

Transformer Based Arrival Time Prediction for a Target Bus Stop Using Single Stop Information

Ga-On Cheon*, Young-Sup Hwang**

*Student, Division of Computer Science and Engineering, Sunmoon University, Asan, Korea

**Professor, Division of Computer Science and Engineering, Sunmoon University, Asan, Korea

[Abstract]

This study proposes a prediction method for estimating the arrival time at a destination stop when a passenger boards at any stop on the route. The data used in this work are DTG records collected every second, and we transformed these records into arrival times and travel times for each stop through preprocessing steps such as stop matching, order correction, and outlier removal. The proposed model is based on a Transformer Encoder and receives the arrival time, weather information, and location of the single stop where the passenger boards. The model predicts the travel time for each section between stops and computes the ETA to the destination by accumulating these predicted times. With this design, the model achieved superior prediction accuracy on routes with longer stop sequences compared to Curriculum learning based LSTM Model, while reducing training time by more than 50%. Experiments confirmed that destination arrival time can be accurately predicted using information from a single stop.

▶ **Key words:** Arrival Time Prediction, Transformer Model, Deep learning, DTG, Preprocessing

[요 약]

본 논문은 이용자가 노선의 어느 정류장에서든 임의로 탑승했을 때, 목적지 정류장의 도착 예정 시각(ETA)을 추정하기 위한 새로운 예측 방식을 제안한다. 본 연구에서 사용한 데이터는 초단위로 기록된 DTG 운행 데이터이며, 이를 정류장 단위의 도착 시각과 구간 이동 시간으로 재구성하기 위해 정류장 매칭, 순서 보정, 이상치 제거로 구성된 전처리 절차를 설계하였다. 제안 모델은 Transformer Encoder 구조를 기반으로 하며, 이용자가 실제로 탑승하는 단일 정류장의 도착 시각과 기상정보 그리고 위치를 입력으로 받는다. 모델은 출력으로 전체 정류장의 구간별 이동 시간을 예측하고, 예측된 구간 이동 시간을 누적하여 목적지까지의 ETA를 산출한다. 이러한 설계를 통해 제안 모델은 Curriculum learning 기반 LSTM 모델과 비교하여 상대적으로 긴 노선에서 더 우수한 예측 정확도를 유지하면서도, 학습 시간은 약 50% 단축하였다. 실험을 통해 단일 정류장 정보만으로도 목적지 도착 시간을 효과적으로 예측할 수 있음을 검증하였다.

▶ **주제어:** 도착 시간 예측, 트랜스포머 모델, 딥러닝, DTG, 전처리

-
- First Author: Ga-On Cheon, Corresponding Author: Young-Sup Hwang
 - *Ga-On Cheon (gaon03425@gmail.com), Division of Computer Science and Engineering, Sunmoon University
 - **Young-Sup Hwang (young@sunmoon.ac.kr), Division of Computer Science and Engineering, Sunmoon University
 - Received: 2025. 12. 26, Revised: 2026. 01. 17, Accepted: 2026. 02. 16.

I. Introduction

현대 도시에서 버스는 상대적으로 낮은 비용과 넓은 운행 범위를 바탕으로 가장 보편적인 교통수단으로 자리 잡고 있다. 이에 따라 대중교통의 운영 효율을 높이기 위한 노선 설계, 배차 계획, 정시성 확보, 정보 제공 시스템, 수요 예측 등이 교통 운영 체계의 핵심 주제로 다뤄지고 있으며, 이러한 요소들은 전체 운영 성능에도 직접적인 영향을 미친다[1-4]. 이러한 요소들 중에서 정확한 정류장 도착 시간 예측은 승객의 만족도[5]뿐 아니라 운영 효율성, 나아가 스마트시티형 교통관리 체계의 기반 기술로서 매우 중요한 역할을 한다[6-7]. 그러나 실제 도로 위에서의 버스 운행은 다양한 변수, 예를 들어 차량 정체, 신호 체계, 기상 변화, 정류장별 승하차 인원 등에 의해 영향을 받으며, 이는 운행 시간의 불확실성을 유발한다[8].

이러한 필요성에 대응하여, 최근에는 GPS 기반의 공공 교통 데이터를 활용하여, 특정 정류장에 버스가 언제 도착할 것인지를 예측하는 연구들이 활발히 이루어지고 있다. 그러나 대부분의 연구는 출발지부터 종착지까지 전체 경로의 도착 시각을 예측하거나[9-13], 각 정류장을 구간으로 나누어서 구간마다 소요 시간[14-15]을 예측한다. 구간 기반 방식은 정류장을 링크 단위로 나누어 예측한 후 누적하는 구조이기 때문에, 구간마다의 입력 정보가 필요하고, 예측 단위가 구간 소요 시간에 한정된다. 이러한 방식은 구간 단위의 변화는 반영할 수 있으나 정류장 체류 시간이나 승하차 지연과 같은 요소를 고려할 수 없다. 또는 현재 위치를 기준으로 다음 정류장 또는 인접 정류장까지의 도착 시간을 예측한다[16-17]. 이는 출발지-도착지 쌍이 고정되어 있거나, 사용자가 탑승하는 시점을 출발 지점으로 간주하는 접근 방식을 따른다. 이러한 설정은 자연스럽지만, 기존 연구들은 주로 이 출발 지점을 기준으로 인접한 몇 개의 정류장에 대해서 예측하므로 예측 범위가 제한적이다. 현실적인 상황에서는 사용자가 임의의 정류장에 도착한 상태에서 버스에 탑승 후 특정 정류장에 언제 도착할지를 알고자 하는 경우가 일반적이다. 특히 도로 정체, 신호 대기, 날씨 등 실시간 교통 상황에 따라 버스는 각 정류장에 도착할 때마다 이후 도착 시간이 유동적으로 변동된다[18-19]. 따라서 이러한 문제는 기존 연구에서 가정하는 고정된 출발-도착 구조와는 다르며, 도착 시점을 기준으로 이후 경로의 누적 도착 시간을 실시간 맥락에서 추정해야 한다는 점에서 새로운 문제 정의가 필요하다. 최근 연구에서는 운행 과정에서 관측되는 데이터를 반영하여 오차를 줄이는 실시간 업데이트 기반 접근이 제안되고 있으며, 이

는 동적 교통 상황에 대응하기 위한 중요한 방향으로 주목받고 있다[20-22].

이러한 현실적인 시나리오를 고려하여, 임의의 정류장에 도착한 시점을 입력으로 하여, 이후 목적지 정류장까지의 도착 시간을 예측하는 모델을 제안한다. 이는 단일 정류장의 정보 (정류장 위치, 탑승 시간, 날씨, 요일, 날씨 등)만으로 정류장 간 소요 시간을 예측하고 일부 구간의 누적 도착 시간을 추정하는 문제로 정의할 수 있다.

본 연구는 이러한 문제를 해결하기 위해 Transformer Encoder 기반 구조를 적용하였다. 천안시 1번과 13번 버스 노선의 1년간 수집된 DTG 데이터를 활용하여, 정류장 간 거리, 시간, 기상 등 다양한 특성을 통합한 시계열 입력을 구성하고, 단일 정류장의 도착 정보를 기반으로 전체 경로 중 특정 구간의 도착 시간을 예측하는 모델을 설계하였다. 제안하는 모델은 단일 정류장 정보만을 입력으로 사용하면서도, Curriculum learning 기반 LSTM 및 Transformer Encoder 모델과 동등하거나 더 우수한 예측 정확도를 보였으며, 학습 시간은 약 50% 단축되었다. 또한 누적 구간이 증가하는 긴 노선 조건에서도 예측 오차가 급격히 증가하지 않았으며, 특히 누적 구간이 길어질수록 예측 오차 증가 폭이 상대적으로 작게 나타나 장거리 노선에 대한 안정적인 적용 가능성을 보인다.

2장에서는 버스 도착 시간 예측과 관련된 선행 연구를 검토하고, 3장에서는 본 연구에서 제안하는 데이터 처리 프레임워크, 모델 구조, 실험 설계 및 결과를 제시한다. 마지막으로 4장에서는 본 연구의 결론을 정리한다.

