

Article

Predicting Bus Travel Time in Cheonan City through Deep Learning Utilizing Digital Tachograph Data

Ghulam Mustafa ¹, Youngsup Hwang ^{1,*} and Seong-Je Cho ²

¹ Division of Computer Science and Engineering, Sun Moon University, Asan-si 31461, Republic of Korea; ghulammustafa@sunmoon.ac.kr

² Department of Software Science, Dankook University, Yongin-si 16890, Republic of Korea; sjcho@dankook.ac.kr

* Correspondence: young@sunmoon.ac.kr

Abstract: Urban transportation systems are increasingly burdened by traffic congestion, a consequence of population growth and heightened reliance on private vehicles. This congestion not only disrupts travel efficiency but also undermines productivity and urban resident's overall well-being. A critical step in addressing this challenge is the accurate prediction of bus travel times, which is essential for mitigating congestion and improving the experience of public transport users. To tackle this issue, this study introduces the Hybrid Temporal Forecasting Network (HTF-NET) model, a framework that integrates machine learning techniques. The model combines an attention mechanism with Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) layers, enhancing its predictive capabilities. Further refinement is achieved through a Support Vector Regressor (SVR), enabling the generation of precise bus travel time predictions. To evaluate the performance of the HTF-NET model, comparative analyses are conducted with six deep learning models using real-world digital tachograph (DTG) data obtained from intracity buses in Cheonan City, South Korea. These models includes various architectures, including different configurations of LSTM and GRU, such as bidirectional and stacked architectures. The primary focus of the study is on predicting travel times from the Namchang Village bus stop to the Dongnam-gu Public Health Center, a crucial route in the urban transport network. Various experimental scenarios are explored, incorporating overall test data, and weekday and weekend data, with and without weather information, and considering different route lengths. Comparative evaluations against a baseline ARIMA model underscore the performance of the HTF-NET model. Particularly noteworthy is the significant improvement in prediction accuracy achieved through the incorporation of weather data. Evaluation metrics, including root mean squared error (RMSE), mean absolute error (MAE), and mean squared error (MSE), consistently highlight the superiority of the HTF-NET model, outperforming the baseline ARIMA model by a margin of 63.27% in terms of the RMSE. These findings provide valuable insights for transit agencies and policymakers, facilitating informed decisions regarding the management and optimization of public transportation systems.

Keywords: bus travel time prediction; long short-term memory (LSTM); gated recurrent unit (GRU); autoregressive integrated moving average (ARIMA); digital tachograph data (DTG); real-world data



Citation: Mustafa, G.; Hwang, Y.; Cho, S.-J. Predicting Bus Travel Time in Cheonan City through Deep Learning Utilizing Digital Tachograph Data. *Electronics* **2024**, *13*, 1771. <https://doi.org/10.3390/electronics13091771>

Academic Editor: Daniel Gutiérrez Reina

Received: 28 March 2024

Revised: 23 April 2024

Accepted: 1 May 2024

Published: 3 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The development of an "Intelligent Transportation System" (ITS) has become increasingly crucial in modern transportation. An ITS aims to provide innovative services for efficient traffic management and diverse modes of transportation while ensuring user safety [1]. This cutting-edge technology encompasses emergency services, passenger travel time prediction, and the use of cameras to enforce traffic regulations or dynamically adjust speed limits based on real-time traffic conditions.

Among these transportation modes, buses play an important role in urban areas, serving as a sustainable means of public transportation. Buses effectively address issues

such as traffic congestion, parking challenges, and environmental pollution stemming from private vehicles. Public transportation stands as an integral component of efficient urban planning, consuming fewer resources and emitting fewer pollutants compared to private transportation. By embracing public transportation, cities can enhance air quality, alleviate traffic congestion, and improve the overall quality of life for citizens, fostering a more sustainable and livable urban environment.

Buses serve as a crucial means of public transport in cities, providing convenient mobility for citizens, including commuters and students. However, during peak traffic periods, they encounter persistent issues such as lengthy travel times and a lack of punctuality, which results in passenger dissatisfaction and decreased ridership. To tackle this problem, optimizing bus routes and schedules has become a top priority. Accurate prediction of bus travel times plays a vital role in achieving this objective [2]. Precise predictions can assist in optimizing bus scheduling systems, thereby enhancing the overall efficiency of the transportation network. Moreover, accurate travel time predictions meet passenger's expectations and foster their trust in the public transport system.

Ensuring reliable predictions is a significant challenge due to the multitude of factors that can influence bus travel times, including traffic conditions, weather, and passenger load. Inaccurate travel time predictions can have adverse effects on the system's efficiency, resulting in operational inefficiencies and increased costs for the bus company. To overcome this challenge, the exploration of advanced technologies, such as deep learning models, can be beneficial in providing more accurate predictions. These technologies can utilize real-time data from various sources, including GPS and digital tachographs, to enhance prediction accuracy. By addressing the issue of predicting bus travel times, public transportation agencies can improve the reliability and punctuality of their services, leading to increased ridership and enhanced mobility for citizens.

