XAI 기반 2025년도 KTX 수요 예측 및 정책적 의사결정

차명주a, 오영택b, 이승연b, 김경원a,*∗*

*a인천대학교 글로벌정경대학 무역학부*

*b한국철도공사 철도연구원*

# Abstract

고속철도는 전기차보다 12배, 내연기관 차량보다 26배 낮은 CO2 배출량을 기록하며, 지속 가능한 교통수단의 대표 사례로 자리 잡고 있다. 고속철도의 수요 변화를 정밀하게 예측하는 것은 단순한 운영 효율성 개선을 넘어, 장기적인 교통 정책 수립과 지속 가능한 인프라 구축을 위한 핵심 요소로 작용한다. 본 연구는 고성능 인공지능 알고리즘과 설명 가능한 인공지능(XAI, Explainable AI)을 활용하여 2025년도 KTX 수요를 정밀하게 예측하고, 예측 결과를 바탕으로 실질적인 비즈니스 및 정책적 의사결정을 지원하는 것을 주된 목적으로 한다. 대표적인 AI 알고리즘을 활용하여 KTX 수요 예측의 오류를 최대 2.49%까지 낮추며 정확도를 대폭 향상시켰다. 또한, 대표적 XAI 기술인 SHAP 알고리즘을 활용하여 예측 결과와 변수들의 기여 정도를 시각화 함으로써, 비즈니스적 정책 설계 및 자원 배분 의사결정 과정에 신뢰도를 높였다. KTX 노선에 따라 기여하는 변수들은 다양하게 변화될 수 있는 가능성을 제시하며 실시간 활용가능한 비즈니스 애널리틱스의 방법론의 필요성을 확인하였다. 최종적으로 2025년도 KTX 수요는 작년보다는 최대 9.45% 정도 감소할 수 있지만, 코로나 이전보다 최대 18.47%까지 상승하는 수치로 수요가 점진적으로 증가되며 안정화 될 것으로 예상된다.

*Keywords:* Demand Forecasting of KTX, Explainable Prediction, Machine and Deep Learning, SHAP, Decision Support

*∗*Corresponding author: thekimk.kr@gmail.com

*ORCID(s):* 0000-0001-6530-8426 (K. Kim)

# Introduction

고속철도는 전 세계 교통 시스템에서 지속 가능한 발전의 핵심 축으로 자리 잡고 있다. 프랑스의 TGV, 일본의 신칸센, 중국의 고속철도와 같은 사례는 고속철도가 국가 경제와 사회적 연결성 강화에 미치는 긍정적 영향을 잘 보여준다. 이들 사례는 고속철도가 단순히 국가 내 이동 수단에 국한되지 않고, 경제적 연결성과 국제적 경쟁력을 강화하는 주요 인프라로 기능할 수 있음을 입증한다. 현재 25개국 이상이 고속철도를 도입하여 주요 교통수단으로 활용하고 있으며, 이러한 국가는 고속철도를 통해 물류와 인구의 이동성을 극대화하며 교통체계의 효율성과 환경적 지속 가능성을 동시에 달성하고 있다. 특히, 고속철도는 전기차보다 12배, 내연기관 차량보다 26배 낮은 CO2 배출량을 기록하며, 지속 가능한 교통수단의 대표 사례로 자리 잡고 있다 (Union, 2023).

글로벌 고속철도의 발전 속에서 KTX는 한국의 대표적인 고속철도 시스템으로 자리 잡았다. 2004년 개통 이후 교통 편리성과 시간 절약이라는 두 가지 핵심 가치를 통해 빠르게 성장해 왔다. 2022년 기준 약 7,500만 명의 이용객을 기록한 이후 지속적인 증가세를 보이고 있다. 2024년에는 연간 이용객 수가 총 1억 1,658만 명으로 전년 대비 5.4% 증가하며, KTX가 단순한 교통수단을 넘어 국가 기반 교통망의 중심축으로 기능하고 있음을 입증하고 있다 (국토교통부, 2024). 그러나 이용객 증가에 따른 운영상의 도전 과제도 점차 복잡해지고 있다.

특히, 최근 화제가 된 ‘수서역 KTX’ 논의에서도 확인할 수 있듯, 특정 시간대와 주요 노선에서 발생하는 혼잡 문제와 자원 배분의 불균형이 운영 효율성에 중요한 영향을 미치고 있다. 이러한 문제를 효과적으로 해결하기 위해서는 수요 변화를 사전에 예측하고, 이를 기반으로 최적화된 운영 전략을 수립하는 것이 필수적이다. 나아가, KTX 교통 분야의 수요 예측은 단순한 운영 최적화 차원을 넘어, 정책적 의사결정을 지원하는 중요한 도구로 기능할 수 있으며, 교통 인프라의 지속 가능한 발전에도 기여할 수 있다. 이에 따라, 국토교통부는 미래 교통 시스템의 효율성을 제고하기 위해 수요 예측과 연계된 미래 선도 기술 개발을 적극적으로 추진하고 있다 (기술과가치, 2020). 이러한 맥락에서, 고속철도의 수요 변화를 정밀하게 예측하는 것은 단순한 운영 효율성 개선을 넘어, 장기적인 교통 정책 수립과 지속 가능한 인프라 구축을 위한 핵심 요소로 작용한다. 그러나 효과적인 수요 예측을 위해서는 고속철도 이용 패턴을 정확히 반영할 수 있는 정교한 분석 기법이 필수적이며, 이에 대한 연구는 지속적으로 이루어져 왔다.

고속철도 수요 예측은 주로 전통적인 통계적 접근에 기반한 시계열 분석 기법을 활용해 이루어져 왔다. (Kim과 Kim, 2011) 는 개입 ARIMA 모형을 사용하여 경부고속철도 2단계 개통과 같은 정책적 개입이 수요에 미치는 영향을 분석하였다. 이 연구는 정책 변화에 따른 단기적 수요 변화를 효과적으로 포착했지만, 열차 운임, 운행시간, 서비스 품질 등 다양한 변수 간의 상호작용을 반영하지 못했고, 데이터의 선형적 구조를 가정하여 설명력에 한계를 드러냈다. 이러한 한계는 고속철도 수요가 시간적, 공간적으로 복잡한 비선형적 특성을 가지며, 다양한 외부 요인의 영향을 받는다는 점에서 더욱 두드러진다. (Cha et al., 2019) 은 다중 개입 계절형 ARIMA 모형을 활용하여 고속철도 개통이나 국가 전염병과 같은 외부 환경변화를 반영한 수요 예측을 수행하였다. 계절적 요인과 외부 충격을 함께 고려해 이전보다 개선된 결과를 도출했지만, 전통적인 시계열 분석 기법이 가진 구조적 한계에서 벗어나지 못했다. 이러한 방법론은 데이터가 선형적 관계를 따르며, 정상성(stationarity)을 만족해야 한다는 전제 조건을 가지므로 현실 세계의 복잡한 데이터 특성을 충분히 반영하지 못하고 정보 손실 가능성을 내포하고 있다. 결과적으로, ARIMA와 같은 시계열 분석 기법은 과거 데이터를 기반으로 수요 예측에 널리 활용되어 왔으나, 선형적 관계를 기반으로 한다는 점에서 현실 세계의 복잡하고 비선형적인 수요 변화를 효과적으로 예측하기에는 한계가 있다(Sima et al., 2018). 이러한 전통적 통계 기반 기법은 대규모 데이터 처리와 실시간 분석이 요구되는 현대 고속철도 수요 예측 문제에서 특히 두드러진 한계를 보인다.

머신러닝 및 딥러닝과 같은 고성능 인공지능 알고리즘은 데이터 내에 내재된 복잡한 패턴을 학습하고, 변수 간의 관계를 자동으로 탐지하며, 대규모 데이터를 처리하는 데 강점을 지닌다. 특히, 인공지능 기반 수요 예측 기법은 높은 예측 정확도와 계산 효율성을 제공하며, 비선형 데이터 패턴의 학습과 다양한 변수 간의 관계 탐지가 가능하다는 점에서 기존 통계적 접근법의 한계를 효과적으로 보완한다 (Jeong과 Lim, 2019). 또한 최근에는 머신러닝 및 딥러닝을 활용하여 고속철도 수요 예측에서 더 높은 정확도를 달성하고, 효율적인 운영 전략 수립에 기여할 가능성을 제시해왔다. 예를 들어, LSTM(Long Short-Term Memory)과 XGBoost를 적용한 연구에서는 비선형 데이터 패턴을 효과적으로 학습하고, 기존 통계 기반 모델보다 더 높은 예측 성능을 보임으로써 알고리즘의 실효성을 입증하였다 (심진호 et al., 2024). 하지만 인공지능 알고리즘의 발전에도 불구하고 고속철도 수요예측에 활용되는 사례나 연구는 아직 부족한 실정이다.

인공지능의 알고리즘의 높은 성능에도 불구하고 구조가 매우 복잡하여 왜 그러한 결과가 도출되었는지 설명하지 못하는 한계가 있다. 이를 “블랙박스” 이슈라고도 하며 의사결정 과정에서 예측 결과의 신뢰성을 낮추고, 정책 설계나 자원 배분과 같은 실제 활용에 제약을 초래할 수 있다. 이러한 한계는 특히, 교통망 운영과 같은 대규모 인프라에서 심각한 문제로 작용할 수 있다. 이와 같은 문제를 해결하기 위해 최근에는 설명 가능한 인공지능(XAI, Explainable AI)이 주목받고 있다. 설명 가능한 인공지능(XAI)는 인공지능 시스템이 수행하는 예측 및 의사결정 과정을 인간이 이해할 수 있도록 설명하는 기술로, AI 시스템의 행동과 상태를 명확히 전달하여 신뢰성을 높이는 것을 목표로 한다 (David et al., 2019). 이 기술은 금융, 의료, 제조 등 다양한 분야에서 활용 사례를 통해 그 중요성을 입증하고 있다. 예를 들어, 금융 분야에서는 SHAP(Shapley Additive Explanations)와 LIME(Local Interpretable Model-agnostic Explanations) 같은 도구를 활용하여 신용등급 평가와 대출 의사결정에서 예측 결과의 해석 가능성을 높이고, 투명한 의사결정을 지원한 사례가 있다 (Kwon, 2023). 의료 분야에서는 딥러닝 기반의 무릎 골관절염 진단 모델에 XAI를 적용하여 진단 근거를 명확히 제시함으로써 의료진의 신뢰를 확보하고 진단 정확도를 향상시킨 사례가 있다 (Rafique과 Ali, 2024). 또한, 제조 분야에서는 XAI를 통해 수주량 변화의 주요 요인을 분석하고, 이를 기반으로 자원 배분 및 운영 최적화를 실현하여 비용 절감과 생산성 향상에 기여한 사례가 있다 (Jung과 Cheolhee, 2024).