II. Preliminaries

1. Related works

1.1 Prediction Based on OD(Origin-Destination)

OD 기반 연구에서는 출발 정류장에서 도착 정류장까지 전체 경로의 각 정류장 도착 시간을 한 번에 예측한다. 출발 시간이 입력으로 주어지고 도착까지의 각 정류장까지의 소요 시간을 예측한다. 이 방식은 경로 전체의 예측을 한 번에 수행할 수 있다는 장점이 있으나, 중간 구간의 교통 변화 반영에는 한계가 있다. 일부 연구에서는 LSTM 기반 하이브리드 모델로 시간적 특성을 추출하고, ANN으로 공간적 특성을 학습한 뒤 PCA를 통해 통합하여 각 정류장의 도착 시간을 예측하였다[11]. 다른 연구에서는 HTF-NET을 설계하여, 기상 데이터를 입력에 포함함으로써 도착 시간 예측 정확도를 향상했다[12]. 또 다른 연구에

서는 ConVLSTM을 활용하여 주행 시간과 대기 시간을 분리해 모델링하고, Attention 메커니즘과 결합하여 정류장별 소요 시간을 추정하였다[13].

1.2 Prediction Based on Link

Link 기반 접근은 전체 경로를 여러 개의 구간(정류장 간 링크)으로 나누고 각 링크의 소요 시간을 예측한 후 합산하여 전체 도착 시간을 산출한다. 즉, 예측의 단위가 구간이며, 구간별 예측값을 합쳐 최종 경로 도착 시간을 산출하는 구조이다. 이 접근은 구간별 변화를 반영할 수 있으나, 각 링크 예측 정확도가 전체 예측 성능에 직접적인 영향을 미친다. 기존 연구에는 전체 경로를 정류장 간 링크 단위로 분할하고, 각 링크의 소요 시간을 예측하기 위해 SVM과 Kalman Filter를 결합하는 방법을 제안하였다[14]. 다른 연구에서는 각 링크의 소요 시간을 예측하기 위해 최근 및 과거 운행 데이터를 활용한 텐서 분해 기반의 결측치 보완 기법과 최적 구간 연결 알고리즘을 결합하여 예측하였다[15]. 그러나 링크 기반 방식은 각 구간의 예측을 위해 구간마다의 GPS 위치, 과거 링크 소요 시간, 속도 정보 등 다양한 입력이 요구된다. 또한 각 구간에서 발생한 예측 오차나 잘못된 입력이 누적되면서 전체 경로 도착 시간에 잘못된 정보가 반영될 수 있다. 이는 단일 정류장의 정보만으로 전체 경로 도착 시간을 예측하는 본 연구와는 차이가 있다.

1.3 Prediction Based on Adjacent Stops

인접 정류장 기반 접근은 현재 위치 또는 직전 정류장의 도착 시각을 입력으로 사용하여, 다음 정류장까지의 도착 시간을 추정하는 방식이다. 이 접근은 인접 정류장의 도착은 예측할 수 있지만, 비인접 정류장의 도착 시간은 사전에 추정하기 어렵다는 한계가 존재한다. 일부 연구에서는 은닉 마르코프 모델을 적용하여 관측된 구간 속도로부터 이후 교통 상태를 추정하고, 그 결과를 기반으로 인접 정류장의 도착 시간을 산출하였다[16]. 또한 서울시 BIS 데이터를 활용하여 LSTM 기반 모델을 적용하여 Kalman Filter나 신경망 기반 방법보다 높은 정확도를 기록하며, 인접 정류장 단위 예측 성능을 개선하였다[17]. 그러나 이러한 접근은 가까운 몇 개의 정류장만을 예측 대상으로 삼기 때문에, 멀리 떨어진 목적지에 대한 누적 도착 시간을 추정하기 어렵다는 한계를 갖는다.

1.4 Prediction Based on Real-Time Update

실시간 업데이트 기반 접근은 운행 중 새로운 관측 데이

터가 들어올 때마다 예측을 보정하여 누적 오차를 줄이는 방식이다. 이 방법은 지연 시간을 실시간으로 반영할 수 있어 상대적으로 높은 정확도를 확보할 수 있다는 장점이 있다. 한 연구에서는 운행에서 이전 정류장 도착 시간을 입력으로 사용하여 남은 모든 정류장의 도착 시간을 재예측하는 방식을 사용하여 여러 모델을 비교하였다[20]. 다른 연구에서는 출발 전에 Attention과 LSTM을 통해 초기 예측을 수행하고, 운행 중에는 Kalman Filter를 적용하여 직전 정류장의 예측 오차를 다음 정류장 도착 시간 예측에 반영함으로써 누적 오차를 완화하였다[21]. 또 다른 연구에서는 ConVLSTM을 활용하여 경로를 링크로 분할하고, 각 링크의 소요 시간을 실시간 데이터로 업데이트하여 다른 정류장의 도착 시간을 예측하였다[22]. 기존의 실시간 업데이트 기반 연구들은 운행 과정에서 지속적인 새로운 관측 데이터가 필요하지만, 본 연구는 단일 정류장 정보만으로도 임의의 정류장까지의 도착 시간을 예측할 수 있다는 점에서 차별성을 갖는다. 이러한 특성은 관측 정보가 부족하거나 일부만 제공되는 환경에서도 모델을 적용할 수 있다.

III. The Proposed Scheme

Table 1. Example of input features for a bus stop

Category	Feature	Example Value
Climatic	Relative Humidity	78.1 %
	Temperature	25.4 °C
	Wind direction	93 °
	Wind speed	0.8 m/s
Temporal	Departure time	2020-06-01 09:32
	Weekend/Holiday	False
	Day of week	Monday
Spatial	Prev stop distance	87 m
	Next stop distance	697 m
	Latitude	36.79431
	Longitude	127.10368
Other	Stop ID	288002022
	Bus number	1445
	Last stop(1 if true)	1

1. Dataset

1.1 Selection of Bus Routes

본 연구에서는 실제 운행 중인 천안시 도시 버스의 1번

및 13번 노선을 선정하였다. 1번 노선은 천안 지역의 공공 및 민간 시설을 고루 포함하며, 보건소, 병원, 백화점, 초·중·고등학교, 한국전력공사, 고속터미널 등 생활·교육·행정·교통 시설이 혼합된 복합적인 경로 특성을 가진다. 또한 왕복 약 43km의 운행 거리를 포함하고 있다. 13번 노선은 총 16.1km를 운행하며, 평균 소요 시간은 약 40분이다. 이 노선은 경찰서, 대학교, 두 개의 병원, 시장, 도서관, 초·중학교, 광장 등 주요 거점을 연결하는 중심 대중교통 노선으로 기능하고 있다. 실험에서는 길이와 특성이 다른 두 노선을 사용하여 적절한 비교 균형을 제공한다. 또한 지나치게 짧거나 긴 노선을 피함으로써 현실적인 교통 상황에서의 예측 가능성을 검증할 수 있는 노선을 선정하였다.

1.2 Data Description

표 1은 버스 정류장 단위의 입력 데이터를 구성하는 주요 변수들을 정리한 것이다. 사용된 운행 데이터는 2020년 동안 천안시에서 운행된 버스의 GPS 데이터를 기반으로 하며, 정부 공공데이터 요청을 통해 수집되었다. 또한 운행 시간에 영향을 미칠 수 있는 외부 요인을 반영하기 위해 기상청(KMA)의 기상 데이터와 공휴일 정보, GPS 좌표 기반의 정류장 간 거리 데이터를 추가로 통합하였다.