South Korea has implemented comprehensive measures known as the "Management Guidelines for Automobile Operation Records and Devices" to efficiently manage automobile operation records. These guidelines, enforced through the Traffic Safety Act Article 55, the Enforcement Ordinance Article 45, and the Enforcement Regulation Articles 29 and 30, encompass various aspects such as storage, submission, inspection, analysis, and utilization [3].

Since 2005, the installation of digital tachograph (DTG) devices has been mandatory for commercial vehicles, including buses and trucks, in compliance with these guidelines. Additionally, since 2011, newly registered cargo vehicles weighing 1 ton and above have been required to install DTG devices. The primary objective of these devices is to promote safe driving practices and discourage reckless behavior by recording and monitoring various aspects of vehicle operation. The DTG data recorded by the DTG devices have proven to be invaluable for a wide range of applications. They enable the analysis of work conditions, facilitate the detection of road slipperiness, and aid in the development of representative driving cycles for delivery trucks, among various applications [4–7]. As a result, the DTG device has emerged as a crucial tool for ensuring the safety and efficiency of vehicle operations in South Korea.

In this paper, we utilize DTG data and state-of-the-art deep learning models to predict bus travel time. The DTG data are obtained from intracity buses in Cheonan City, South Korea, while the deep learning models include pure Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), bidirectional LSTM and GRU, Stacked-LSTM and -GRU, and Hybrid Temporal Forecasting Network (HTF-NET). We have conducted several experiments to evaluate the model's performance on overall test data, weekday data, weekend data, and with and without weather information. Furthermore, we can successfully predict travel times for different types of routes, including both short and long routes. To evaluate the model's performance, we compare it with the autoregressive integrated moving average (ARIMA) time series model, which has been commonly used in previous studies [8–11]. The experimental results demonstrate the effectiveness of the HTF-NET model in predicting bus travel times. Additionally, the inclusion of weather information enhances prediction accuracy. We use evaluation metrics such as root mean

squared error (RMSE), mean absolute error (MAE), and mean squared error (MSE) to assess the model's performance. Notably, the HTF-NET model outperforms the baseline ARIMA model by an impressive 63.27% in terms of the RMSE.

Our main contributions are as follows:

- To the best of our knowledge, this is the first work aiming at employing DTG data to predict bus travel time. We demonstrate that DTG data from intracity buses in Cheonan City are effective in predicting bus travel times.
- Our approach introduces a hybrid model that integrates various deep learning architectures, including attention, LSTM, and GRU layers. This ensemble model, in conjunction with the Support Vector Regressor (SVR), demonstrates outstanding performance, surpassing all other models in terms of the RMSE, MAE, and MSE.
- We extract novel temporal features, such as days of the week, and holidays, from the existing dataset. This enhancement contributes to the robustness of our model, leading to more accurate bus travel time predictions.
- Our developed deep learning model is versatile and applicable to various real-world traffic scenarios, encompassing both rush and non-rush hour periods in Cheonan, South Korea. By accounting for specific characteristics and patterns associated with different time periods, our models can adapt to the dynamic nature of bus travel times, resulting in more precise predictions. Furthermore, we successfully predict travel times for both short and long routes.
- We conducted thorough experiments to evaluate the proposed model. The results demonstrate that our model significantly enhances the accuracy of predicting bus travel time, affirming its effectiveness in diverse traffic scenarios.

The remainder of this paper is structured as follows: In Section 2, we provide a review of the related work. Section 3 describes the data collection process and the preprocessing steps employed to obtain the travel time dataset used in our study. Section 4 outlines the methodology adopted for developing the travel time prediction models, which leverage deep learning techniques. In Section 5, we present the results of the comparative analysis of the algorithm performance in our study, based on the RMSE, MAE, and MSE, emphasizing the effectiveness of the deep learning algorithms utilized. Finally, in Section 6, we provide concluding remarks and discuss future directions.

2. Related Work

To address the challenge associated with accurate and reliable travel time prediction (TTP) models, the integration of advanced machine learning techniques has gained substantial attention in recent years. Deep learning models, including recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have demonstrated significant potential for capturing the intricate temporal dependencies and spatial characteristics of travel time data. These models can automatically extract and learn meaningful representations from vast amounts of historical data, enabling more accurate and reliable travel time predictions. Moreover, the fusion of multiple data sources, such as weather data, traffic sensor data, and historical travel patterns, has been explored to enrich the input features and enhance the predictive capabilities of TTP models.

Several researchers use DTG data for many purposes. For example, Kim et al. [3] presented an algorithm that used blockchain technology to improve the security and reliability of the data from a DTG device. M.-H. Jeong et al. [4] proposed a model that utilized GRUs to consider historical traffic speed data and weather conditions for forecasting highway speeds. H. Jeong et al. [5] suggested a data integration process that combined GPS data with other vehicle sensor data to create a vehicle trajectory database for livestock vehicles. Ahn and Shin. [6] analyzed the travel patterns of taxi passengers in Busan, South Korea, using DTG data. Seung-Bae Jeon et al. [7] utilized DTG data integrated with road link data to predict bus travel speed. They employed an LSTM neural network, which shows the potential of this approach for enhancing bus travel speed prediction. These valuable

contributions exemplify the diverse range of research endeavors to explore the potential and address the intricacies associated with DTG data in various contexts.