이러한 연구사례들은 XAI가 단순히 예측 결과를 제공하는 데 그치지 않고, 그 결과를 해석하고 시각화함으로써 다양한 산업에서 실질적인 의사결정을 지원하는 도구로 자리 잡고 있음을 보여준다. XAI의 이러한 특성은 고속철도 수요 예측과 같은 대규모 교통 인프라 운영에서도 중요한 기여를 할 수 있다. 특히, KTX와 같은 고속철도망은 수요 변동성이 높고, 다양한 외부 요인의 영향을 받는 복잡한 시스템이므로, 예측 결과에 대한 명확한 해석과 신뢰성이 확보되지 않는다면, 정책 설계와 자원 배분의 효과성이 크게 저하될 수 있다. 따라서 XAI를 활용하여 AI 기반 수요 예측 결과를 해석 가능하게 만들고, 이를 시각화하여 운영 전략 수립과 정책 결정 과정에서 보다 신뢰도 높은 의사결정을 지원할 수 있다.

이후의 내용은 한국철도공사가 제공한 데이터와 전처리 과정을 소개하고, 승차인원수 예측을 위해 사용된 AI와 XAI 알고리즘의 소개, 그리고 마지막으로 2025년도 예측에 대한 연구결과를 제시하며 마무리한다.

# Methods

* 1. *Data Preprocessing*

본 연구에서는 경부선, 경전선, 동해선, 전라선, 호남선 총 5개의 월별 승차인원수 요를 예측하는 것이 목적이다. 한국철도공사 철도연구원으로부터 제공받은 2015년 1월부터 2024년 3월까지 약 10년간의 “수송-운행일-주운행” 그리고 “시종착역별 열차운행” 정보가 담긴 데이터베이스를 결합하여 2025년 12월까지의 월별 KTX 수송수요를 예측하는데 활용하였다. 그리고 수요예측에 도움이 될 수 있는 다양한 파생 변수들을 생성하였다. 첫째로, 과거의 수요가 현재 또는 미래의 수요에 영향을 줄 수 있기 때문에 과거 1개월~12개월 전의 수요를 “과거 승차인원수” 파생변수로 생성하였다. 둘째로, 버스나 자하철과 같은 대중교통과 달리 요일이나 이벤트에 따라 수요의 변화가 느리게 발생하기 때문에, 시계열데이터에서 각 월의 실제 날짜수, 주말수, 주중수, 공유일수, 명절 수 등의 “시간정보” 파생변수를 결합하였다. 셋째로, 대외적인 경제상황과 환경변화를 반영하기 위해서 한국의 주식 시장 지표와 소비자의 물가수준, 그리고 코로나 시기의 예방접종 인원수, 격리자수, 사망자수, 정부대응 지수 등의 “외부환경” 파생변수를 반영하여 정교함을 높였다. 마지막으로 한국철도공사에서 제공받은 공급좌석 정보, 열차 정보, 운행정보 등을 재계산하여 “좌석 및 운행정보” 변수로 반영하여 수요예측에 활용하였다. 표 1에 생성된 기본적인 파생변수들을 포함하여 수요예측에 사용한 변수들의 원천데이터의 예시를 정리하였다.

표 1. KTX 수요예측을 위해 사용한 변수명과 의미

|  |  |  |  |
| --- | --- | --- | --- |
| **특성** | **변수명** | **예시** | **정의** |
| **열차종** | **주운행선** | 경부선 | KTX 주요 5개 노선 경부/경전/동해/전라/호남선 |
| **과거**  **승차인원수** | **승차인원수\_Lag1** | 3,464,111.00 | 1개월 전 승차인원수 |
| **승차인원수\_Lag2** | 3,318,669.00 | 2개월 전 승차인원수 |
| **승차인원수\_Lag3** | 3,647,548.00 | 3개월 전 승차인원수 |
| **승차인원수\_Lag4** | 3,547,176.00 | 4개월 전 승차인원수 |
| **승차인원수\_Lag5** | 3,643,417.00 | 5개월 전 승차인원수 |
| **승차인원수\_Lag6** | 3,291,100.00 | 6개월 전 승차인원수 |
| **승차인원수\_Lag7** | 3,352,224.00 | 7개월 전 승차인원수 |
| **승차인원수\_Lag8** | 3,279,768.00 | 8개월 전 승차인원수 |
| **승차인원수\_Lag9** | 3,274,361.00 | 9개월 전 승차인원수 |
| **승차인원수\_Lag10** | 6,921,600.00 | 10개월 전 승차인원수 |
| **승차인원수\_Lag11** | 3,281,372.00 | 11개월 전 승차인원수 |
| **승차인원수\_Lag12** | 3,152,500.00 | 1년 전 승차인원수 |
| **시간정보** | **운행년월** | 45,352.00 | 해당 열차가 운행된 연도와 월 |
| **일수** | 31.00 | 해당 월에 포함된 날짜의 수 |
| **주말수** | 15.00 | 해당 월에 포함된 주말(금토일) 수 |
| **주중수** | 16.00 | 해당 월에 포함된 주중(월화수목) 수 |
| **공휴일수** | 1.00 | 해당 월에 포함된 대체휴일을 포함한 공휴일의 수 |
| **명절수** | - | 해당 월에 포함된 대체휴일을 포함한 명절의 수 |
| **좌석 및 운행정보** | **공급차량수** | 62,658.00 | 운행될 수 있는 총 열차의 수 |
| **공급좌석합계수** | 3,301,402.00 | 열차 내에 판매할 수 있는 좌석의 수 |
| **승차수입금액** | 113,298,033,576.00 | 판매된 좌석의 총금액 |
| **승차인원수** | 3,473,501.00 | 판매된 좌석의 수 |
| **승차연인거리** | 8,239,441,412.00 | 승객이 타고간 거리 |
| **좌석거리** | 13,516,340,607.00 | 공급좌석이 이동가능한 거리 |
| **1인당수입율** | 1,006,017.52 | "승차수입금액"을 "승차인원수"로 나눈 값으로, 1인당 평균 수입미 |
| **공급대비승차율** | 32.33 | "승차인원수"를 "공급좌석합계수"로 나눈 값으로, 좌석 판매의 집중도 |
| **운행대비고객이동** | 52.84 | "좌석거리"를 "승차연인거리"로 나눈 값으로, 승객의 이용 집중도 |
| **관광** | - | 관광으로 편성된 열차 수 |
| **일반** | 97.00 | 일반으로 편성된 열차 수 |
| **일반/관광** | 408.00 | 일반/관광으로 편성된 열차 수 |
| **대수송** | - | 대수송으로 편성된 열차 수 |
| **임시** | - | 임시로 편성된 열차 수 |
| **확정** | 505.00 | 확정으로 편성된 열차 수 |
| **시발역** | 155.00 | 열차가 출발하는 시발역 종류의 수 |
| **종착역** | 155.00 | 열차가 도착하는 종착역 종류의 수 |
| **시발종착역** | 248.00 | 열차가 운행하는 "시발역+종착역" 노선의 수 |
| **열차운행횟수** | 3,791.00 | 운행을 한 총 열차의 수 |
| **외부환경** | **Stringency Index** | 1,463.82 | 코로나 진행정도 지수 |
| **Government Response Index** | 1,775.99 | 정부의 코로나 대응정도 지수 |
| **International Movement Restrictions** | 62.00 | 국가간 이동 제한정도 지수 |
| **Death People** | 82,667.00 | 코로나 사망자 수 |
| **Vaccinated People** | 1,246,537,550.00 | 백신접종을 시작한 인원수 |
| **Fully Vaccinated People** | 1,014,704,524.00 | 백신접종이 완료된 인원수 |
| **Containment People** | 1,918.90 | 격리된 인원수 |
| **Confirmed People** | 10,586,338.00 | 코로나 확진자 수 |

예측 대상인 승차인원수는 종속변수로 사용되고 나머지 44개의 변수들은 독립변수로 사용된다. 분석기간은 모든 노선의 2015년 1월부터 2024년 3월까지이며, 그 중 시계열 순서로 2023년 4월부터 2024년 3월까지의 데이터는 검증(Validation)셋으로 나머지는 학습(Training)셋으로 사용된다. 그리고 최종 예측 목표인 2024년 4월부터 2025년 12월까지가 테스트(Test)셋이다. 데이터에 특별한 결측치는 존재하지 않으며 변수들의 값의 범위가 다양하기 때문에 동일하게 조정하기 위해 스케일링(Scaling)을 적용하였다. 마지막으로 전처리가 완료된 원천데이터를 기반으로 한국철도공사에서 중시하는 파생지표들로 변경한 후 최종적으로 모델링의 입력으로 활용된다

* 1. *Machine Learning Algorithm: Bagging and Boosting*

기본적으로 머신러닝 알고리즘은 예측 오차를 줄이는 방향으로 설계되었다. 오차는 bias and variance로 분리될 수 있는데, 예측의 안정성에 초점을 두어 variance를 줄이기 위해 샘플링 기법을 활용하는 Bagging과 성능에 초점을 두어 bias를 줄이기 위해 반복적인 모델링을 활용하는 Boosting으로 구분될 수 있다. Bagging은 여러 개의 샘플을 추출하여 각 샘플마다 모델링한 후 outputs의 투표로 최종 output의 label을 결정하는 방식이기에 bootstrap aggregating으로 불린다. 본 연구에 활용된 대표적 알고리즘은 Random Forest(RF)다. RF는 samples를 병렬처리로 한꺼번에 모델링 할 수 있기 때문에 속도도 빠르다며 training 데이터의 학습 성능이 우수한 편이고 noise data에도 성능이 크게 변하지 않는 robust algorithm이다 (Michie, Spiegelhalter, and Taylor 1999). 변수의 중요도를 제공하긴 하지만 각 샘플별 모델링 과정에서 각 변수들의 우선순위를 평균한 것으로 positive or negative와 같은 영향력의 방향성을 포함하지는 못하는 단점이 있다.

Bagging처럼 Boosting도 classification and regression 문제 모두 활용할 수 있는 supervised learning이다 (Chen and Guestrin 2016). Gradient boosting methods(GBM)를 기반으로 Extreme gradient boosting (XGBoost) and LightGBM, CatBoost 등의 알고리즘으로 확장되며 단점들이 개선되고 다양한 기능이 추가되었다. 본 연구에서는 XGBoost, LightGBM and CatBoost를 대표적인 boosting 알고리즘으로 활용하여 자하철 혼잡도 예측에 사용한다. 이 알고리즘들은 전체 데이터를 학습할 때 잘 학습되지 않은 에러들을 가중치를 높여 재학습을 하면서 성능을 향상시키는 방향으로 설계되었다. 메모리를 효율적으로 사용하거나 computation 속도를 높이기 위한 많은 기능들이 추가되었고, iterative learning과 병렬처리 과정에서 성능이 높아질 수 있었다 (Alsubari et al. 2021).



그림 1. 배깅(Bagging)과 부스팅(Boosting) 알고리즘의 구조 비교 (Cha, Moon, and Kim 2021).