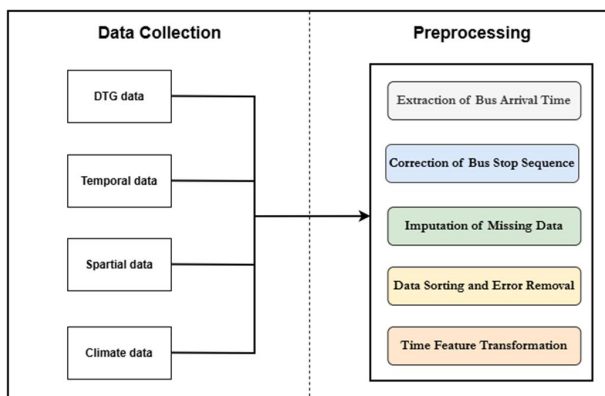


Fig. 1. Data Preprocessing Framework

1.3 Data Preprocessing

본 연구에 사용된 DTG(Digital Tachograph) 데이터는 운행 기록이 초 단위로 수집된 대용량 시계열 데이터이다. 특히 DTG 데이터는 GPS, 속도, RPM 등 차량 상태를 1초 단위로 기록하므로 대량의 데이터를 효과적으로 분석하기 위해서는 전처리와 변환 과정이 필수적이다[23]. 이러한 방대한 데이터는 운행 단위가 아닌 차량 단위로 제공되며, 정류장 위치와의 매칭, 순서 보정, 이상치 처리 등

의 정교한 전처리 과정을 요구한다. 이에 본 연구에서는 그림 1과 같이 총 6단계의 전처리 절차를 수행하였다.

1.3.1 Extraction of Arrival Time at Bus Stops

버스의 GPS 위치 정보를 활용하여 버스의 각 정류장 도착 시점을 추출하였다. 이를 위해 각 정류장의 위도·경도 좌표를 기준으로, 버스 위치와의 유클리드 거리를 계산하였다. 예를 들어, 정류장 A의 좌표에서 2.5m 떨어진 위치에서 기록된 GPS 정보는 A 정류장에 도착한 것으로 처리된다.

1.3.2 Adjustment of Bus Stop Indices

데이터는 노선 단위 데이터가 아닌 정류장 도착 데이터이기 때문에 정류장 도착 데이터로부터 ETA 예측 모델 학습을 위한 운행 단위 시퀀스를 구성하기 위해, 정류장 도착 데이터들을 노선의 1번 정류장부터 시작하여 버스의 순차적인 정류장 도착 정보를 확인한다. 이를 위해 각 버스의 GPS 위치 기록을 시간순으로 정렬한 후, 노선의 1번 정류장부터 시작하여 버스의 순차적인 정류장 도착 여부를 판단하였다. 천안시 버스 노선은 상행을 따라 이동한 뒤 종점을 지나 다시 하행 방향으로 되돌아오는 형태로 구성되어 있어, 실제 도로에서는 상·하행 정류장이 서로 가까운 위치에 배치되는 경우가 많다. 상·하행 노선의 정류장이 매우 가까운 위치에 있는 경우, GPS 오차로 인해 상행 정류장을 하행 정류장으로 잘못 인식하는 문제가 발생할 수 있다. 이러한 상·하행 정류장 오인식을 완화하기 위해 정류장 순서를 고려한 보정 규칙을 적용하였다. 현재 정류장 번호 S_{curr} 가 이전 정류장의 번호 S_{prev} 보다 크고 그 차이가 최대 허용 범위 k 이내일 경우에는 정상적인 순서로 판단하여 그대로 유지하였다:

$$\text{If } S_{curr} > S_{prev} \text{ and } (S_{curr} - S_{prev} \leq k) \quad (1)$$

실제 DTG 데이터 분석 결과, 전체 운행 시퀀스의 99.57%에서 상·하행 정류장 오인식으로 인한 정류장 번호 보정이 최소 한 번 이상 발생하였으며, 전체 데이터의 6.2%에서 보정이 발생하였다. 정류장 순서 보정에서 임계값 k 는 이전 정류장 대비 현재 정류장 번호 차이를 허용하는 범위를 의미한다. k 를 1과 2와 같은 작은 값을 적용했을 때 실제 DTG 데이터에서 정상 운행임에도 불구하고 GPS 오차 및 기록 간격 등의 영향으로 정상 운행 데이터가 순서 오류로 과도하게 제거되는 문제가 발생하였다. 반면, $k = 6, 9$ 와 같이 큰 값을 적용했을 때 상·하행 정류장 오인식이 정상 이동으로 허용되어 정류장 번호가 잘못 지

정되었고, 이에 따라 보간에 필요한 이전·이후 정류장 정보가 소실되면서 실제 사용할 수 있는 운행 데이터 수가 감소하였다. 이에 본 연구에서는 정상 운행 데이터의 불필요한 제거를 방지하면서도 보간 안정성을 유지할 수 있는 균형점으로 $k=3$ 을 임계값으로 선정하였다.

식 (1)의 조건을 만족하지 않을 경우, 반대 노선으로 잘못 분류된 것으로 간주하고, 마지막 정류장 S_{last} 에서 현재 판단된 정류장의 값에서 1을 빼서 정류장 번호를 수정하였다. 예를 들어, 56개의 정류장이 있는 노선에서 하행 운행 중 45번 정류장에 도착했지만, 현재 판단된 정류장 번호가 상행 번호 12인 경우, 이를 $56-(12-1)=45$ 로 보정하여 상·하행 오류를 조정한다.

$$S_{corrected} = S_{last} - (S_{curr} - 1) \quad (2)$$

1.3.3 Interpolation of Missing Arrival Times

DTG(Digital Tachograph) 데이터는 GPS 수신 불량, 센서 오류 등 다양한 원인으로 인해 일부 정류장의 도착 시간이 누락되는 경우가 발생한다. 이런 결측값은 시계열 흐름의 불연속성을 유발하여 예측 모델의 안정성과 성능에 영향을 줄 수 있으므로 결측된 값을 보간하여 운행 데이터의 연속성을 유지하였다. 또한 본 연구에서 정류장 도착 시간의 누락 비율은 전체 데이터의 약 0.58%로 나타났고, 이러한 환경에서는 평균 속도 기반 보간이 전체 학습 및 평가 결과에 미치는 영향이 제한적이라고 판단하였다. 이에 전처리 안정성을 위해 평균 속도 기반 보간 방식을 적용하였다. 먼저, 결측 이전 정류장 도착 시간과 다음 정류장 도착 시간의 간격 Δt 를 계산하였다.

이 구간의 총 이동거리 D_{total} 은 해당 구간을 구성하는 개별 정류장 간 거리 d_i 의 합으로 계산되며,

$$D_{total} = \sum_{i=1}^n d_i \quad (3)$$

이를 바탕으로 평균 속도 v_{avg} 는 다음과 같이 정의된다.

$$v_{avg} = \frac{D_{total}}{\Delta t} \quad (4)$$

이때, 도착 시간이 누락된 정류장 구간의 거리 $d_{missing}$ 을 곱하여 해당 구간의 예상 소요 시간 Δt_{fill} 을 추정한다.

$$\Delta t_{fill} = v_{avg} \cdot d_{missing} \quad (5)$$

예를 들어, 빠진 정류장의 이전 정류장과 다음 정류장의 도착 시각이 각각 09:00:00과 09:04:00이며, 정류장 사이 이동 거리가 600m일 경우, 평균 속도는 2.5m/s가 되며,

빠진 정류장까지의 거리가 400m일 때, 보간된 시간은 약 160초로 계산된다.

1.3.4 Data Sorting and Outlier Removal

데이터 정렬 과정에서는 중복된 정류장 번호, 누락된 번호, 비정상적인 운행을 식별하여 제거하고 데이터를 시간 순서로 다시 정렬하였다. 또한 일부 버스 운행 기록에서는 장시간 동안 도착 정보가 기록되지 않은 구간이 존재하였는데, 이는 비정상적인 데이터로 간주하여 연속된 정류장 간 도착 시간 차이가 6분(360초)을 초과하는 데이터를 삭제하였다. 예를 들어, 특정 버스가 10번 정류장에서 08:10에 도착하고 다음 11번 정류장에서 08:19에 도착한 경우, 소요 시간이 9분으로 비정상적으로 판단된다. 이 데이터는 사고나 특정 상황에서 실제로 발생하지만, 학습에 불필요한 영향을 줄 수 있으므로 이상치로 간주하고 제거하였다.