An accurate prediction of travel time is a crucial aspect of transportation planning. Therefore, extensive research has been conducted to explore two primary approaches: route-based and data-driven approaches. The existing TTP techniques can be categorized into these two approaches. Route-based approaches calculate the overall travel time by combining the segment time and transition time, which includes waiting time resulting from signals, turns, and other factors between segments. Based on the formulation of the overall travel time, the route-based approaches can be further divided into segment-based methods that utilize segment time while disregarding inter-segment correlation [12], and path-based methods that consider both segment time and intersection delays [13,14]. On the other hand, data-driven approaches treat travel time as a regression task and estimate the travel time for an entire path or route based on historical data, implicitly capturing the complexities of traffic patterns. Data-driven approaches can be further classified into trajectory-based methods [15,16] and origin–destination (OD)-based methods [17]. Trajectory-based methods utilize road network and trajectory data to predict travel time, while OD-based methods solely consider pickup and drop-off location data for travel time estimation.

In recent years, the utilization of data-driven approaches for travel time estimation and prediction has gained substantial momentum. These approaches have emerged as powerful tools capable of uncovering hidden patterns and relationships within vast volumes of traffic data. Leveraging technological advancements and a diverse array of machine learning algorithms, including linear regression (LR), decision trees (DTs), random forests (RFs), gradient boosting regressors (GBRs), and ARIMA, as well as deep learning techniques such as GRU and LSTM, these sophisticated models offer remarkable capabilities. The key strength of these models lies in their ability to capture intricate and underlying relationships among various factors, even when such connections are not readily apparent. By effectively leveraging the vast amounts of available data, they excel at identifying complex temporal dependencies and nonlinear relationships within the data, ultimately contributing to improved travel time predictions. The flexibility and adaptability of these models allow them to handle diverse and dynamic traffic scenarios, providing valuable insights into travel time variations under different conditions. In Table 1, we provide a comprehensive overview of recent studies that have employed modern machine learning techniques for travel time prediction. The table shows the wide range of methodologies and algorithms utilized in the TTP research field, highlighting the diversity and richness of approaches. From traditional regression-based models to sophisticated deep learning architectures, researchers have explored various avenues to enhance travel time prediction accuracy and robustness.

Table 1. Summary of travel time prediction studies using machine learning algorithms.

Author	Year	Country	Data Source	Analysis Area	Method	Vehicle	Data Type
Sun, Xiaoyu et al. [18]	2018	China	FWOM, OPATDS	Fixed Road	K-NN, SVM, RF	Trucks	TT
Wang, Zheng et al. [19]	2018	China	GPS	Urban Road	LSTM	Taxi	Taxi TS
Zheng, Fang et al. [20]	2013	Netherlands	GPS	Urban Road	State-Space NN	Car	Vehicle Position, TS
Mendes-Moreira et al. [21]	2015	Portugal	STCP System	Urban Road	Regression and SVM	Bus	TT
Yu, Bin et al. [22]	2018	China	AVLS	Bus Route	RF and K-NN	Bus	Bus TT
Gupta, Bharat et al. [23]	2018	Portugal	GPS	Urban Road	RF, GB	Taxis	Taxi TS
Crist, Teresa et al. [24]	2019	Spain	PTN	Urban Road	K-Mean Clustering	Bus	Travel Time
Zhi-jian, Wang et al. [25]	2020	China	GIS GPS	Urban Road	LSTM	Taxis	TT, Taxi TS
Chughtai, J.-U.-R et al [26]	2022	Pakistan	FCD	Urban Road	LSTM+GRU	Car	TT
Lee, G. et al [27]	2022	South Korea	STD	Urban Road	Geo-conv LSTM	Bus	TT
Lee, C. et al [28]	2022	South Korea	BMS & BIS	Urban Road	LSTM, GRU, ALSTM	Bus	TT
Shaji, H. et al [29]	2023	India	GPS	Urban Road	RF, ANN, Clustering	Bus	TT

3. Data Collection and Preprocessing

3.1. Study Area

In this study, our focus was on collecting travel time data specifically from public transport buses operating in Cheonan, a significant transportation hub in the central region

of South Korea, located approximately 83 km south of Seoul, the capital city. With a population of 689,881 residents as of the end of May 2023. Cheonan is a bustling industrial city, housing renowned companies such as Samsung SDI and Samsung Display. Buses play a pivotal role in the transportation system of Cheonan, with the bus transport authority assigning specific routes to individual buses. Currently, there are over 150 designated routes available for passengers to travel from their source locations to their desired destinations. For our research, we focused on a specific sub-route of route number 200, which spans from the Ibjang bus stop to the Cheonan Station, as shown in Figure 1.