* 1. *Deep Learning Algorithm: MLP, RNN*

딥러닝은 인공지능의 한 방법론으로, 인간의 두뇌 구조에서 영감을 받아 개발되었다. 데이터의 복잡한 패턴들을 학습하기 위해 연속된 layer를 중첩하여 변수들의 모든 상호작용을 포함하여 의미있는 규칙들을 학습해 내는데 강점이 있다. 이러한 구조의 기본이 되는 알고리즘으로 multilayer perceptron (MLP)가 있으며, 이미지나 시계열 등의 데이터도 학습해 낼 수 있도록 layer의 구조나 흐름을 개선하여 convolutional neural network (CNN)과 recurrent neural network (RNN) 등으로 확장되었다. neural network structure를 가진 모든 알고리즘들은 앞서 소개한 머신러닝의 알고리즘 처럼 activation function만 선택적으로 변경함으로써 classification and regression 문제 모두에 활용가능한 universal approximator이다. 모델의 구조가 복잡하기에 수많은 가중치들을 한꺼번에 추정해야 하는데 feed-forward and backpropagation 을 사용하여 loss function을 최소화하기 위해 gradient descent optimizer를 사용하였다 (see Figure 2) (Viswavandya, Patel, and Sahoo 2021). 머신러닝과 마찬가지로 다양한 nonlinear patterns를 학습함으로써 기존의 알고리즘보다 성능이 대폭 향상되어 인간이 인지하기 어려운 것들도 학습해내는데 유용하다.



그림 2. 대표적인 딥러닝 알고리즘인 MLP의 구조와 데이터 학습을 통한 수많은 가중치 추정 과정

본 연구에서는 MLP and CNN 알고리즘을 활용하여 지하철 혼잡도를 예측한다. 특히 CNN은 이미지 분류 특화 알고리즘이지만 내부 구조가 결국 feature를 요약하여 것이기 때문에 regression 문제에도 활용이 가능하다. Feature를 요약하기 위해 convolution, pooling, and fully connected layers를 활용하여 인접한 feature values들의 관련성에 더욱 초점을 두어 학습한다. 이러한 과정에서 여러 개의 neurons로 구성된 layer들을 계속적으로 통과하는데, 결국 low-level feature vector에서 점차 high-level feature vector로 변수들의 특성이 변환되어 학습된다 (Shustanov and Yakimov 2017). 따라서 다양한 변수들의 상호작용을 포함한 특징들이 학습될 것이고 마지막 output은 지하철 혼잡도 값이 출력되며 regression 문제를 해결한다.

* 1. *MLP 기반 알고리즘*

*2.4.1* NBEATS (2020)

N-BEATS(Neural Basis Expansion Analysis for Time Series Forecasting)는 기존의 시계열 예측 모델과 달리 RNN(Recurrent Neural Network)이나 CNN(Convolutional Neural Network)과 같은 복잡한 구조 없이 완전연결층(Fully Connected Layers)만을 활용하여 높은 예측 성능을 달성하는 모델이다.

일반적으로 시계열 예측 모델은 시간 종속성을 학습하기 위해 순환 구조(RNN 계열)를 사용하거나, 합성곱 연산(CNN 계열)을 활용하여 패턴을 추출하는 방식으로 학습을 수행한다. 그러나 이러한 방식은 계산 비용이 높고 학습 과정이 복잡하여 모델의 해석 가능성이 낮아지는 문제가 있다. 반면, 해당 모델은 다층 완전연결 신경망(MLP) 기반의 구조를 유지하면서도 강력한 예측 성능을 제공하는 것이 특징이다. 특히 트렌드(trend)와 계절성(seasonality) 등의 패턴을 별도로 학습할 수 있는 구조를 통해 예측 결과의 해석 가능성을 높인다 (Jang과 Kim, 2022). 이를 통해 모델은 단순히 미래 값을 예측하는 것을 넘어, 시간에 따른 변화 요인을 분리하여 분석할 수 있는 장점을 제공한다. 또한, 잔차 기반 학습(residual learning)과 계층적 학습 구조를 통해 단순한 선형 회귀 모델을 뛰어넘어, 보다 정교한 시계열 분석을 가능하게 한다 (Boris et al., 2019).

모델은 기본적으로 다층 완전연결층을 포함하는 블록(Block)으로 구성되며, 이러한 블록들이 여러 개 결합되어 스택(Stack)을 형성한다. 각 블록은 네 개의 완전연결층을 포함하며, 두 가지 출력을 생성한다. 여기서 출력 중 하나는 입력 데이터에서 불필요한 부분을 제거하는 Backcast, 다른 하나는 미래 데이터를 예측하는 Forecast이다.

또한, 모델의 구조적 특징 중 하나는 블록들이 결합된 Stack 구조이다. 그림 3에서 좌측 파란 박스는 FC Stack의 구조를 나타내며, 각 블록은 함수 를 학습하여 각각 Backcast와 Forecast를 생성하는 방식으로 작동한다. Backcast는 현재 블록이 입력받은 데이터를 기반으로 기존 패턴을 복원하는 역할을 하며, Forecast는 미래 시점에 대한 예측 값을 생성하는 역할을 한다.

*텍스트, 도표, 스크린샷, 평면도이(가) 표시된 사진

자동 생성된 설명*

그림 3. NBEATS 구조

그림 3의 중앙 부분을 보면, 여러 개의 블록이 쌓여 있는 구조를 확인할 수 있으며, 각 블록이 생성한 Forecast 값이 점진적으로 최종 예측값에 반영된다. 스택 내부에서는 각 블록이 독립적으로 예측을 수행하는 것이 아니라, Stack residual connection을 활용하여 이전 블록이 예측한 Backcast 값을 제거하고, 남은 정보를 다음 블록의 입력으로 전달하는 방식으로 학습이 이루어진다. 이러한 방식은 모델이 점진적으로 더욱 정교한 예측을 수행할 수 있도록 하며, 장기적인 트렌드와 단기적인 변동성을 동시에 학습할 수 있도록 돕는다. 또한, Doubly Residual Stacking 기법을 활용하여 학습 성능을 더욱 향상시킨다. 일반적인 Residual Network는 입력과 출력을 직접 연결하여 학습을 원활하게 하지만, 이를 확장하여 Backcast와 Forecast 두 가지 방식으로 잔차를 계산하는 구조를 도입하였다. 우측 부분을 보면 각 스택(Stack)이 개별적으로 Forecast를 생성하며, 이들이 종합되어 최종적인 Global Forecast가 도출되는 모습을 확인할 수 있다. 이러한 방식은 기존의 단순한 잔차 학습(residual learning)보다 더욱 효과적으로 데이터를 학습할 수 있도록 하며, 특히 장기적인 시계열 패턴을 보다 안정적으로 모델링할 수 있도록 한다.

결국, 해당 모델은 기본적으로 과거 데이터를 활용하여 미래 값을 예측하는 방식으로 동작한다. 모델은 Lookback Window를 설정하여 일정 기간의 과거 데이터를 입력으로 받는다. 그림 3의 상단에서 확인할 수 있듯이, Lookback Period는 모델이 입력으로 받는 데이터의 범위를 나타내며, 이는 보통 예측하려는 기간(Forecast Period)보다 길게 설정된다. 이후, 입력 데이터는 첫 번째 스택으로 전달되며, 개별 블록들은 잔차(Residual)를 계산하여 Backcast와 Forecast를 생성한다. 생성된 Backcast 값은 입력 데이터에서 제거되며, 남은 데이터가 다음 블록의 입력으로 전달된다. 이러한 과정이 반복되면서 모델은 점진적으로 예측 성능을 향상시키고, 마지막으로 모든 스택에서 생성된 Forecast 값을 종합하여 최종적인 Global Forecast 예측 결과를 생성한다.

*2.4.2* NBEATSx (2021)

NBEATSx(Neural Basis Expansion Analysis with Exogenous Variables)는 앞선 N-BEATS 모델을 확장하여 외생 변수(Exogenous Variables)를 입력으로 활용할 수 있도록 설계된 시계열 예측 모델이다. N-BEATS는 시계열 데이터 자체의 패턴을 분석하여 예측을 수행하는 반면, NBEATSx는 추가적인 외생 변수를 고려하여 보다 정교한 예측이 가능하도록 한다. 이를 통해 단순한 과거 데이터의 패턴 학습을 넘어, 기온, 경제 지표, 계절성 등 외부 요인이 미래 값에 미치는 영향을 반영할 수 있다.

모델의 입력은 예측 대상 시계열 데이터 와 함께 여러 개의 외생 변수 등을 포함하며, Backcast Period(과거 데이터 학습 구간)와 Forecast Period(미래 예측 구간)으로 구성된다. 이러한 구조를 통해 모델은 시간 흐름에 따른 변화뿐만 아니라 외부 요인 간의 상호작용도 고려하여 보다 신뢰성 있는 예측을 수행할 수 있다 (Kin et al., 2023).

. 텍스트, 도표, 평행, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그림 4. NBEATSx 구조

모델의 학습 구조는 계층적 스택(Stack)과 블록(Block) 구조를 기반으로 한다. 입력 데이터는 첫 번째 스택(Stack 1)으로 전달되며, 이후 여러 개의 스택을 거치면서 예측 성능을 개선하는 방식으로 동작한다. 각 스택은 내부적으로 다수의 블록을 포함하며, 블록마다 개별적인 예측을 수행한 후 이를 조합하여 최종 결과를 도출한다. 그림 4에서 각 블록 내부에는 완전연결층(FC Stack)이 배치되어 있으며, 이 과정에서 Backcast()와 Forecast() 두 가지 출력을 생성한다. 여기서 Backcast는 입력 데이터에서 학습된 패턴을 복원하는 역할을 하며, Forecast는 미래 값을 예측하는 기능을 수행한다. 블록에서 생성된 결과는 다음 블록으로 전달되며, 이 과정이 반복되면서 모델은 점진적으로 예측 성능을 향상시킨다.

또한, Stack Residual Connection을 통해 이전 스택이 학습하지 못한 부분을 다음 스택이 보완할 수 있도록 하여, 예측 오류를 줄이고 보다 정밀한 결과를 도출할 수 있다. 마지막으로, 모든 스택에서 생성된 예측을 종합하여 Global Forecast(모델의 최종 예측 값)을 생성한다. 이러한 구조적 특징 덕분에 해당 모델은 외부 요인의 영향을 강하게 받는 예측 문제에서 우수한 성능을 발휘할 수 있다.

*2.4.3* NHITS (2023)

NHITS(Neural Hierarchical Interpolation for Time Series Forecasting)는 장기 시계열 예측(Long-Horizon Forecasting, LHF)의 성능을 향상시키기 위해 개발된 모델로, 다중 해상도 학습과 계층적 보간(Hierarchical Interpolation)을 결합한 것이 특징이다.

기존 Transformer 기반의 시계열 예측 모델은 장기 예측에서 연산 비용이 증가하고, 예측 오차가 누적되는 문제가 발생하는데, 해당 모델은 이를 해결하기 위해 다중 비율 입력 샘플링(Multi-Rate Input Sampling)과 계층적 보간 기법을 도입하여 예측 정확도를 향상시키면서도 계산 비용을 줄였다 (Cristian et al., 2023). 여기서 계층적 보간 기법이란, 시계열 데이터를 다양한 해상도로 분해하여 낮은 해상도의 예측을 기반으로 점진적으로 높은 해상도의 예측을 보완하는 방식을 의미한다. 이를 통해 단기적 변동성을 포착하는 동시에 장기적인 트렌드를 보다 안정적으로 학습할 수 있다.

