,” in *Sivakumar Babu, G. L., Mulangi, R. H., Kolathayar, S. (eds) Technol. Sustainable Trans. Infrastructures, SIIOC 2023*, Lecture Notes in Civil Engineering, vol. 529. Springer, Singapore, 2024. (https://doi.org/10.1007/978-981-97-4852-5_25)

[10] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenanm, “Beyond throughput: A 4G LTE dataset with channel and context metrics,” in *Proc. ACM MMSys 2018*, pp. 12-15, Amsterdam, The Netherlands, Jun. 2018.

[11] K. Spiteri, R. Sitaraman, and D. Sparacio, “From theory to practice: Improving bitrate

adaptation in the DASH reference player," in *Proc. 9th ACM MMSys 2018,* pp. 123-137, Association for Computing Machinery, New York, NY, USA, 2018. (https://doi.org/10.1145/3204949.3204953)

[12] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," in *Proc 8th Int. Conf. Emerging Netw. Experiments and Technol. (CoNEXT '12)*, pp. 97-108, Association for Computing Machinery, New York, NY, USA, 2012. (https://doi.org/10.1145/2413176.2413189)

[13] P. G. Pereira, A. Schmidt, and T. Herfet, "Cross-layer effects on training neural algorithms for video streaming," in *Proc. 28th ACM SIGMM Wkshp. Netw. and Operating Syst. Support for Digital Audio and Video (NOSSDAV '18)*, pp. 43-48, Association for Computing Machinery, New York, NY, USA, 2018. (https://doi.org/10.1145/3210445.3210453)

[14] J. Ye, M. Dan, and W. Jiang, "A visual sensitivity aware ABR algorithm for DASH via deep reinforcement learning," *ACM Trans. Multimedia Comput. Commun. Appl.,* vol. 20, 3, Article 77 (Mar. 2024), p. 22, 2023. (https://doi.org/10.1145/3591108)

[15] Z. Zhang, et al., "Anableps: Adapting bitrate for real-time communication using VBR-encoded video," *2023 IEEE ICME*, IEEE, 2023.

[16] M. Liu, et al., "EVAN: Evolutional video streaming adaptation via neural representation," *2024 IEEE ICME*, IEEE, 2024.

[17] H. Zhou, et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 35, no. 12, 2021.

[18] B. Lim, et al., "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748-1764, 2021.

[19] G. Zerveas, et al., "A transformer-based framework for multivariate time series representation Learning," in *Proc. 27th ACM SIGKDD Conf. on Knowledge Discovery & Data Mining*, 2021.

[20] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: a 5g dataset with channel and context metrics," in *Proc. 11th ACM Multimedia Syst. Conf.*, pp. 303-308, Istanbul, Turkey, 2020.

[21] UCC MISL 5G dataset, https://github.com/uccmisl/5Gdataset

문 이 빈 (Ie-bin Moon)

2024년 2월 : 국립창원대학교 컴퓨터공학과 학사

2024년 3월~현재 : 국립창원대학교 컴퓨터공학과 석사과정

<관심분야> 컴퓨터네트워크, 딥러닝

[ORCID:0009-0007-0786-5209]

안 동 혁 (Donghyeok An)

2006년 2월 : 한동대학교 전산전자공학부 학사

2013년 2월 : KAIST 전산학과 박사

2023년 3월~2014년 2월 : 성균관대학교 박사후연구원

2014년 3월~2015년 2월 : 삼성전자 책임연구원

2015년 3월~2017년 8월 : 계명대학교 컴퓨터공학과

2017년 9월~현재 : 국립창원대학교 컴퓨터공학과 부교수

<관심분야> 5G 및 6G, 저지연 통신, 딥러닝

[ORCID:0000-0001-6703-9311]