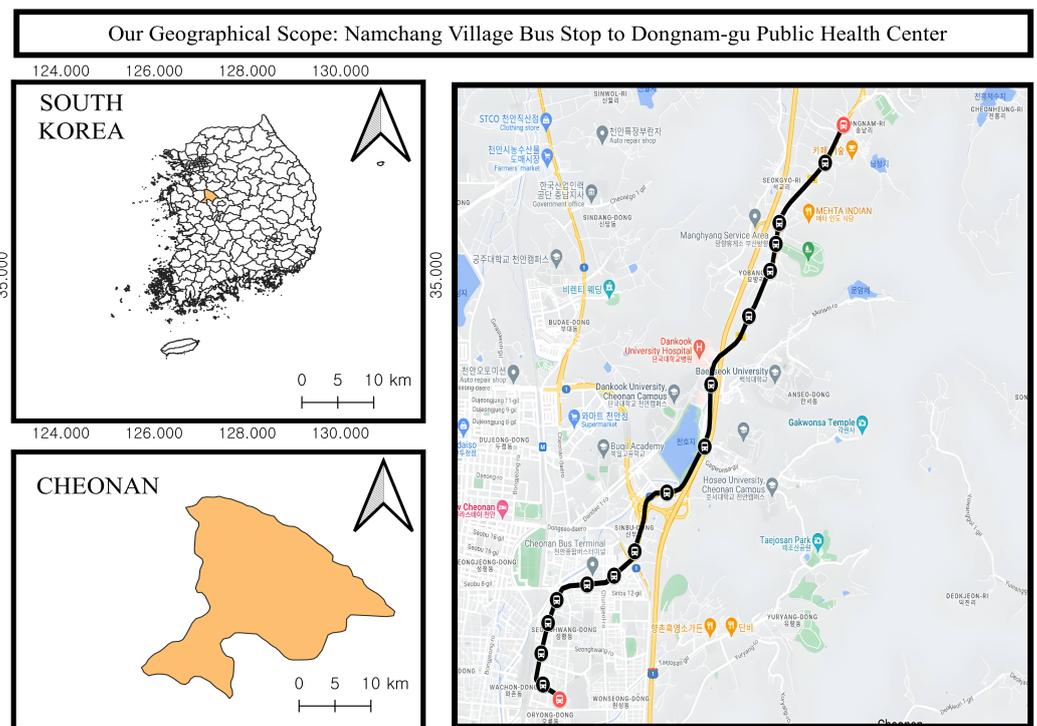


Figure 1. Geographical Scope: Namchang Village bus stop to Dongnam-gu Public Health Center.

This sub-route covers a distance of 8.5 km, stretching from the Namchang Village bus stop to the Dongnam-gu Public Health Center. The choice of this particular sub-route was driven by its reputation for high traffic volume, making it one of the busiest routes in the Cheonan area. Moreover, this sub-route encompasses a diverse range of public and private institutions, including several universities (such as Dankook University, Sangmyung University, Hoseo University, and Baekseok University), public and private hospitals (including Dankook Hospital and Dongnam-gu Public Health Center), large shopping malls, and various public areas. The commuter volume on this sub-route varies depending on the day of the week. On weekdays, there are significant increases during morning and evening rush hours as students travel to and from work and school. On the other hand, weekends generally witness lower congestion levels due to the closure of universities and hospitals in the area. These distinct scenarios contribute to the unique and multifaceted nature of this sub-route, making it an ideal choice for our analysis. The scheduled travel time for this sub-route is estimated to be approximately 27 min.

3.2. DTG Device

This study uses a data-driven methodology that uses portable devices and sensors to collect vehicle information. In the city of Cheonan, an array of commercial vehicles, including buses, cars, and taxis, are equipped with state-of-the-art sensors and devices, most notably DTG. The mandatory installation of DTG devices in all commercial vehicles across South Korea, driven by the overarching goal of enhancing road traffic safety, has been discussed in the introduction Section 1. The DTG device plays an important role

in recording real-time data, capturing essential parameters such as GPS location, brake signals, acceleration, and time stamps with a granularity of one-second intervals. It is important to note that the DTG device strictly adheres to privacy regulations, ensuring the exclusion of any personally identifiable information about the driver. To ensure the reliability and accuracy of data collection, the DTG device adheres to the necessary security standards. A multifaceted approach has been implemented, incorporating various security measures to safeguard the confidentiality, integrity, and availability of the collected data. For instance, the DTG device operates as an offline, secure device, minimizing the risk of unauthorized access or data breaches. The data from commercial vehicles are securely collected under the supervision of the South Korean government and safely transferred to the designated government office. Moreover, access to and downloading of data from the DTG device is restricted to registered individuals only, further bolstering data security and preventing unauthorized tampering or manipulation. Figure 2 shows a sample of a DTG device. Overall, the utilization of DTG devices ensures the collection of reliable and accurate data, enabling a robust analysis of vehicle-related parameters in the context of this study. The digital tachograph (DTG) data used in this study are securely managed by the Korea Transportation Safety Authority. While the DTG data can be obtained through a request to open access data in Korea, access to these data is limited to authorized personnel within South Korea. The data collection process is conducted with robust security measures, including offline operations, supervised data collection, and stringent access controls.