텍스트, 도표, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

그림 . NHITS 구조

모델은 낮은 주파수(low-frequency)에서 먼저 장기적인 패턴을 학습한 후, 고주파(high-frequency) 요소를 점진적으로 추가하여 예측을 정밀하게 보정한다. 이 과정에서 예측 값은 계층적으로 상위 해상도에서 보간(interpolation)되며, 기존 시계열 모델이 단일 해상도로 학습하는 방식과 비교하여 보다 정밀한 예측을 수행할 수 있도록 한다. 입력은 시계열 데이터 ​와 다양한 해상도의 외부 변수를 포함하며, Backcast Period(과거 데이터 학습 구간)와 Forecast Period(미래 예측 구간)으로 구분된다.

그림 5를 보면, NHITS는 여러 개의 스택(Stack)으로 구성되어 있으며, 각 스택은 서로 다른 주파수 성분을 학습할 수 있도록 설계되어 있다. 예측 과정에서 각 스택은 내부적으로 다수의 블록(Block)으로 구성되며, 블록 내부에는 다층 퍼셉트론(MLP Stack)과 MaxPool 연산이 포함되어 있다. 특히, 각 블록은 Backcast( )와 Forecast() 두 가지 출력을 생성하며, 이전 블록의 예측을 수정하고 점진적으로 예측 성능을 개선하는 역할을 한다. 또한, Stack Residual Connection을 통해 각 스택은 이전 스택이 학습하지 못한 정보를 보완하며, 최종적으로 모든 스택의 예측 결과를 결합하여 Global Forecast()을 생성한다. 이러한 구조는 장기 예측에서 발생하는 누적 오차를 줄이고, 다양한 주파수 대역에서 시계열 데이터를 효과적으로 분해하여 모델의 해석 가능성을 높인다.

*2.4.4* TiDE (2023)

TiDE(Time-series Dense Encoder)는 장기 시계열 예측을 위해 개발된 모델로, 기존 Transformer 기반 모델의 높은 연산 비용과 성능 저하 문제를 해결하면서 다양한 변수 간의 복잡한 관계를 효과적으로 반영할 수 있도록 설계되었다. 특히, 시간에 따라 변화하는 외부 요인으로 시계열 데이터의 변동성에 영향을 미치는 주요 요소인 공변량(covariates)을 활용하여 시계열 데이터의 변화를 보다 정밀하게 예측할 수 있도록 한다.

기존의 선형 회귀 모델은 변수 간의 관계를 단순한 선형 함수로 학습하는 반면, 해당 모델은 다층 퍼셉트론(MLP) 기반의 구조로 입력 데이터를 다양한 차원에서 변환하고 비선형적인 특징을 학습할 수 있도록 하는 Dense Encoder를 활용하여 공변량과 대상 변수 간의 비선형적인 의존성을 학습할 수 있도록 한다 (Abhimanyu et al., 2023). 이를 통해, 시계열의 복잡한 패턴을 효과적으로 포착하고, 다양한 외부 요인이 데이터에 미치는 영향을 정교하게 반영할 수 있다. 또한, 연산 효율성을 유지하면서도 장기 예측에서의 신뢰성을 높여, 수요 예측, 교통량 분석, 금융 데이터 예측 등 다양한 응용 분야에서 활용될 수 있다.

.

텍스트, 스크린샷, 도표, 평면도이(가) 표시된 사진

자동 생성된 설명

그림 6. TiDE 구조

모델은 과거 시계열 데이터(Lookback)뿐만 아니라 다양한 속성(Attribute) 및 동적 공변량(Dynamic Covariates)을 함께 입력받으며, Feature Projection 단계를 거쳐 고유한 특징을 추출하고 모델의 입력 차원을 최적화한다. 그림 6을 보면, 인코딩 과정에서 Dense Encoder는 여러 개의 Residual Block으로 구성되며, 여기서 Residual Block은 깊은 신경망에서 정보 손실을 줄이고 학습 속도를 개선하기 위해 도입되어, 입력 데이터를 원래 출력값과 더하는 Skip Connection을 포함한다. 이러한 구조는 Gradient Vanishing 문제를 완화하며, 네트워크가 깊어지더라도 중요한 정보를 유지할 수 있도록 돕는다. 또한, 각각의 블록은 Skip Connection, Layer Normalization, Dropout 기법을 적용하여 안정적인 학습을 지원하며, 이를 통해 다층 비선형 변환을 수행하여 공변량과 예측 대상 변수 간의 복잡한 관계를 효과적으로 학습한다. 결론적으로 이러한 방식은 단순한 선형 모델보다 더 높은 예측 성능을 제공하며, 다양한 외부 요인의 영향을 정밀하게 반영할 수 있도록 한다.

모델의 예측 과정은 인코딩, 디코딩, 그리고 최종 예측 단계로 이루어진다. 먼저, 과거 시계열 데이터 , 속성 변수 , 그리고 동적 공변량 ​이 결합되어 Dense Encoder를 통해 핵심 특징이 추출되며, 이를 통해 인코딩 벡터 가 생성되고, 이는 수식으로 다음과 같이 표현된다.

이후, Dense Decoder는 인코딩된 벡터 를 기반으로 예측 벡터 를 생성한다. 이는 미래 시점의 데이터를 예측하기 위한 중간 과정이며, 수식으로 다음과 같이 정의된다.

마지막으로, Temporal Decoder는 예측 벡터를 시계열 흐름에 맞게 조정하여 최종적인 시계열 예측값 을 도출한다.

위 수식에서 는 예측 기간 동안의 동적 공변량을 의미하며, 모델이 예측 과정에서 시계열 데이터뿐만 아니라 외부 요인의 변동성을 반영할 수 있도록 돕는다. 이를 통해 해당 모델은 단순한 패턴 학습을 넘어, 시계열 데이터와 외부 요인 간의 관계를 정교하게 모델링할 수 있다. 또한, Residual Block을 활용하여 누락된 정보를 보완하고, 원래 입력 데이터와의 관계를 유지하면서도 장기적인 시계열 패턴을 효과적으로 반영할 수 있도록 한다.

*2.4.5* DeepNPTS (2023)

DeepNPTS(Deep Non-Parametric Time Series Forecaster)는 특정한 확률 분포를 가정하지 않고 데이터를 기반으로 학습하는 비모수적(non-parametric) 시계열 예측 모델이다. 기존의 확률 기반 시계열 모델들은 특정한 분포를 전제로 하지만, 해당 모델은 이러한 제한 없이 다양한 시계열 환경에서 유연하게 적용될 수 있는 특징을 가진다.

또한, 모델은 샘플링 기반 접근법(Sampling-based Approach)을 활용하여 단일한 예측 값이 아닌 확률 분포를 직접 학습한다 (Syama et al., 2023)​. 즉, 특정 시점에서 하나의 값을 예측하는 것이 아니라, 모든 가능한 예측 값에 대한 확률 분포를 학습함으로써 예측의 신뢰도를 정량적으로 표현할 수 있다. 이를 위해 Importance Sampling(IS), Monte Carlo Sampling(MC), Kernel Density Estimation(KDE) 등의 샘플링 기법을 활용하여 데이터의 변동성을 반영하며, Adaptive Sampling Mechanism을 적용하여 샘플링 확률을 데이터 분포에 맞게 최적화함으로써, 단순한 계절성(seasonality)이나 추세(trend)뿐만 아니라 예측이 어려운 비정상적인 패턴까지도 반영할 수 있다.

또한, 글로벌 학습 방식(Global Learning Approach)을 적용하여, 다수의 시계열 데이터를 하나의 모델에서 동시에 학습할 수 있다. 기존의 개별 학습 방식(Local Learning Approach)은 단일 시계열 데이터만을 독립적으로 학습하는 반면, 글로벌 학습 방식은 여러 시계열 데이터 간의 공통된 패턴을 효과적으로 학습하여 일반화 성능을 향상시킨다. 이러한 특성 덕분에 다양한 시계열 데이터 환경에서 높은 예측 성능을 유지하며, 새로운 데이터에도 빠르게 적응할 수 있다.

모델은 다층 퍼셉트론(MLP, Multi-Layer Perceptron) 기반의 구조를 활용하여, 시계열 데이터의 과거 값을 입력으로 받아 특정 시점에서 샘플링할 확률 분포를 학습한다. 과거 시계열 데이터와 공변량(covariates) 정보를 함께 입력받아, 시간적 패턴뿐만 아니라 외부 요인의 영향을 반영할 수 있도록 한다.

구체적으로, 모델은 주어진 시점까지의 데이터 과 공변량 ​을 입력으로 받아 샘플링 확률 분포 를 출력하며, 이를 통해 예측 시점 에서의 값을 추론한다. 이를 수식으로 표현하면 다음과 같다.

여기서 는 피드포워드 신경망을 의미하며, 는 학습된 가중치 파라미터를 나타낸다. 이 신경망은 다중 시계열 데이터를 학습하면서 개별 시계열에 적합한 샘플링 확률을 동적으로 생성하는 방식으로 동작한다. 예측 과정에서는 샘플링된 확률을 기반으로 향후 시점의 값을 추론하며, 이를 통해 데이터의 장기적인 패턴과 비정상적인 변동성을 반영할 수 있다.

다음으로, 모델은 훈련 과정에서 Ranked Probability Score(RPS)를 손실 함수로 활용한다. 이는 예측된 분포와 실제 값 간의 차이를 측정하는 척도로, 다음과 수식과 같이 정의된다.

여기서 는 특정 분위수 에서의 분위수 손실(Quantile Loss)로 정의되며, 이는 다음 수식과 같이 표현된다.

이 손실 함수는 예측된 확률 분포가 실제 관측된 값과 얼마나 잘 일치하는지를 평가하며, 모델이 학습 과정에서 더욱 정교한 샘플링 전략을 형성할 수 있도록 돕는다. 또한, 모델의 예측 과정에서는 학습된 샘플링 확률을 기반으로 다중 샘플링을 수행하여 예측 분포를 구성한다. 즉, 특정 시점 에서 단일한 예측 값을 제공하는 것이 아니라, 확률적 분포를 생성함으로써 예측의 불확실성을 정량적으로 표현할 수 있다.

결론적으로, 모델은 특정 시점에서의 샘플링 확률을 조정하는 방식으로 시계열 데이터 내 존재하는 다양한 패턴을 학습하며, 데이터의 변동성이 높은 환경에서도 안정적인 예측 성능을 유지할 수 있다.

* 1. *RNN 기반 알고리즘’*

*2.5.1 LSTM(1997)*

LSTM(Long Short-Term Memory)은 Recurrent Neural Network(RNN)의 확장된 형태로 개발되었다. 기본적으로 RNN 기반 구조를 가지며, 순환 연결을 통해 이전 시점의 정보를 다음 시점으로 전달하는 특성을 가진다. 기존 RNN에서는 시간이 지남에 따라 기울기가 소실되거나 폭발하는 문제가 발생하여 장기적인 의존성을 학습하는 것이 어렵기 때문에, 이를 해결하기 위해 모델은 특별한 게이트 구조와 셀 상태(cell state) 메커니즘을 도입하였다 (Alex과 Alex, 2012).