1.3.5 Transformation of Time Features

모델이 시간의 흐름에 따른 패턴을 효과적으로 학습할 수 있도록, 정류장 도착 시간 정보를 하루의 초 단위로 분해하였다. 이후 시간 데이터의 주기성을 반영하기 위해 시간 단위에 대해 \sin 변환과 \cos 변환을 적용하였다. 0시와 23시는 숫자로 보면 0과 23으로 멀리 떨어져 있지만, 실제로 연속된 시간이므로, 이를 주기성을 고려하여 시간 값을 \sin 과 \cos 한 주기 안에 매핑하여 전처리하였다.

$$TimeEnc(t) = \left[\sin\left(2\pi \frac{t}{T}\right), \cos\left(2\pi \frac{t}{T}\right) \right] \quad (6)$$

이때 T 는 하루를 초 단위로 환산한 86,400을 의미하고 t 는 변환되는 현재 시각을 의미한다. \sin 과 \cos 변환을 함께 사용하는 이유는 하나의 값만 사용하면 같은 값으로 표현될 수 있기 때문이다. 예를 들어 자정과 정오는 전혀 다른 시간대이지만, \sin 변환에서는 모두 0으로 표현된다. 이를 해결하기 위해 \cos 변환도 같이 사용하여 주기 안에서 표현될 수 있도록 하였다. 변환의 예는 오전 5시 30분은 하루 중 약 19,800초에 해당하며, 이 값을 변환하면 대략 0.99, 0.13 두 개의 값으로 표현된다.

1.3.6 Construction of Training Data

학습 데이터는 각 운행의 흐름을 효과적으로 학습시키기 위해서 개별 버스 운행을 기준으로 정류장 도착 시간 순서에 따라 하나의 시계열로 구성하였다. 각 시계열 데이터는 하나의 운행에서 발생한 정류장 방문 기록을 나열한 것으로 상행과 하행 중 데이터 수가 더 많은 운행을 선택하였다. 1번 노선의 실험에서는 상행 운행인 1번부터 48

번 정류장까지의 형태를 한 시계열 데이터로 구성하였으며, 13번 노선은 하행인 29번부터 56번 정류장까지 한 시계열 데이터로 구성하였다.

2. Proposed Model

본 연구는 단일 정류장의 도착 시간과 관련 정보만을 입력으로 하여, 동일 노선 내 다른 정류장의 도착 시간을 예측하는 새로운 방식을 제안한다. 기존 연구들이 운행 구간 예측이나 출발 정류장 기준 예측에 집중한 것과 달리, 단 하나의 정류장 정보로 전체 노선의 도착 시간을 복원하고, 이 중 사용자가 원하는 정류장까지의 도착 시간을 산출한다.

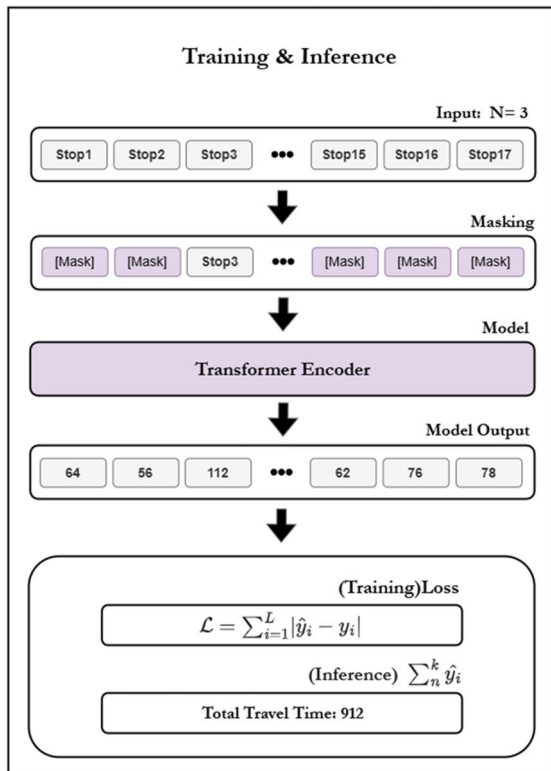


Fig. 2. Training and Inference process of the proposed model

2.1 Model Architecture

단일 정류장의 정보를 입력받고 다른 정류장의 도착 시간을 예측하기 위해서는, 모델이 노선 전체의 흐름을 스스로 추론할 수 있어야 한다. 이를 위해 입력 정류장 중에서 특정 정류장만을 남기고 나머지 정류장의 정보를 마스킹하는 방식을 사용하였다. 이러한 마스킹 기반 학습은 모델이 단일 지점의 정보에만 의존하지 않고 노선 전체의 흐름과 정류장 간 시간적 관계를 학습하게 한다. 또한 이러한 예측을 효과적으로 수행하기 위해 과거와 미래의 정보를 모두 활용하는 self-attention 메커니즘을 적용하여,

Transformer Encoder 기반의 모델 구조를 채택하였다 [24]. 학습 단계에서는 그림 2의 Training 부분과 같이 입력받은 정류장들의 데이터 중 3번 정류장의 정보를 제외하고, 나머지 정류장의 정보는 마스킹 처리한다. 이후 전체 정류장에서의 도착 시간을 예측하고 전체 정류장의 도착 시간 오차를 loss로 사용하여 학습한다. 예를 들면 모델의 결과는 1번 정류장에서 2번 정류장까지 56초, 2번 정류장에서 3번 정류장까지 112초 등 모든 정류장 사이에 걸린 시간이 출력된다. 실제 환경의 추론에서는 그림 2의 Inference 부분과 같이 N 과 K 의 입력을 받게 되면 나머지 정류장을 마스킹 처리하여 전체 정류장에서의 도착 시간을 예측하고 입력된 정류장들 사이의 운행 시간을 누적함으로써 K 번째 정류장의 도착 시각을 제공한다. 예를 들면 3번과 17번 사이의 소요 시간을 누적하여 예측한다.

Table 2. Model Hyperparameters

Category	Environment
d_{model}	128
d_{ff}	128
Attention head	8
Encoder Layer	2
dropout	0.1
Learning rate	10×10^{-4}

2.1.1 Linear Projection and Scaling

먼저 입력 피처의 차원 불일치를 해소하고 동일한 차원에서 학습이 이루어지도록, 입력값을 linear projection을 통해 Transformer Encoder의 입력 크기인 128차원으로 변환하였다. 이후 변환된 벡터는 모델 차원의 제곱근으로 스케일링하여 초기 계산에서 값의 분산이 불필요하게 증가하는 것을 방지하였다.

2.1.2 Positional Embedding

정류장 간의 순서 정보는 전체 정류장의 도착 시간을 예측하는 과정에서 필수적으로 고려되어야 하므로, 각 정류장 위치에 대한 고유한 순서 정보를 모델에 제공하기 위해 positional embedding을 적용하였다. 모델은 전체 정류장의 정보를 입력으로 받지만, 단일 정류장의 정보를 제외한 나머지 정류장은 마스킹하여 제공된다. 이에 따라 모든 정류장에 고유한 positional embedding을 부여함으로써 모델은 노선의 구조와 정류장 간 상대적 위치 관계를 파악할 수 있게 하였다. 예를 들어 전체 노선의 길이가 28인

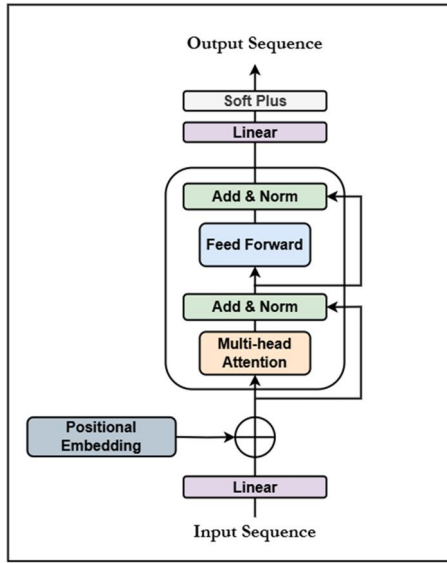


Fig. 3. Model Architecture

경우를 가정하고 13번 정류장의 정보를 입력으로 받았을 때, 단일 정류장의 정보만으로는 해당 정류장이 노선상에서 어느 위치에 해당하는지 구분할 수 없다. 그러므로 positional embedding을 부여하여 단일 정류장의 정보만으로도 노선상에서 해당 정류장의 상대적 위치와 정류장 간 관계를 함께 학습할 수 있게 하였다. 이는 입력으로 특정 정류장의 위치를 표시하는 기존 ETA 모델[11, 21, 22, 25]과 구별된다. 생성된 positional embedding은 정류장별 입력에 더해져, self-attention이 정류장 간 연결 관계와 시간적 흐름을 보다 안정적으로 학습할 수 있도록 한다.