Figure 2. Sample of a DTG device. This figure presents a sample of a DTG (digital tachograph) device used for collecting transportation data.

3.3. Dataset Description

Raw data were gathered daily from 1 January 2020 to 30 May 2020 using DTG devices installed on all buses operating within Cheonan City. The DTG devices are designed to record various parameters, providing a comprehensive dataset for analysis. However, for our study, we selected specific information deemed relevant for predicting travel time. Table 2 shows the information recorded and stored by the DTG device. Among the available data, we narrowed our focus to six key variables: trip number, bus registration number, distance covered, longitude, latitude, and information on occurrence. To show our travel time prediction task, we selectively include a portion of the collected dataset in Table 3.

Table 2. Details of the digital tachograph (DTG) data used in the study.

Column Name	Description
Trip Key	An identifier for a specific trip taken by a vehicle. It uniquely identifies a particular journey made by the vehicle.
Device Model	The model of the DTG device used to record data about the vehicle's journey.
Bus Number	The unique identifier for the vehicle. It distinguishes one vehicle from another in the fleet.
Bus Type	The classification of the vehicle, such as rural bus, commercial bus, or minibus.
Business Number	The unique registration number assigned to the business owning the vehicle.
Distance covered	The distance covered by the vehicle during the trip.
Total Distance	The total distance the vehicle has covered since it was last serviced or had its odometer reset.
Bus Speed	The speed of the vehicle at a particular point during the trip.
Engine Rotation	The rotation speed of the vehicle's engine at a particular point during the trip.
Brake Signal	An indicator of whether the vehicle's brakes were applied during the trip.

Table 2. Cont.

Column Name	Description
Longitude	The longitude coordinates of the vehicle's location at a particular point during the trip.
Latitude	The latitude coordinates of the vehicle's location at a particular point during the trip.
Azimuth	The azimuth means direction of the vehicle's movement, as determined by GPS.
Device Status	An indicator of the vehicle's operational status, e.g., whether it is in service or out of service.
Operating Area	The geographic area in which the vehicle is authorized to operate.
Timestamp	A timestamp indicating when the data were recorded.

Table 3. Description of the travel time data extracted from raw digital tachograph (DTG) data.

Bus No.	Date	Time	Stop Name	Longitude	Latitude	Distance Covered (Km)
1216	9 January 2020	8:03:33 AM	Namchang Village bus stop	127.194035	36.265738	Trip Start
1216	9 January 2020	8:04:08 AM	Songnam-ri bus stop	127.191316	36.861926	0.5
1216	9 January 2020	8:04:41 AM	Cheonggu Villa bus stop	127.184462	36.85576	0.5
1216	9 January 2020	8:05:37 AM	Seokgyo 2-ri bus stop	127.184462	36.85576	0.4
1216	9 January 2020	8:06:01 AM	National Manghyang Cemetery bus stop	127.183946	36.85363	0.2
1216	9 January 2020	8:06:22 AM	Seonggeo Yukyeong A bus stop	127.183067	36.850893	0.3
1216	9 January 2020	8:07:01 AM	Yobang 3-ri bus stop	127.179921	36.846285	0.6
1216	9 January 2020	8:09:44 AM	Dankook Hospital bus stop	127.174243	36.83936	1.0
1216	9 January 2020	8:10:35 AM	Sangmyung University bus stop	127.173299	36.833026	0.7
1216	9 January 2020	8:11:31 AM	Cheonan Toll Gate bus stop	127.167579	36.828419	0.8
1216	9 January 2020	8:12:24 AM	Dosol Square bus stop	127.162813	36.822393	0.8
1216	9 January 2020	8:14:34 AM	Daelim Hansup Apartment bus stop	127.159746	36.819947	0.5
1216	9 January 2020	8:16:34 AM	Cheonan General Terminal bus stop	127.155642	36.819008	0.4
1216	9 January 2020	8:18:37 AM	Bangjukan Five-way Street bus stop	127.151097	36.817461	0.5
1216	9 January 2020	8:19:37 AM	Bokja Girls Middle and High School bus stop	127.149857	36.815137	0.3
1216	9 January 2020	8:20:35 AM	Samdo Shopping Mall bus stop	127.148792	36.812047	0.3
1216	9 January 2020	8:22:59 AM	Cheonan Station bus stop	127.149029	36.808867	0.4
1216	9 January 2020	8:23:41 AM	Dongnam-gu Public Health Center bus stop	127.151555	36.807345	0.3