핵심 구성 요소는 셀 상태와 세 가지 주요 게이트인 입력 게이트(input gate), 삭제 게이트(forget gate), 출력 게이트(output gate)이다. 입력 게이트는 새로운 정보를 셀 상태에 추가할지를 결정하고, 삭제 게이트는 기존 정보를 유지하거나 제거하는 역할을 하며, 출력 게이트는 최종적으로 출력할 정보를 결정하는 기능을 수행한다. 이러한 구조를 통해 기존 RNN과 비교하여 장기 의존성을 보다 효과적으로 학습할 수 있다. 모델의 셀 상태는 다음과 같이 수식으로 표현된다

여기서 ​는 삭제 게이트, 는 입력 게이트, ​는 새로운 기억 정보이며, 는 요소별 곱을 의미한다. 또한, 최종 출력은 다음과 같은 수식으로 정의된다.

또한, 모델은 가중치를 업데이트하기 위해 역전파 알고리즘을 사용하고, 시퀀스 데이터를 다루기 때문에 시간에 따른 역전파(Backpropagation Through Time, BPTT)가 적용된다. 손실 함수의 기울기를 계산하고, 이를 바탕으로 가중치를 업데이트하여 모델의 성능을 최적화한다.

이러한 구조 덕분에 해당 모델은 장기적인 패턴을 보다 효과적으로 학습할 수 있으며, 자연어 처리(NLP), 음성 및 영상 처리, 금융 데이터 분석, 생체 신호 분석, 수요 예측(demand forecasting)과 같은 시계열 분석 문제에서 활용되고, 특정한 시계열 데이터 처리에서 강력한 성능을 유지하고 다양한 최적화 기법과 결합하여 지속적으로 발전하고 있다.

*2.5.2 GRU(2014)*

Gated Recurrent Unit(GRU)는 LSTM(Long Short-Term Memory)의 경량화된 변형으로, 동일하게 기존 RNN이 가진 기울기 소실 및 폭발 문제를 해결하기 위해 설계되었으며, 순환 연결을 통해 시간에 따른 정보를 유지하고 전달하는 특성을 갖는다. LSTM은 게이트 구조(입력 게이트, 삭제 게이트, 출력 게이트)를 통해 장기 의존성 문제를 해결하는 반면, GRU는 비교적 단순화된 게이트 구조(업데이트 게이트와 리셋 게이트)를 채택하여 계산 효율성을 높였다. 별도의 메모리 셀을 사용하지 않고 은닉 상태를 직접 조절하며, 적은 파라미터 수로도 효과적인 학습이 가능하여 작은 데이터셋에서도 성능 저하 없이 빠르게 학습할 수 있다 (Kyunghyun 등, 2014).

GRU의 핵심 요소는 업데이트 게이트(update gate) 와 리셋 게이트(reset gate) 이며, 업데이트 게이트는 기존 정보를 유지할지 새로운 정보를 반영할지를 결정하고, 리셋 게이트는 이전 상태 정보를 어느 정도 잊을지를 조절한다. 업데이트 게이트 는 이전 상태 와 현재 입력 을 기반으로 현재 상태를 얼마나 업데이트할지를 결정한다. 리셋 게이트 는 이전 은닉 상태의 정보를 얼마나 유지할지를 조절하며, 이를 통해 중요한 정보를 선택적으로 보존하고 불필요한 정보를 제거할 수 있다. 후보 활성화 상태 는 현재 입력과 이전 상태를 기반으로 계산되며, 리셋 게이트의 영향을 받는다. 마지막으로, 최종 은닉 상태 는 이전 상태와 후보 활성화 상태 사이에서 업데이트 게이트에 의해 가중 평균되어 결정된다. 모델의 동작 방식은 다음과 같이 수식으로 표현된다.

여기서 는 시그모이드 활성화 함수이며, 는 하이퍼볼릭 탄젠트 활성화 함수를 의미한다. 이를 통해 시계열 문제에서 입력 데이터 가 현재 상태에 반영되고, 과거의 은닉 상태 와 결합되어 최종 은닉 상태 가 결정되는지 설명할 수 있다.

결론적으로, 해당 모델은 LSTM보다 구조가 단순하고 연산량이 적어 학습 속도가 빠르며, 장기 의존성을 효과적으로 학습할 수 있는 업데이트 게이트와 불필요한 정보를 조절하는 리셋 게이트를 활용하는 구조적 특성을 가지고 있어 기계 번역, 감성 분석, 텍스트 요약 등의 자연어 처리(NLP) 분야에 활용되고, 수요 예측 및 심전도, 뇌파 분석과 같은 시계열 데이터 처리에서도 높은 성능을 보인다.

*2.5.3 DilatedRNN(2017)*

Dilated Recurrent Neural Network는 Recurrent Neural Network(RNN)의 구조를 확장하여 발전된 모델로, dilated(확장된) 연결을 도입하여 특정 시간 간격을 늘려 데이터를 처리함으로써 긴 시계열 패턴을 효과적으로 학습할 수 있다 (Shiyu 등, 2017). 해당 모델의 핵심 개념은 Dilated recurrent skip connection으로, RNN의 순차적 연결을 확장하여 특정 시점 간의 간격을 늘려 데이터를 처리하는 방식이다. 일반적인 RNN에서는 은닉 상태가 다음과 같이 순차적으로 업데이트된다.

하지만, Dilated Skip Connection을 적용하면 특정 시간 간격을 확장하여 더 멀리 떨어진 이전 상태와 연결할 수 있다. 이러한 구조를 수식으로 표현하면 다음과 같다.

여기서, 는 번째 층에서의 hidden state를 나타내며, 는 해당 층에서의 입력 데이터이다. 또한, 는 해당 층의 dilation skip 간격을 의미하며, 이는 특정 시간 간격마다 연결되는 구조를 형성한다. 즉, 일반 RNN은 모든 시점에서 과 연결되는 반면, Dilated Skip Connection은 일정한 간격 을 두고 연결되므로, 더 긴 시퀀스 정보를 효과적으로 학습할 수 있다.

모델은 입력 시 각 층의 dilation 값을 1, 2, 4등 지수적으로 증가시키는 방식으로 설정한다. 이를 수식으로 표현하면 다음과 같다.

여기서 은 는 번째 층에서 적용되는 dilation 값이고, 은 dilation 증가율, 은 총 층의 개수이다. DilatedRNN에서는 층(layer)이 깊어질수록 dilation 값이 증가하며, 더 넓은 시간 간격을 학습할 수 있다. 각 층의 dilation은 증가하여, 낮은 층에서는 짧은 시간 범위를 학습, 높은 층에서는 더 넓은 시간 범위를 학습할 수 있다.

라인, 도표, 폰트, 그래프이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

그림 7. DilatedRNN 구조

그림 7에서 좌측 구조는 세 개의 layer으로 구성된 해당 모델의 예시이며, 각 layer의 dilation 값이 1, 2, 4로 설정된 것을 보여준다. 이 구조는 기존 RNN보다 더 넓은 시간 간격을 학습할 수 있도록 설계되어 있으며, 특정 패턴이 장기적으로 유지되는 데이터에서 효과적으로 작동할 수 있다. 우측 구조는 두 개의 layer으로 구성된 모습을 보여주며, 첫 번째 layer에서 dilation 값이 2로 설정되어 있다. 이 경우, 일부 데이터가 손실될 수 있어, 추가 임베딩 연결(빨간색 화살표)이 적용된다. 이러한 보완적 연결을 통해 데이터 의존성을 유지하며, 학습의 안정성을 향상시킬 수 있다.

결론적으로, 해당 모델은 RNN의 단점을 보완하면서 장기적인 시계열 패턴을 효과적으로 학습할 수 있도록 설계되었다. 일반적인 RNN은 모든 시점을 순차적으로 업데이트해야 하지만, 모델의 Dilated Skip Connection을 활용하면 과거 정보를 넓은 시간 간격으로 직접 연결할 수 있으며, 모든 시점을 거치지 않고 학습할 수 있어 긴 시퀀스를 처리할 때 발생하는 RNN의 vanishing gradient문제를 완화한다.

*2.5.4 TCN(2018)*

Temporal Convolutional Network(TCN)는 2018년 발표된 시계열 데이터 처리에 특화된 합성곱 신경망(Convolutional Neural Network, CNN) 기반의 모델이다 (Shaojie 등, 2018). 해당 모델은 세 가지 특징을 바탕으로 장기적인 시간 의존성을 효과적으로 학습할 수 있다는 장점이 있다.

앞서 언급한 모델의 세 가지 주요 특징으로 Causal Convolution, Dilated Convolution, Residual Connections가 있다. Causal Convolution은 미래 시점의 정보가 현재 시점에 반영되지 않도록 설계되어 있으며, Dilated Convolution은 시간 간격을 확장하는 방식으로 긴 시퀀스 내 장기 의존성을 학습할 수 있도록 한다. 또한, Residual Connections는 깊은 네트워크의 학습을 안정화하고 성능을 향상시키는 역할을 한다.

모델의 구조는 기본적으로 1D CNN과 유사하다. 1D CNN은 시간축을 따라 합성곱을 수행하며, 각 입력 벡터는 특정 시점의 데이터 포인트를 나타낸다. 이와 유사하게, 해당 모델 또한 시간 순서를 유지하며 합성곱을 수행하여 과거의 정보를 바탕으로 현재의 출력을 계산한다. 하지만, 모델이 시계열 데이터를 효과적으로 처리하기 위해 Causal Convolution과 Dilated Convolution을 도입하였고, 깊은 신경망 구조에서 안정적인 학습이 가능하도록 Residual Connection을 적용하였다.

자세히 살펴보면, Causal Convolution은 시간적 순서를 보존하는 방식으로 특정 시점의 출력이 현재 및 과거 입력 값에만 의존하도록 한다. 이를 통해 RNN처럼 과거에서 미래로 정보를 전달하는 기능을 CNN에서 구현할 수 있다. 수식으로 다음과 같이 표현되며, 는 커널 크기, 는 현재 시점 보다 과거의 입력 값만을 포함한다.

다음으로, Dilated Convolution은 단순한 합성곱만으로는 장기 의존성을 학습하는 데 한계가 있기 때문에 사용된다. 이 방법은 커널 적용 간격을 조정하여 더 넓은 수용 영역을 확보하도록 한다. 층이 깊어질수록 dilation factor가 지수적으로 증가하게 되어, 비교적 적은 층수로도 RNN보다 긴 시퀀스 의존성을 학습할 수 있다. 수식으로 아래와 같으며, 는 dilation factor이다.

Residual Block은 깊은 네트워크 구조에서 학습이 어려워지는 문제를 해결하기 위해 사용되며, 두 개의 Dilated Causal Convolution 레이어를 포함한다. 블록은 입력과 출력을 더해주는 방식(skip connection)으로 구성되며, vanishing gradient문제를 완화한다. 수식은 아래와 같다.