2.1.3 Output Layer and Softplus Activation

Transformer Encoder를 거쳐 얻어진 정류장별 최종 은닉 벡터는 linear projection을 통해 각 정류장의 이동 시간 또는 도착 시간을 예측한다. 이동 시간은 누적되어 전체 도착 시간을 구성하는 양이며, 음수가 될 수 없으므로, 출력층 활성화 함수로 softplus를 적용하여 모든 예측값이 비음수 범위에서만 생성되도록 설계하였다.

$$\text{softplus}(x) = \ln(1 + e^x) \quad (7)$$

softplus는 음수가 나오지 않으면서도 매끄러운 기울기를 제공하므로, 이동 시간 예측에 안정적으로 활용할 수 있다. 최종적으로 예측된 전체 운행 시간은 정류장 순서에 따라 누적되어, 임의의 정류장 N 에서 K 까지의 ETA를 직접 산출할 수 있다.

3. Experiment

본 장에서는 두 가지의 실험을 수행하였다. 첫 번째 실험에서는 단일 정류장 입력만으로 전체 노선의 도착 시각

을 복원하는 모델의 기본 성능을 평가하였다. 두 번째 실험에서는 실제 운행 구간에서 예측된 누적 도착 시각과 실제 도착 시각의 차이를 통해 모델의 구간 단위 예측 정확도를 검증하였다. 실험에 사용된 데이터는 천안시 1번과 13번 노선의 1년간 DTG 운행 기록으로, 이를 정류장 단위로 재구성하여 1번 노선의 경우 19,106개의 운행 데이터가 생성되었으며, 13번 노선의 경우 3,404개의 운행 데이터가 생성됐다. 각 노선 데이터는 시퀀스 단위인 운행 데이터를 운행 시간 순서를 유지한 채 과거 구간 70%를 학습, 이후 10%를 검증, 가장 최근 구간 20%는 테스트 세트로 나누었다. 1번 노선의 경우 13,374개, 1,910개, 3,822개로 나누어졌고, 13번 노선의 경우 2,382개, 340개, 682개로 나누어 사용하였다.

Table 3. Experiment Environment

Category	Environment
GPU	NVIDIA RTX 3060
Python	3.10.5
PyTorch	2.5.0
Optimizer	Adam
Epochs	50
Loss	L1 Loss (MAE)

모든 실험은 표 3과 같이 Python 3.10.5와 PyTorch 기반 환경에서 수행되었으며, 학습은 NVIDIA RTX 3060 GPU를 활용하여 진행되었다. 손실 함수로는 L1 Loss (MAE)를 사용하였다. L2 loss는 이상치에 과도하게 민감해질 수 있으므로, 민감도가 낮아 도착 시간을 안정적으로 반영할 수 있는 L1 Loss를 사용하였다. 또한 최적화 기법으로는 적응적 학습률을 통해 Transformer 기반 구조에서 안정적인 수렴 특성을 보이는 Adam optimizer를 사용하였다.

3.1 Single-Stop-Based Full Route Arrival Time Prediction

본 실험에서는 1번과 13번 노선 운행 데이터를 대상으로, 단일 정류장 정보만으로 전체 정류장의 도착 시간을 예측하는 각 모델의 성능을 비교 분석하였다. 기본 모델의 경우 LSTM과 Transformer Encoder 구조를 사용하였다. LSTM은 단일 시점의 입력만으로는 전체 운행 시간을 예측하지 못하여, 학습이 안정적으로 수렴하지 못하였으며 높은 손실 값을 보였다. 이러한 한계를 보완하기 위해서 학습 난이도를 조절할 수 있는 커리큘럼 러닝(Curriculum learning) 전략을 도입했다[26]. 커리큘럼 러닝은 모델이 처음부터 어려운 문제를 학습하는 대신, 쉬운 문제부터 시작하여 점진적으로 난이도를 올리는 방식으로 학습하는

전략이다. 또한 Transformer Encoder에도 적용하여 성능 향상을 검증하였다. 실험에서는 C-LSTM(Curriculum learning LSTM), Transformer Encoder, C-Transformer Encoder(Curriculum learning Transformer Encoder) 모델을 비교하여 RMSE, MSE, MAE, 훈련 시간 4가지 지표로 비교하였다. 각 모델은 1번 노선은 100 epoch, 13번 노선은 50 epoch씩 5회 반복 학습하였으며, 결과는 이 다섯 번의 실험에서 얻은 지표의 평균값을 기준으로 종합하였다. 모델의 입력은 단일 정류장을 제외한 나머지 정류장 정보를 마스킹한 시계열 데이터이며, 출력으로는 모든 정류장 사이에 걸린 운행 시간이다.

3.1.1 C-LSTM Model

초기 실험에서는 시계열 데이터 처리에 널리 사용되는 LSTM (Long Short-Term Memory) 구조를 적용하였다 [27]. 또한 본 연구에서는 모델에 Curriculum learning을 적용하여, 에포크(epoch)에 따라 마스킹 되는 정류장 수를 점차 늘리는 방식을 도입하였다. 이 방식에서는 학습 초기에는 전체 정류장 정보를 입력으로 제공하지만, 에포크가 진행될수록 입력으로 남겨 두는 사용 가능한 정류장의 개수 *usable stops*가 점진적으로 감소하는 구조로 설계하였고 다음과 같은 식으로 계산하였다:

$$usable\ stops = [S_{last} - (\frac{S_{last} - 1}{epochs}) \cdot epoch] \quad (8)$$

여기서 *epoch*는 현재 에포크를 의미하고 *epochs*는 총 학습 에포크 수를 의미한다. 학습이 진행될수록 모델이 접근할 수 있는 정류장 수가 감소하게 되며, 모델은 점차 어려운 조건에서의 예측에 적응하게 된다. 예를 들어 전체 정류장이 28개일 때 학습 초반에는 모든 정류장을 입력으로 사용하지만, 학습 후반에는 단 1개만 남기고 나머지는 랜덤으로 마스킹 된다. 커리큘럼 러닝을 적용한 결과, 기존 LSTM이 초기 학습 단계에서 전체 정류장의 이동 시간을 안정적으로 예측하지 못하던 문제를 해결할 수 있었다.

Table 4. Average MAE of Different LSTM Models

Route No	Model	MAE
1	Residual	0.72
	C-Stacked	0.70
	C-Residual	0.41
13	Residual	0.59
	C-Stacked	0.62
	C-Residual	0.37

또한 단순한 LSTM 구조만으로는 구간별 패턴 반응이 어렵기 때문에, Residual connection과 Layer Normalization을 사용하였다. 이러한 선택의 효과를 검증하기 위해, Residual LSTM과 Curriculum learning을 적용한 C-Residual 그리고 C-Stacked(Stacked LSTM)를 비교하는 실험을 진행하였다. 표 4와 같이 1번 노선은 전체 정류장 수가 더 많아 학습이 어렵기 때문에 C-Residual은 가장 좋은 성능을 보였고, C-Stacked가 Residual보다 근소하게 좋은 성능을 보였다. 13번 노선에서는 Residual이 C-Stacked보다 더 좋은 성능을 보이며, 이를 통해 Curriculum learning이 장거리 운행 패턴을 학습하는 데 효과적임을 확인할 수 있다.