The accuracy of travel time estimation, whether in urban or rural areas, is profoundly influenced by prevailing weather conditions [30]. Previous studies highlighted the negative impact of severe weather on travel time reliability [31]. To address this concern, we devised a mapping approach that integrates weather data with travel time information, accounting for any temporal discrepancies that may arise between the two datasets. To ensure precise analysis, we established a common time and date column to establish a robust link between the weather and travel time data, ensuring that the datasets were aligned for accurate analysis. As an integral part of our feature set, we incorporated a rich assortment of weather conditions sourced from the esteemed Korea Meteorological Administration (KMA) dataset. This comprehensive dataset encompasses crucial meteorological variables, including temperature, air pressure, humidity, and precipitation. To comprehensively assess the varying impact of weather conditions across distinct seasons and their particular significance for travel, we meticulously gathered weather data over five months, with a specific emphasis on January, February, March, April, and May. These months were deliberately selected to exemplify the distinct weather patterns commonly observed in Korea. In Korea, January is a winter month with frequent snowfall, February sees a decrease in snowfall and occasional rain, March has heavy traffic as schools commence a new semester, and April and May are the middle of spring, characterized by lower rainfall and mild weather, making it an enticing time to visit South Korea. This extension allows us to explore the implications of various weather conditions across different seasons in a more comprehensive manner. An example of the raw weather data is shown in Table 4. Finally, we combined the weather features with the travel time data to create a final feature set, which includes bus number, date, time, stop name, longitude, latitude, distance covered, temperature, air pressure, humidity, and precipitation. By incorporating these diverse dimensions into our analysis, we aim to unveil the intricate interplay between weather conditions and travel time dynamics, thus paving the way for more accurate and reliable travel time predictions.

Table 4. Weather conditions information used in the study.

Date	Temperature	Air Pressure	Humidity	Precipitation
Sunday, 1 March 2020	3.1	1.9	63.4	1.5
Monday, 2 March 2020	2.4	2	62.9	1.5
Tuesday, 3 March 2020	3.1	2.1	62.9	1.4
Wednesday, 4 March 2020	3.9	2.1	62.8	1.3
Thursday, 5 March 2020	3.2	2.2	62.8	1.4

3.4. Data Preprocessing

The data collected from the DTG device underwent several preprocessing steps, including data scrubbing, matching weather data, data standardization, and partitioning the data into specific study areas for analysis. These preprocessing techniques ensure that the data are in a suitable format for further investigation. To optimize the performance of the deep learning models, hyperparameter tuning was conducted using the preprocessed data. This involved selecting the appropriate batch size, determining optimal factors, tuning the number of hidden layers, and selecting the number of epochs. By fine-tuning these hyperparameters, we aimed to achieve the best possible performance from the models. Once the hyperparameters were optimized, a comprehensive comparison was performed among different deep learning models. These models included pure LSTM, pure GRU, LSTM bidirectional, GRU bidirectional, Stacked-LSTM, Stacked-GRU, and our proposed model HTF-NET. By evaluating and contrasting the performance of these models, we sought to identify the most effective approach for bus travel time prediction. The flow chart of the study is shown in Figure 3.

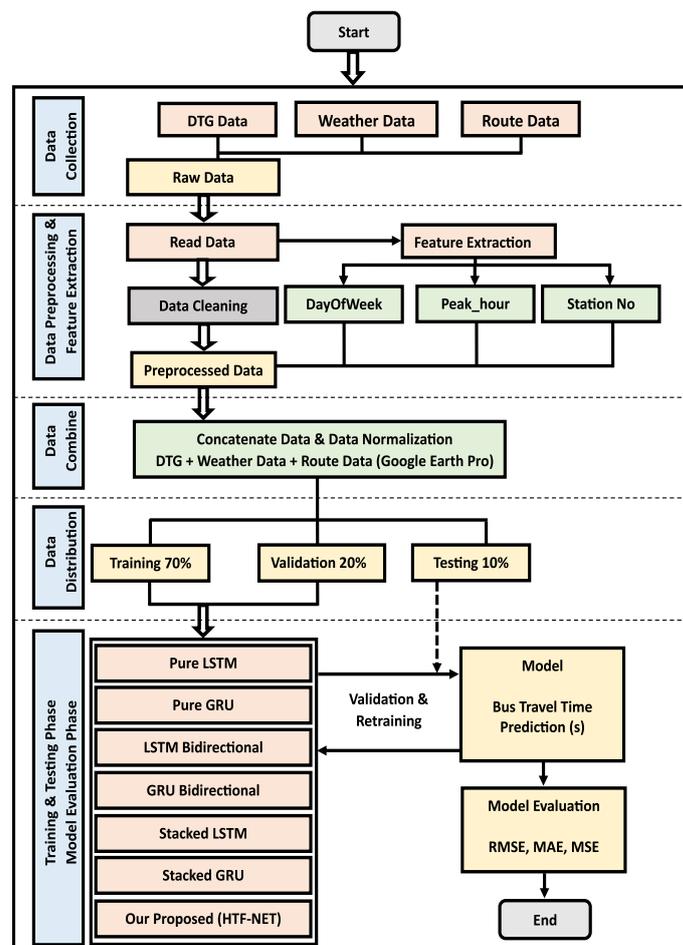


Figure 3. Flowchart of the study methodology, depicting the steps and processes followed throughout the research.