위 수식에서 는 입력, 는 변환된 출력값이다. 이러한 Residual Block 구조는 모델의 학습 안정성을 높이고, 보다 깊은 신경망을 효과적으로 학습할 수 있도록 한다. 또한, 이러한 블록은 학습 과정에서 정보의 손실을 최소화하고, 네트워크가 장기 의존성을 더욱 효과적으로 학습할 수 있도록 돕는다. 이는 입력과 출력 사이에 직접적인 연결(skip connection)을 제공함으로써, 중요한 정보가 네트워크의 깊은 층을 통과하면서 손실되지 않도록 하기 때문이다.

텍스트, 라인, 스크린샷, 도표이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

그림 . TCN 구조

그림 8은 좌측부터 Dilated Causal Convolution의 구조, Residual Block의 요소, 그리고 Residual Connection의 예시를 보여준다. Dilated Causal Convolution 구조에서는 서로 다른 dilation factor를 사용하여 입력 데이터에서 점진적으로 더 넓은 수용 영역을 확보한다. Residual Block 구조는 Dropout, ReLU, Weight Normalization, Dilated Causal Convolution 등의 요소를 포함하여 학습을 안정화하고 성능을 향상시키며, Residual Connection 예시에서는 1x1 합성곱 또는 Identity Map을 사용하여 입력 차원과 출력 차원을 정렬하고 정보 손실 없이 학습할 수 있다.

결론적으로, 해당 시계열 데이터 처리 모델로, 장기 의존성을 효과적으로 학습할 수 있는 특징을 가진다. 또한, vanishing gradient문제를 완화하고 학습을 안정화하는 Residual Block을 포함하여 깊은 네트워크에서도 성능을 유지할 수 있다. 이러한 특징으로 다양한 시계열 분석 및 예측 작업에서 우수한 성능을 보이며, 특히 장기적인 패턴을 학습해야 하는 문제에서 효율적이다.

*2.5.5 DeepAR(2019)*

DeepAR은 아마존(Amazon)에서 개발한 확률적 시계열 예측(probabilistic time series forecasting) 모델로, LSTM(Long Short-Term Memory) 기반의 오토회귀 순환 신경망(Autoregressive Recurrent Network, ARN) 구조를 활용하여 다중 시계열 데이터를 학습하고, 이를 바탕으로 확률 분포 기반의 예측을 수행하는 딥러닝 모델이다 (David 등, 2020). 기존의 전통적인 시계열 예측 모델(예: ARIMA, ETS, 상태공간 모델)은 개별 시계열을 독립적으로 모델링하는 반면, DeepAR은 다중 시계열 데이터를 학습하여 공통적인 패턴을 포착하고, 개별 시계열의 데이터가 부족한 경우에도 일반화된 예측이 가능하도록 설계되었다.

앞서 언급하였듯, 모델은 오토회귀(Auto-Regressive) 구조를 기반으로 하는 순환 신경망(RNN) 으로, 과거 시계열 값을 이용하여 미래 값을 점진적으로 예측하는 방식을 따른다. 즉, 예측 시점에서는 실제 값이 존재하지 않으므로, 모델이 생성한 샘플링된 값을 입력으로 사용하여 점진적인 예측을 수행한다.

모델은 수요 예측, 공급망 관리, 금융 시장 분석 등 다양한 응용 분야에서 활용되며, 특히 샘플 개수가 적거나 새로운 시계열 데이터가 등장하는 경우에도 효과적으로 동작할 수 있는 강점을 가진다. 단순한 점 예측(point forecast) 대신 확률적 예측(probabilistic forecasting) 을 제공하여, 신뢰 구간을 고려한 보다 정교한 예측이 가능하다. 이를 위해 모델은 조건부 확률 분포(Conditional Probabilistic Distribution) 를 학습하여 시계열 데이터를 모델링하며, 주어진 상황에서 특정 시계열의 미래 값을 예측하는 확률 분포를 추정한다. 이는 다음과 같이 수식으로 표현된다.

여기서 는 시계열의 관측값, 는 보조 변수이며, 이후의 미래 시점에 대한 확률 분포를 과거 데이터 를 기반으로 학습한다.

이후 LSTM 기반의 다층 순환 신경망을 사용하여 과거 데이터를 요약하는 은닉 상태 를 생성하며, 네트워크의 은닉 상태는 다음과 같이 정의된다.

여기서 는 네트워크의 학습 가능한 파라미터이다. 이 구조는 오토회귀(Auto-Regressive) 방식을 따르며, 과거 데이터를 활용하여 점진적으로 미래 시점을 예측한다.

예측 단계에서는 확률 분포를 활용하여 미래 값을 Monte Carlo 샘플링을 통해 생성한다. 예측 값은 다음과 같이 확률 분포에서 샘플링된다.

여기서 는 특정 시점에서의 확률 분포를 의미하며, 모델이 학습한 조건부 확률 분포를 따른다. 즉, 모델은 다수의 샘플을 생성하여 확률적 예측을 수행하며, 이를 통해 신뢰 구간을 포함한 예측이 가능하다. 모델은 로그 가능도(log-likelihood)를 최대화하는 방식으로 학습되며, 손실 함수는 다음과 같이 정의된다.

특히 해당 모델은 다양한 유형의 시계열 데이터에 적용될 수 있도록, Gaussian, Negative Binomial 등 다양한 확률 분포를 적용할 수 있도록 설계되었다. Gaussian 분포는 연속적인 수치를 가지는 시계열 데이터에 적합하며, Negative Binomial 분포는 비정규 분포를 가지는 count data를 효과적으로 모델링할 수 있다.

Negative Binomial likelihood는 포아송 분포보다 더 유연한 분산을 가지며, 시계열 데이터에서 흔히 발생하는 오버디스퍼전(overdispersion, 분산이 평균보다 큰 경우) 을 효과적으로 처리할 수 있다. 이를 통해 다양한 데이터 특성을 반영한 보다 강건한 예측을 수행할 수 있다.

*텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.*

그림 9. DeepAR 구조

그림 9는 DeepAR의 훈련 및 예측 과정의 차이를 시각적으로 나타낸다. 좌측은 학습 과정을, 우측은 예측 과정을 나타낸다. 학습 과정에서는 각 시점 에서 보조 변수 , 이전 시점의 타겟 값 , 그리고 이전 네트워크 출력 를 입력으로 사용한다. 이를 기반으로 현재 시점의 은닉 상태 가 생성되며, 이후 해당 은닉 상태를 사용하여 확률 분포의 매개변수 를 계산한다. 이 확률 분포는 학습을 위한 손실 함수를 결정하는 데 사용된다.

예측 시에는 실제 값이 존재하지 않으므로 샘플링된 값을 입력으로 사용하여 점진적으로 시계열을 확장하는 오토회귀(auto-regressive) 방식으로 예측이 수행된다. 구간에서는 기존의 실제 데이터 를 입력으로 사용하지만 이후의 예측 구간에서는 모델이 샘플링된 값 를 입력으로 사용하여 점진적으로 예측을 생성한다. 이 과정은 예측 범위 부터 까지 반복되며, 하나의 샘플 경로 (sample trace)를 생성한다. 동일한 예측 과정을 여러 번 수행하여 Monte Carlo 샘플링을 통해 다수의 예측 경로를 생성하면, 미래 시점의 확률 분포를 근사하는 방법으로 활용할 수 있다. 이를 통해 모델은 단순한 점 예측(point forecast)이 아닌, 신뢰 구간을 포함한 확률적 예측(probabilistic forecasting) 을 수행할 수 있다.

따라서, 결론적인 모델의 주요 장점은 다중 시계열 데이터를 학습하여 예측 정확도를 높일 수 있다는 점이다. 점 예측 대신 확률적 예측을 수행함으로써 신뢰 구간을 포함한 예측이 가능하다. 오토회귀 기반의 예측을 사용하여 과거 데이터를 기반으로 점진적으로 시계열을 확장할 수 있으며, 외부 변수를 반영하여 더욱 정교한 예측을 수행할 수 있다. 특히, 수요 예측과 같이 간헐적(intermittent) 수요를 가지는 데이터에도 강건하게 동작하는 특성을 가진다.

* 1. *Model Explainability: SHapley Additive exPlanations (SHAP)*

블랙박스와 같은 인공지능의 예측 결과를 설명하기 위한 알고리즘인 SHAP(Shapley Additive exPlanations)는 LIME(local interpretable model-agnostic explanations)과 셰플리 값 (Shapley Value)를 연결하여 KTX 승차인원수 수요 예측에 대한 변수들의 기여 정도와 방향을 설명해준다. LIME은 임의로 데이터의 값을 변화시켰을 때 모델의 예측 결과의 변화를 추정하며 관련성을 계산한다. 그리고 게임 이론을 기반으로 개발된 셰플리 값은 이러한 변수들의 기여 정도를 계산하는 지표로 볼 수 있다. 따라서 변수들이 모든 조합에서 실제 값의 입력을 통해 생성되는 예측값의 변화 정도로 기여도를 추정한다. SHAP를 사용하여 KTX 승차인원수 예측에 기여하는 변수들의 정도와 기여 방향을 알 수 있고 복잡한 인공지능 알고리즘의 신뢰성을 높인다.

* 1. *Evaluation Metrics*

KTX 승차인원수의 예측 성능을 확인하고 설명의 신뢰성을 높이기 위해 6개의 검증지표로 확인한다. 이들은 RMSE(Root Mean Squared Error), MSPE(Mean Squared Percentage Error), MAE(Mean Absolute Error), MAPE(Mean Absolute Percentage Error), MedAE(Median Absolute Error), MedAPE(Median Absolute Percentage Error)이다. 모든 검증지표들은 실제 승차인원수와 예측된 값의 차이를 방정식으로 표현한 것으로써, 예측 성능이 좋을수록 낮은 수치들이 나오도록 구성되어 있다. 따라서 모든 검증지표의 수치가 낮을수록 KTX 승차인원수를 잘 예측하는 모델이라 볼 수 있다.

단, 승차인원수는 경부선의 경우 백만이 넘는 큰 수치이기 때문에 예측이 어느 정도 잘 되더라도 검증지표는 수치가 크게 나타날 수도 있을 것이다. 따라서 최적 변수의 조합이나 모델 선택, 그리고 한국철도공사 측에 예측 설명력을 이해시키는 경우에는 MSPE, MAPE, MedAPE 3가지의 검증지표를 주로 사용할 것이다. 왜냐하면 이 3가지의 지표들은 실제 승차인원수 대비 상대적인 차이를 퍼센트로 표현하기 때문에 이해하기가 수월하기 때문이다.