3.1.2 Transformer Encoder

본 연구에서는 Transformer Encoder 구조를 기반으로 단일 정류장 정보만을 입력으로 활용해 전체 정류장 도착 시간을 예측하는 모델을 설계하였다. Transformer Encoder는 self-attention을 통해 입력 간 전역적인 관계를 동시에 학습할 수 있어, 입력이 제한적인 환경에서도 높은 예측 성능을 보였다. 실험에서는 무작위의 단일 정류장의 도착 시간과 보조 특성(시간, 기후, 거리 등)을 포함한 입력을 제공하고, 출력으로는 전체 노선의 정류장별 도착 시간을 예측한다.

3.1.3 C-Transformer Encoder

Transformer Encoder 모델에도 커리큘럼 러닝 방식을 적용하여 실험을 진행하였다. Transformer Encoder는 단일 입력만으로도 좋은 성능을 보이기 때문에 출력을 해당 정류장을 기준으로 연속된 예측 구간을 점진적으로 늘리는 방법을 채택하였다. 초기 학습 단계에서는 짧은 예측 구간으로 학습을 진행하며, 학습이 진행될수록 예측 대상 범위가 선형적으로 증가하는 구조이다. 입력되는 중심 정류장을 기준으로, 좌우로 확장되는 연속된 구간을 예측하는 구조이며, 다음과 같이 설계하였다:

$$l_t = \min(S_{last}, l_{min} + [\frac{epoch}{epochs} \cdot (S_{last} - l_{min})]) \quad (9)$$

이때, l_t 는 epoch에서의 예측 길이이고 l_{min} 은 초기 예측 길이를 의미한다. 전체 노선의 길이가 28인 경우를 가정하고 임의의 정류장 14번째 정류장이 랜덤으로 설정됐을 때, 14번 정류장의 단일 도착 시간과 보조 특성을 포함한 입력을 제공하고, 학습 초반인 0번째 에포크에서는 중심을 기준으로 ± 1 개씩, l_{min} 의 값인 총 3개의 정류장 (13번, 14번, 15번 정류장)에 대해서만 예측하도록 한다.

Table 5. Overall performance comparison of ETA prediction models

Route No	Model	RMSE	MSE	MAE	Time
1	C-LSTM	0.71	0.50	0.41	210.21
	C-Transformer Encoder	0.69	0.48	0.41	263.46
	Transformer Encoder	0.69	0.48	0.40	98.32
13	C-LSTM	0.56	0.31	0.37	34.79
	C-Transformer Encoder	0.55	0.31	0.37	47.42
	Transformer Encoder	0.55	0.30	0.37	32.66

Table 6. Statistical comparison of MAE for Route 1

Model A	Model B	Mean difference	p	Hedges' g
Transformer Encoder	C-LSTM	-0.0067	<0.01	-0.140
Transformer Encoder	C-Transformer Encoder	-0.0137	<0.01	-0.333
C-LSTM	C-Transformer Encoder	-0.0070	<0.01	-0.188

Table 7. Statistical comparison of MAE for Route 13

Model A	Model B	Mean difference	p	Hedges' g
Transformer Encoder	C-LSTM	-0.0002	0.920	-0.003
Transformer Encoder	C-Transformer Encoder	-0.0137	<0.01	-0.120
C-LSTM	C-Transformer Encoder	-0.0070	0.306	-0.032

이후 1 에포크 이후에는 5개의 정류장의 범위를 예측하도록 점진적으로 확장된다.

3.1.4 Results and Discussion

비교 실험에서는 세 개의 모델을 RMSE, MSE, MAE 기준으로 비교하였으며, 표 5에 명시된 값의 단위는 Min-Max 정규화된 ETA 값을 기준으로 계산된 오차 지표이다. 여기서 Transformer Encoder 모델은 다른 모델들과 달리 한 개의 단일 정류장의 정보만을 입력으로 주고 학습하게 된다. 나머지 모델들은 이와 다르게 Curriculum learning에 따라 초반에는 여러 정류장의 정보를 제공받고, 이후 단계적으로 정류장의 수를 줄이는 방식으로 학습을 진행하였다.

표 5의 실험 결과를 종합적으로 분석한 결과, Transformer Encoder 모델은 전체적으로 우수한 성능과 효율성을 동시에 보여주었다. 13번 버스 노선 기준으로는 다른 모든 모델과 동등하거나 더 나은 예측 정확도를 달성하였으며, 학습 속도 측면에서도 다른 모델들과 비교하였을 때 더 빠른 학습 속도를 보였다. 1번 노선에서도 Transformer Encoder는 준수한 성능을 보이며, 복잡한 정류장 구성에서도 비교적 안정적인 예측 능력을 유지했다. 반면, C-LSTM은 1번 노선에서 가장 높은 RMSE와 전체적으로 높은 210.21초의 학습 시간을 보이며, 실용성에서 제한이 있었다. C-Transf

ormer Encoder는 263.46초로 측정되었다. 반면, Transformer Encoder는 98.32초로 두 모델에 비해 크게 단축된 학습 시간을 기록하였다. 이 결과는 Transformer Encoder가 C-LSTM보다 약 53.2%, C-Transformer Encoder보다 약 62.7% 빠른 학습 속도를 보였음을 의미한다. 이러한 차이는 트랜스포머의 병렬 연산 구조 및 학습 안정성에 기반한 것으로, 학습 시간 측면에서 우수한 성능을 보였다. 또한 모델 간 성능 차이의 통계적 유의성을 보다 명확히 분석하기 위해 추가적인 통계 검정을 수행하였다. 노선 특성에 따른 분포 차이를 고려하여, 1번 노선과 13번 노선에 대해 각각 독립적으로 통계 분석을 수행하였으며, 그 결과를 표 6과 표 7에 제시하였다.

각 모델 간 비교는 테스트 세트에 포함된 3,822개의 개별 운행 시퀀스별로 실제 도착 시간과 예측 도착 시간 간의 오차를 계산하여 얻어진 MAE 분포를 기반으로 수행되었다. 모델 간 비교는 두 모델을 한 쌍으로 설정하여 진행하였으며, 각 비교에서 Model A와 Model B는 표에 제시된 모델 순서를 따른다. 모델 쌍 간 평균 차이에 대해 대응 표본 t-검정을 적용하였다. 또한 단순한 유의성 검정 결과에 더해, 실제 효과의 크기를 함께 해석하기 위해 Hedges' g를 효과 크기 지표로 함께 산출하였다.

표 6은 1번 노선에 대한 통계 검정 결과를 나타낸다. Transformer Encoder는 C-LSTM 및 C-Transformer En

coder 대비 유의하게 낮은 MAE를 보였으며($p < 0.01$), 효과 크기는 작은 수준으로 나타났다. 특히 C-Transformer Encoder와의 비교에서는 상대적으로 더 큰 효과 크기가 관찰된다.

표 7은 13번 노선에 대한 통계 검정 결과를 나타낸다. C-LSTM과 C-Transformer Encoder 간 비교에서는 통계적으로 유의한 차이가 관찰되지 않았지만, Transformer Encoder와 C-Transformer Encoder 간 비교에서는 유의한 차이($p < 0.01$)가 나타났으나, 효과 크기는 매우 작은 수준으로 확인되었다. 이는 13번 노선이 상대적으로 짧고 운행 패턴이 단순한 노선 특성을 가지기 때문에, 모델 간 예측 성능이 유사하게 나타나는 것으로 해석된다. 이러한 조건에서는 단일 정류장 기반 Transformer Encoder가 기존 모델들과 비교하여 동등한 수준의 예측 성능을 유지하면서도, 일부 비교에서는 미세한 성능 차이를 보였음을 의미한다.

종합적으로, 통계 검정 결과는 표 5에서 관찰된 평균 성능 비교 결과를 보완하며, Transformer Encoder 모델이 특히 복잡한 노선 조건에서 안정적이고 일관된 성능 향상을 보이며, 빠른 속도의 학습이 가능하다.