To ensure the integrity and accuracy of our analysis, we implemented several steps in our research:

- **Data scrubbing:** We conducted data scrubbing to eliminate duplicates, address missing values, correct inaccuracies, and remove outliers, ensuring the reliability of our dataset. Notably, approximately 5–10% of data are missing when buses begin their journeys from the initial stops, which we exclude during preprocessing. Importantly, we encounter minimal to no missing data while buses are in transit within our study area, covering Namchang Village bus stop to Dongnam-gu Public Health Center bus stop.
- **Matching weather data:** To assess the influence of weather conditions on bus travel time, we synchronized our datasets using a 10 s interval. This synchronization is crucial for aligning timestamps between the travel time and weather data by utilizing common date and time columns. By merging these datasets, we effectively analyzed the correlation between weather conditions and bus travel time.
- **Data standardization:** For data consistency and interpretability, we rigorously standardized the variables by scaling them to a common range, normalizing their values, and enhancing the data format. This leveled the playing field for all variables in our analytical models, improving accuracy, reliability, and our ability to detect meaningful patterns and trends in the dataset.
- **Partitioning the data for analysis:** We partitioned the data for analysis in a specific study area for bus travel time. This involved selecting the study area, partitioning the data, determining the time period, considering the sample size, and ensuring data quality. The data were also divided into training, validation, and testing sets.

3.5. Feature Selection

A series of steps were undertaken to preprocess the GPS dataset specifically for the study area of Cheonan. Initially, a subset of the data was created that exclusively comprised GPS trajectories from trips taken within the Cheonan study area. This subset was generated by visualizing the dataset on Google Earth Pro and manually selecting data points that fell within the geographical boundaries of Cheonan. Subsequently, the longitude and latitude values of the bus stops were extracted from the route information and plotted accurately on Google Earth Pro. This allowed for precise visualization of the exact positioning of the buses at each stop. An imputation strategy was implemented to address any missing or incomplete records within the dataset. Specifically, missing records were filled in with the mean values derived from the closest surrounding records. Consequently, through these preprocessing steps, the GPS dataset was cleansed and made ready for input into the deep learning algorithm. This preparation ensures that the subsequent analysis and prediction tasks can be conducted with reliable and accurate data. The analysis included several parameters related to the bus transportation system, such as the bus number, number of stops, distances between stops, days of the week, arrival and departure times, and weather conditions. These parameters were categorized into two groups, namely, dynamic and static variables. Dynamic variables consisted of travel times between stops, duration of stays, and weather, while static variables included the bus route, vehicle model, days of the week, holidays, and working days. The input features consisted of route number, starting geographical location, ending geographical location, bus number, and departure time (hours, minutes, and seconds), which is converted into a Unix timestamp, days of the week, holidays, distance, and weather conditions such as temperature, humidity, and air pressure. The output prediction series was the travel time in seconds.

4. Travel Time Prediction Models

4.1. Long Short-Term Memory

Hochreiter and Schmidhuber [32] introduced the LSTM model as an effective tool for learning long-term dependencies. This model has demonstrated remarkable success across diverse domains, including finance, healthcare, and transportation. Its applications range from predicting stock prices and diagnosing diseases to forecasting traffic flow [33,34].

Notably, LSTM has gained significant traction among researchers for predicting bus travel times [35–37]. LSTM excels at capturing segment-level and long-term information in traffic data due to its intricate structure, as illustrated in Figure 4. This complexity arises from its gating mechanism, encompassing the forget, input, and output gates. These gates, defined by Equations (1)–(3), empower LSTM to address long-term dependencies by extending the memory cycle of the network.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (2)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (3)$$

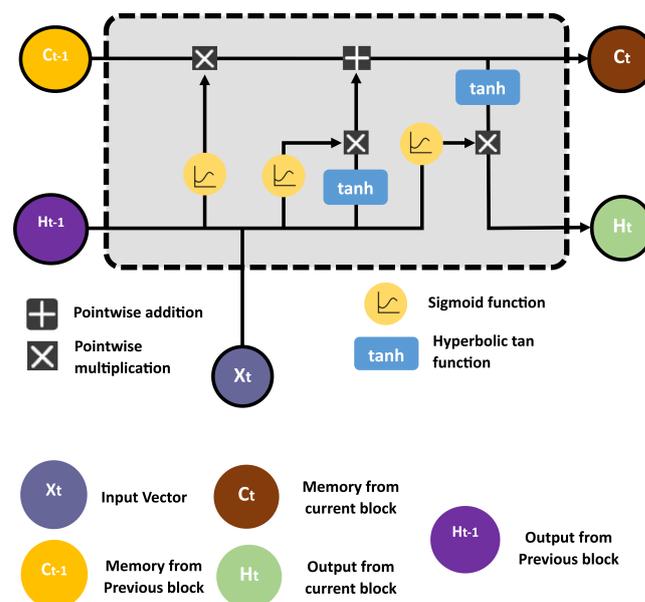


Figure 4. Structure of an LSTM cell, displaying the key components of a Long Short-Term Memory (LSTM) architecture.