# Results

이번 섹션에서는 다양한 파생변수를 포함하는 44개의 독립변수와 종속변수인 승차인원수를 모델에 학습시킨 후 2025년도의 승차인원수 수요 예측 결과를 제시한다. 실제 승차인원수가 정확하게 예측이 되어야 왜 그러한 수치가 나타나게 되었는지 변수들의 기여도 또는 관련성과 같은 설명력의 신뢰성이 높아질 것이다. 선형회귀분석과 같은 전통적 모델은 예측수치를 위한 변수들의 가중치가 쉽게 설명되지만 실제 미래를 예측할 경우 성능이 낮아서 설명만 될 뿐 실제 비즈니스에 활용되기는 어렵다. 사회과학 분야는 과거 설명력에 집중하느라 미래에 얼마나 신뢰할 수 있는 수치인지 다소 경시하는 경향이 있다. 따라서 머신러닝과 딥러닝 알고리즘으로 미래 예측 결과를 우선적으로 확인하여 모델링 성능을 확인 후 다음으로 SHAP 알고리즘을 통해 변수들의 승차인원수 예측 설명력을 확인한다. 데이터 준비와 전처리, 그리고 모델링과 검증의 모든 데이터분석 프로세스는 python 3.9.20 버전을 사용하였다. 그리고 머신러닝과 딥러닝 알고리즘은 sklearn 1.5.2과 tensorflow 2.17.0 버전을 사용하였다. 마지막으로 설명력 제공을 위한 SHAP 알고리즘은 0.46.0 버전이 사용되었다.

* 1. *Performance and Prediction Process*

2025년도 승차인원수의 예측 전에 과거데이터로 간주되는 학습데이터와 미래데이터로 간주되는 검증데이터를 사용하여 머신러닝 및 딥러닝 알고리즘 모델 예측 성능을 검증한다. 대표적인 머신러닝 알고리즘인 Random Forest, XGBoost, LightGBM, 그리고 CatBoost와 대표적인 딥러닝 알고리즘인 MLP, RNN, LSTM, 그리고 GRU 알고리즘의 성능을 검증하였다. 공정한 비교를 위해 알고리즘의 파라미터들은 모두 동일하게 사용하였다. 총 8가지의 알고리즘으로 모두 검증기간에 대해 예측을 한 후 검증기간의 예측성능이 높은 알고리즘을 선택하여 테스트기간동안 예측을 하였다. 그림 3과 표 4에 경부선 승차인원수에 대한 검증기간 상위 5종 예측 시각화와 검증지표를 내림차순으로 제시하였다. 그림 3 (a)에 따르면 학습에 활용되지 않은 검증기간에서의 승차인원수를 가장 정확하게 예측한 알고리즘은 LightGBM 알고리즘이다. 그리고 실제 LightGBM 알고리즘을 사용하여 예측을 할 때, 미래 발생가능한 시나리오를 생성하기 위해 독립변수들의 값을 특정 표준편차 범위 내로 랜덤하게 변화시켰다. 총 10000번의 예측을 통해 각 월별 예측값의 범위인 신뢰구간이 측정되며 그 평균값을 계산할 수가 있다. 이것이 그림 3 (b)에서 노란색 음영과 빨간색 점선으로 표시되는 것이다. 이러한 과정을 통해 예측값이 발생할 수 있는 범위를 알 수 있고 여러가지 미래 시나리오를 반영 및 설명하는 예측이 가능하다. 퍼센트 검증지표인 MSPE, MAPE, 그리고 MedAPE의 평균치 기준 2.49% 밖에 되지 않은 낮은 수준의 예측 오류를 보인다 (표 4).

|  |
| --- |
| (a) 예측성능 상위5종의 승차인원수 예측 비교 |
| (b) 최고 예측성능인 LightGBM 알고리즘의 예측평균과 신뢰구간 |

그림 10. 경부선의 검증기간(2023년 4월 ~ 2024년 3월)에서의 예측성능 상위 5종 승차인원수 예측 시각화

표 2. 경부선의 검증기간(2024년 4월 ~ 2024년 3월)에서의 예측 성능지표 6종 순위

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **RMSE** | **MSPE** | **MAE** | **MAPE** | **MedAE** | **MedAPE** | **MSPE+MAPE+MedAPE** |
| **LightGBM** | **183,310** | **0.30%** | **148,672** | **4.39%** | **99,094** | **2.79%** | **2.49%** |
| **RandomForest** | 200,483 | 0.35% | 174,521 | 5.13% | 181,347 | 5.49% | 3.66% |
| **CatBoost** | 201,252 | 0.35% | 180,439 | 5.27% | 189,825 | 5.41% | 3.68% |
| **XGBoost** | 235,083 | 0.47% | 220,373 | 6.44% | 240,598 | 7.03% | 4.65% |
| **RNN** | 257,769 | 0.56% | 254,829 | 7.44% | 266,759 | 7.73% | 5.24% |
| **MLP** | 369,436 | 1.08% | 283,462 | 8.07% | 255,093 | 7.40% | 5.52% |
| **GRU** | 301,891 | 0.78% | 297,093 | 8.68% | 300,684 | 8.81% | 6.09% |
| **LSTM** | 307,550 | 0.80% | 299,494 | 8.74% | 311,268 | 9.20% | 6.25% |

검증 성능이 가장 높은 LightGBM 알고리즘을 사용하여 2025년의 승차인원수를 예측한다. 단, 학습된 모델을 그대로 사용하여 테스트 기간을 예측하는 것이 아니라, 학습 및 검증 기간을 모두 사용하여 LightGBM 알고리즘을 재학습 한 후 테스트기간을 예측하였다. 왜냐하면 이미 학습이 잘 된 모델링이라고 하더라도 검증기간을 제외하고 바로 테스트기간을 예측할 수야 있지만 너무 먼 미래를 예측하는 것이기 때문에 성능도 떨어질 뿐 아니라 최근 데이터를 학습에 사용하지 못하는 단점이 있기 때문이다. 따라서 학습데이터로 모델링 후 검증기간을 예측한 것처럼 학습 및 검증데이터를 모두 사용하여 모델링 후 예측기간의 수치를 확인하는 시계열 모델링을 진행하였다. 그 결과 경부선의 경우, 과거 350만명 정도의 승차인원수가 코로나 이슈를 통해 감소하다 회복중에 있으며 2025년도 2024년보다 근소하게 상승할 것으로 나타나고 있다. 그리고 약 325만~350만명 사이의 승차인원수 수요가 예상된다 (그림 4).

|  |
| --- |
| (a) 학습, 검증, 테스트기간 전체의 실제 승차인원수와 예측치 |
| (b) 테스트 기간인 2024년 3월부터 2025년 전체의 월별 승차인원수 예측치 |

그림 11. 경부선의 전체기간(2015년 1월 ~ 2025년 12월)에서의 월별 승차인원수 예측 시각화

* 1. *Forecasting of 2025 Year*

위와 같은 프로세스를 통해 나머지 KTX 노선별 예측결과를 확인한다. 5개 노선의 예측 결과 LightGBM 알고리즘이 2개의 노선에서, XGBoost 알고리즘이 2개의 노선에서, 그리고 LSTM 알고리즘이 1개의 노선에서 최고 검증 예측성능을 보였다 (표 3). 전라선의 경우 딥러닝 알고리즘이 성능이 가장 좋았지만, 나머지 경부선, 경전선, 동해선과 호남선은 머신러닝 알고리즘이 예측력이 좋았다. 그리고 MSPE, MAPE, MedAPE 2가지 검증지표의 평균값을 기준으로 예측 오류는 2.49% ~ 6.89%의 낮은 범위를 보이고 있다. 즉, 머신러닝 또는 인공지능 알고리즘을 통해 KTX 승차인원수를 5% 전후의 낮은 오류의 예측성능을 발휘할 수 있다.

표 3. KTX 각 노선별 예측 검증성능 최고 알고리즘과 검증지표 비교

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KTX** | **Algorithm** | **MSPE** | **MAPE** | **MedAPE** | **Average** |
| **경부선** | **LightGBM** | 0.30% | 4.39% | 2.79% | **2.49%** |
| **경전선** | **LightGBM** | 0.94% | 9.35% | 9.22% | **6.50%** |
| **동해선** | **XGBoost** | 0.91% | 9.18% | 8.40% | **6.16%** |
| **전라선** | **LSTM** | 0.99% | 9.77% | 9.92% | **6.89%** |
| **호남선** | **XGBoost** | 0.37% | 5.22% | 5.17% | **3.59%** |

그리고 경부선을 제외한 나머지 경전선, 동해선, 전라선, 그리고 호남선의 검증 및 테스트 예측 결과를 그림 5에 제시하였다. 모든 노선에서 머신러닝과 딥러닝 알고리즘이 전반적으로 승차인원수의 추세 뿐만 아니라 증가와 감소폭도 최대한 따라가기 위해 애쓰고 있음이 나타난다. 그리고 최종적으로 2025년의 KTX 승차인원수 수요는 경부선은 평균적으로 약 326만명, 경전선은 62만명, 동해선은 52만명, 전라선은 68만명, 그리고 호남선은 96만면 수준으로 예측되었다. 코로나 이전으로 볼 수 있는 약 5년 전인 2019년 대비 6.61~ 13.13% 수준으로 증가할 것으로 예상되며, 작년인 2024년 대비는 약 3.01 ~ 9.45% 수준의 감소가 예상되었다 .

|  |
| --- |
| (a) 경전선의 검증성능 및 예측 |
| (b) 동해선의 검증성능 및 예측 |
| (c) 전라선의 검증성능 및 예측 |
| (d) 호남선의 검증성능 및 예측 |

그림 12. 경전선, 동해선, 전라선, 호남선의 검증기간 및 테스트기간에서의 월별 승차인원수 예측 시각화

표 4. KTX 고속철도의 2025년도 승차인원수 예측 평균치와 코로나 전후의 과거 대비 증감율 비교

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| KTX | 2019년 | 2024년 | 2025년 | 증감율%(2019-2025) | 증감율%(2024-2025) |
| 경부선 | 3,000,064 | 3,418,760 | 3,261,319 | 8.71% | -4.61% |
| 경전선 | 552,302 | 675,125 | 624,792 | 13.13% | -7.46% |
| 동해선 | 470,231 | 573,080 | 521,482 | 10.90% | -9.00% |
| 전라선 | 580,489 | 759,476 | 687,733 | 18.47% | -9.45% |
| 호남선 | 905,066 | 994,880 | 964,904 | 6.61% | -3.01% |

* 1. *Explainability of 승차인원수 Forecating*

인공지능 알고리즘은 전통적인 선형회귀분석 형태의 알고리즘 보다 데이터의 복잡한 패턴과 변수들의 상호작용들을 모두 학습하기 때문에 구조가 복잡하여 승차인원수 예측에 기여하는 독립변수들의 기여도를 쉽게 확인하기 어렵다. 하지만 SHAP 알고리즘을 사용하면 미래 승차인원수 예측치가 어떤 독립변수들의 상대적 기여로 계산되었는지 추정할 수 있다. 따라서 2025년도 KTX 고속철도의 승차인원수 예측 뿐만 아니라 신뢰할 수 있는 설명력을 시각적으로 제공해준다.