Table 8. Results of cumulative ETA prediction

Distance (K-N)	Median case MAE(s)	Mean MAE(s)	IQR(s)	95% CI(s)
10	51.75	56.44	23.84	[55.70, 57.19]
15	65.76	73.72	36.54	[72.58, 74.91]
20	78.70	91.68	53.48	[90.02, 93.44]
25	89.72	108.79	75.54	[106.59, 111.05]
30	101.83	124.74	94.39	[122.15, 127.51]
35	115.02	140.71	111.02	[137.77, 143.79]

3.2 Evaluation of Cumulative ETA Prediction

두 번째 실험에서는 더욱 긴 거리를 가지고 있는 1번 노선의 테스트 데이터 세트의 실제 운행 구간에서 Transformer Encoder 모델이 예측한 누적 도착 시간과 실제 도착 시간 간의 오차를 평가하였다. 3.1의 단일 정류장 기반 전체 정류장 예측 실험과 다르게, 본 실험은 특정 정류장(N)에서 임의의 정류장(K)까지 이동할 때의 누적 도착 시간을 비교함으로써, 모델이 실제 구간 단위 이동 시간 패턴을 얼마나 정확히 반영하는지를 측정하였다. N과 K의 간격 10부터 35까지 5단위로 설정하였으며, 거리별 중앙값 기준 MAE, 평균 MAE, IQR 그리고 95% 신뢰구간의 길이를 비교하여 모델의 성능과 누적 ETA 예측 성능을 종합적으로 분석하였다. 실험 결과를 정리한 표 8에서는 중앙값 기준 MAE는 거리 증가에 따라 점진적으로 상승하는 경향을 보였으나, 35개의 정류장 구간에서도 115.02초로 비교적 안정적인 성능을 유지하였다. 이는 중앙값 기준

MAE(Median Case MAE)가 정류장 간격이 5개씩 증가할 때마다 약 11-14초 범위에서 증가하며, 정류장 1개당 평균 약 2-3초 수준으로 완만하게 증가하였다. 평균 MAE 역시 거리 증가에 따라 함께 증가하였으나, 장거리 구간에서도 오차가 급격히 폭증하는 양상은 관찰되지 않았다. 평균 MAE가 중앙값보다 크게 나타난 것은 일부 구간에서 발생한 큰 오차가 평균을 끌어올린 결과로 해석할 수 있으며, 이는 소수의 이상치에 영향을 받은 현상으로 판단된다. 이러한 해석은 거리 증가에 따라 평균과 중앙값 간의 차이가 확대되지만, 중앙값과 IQR이 점진적인 증가 경향을 유지한 결과에서 확인할 수 있다. 또한 IQR은 거리 증가에 따라 완만한 증가 경향을 보였으며, 특정 거리 구간에서 분산이 급격하게 증가하는 현상은 관찰되지 않았다. 이는 누적 이동 거리가 증가함에 따라 오차 분포의 폭이 점진적으로 확장되는 자연스러운 경향을 반영한 결과로, 장거리 구간에서도 예측 오차의 분포가 불안정하게 증가하지 않고 비교적 통제된 상태로 유지되고 있음을 의미한다. 95% 신뢰구간의 길이 역시 거리 증가에 따라 점진적으로 증가하였으나, 평균의 불확실성이 급격히 커지지 않았다. 또한 그림 4와 그림 5는 각각 거리 간격 15와 25구간에 대한 Median Case의 누적 도착 시간을 시각화한 결과를 나타낸다.

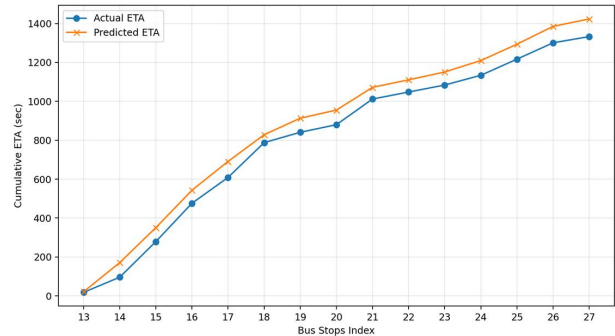


Fig. 4. Middle case result of arrival time prediction(Distance 15)

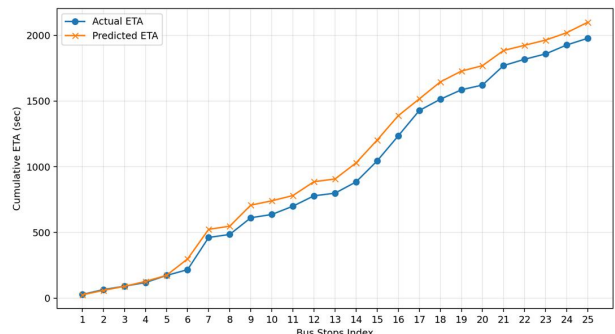


Fig. 5. Middle case result of arrival time prediction(Distance 25)

거리 증가에 따른 누적 ETA 예측 특성을 균형 있게 분석하기 위해, 중거리 구간에 해당하는 15 및 25 정류장 간격의 Median Case를 대표 사례로 선정하였다. 일부 정류장에서 누적 오차가 소폭 발생하였으나, 장거리 구간으로 진행됨에 따라 오차가 급격히 확대되는 현상은 관찰되지 않았으며, 전체적인 누적 곡선의 형태는 실제 운행 결과와 유사하게 유지되었다. 이러한 결과는 제안 모델이 단일 정류장 정보만을 이용하더라도 중·장거리 구간에서의 누적 ETA를 안정적으로 예측할 수 있음을 시각적으로 보여주었다.

IV. Conclusions

본 연구는 실제 버스 운행 환경에서 발생하는 사용자의 요구, 즉 임의의 정류장에 도착한 시점에서 목적지 정류장까지의 도착 시간 예측이라는 현실적인 문제를 다루었다. 천안시 1번과 13번 버스 노선의 1년간 수집된 DTG 데이터를 활용하여 모델을 학습하였다. 복잡한 DTG 데이터를 모델이 활용할 수 있는 입력 구조로 변환하기 위한 데이터 전처리 프레임워크도 제시하였다. 예측 모델 설계 과정에서는 복잡한 ETA 예측 문제를 다루기 위해 Curriculum learning을 도입하고, 이를 LSTM과 Transformer Encoder 모델에 각각 적용하여 예측 성능과 학습 시간을 비교 분석하여 동등하거나 더 우수한 예측 정확도를 유지하면서 50% 이상 적은 학습 시간을 보인 Transformer Encoder를 최종 모델로 선정하였다. 또한 누적 ETA 예측 실험을 통해, 제안한 Transformer Encoder 모델이 단일 정류장 도착 정보를 기반으로 실제 운행 구간에서의 누적 이동 시간 패턴을 안정적으로 반영할 수 있음을 확인하였다. 실험 범위 내에서 중앙값 기준 누적 MAE는 35개 정류장 구간에서도 약 115초로 나타났다. 누적 구간이 증가함에 따라 예측 오차는 점진적으로 증가하였으나, 오차 분포의 폭(IQR)과 평균의 불확실성은 급격히 증가하지 않고 완만하게 변화하는 경향을 보였다. 이는 장거리 구간으로 이동하더라도 예측 결과가 특정 구간에서 불안정하지 않고, 전체 운행 흐름을 일관되게 유지함을 의미하며, 실제 서비스 환경에서 누적 ETA 예측이 실질적으로 적용 가능함을 보여준다.

본 연구는 사전 수집된 정적 DTG 데이터를 기반으로 수행되어, 실시간 교통 상황 변화나 정체 등 외부 요인을 반영하지 못하였다는 한계가 있다. 향후 연구에서는 다양한 노선 구조로의 확장과 실시간 교통 상황에 대한 정보를 통합하여, 실제 서비스 환경에 적용 가능한 실시간 예측 모델로 확장할 예정이다.