SHAP는 특정 시점의 독립변수들의 샘플 수치들에 대해 승차인원수가 얼마일지 예측할 뿐만 아니라 각 독립변수들의 수치값과 기여 방향을 제시해준다. 이러한 결과를 누적한다면 승차인원수에 대한 독립변수들의 일반화된 기여 설명력을 확인할 수 있다. 단, 본 연구에 활용된 샘플 시점들 결과의 누적기준 일반화이기 때문에 샘플 시점이 향후 늘어난다면 해석이 변경될 수도 있다. 샘플의 양이 많아진다면 더욱 신뢰할 수 있는 설명력을 가질 것은 분명하다. 그림 5는 표 3에서 추정된 노선별 예측 최고성능 알고리즘을 기준으로 추정된 SHAP의 독립변수 기여도를 시각화 한 것이다.

(왼쪽)세로축은 승차인원수에 영향을 주는 독립변수의 우선순위를 내림차순으로 정렬한 것이다. 즉, 경부선을 기준으로 “좌석당단가, 좌석회전율, 공급차량수, 1열차당승차인원, 국가이동제한정도” 등의 순서로 승차인원수 예측에 기여를 하고 있다. (오른쪽)세로축은 변수들의 값이 낮은/높은 경우 파란색/빨간색 계열로 표시하여 값에 대응되는 승차인원수의 예측값의 변화를 표시하였다. 마지막으로 가로축에 승차인원수의 예측값을 표시하였다. SHAP 값이 0보다 작으면 부정적인 기여를 0보다 크면 긍정적인 기여를 의미한다. 각 시점마다 변수의 수치는 다양하게 분포할 수 있고 각 수치별 승차인원수에 대한 변수들의 기여 방향도 얼마든지 변화될 수 있다. 따라서 변수들의 값과 대응되는 SHAP 예측치를 모두 누적하여 표현하면 변수 값의 변화에 따른 승차인원수 기여 방향을 확인할 수 있다.

경부선의 “1좌석당단가”는 값이 작을때(파란색계열) 승차인원수의 부정 기여에 많이 분포되어 있고 값이 커지면(빨간색계열) 승차인원수의 긍정 기여에 많이 분포 되어 있다. 따라서 1좌석당단가가 비싸질수록 승차인원수를 높이는 영향을 준다. 반대로 코로나가 발생하여 “국가이동제한정도”가 증가하면 승차인원수를 낮추는데 기여한다. 따라서 미래의 승차인원수에 증가에 가장 기여를 많이 하는 변수의 상위 3가지 변수는 "1좌석당단가, 좌석회전율, 공급차량수"이며 승차인원수 감소에 가장 기여를 많이하는 변수 상위3가지는 "국가이동제한정도, 코로나진행정도, 승차인원수\_lag3"로 볼 수 있다. 그런데 이러한 변수들의 기여정도는 다른 노선에선 다르게 나타난다. 경전선의 경우 승차인원수를 증가시키는 변수 상위 3가지는 경부선과 달리 “승차인원수\_lag1, 1열차당승차인원, 열차운행횟수”로 나타났다. 반면 승차인원수를 감소시키는 변수 상위 3가지는 “승차인원수\_lag12, 격리된자수, 1인당단가”가 해당된다. 마찬가지의 해석을 다른 노선들에서도 가능하지만, 중요한 것은 특정 변수가 모든 노선에서 긍정 또는 부정적 기여를 한다고 일반화 하는것은 위험하다는 것이다. 따라서 더욱 실제 비즈니스에 활용하기 위해선 실시간으로 정량적인 예측과 설명력을 빠르게 추정하여 활용하는 것이 훨씬 용이하다. 그러한 목적을 위해 본 연구는 누구나 현업에서 빠르고 쉽게 사용할 수 있는 설명가능한 인공지능을 활용한 비즈니스 애널리틱스 방법론을 제시하였다. 승차인원수를 높은 성능으로 미리 예측할 뿐 아니라 그 원인을 설명함으로써 비즈니스의 활용도를 높일 수 있다.

|  |  |
| --- | --- |
| (a) 경부선 | (b) 경전선 |
| (c) 동해선 | (d) 전라선 |
| (e) 호남선 | |

그림 13. KTX 노선별 승차인원수 예측에 대한 변수들의 기여 설명력 기반 의사결정

# Conclusions

현재 25개국 이상이 고속철도를 도입하여 주요 교통수단으로 활용하고 있으며, 이러한 국가는 고속철도를 통해 물류와 인구의 이동성을 극대화하며 교통체계의 효율성과 환경적 지속 가능성을 동시에 달성하고 있다. 특히, 고속철도는 전기차보다 12배, 내연기관 차량보다 26배 낮은 CO2 배출량을 기록하며, 지속 가능한 교통수단의 대표 사례로 자리 잡고 있다. 국토교통부는 미래 교통 시스템의 효율성을 제고하기 위해 수요 예측과 연계된 미래 선도 기술 개발을 적극적으로 추진하고 있다. 이러한 맥락에서, 고속철도의 수요 변화를 정밀하게 예측하는 것은 단순한 운영 효율성 개선을 넘어, 장기적인 교통 정책 수립과 지속 가능한 인프라 구축을 위한 핵심 요소로 작용한다. 효과적인 수요 예측을 위해서는 고속철도 이용 패턴을 정확히 반영할 수 있는 정교한 분석 기법이 필수적이다.

본 연구는 고성능 인공지능 알고리즘과 설명 가능한 인공지능(XAI, Explainable AI)을 활용하여 2025년도 KTX 수요를 정밀하게 예측하고, 예측 결과를 바탕으로 실질적인 비즈니스 및 정책적 의사결정을 지원하는 것을 주된 목적으로 3가지 항목에서 기여하고 있다. 첫째, AI 알고리즘을 활용하여 KTX 수요 예측의 정확도를 대폭 향상시킨다. 기존의 통계 기반 모델이 가지는 예측 성능의 한계를 극복하고, 고성능 알고리즘을 통해 비선형적 복잡한 데이터 패턴을 학습함으로써 더욱 현실성 높은 수요를 예측한다. 이를 통해, KTX 수요 변화의 복잡한 양상을 효과적으로 예측하고, 미래 교통 계획 수립에 기여할 것이다. 둘째, XAI 기술을 통해 높은 미래 예측 성능 뿐만 아니라 설명력을 동시에 제공함으로써 신뢰성을 높인다. SHAP(Shapley Additive Explanations) 알고리즘을 활용하여 예측 결과와 변수들의 기여 정도를 시각화 함으로써, 비즈니스적 정책 설계 및 자원 배분 의사결정 과정에서 신뢰도를 높인다. 셋째, 고성능 컴퓨팅 환경 없이도 실행 가능한 대표적인 머신러닝과 딥러닝 알고리즘을 활용하여, 한국철도공사 운영, 관리 및 정책 담당자가 현업에서 쉽고 간편하게 수요 예측 결과를 확인하고 실시간으로 의사결정을 내릴 수 있도록 활용도를 강화하였다.

결론적으로 본 연구는 AI 기반 수요 예측의 높은 정확성과 설명력을 통해 2025년도 KTX 수요 예측의 신뢰성과 실효성을 극대화하였으며, 데이터 기반 의사결정이 보다 효율적으로 활용가능한 방식으로 이루어져 지속 가능한 교통체계 구축에 도움이 될 것이다. 향후 다양한 교통 체계의 수요예측에도 활용가능한 연구 방향을 제시함으로써 교통 전반의 데이터기반 효율성을 높이는 데 중요한 학문적·실무적 시사점을 제공할 것이다. 또한 한국철도공사의 도메인 지식과 추가적인 고성능 AI 알고리즘을 활용하여 더욱 정교한 수요예측을 달성하는 것을 향후 과제로 남겨둔다.

# References

Alex, G., Alex, G. (2012). Long short-term memory. <Supervised sequence labelling with recurrent neural networks>, 37-45.

David, S., Valentin, F., Jan, G., Tim, J. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. <International journal of forecasting>, **36**, 1181-1191.

Kyunghyun, C., Merriënboer,, V., Caglar, G., Dzmitry, B., Fethi, B., Holger, S., Yoshua, B. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. <arXiv preprint arXiv:1406.1078>, .

Shaojie, B., J, K., Vladlen, K. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. <arXiv preprint arXiv:1803.01271>, .

Shiyu, C., Yang, Z., Wei, H., Mo, Y., Xiaoxiao, G., Wei, T., Xiaodong, C., Michael, W., Mark, H. J., Thomas, H. (2017). Dilated recurrent neural networks. <Advances in neural information processing systems>, **30**, .

Union, I. (2023). HIGH-SPEED RAIL, the right speed for our planet - Key Messages. <11th UIC HS Congress>, **1**, 24.

기술과가치 (2020). 국토교통 2050 미래기술 도출을 위한 조사분석 연구. <국토교통과학기술진흥원>, .

Abhimanyu, D., Weihao, K., Andrew, L., Shaan, M., Rajat, S. and Rose, Y. (2023). Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, .

Boris, O., Dmitri, C., Nicolas, C. and Yoshua, B. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, .

Cha, H., Oh, Y., Song, J. and Lee, T. (2019). KTX passenger demand forecast with multiple intervention seasonal ARIMA models. *응용통계연구*, **32**, 139-148.

Cristian, C., Kin, O., Boris, O., Federico, R., Max, C. and Artur, D. (2023). Nhits: Neural hierarchical interpolation for time series forecasting. *Proceedings of the AAAI conference on artificial intelligence*, **37**, 6989-6997.

David, G., Mark, S., Jaesik, C., Timothy, M., Simone, S. and Guang-Zhong, Y. (2019). XAI—Explainable artificial intelligence. *Science robotics*, **4**, eaay7120.

Jang, J. H. and Kim, M. K. (2022). An Explanatory Analysis of Electricity Demand Prediction Using XAI and N-BEATS Models. *대한전기학회 학술대회 논문집*, 277-278.

Jeong, H. and Lim, C. (2019). A review of articial intelligence based demand forecasting techniques. *응용통계연구*, **32**, 795-835.

Jung, Y. and Cheolhee, Y. (2024). A Study on Forecasting Order Quantity from Manufacturing Supply Chain Data using XAI. *Journal of Korean Institute of Information Technology*, **22**, 41-53.

Kim, K. and Kim, H. (2011). KTX Passenger Demand Forecast with Intervention ARIMA Model. *Journal of the Korean Society for Railway*, **14**, 470-476.

Kin, O., Cristian, C., Grzegorz, M., Rafał, W. and Artur, D. (2023). Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, **39**, 884-900.

Kwon, J. (2023). A Study on the Applicability of eXplainable Artificial Intelligence(XAI) Methodology by Industrial District. *Global Business Administration Review*, **20**, 195-208.

Rafique, A. and Ali, I. (2024). Knee Osteoarthritis Analysis Using Deep Learning and XAI on X-rays. *IEEE Access*, .

Sima, S. N., Neda, T. and Akbar, N. (2018). A comparison of ARIMA and LSTM in forecasting time series. *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, 1394-1401.

Syama, R., Jan, G., Lorenzo, S., Valentin, F., David, S., Yuyang, W. and Tim, J. (2023). Deep Non-Parametric Time Series Forecaster. *arXiv preprint arXiv:2312.14657*, .

심진호, 오윤식, 오영택 and 금수희 (2024). A study of high-speed railway demand forecasting using nonlinear data pattern learning. *한국철도학회 학술발표대회논문집*, 114-114.