REFERENCES

- [1] Kim, S., Kim, Y., & Lee, C., "Systematic Error Term Analysis on Bus Arrival Time Estimation," *Journal of the Korean Society of Transportation*, Vol. 24, No. 4, pp. 117-127, June 2006.
- [2] Kim, M.J. & Lee, Y.I., "Study on the Optimum Route Travel Time for Bus to Improve Bus Schedule Reliability," *Journal of Korean Society of Intelligent Transport Systems*, Vol. 16, No. 6, pp. 112-123, September 2017. DOI: 10.12815/kits.2017.16.6.112.
- [3] S. Choi, G. S. Cho, Y. S. Yu, and M. S. Do, "Analysis of the Operation Efficiency of the City Bus Industry: Focusing on Comparison between Semi-public and Private Operating systems. *Korean Society of Transportation*, Vol. 42, No. 1, pp. 94-106, February 2024, DOI: 10.7470/jkst.2024.42.1.094
- [4] S. Chung, "A study of quality attributes for reliability improvement on traffic information," *Journal of the Korea Society of Computer and Information*, Vol. 18, No. 5, pp. 133-145, May 2013. DOI: 10.9708/jksci.2013.18.5.133
- [5] Cats, O. & Loutos, G., "Evaluating the Added-Value of Online Bus Arrival Prediction Schemes," *Transportation Research Part A: Policy and Practice*, Vol. 86, pp. 35-55, April 2016. DOI: 10.1016/j.tra.2016.02.004
- [6] Kuo, Y.-H., Leung, J. M. Y. & Yan, Y., "Public Transport for Smart Cities: Recent Innovations and Future Challenges," *European Journal of Operational Research*, Vol. 306, No. 3, pp. 1001-1026, May 2023. DOI: 10.1016/j.ejor.2022.06.057
- [7] M. Kim and J. Moon, "Deep Learning Trends in Urban Mobility and Energy Management for Smart City Implementation," in *Proceedings of the Korean Society of Computer Information Conference*, pp. 261-264, January. 2025
- [8] Jang, J. M., Lee, Y. I. & Lee, G., "A Study on the Factors Affecting the Stopping Time and Punctuality of Bus Stop: A Case of Bus Stop by Roadside Bus Only Lane," *Journal of Korean Society of Transportation*, Vol. 35, No. 3, pp. 234-246, June 2017. DOI: 10.7470/jkst.2017.35.3.234
- [9] Arhin, S., Manandhar, B. & Baba-Adam, H., "Predicting Travel Times of Bus Transit in Washington, D.C. Using Artificial Neural Networks," *Civil Engineering Journal*, Vol. 6, No. 11, pp. 2245-2261, November 2020. DOI: 10.28991/cej-2020-03091615
- [10] Li, J., "Bus Arrival Time Prediction Based on Random Forest," *Proceedings of the 2017 5th International Conference on Frontiers of Manufacturing Science and Measuring Technology (FMSMT 2017)*, pp. 867-872, April. 2017. DOI: 10.2991/fmsmt-17.2017.1.67
- [11] Liu, H., Xu, H., Yan, Y., Cai, Z., Sun, T. & Li, W., "Bus Arrival Time Prediction Based on LSTM and Spatial-Temporal Feature Vector," *IEEE Access*, Vol. 8, pp. 11917-11929, January 2020. DOI: 10.1109/ACCESS.2020.2965094
- [12] Mustafa, G., Hwang, Y. & Cho, S.-J., "Predicting Bus Travel

- Time in Cheonan City through Deep Learning Utilizing Digital Tachograph Data," *Electronics*, Vol. 13, No. 9, 1771, 2024. DOI: 10.3390/electronics13091771
- [13] Wu, J., Wu, Q., Shen, J. & Cai, C., "Towards Attention-Based Convolutional Long Short-Term Memory for Travel Time Prediction of Bus Journeys," *Sensors*, Vol. 20, No. 12, 3354, May 2020. DOI: 10.3390/s20123354
- [14] Bai, C., Peng, Z.-R., Lu, Q.-C. & Sun, J., "Dynamic Bus Travel Time Prediction Models on Road with Multiple Bus Routes," *Computational Intelligence and Neuroscience*, Vol. 2015, Article ID 432389, July 2015. DOI: 10.1155/2015/432389
- [15] Wang, Y., Zheng, Y. & Xue, Y., "Travel Time Estimation of a Path Using Sparse Trajectories," *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Vol. 20, pp. 25-34, August. 2014. DOI: 10.1145/2623330.2623656
- [16] Park, C. Y., Kim, H. G., Shin, C. S., Cho, Y. Y. & Park, J. W., "Arrival Time Estimation for Bus Information System Using Hidden Markov Model," *KIPS Transactions on Computer and Communication Systems*, Vol. 6, No. 4, pp. 189-196, April 2017. DOI: 10.3745/KTCCS.2017.6.4.189
- [17] Jang, J. Y. & Park, J. T., "A Study on the Application of Machine Learning to Improve BIS (Bus Information System) Accuracy," *The Journal of The Korea Institute of Intelligent Transport Systems*, Vol. 21, No. 3, pp. 42-54, June 2022. DOI: 10.12815/kits.2022.21.3.42
- [18] Chen, X., Saidi, S. & Sun, L., "Understanding Bus Delay Patterns under Different Temporal and Weather Conditions: A Bayesian Gaussian Mixture Model," *Transportation Research Part C: Emerging Technologies*, Vol. 171, February 2025. DOI: 10.1016/j.trc.2025.105000
- [19] Chen, H., Wang, H. & Hu, Y., "Weathering the Wait: Temperature Impacts on School Bus Delays," *Transportation Research Part D: Transport and Environment*, Vol. 136, 104455, November 2024. DOI: 10.1016/j.trd.2024.104455
- [20] Xie, Zhi-Ying, et al. "Multistep prediction of bus arrival time with the recurrent neural network." *Mathematical Problems in Engineering*, Vol. 2021, pp. 1-14, March 2021. DOI: 10.1155/2021/6636367.
- [21] Shen, J., Liu, Q., Zhang, Y. & Yu, M., "A Novel Model Incorporating Deep Learning and Kalman Filter Augmentation for Route-Level Bus Arrival Time Prediction with Error Accumulation Mitigation," *Expert Systems with Applications*, Vol. 281, July 2025. DOI: 10.1016/j.eswa.2025.127622
- [22] Petersen, N. C., Rodrigues, F. & Pereira, F. C., "Multi-Output Bus Travel Time Prediction with Convolutional LSTM Neural Network," *Expert Systems with Applications*, Vol. 120, pp. 426-435, April 2019. DOI: 10.1016/j.eswa.2018.11.028
- [23] B. Kim, "Design and Implementation of a Big Data Analytics Framework based on Cargo DTG Data for Crackdown on Overloaded Trucks," *Journal of the Korea Society of Computer and Information*, Vol. 24, No. 12, pp. 67-74, December 2019. DOI: 10.9708/jksci.2019.24.12.067
- [24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I., "Attention is All You Need," *Advances in Neural Information Processing Systems*, Vol. 30, December 2017. DOI: 10.5555/3295222.3295349
- [25] F. Hafizi, S. Seyedabrishami, and E. Sherafat, "Real-time prediction of bus inter-stop travel time using deep learning approach," *Transportation Research Procedia*, Vol. 82, pp. 3508-3520, January. 2025. DOI: 10.1016/j.trpro.2024.12.079.
- [26] Bengio, Y., Louradour, J., Collobert, R. & Weston, J., "Curriculum Learning," *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*, pp. 41-48, June. 2009. DOI: 10.1145/1553374.1553380
- [27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, November 1997. DOI: 10.1162/neco.1997.9.8.1735

Authors



Ga-On Cheon is an undergraduate student in the Division of Computer Science and Engineering at Sunmoon University, Asan, Korea. He is interested in deep learning and machine learning.



Young-Sup Hwang received the B.S., degree in Computer Engineering from Seoul National University, Korea, in 1989. He received the M.S. and Ph.D. degrees in Computer Engineering from POSTECH, Korea, in 1991

and 1997, respectively. Dr. Hwang joined the faculty of the Division of Computer Science and Engineering at Sun Moon University, Asan, Korea, in 2002. He is currently a Professor in the Department of Computer Science and Engineering, Sun Moon University. He is interested in deep learning, pattern recognition, machine learning and image processing.