In the context of LSTM, important components include the forget gate (f_t), input gate (i_t), and output gate (o_t) at each time step (t), with σ representing the sigmoid activation function. These gates are governed by respective weight matrices (W_f , W_i , W_o) and biases (b_f , b_i , b_o). The LSTM computations involve the previous hidden state (h_{t-1}) and the current input (x_t) to compute the LSTM cell state (C_t) and hidden output (h_t), as referenced from the source [38].

We used different layered LSTM architectures for bus travel time prediction, which are described below:

- **Pure LSTM:** The “Pure LSTM” model is such an architecture that relies solely on the LSTM cells without any additional layers or modifications. The pure LSTM model is made up of two layers: the LSTM layer and the dense layer. The LSTM layer contains 64 units with 20,992 trainable parameters, while the dense layer produces a sequence of one-dimensional vectors with a single element and has 65 trainable parameters. The total number of parameters in the pure LSTM model is 21,057.
- **LSTM bidirectional:** The “LSTM bidirectional” model consists of two bidirectional layers and a dense layer. This architecture has gained popularity due to its ability to model the dependencies of sequential data in both forward and backward directions. Bidirectional layers can capture patterns from past and future contexts, resulting in a more comprehensive understanding of the sequence. The model consists of a dense layer that produces a sequence of one-dimensional vectors with a single element, and the LSTM bidirectional model has 83,265 trainable parameters.

- Stacked-LSTM:** We stacked multiple layers of LSTM. The Stacked-LSTM model can improve the accuracy of bus travel predictions. The LSTM stack model is composed of four LSTM layers and one dense layer. The LSTM stack model has a total of 539,553 trainable parameters. This architecture has been designed to enhance the capabilities of the LSTM network, allowing for a detailed analysis of bus travel time.

4.2. Gated Recurrent Unit

The GRU (Gated Recurrent Unit), another refined variant of recurrent neural networks (RNNs), offers a more streamlined architectural approach compared to LSTM by employing just two gates: the update gate and the reset gate, as opposed to LSTM’s three. This simplification enhances the GRU’s overall efficiency and reduces the number of trainable parameters, as noted in [39]. In this experiment, we utilized a two-layer GRU model. The structural representation of the GRU cell can be observed in Figure 5, and the mathematical expressions governing the functioning of these two gates to regulate information flow within the cell are detailed in Equations (4)–(7). The equations describing the GRU model are sourced from [40].

$$u_t = \sigma(W^u x_t + U^u h_{t-1}) \tag{4}$$

$$r_t = \sigma(W^r x_t + U^r h_{t-1}) \tag{5}$$

$$h'_t = \mu(W x_t + r_t \odot U h_{t-1}) \tag{6}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t \odot h'_t) \tag{7}$$

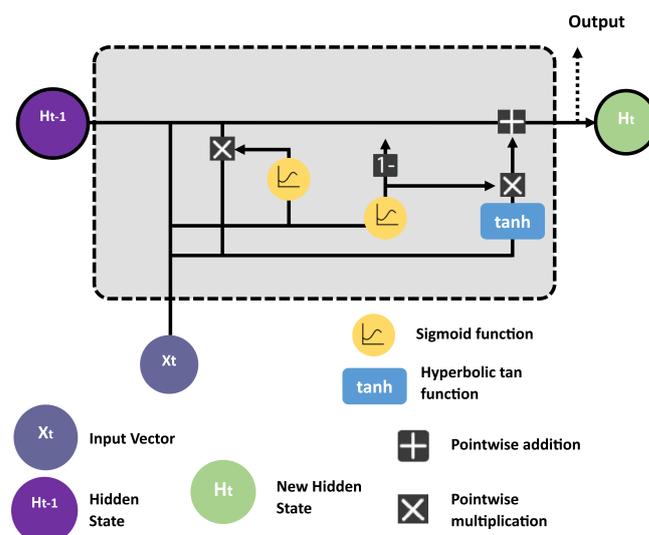


Figure 5. Structure of a GRU cell, depicting the key components of a Gated Recurrent Unit (GRU) architecture.

In this context, u_t signifies the update gate, r_t represents the reset gate, h_{0t} stands for the current memory content, and h_t represents the final memory content at time t . The symbols σ and μ denote the sigmoid and tanh activation functions, respectively. Furthermore, the \odot symbol denotes element-wise multiplication, while W^u and U^u are the weight matrices corresponding to the two gates.

We used different layered GRU architectures for bus travel time prediction, which are described below:

- Pure GRU:** The “Pure GRU” model is a specific type of GRU network that employs only GRU cells, without any additional layers or modifications. This model consists of a GRU layer and a dense layer. The GRU layer has 64 units with 15,936 trainable parameters. The dense layer in this model has 65 trainable parameters and produces

a sequence of one-dimensional vectors with a single element. The total number of parameters in Pure LSTM is 16,001.

- **GRU bidirectional:** The “GRU bidirectional” model consists of two bidirectional layers and a dense layer. It has a dense layer that generates a sequence of one-dimensional vectors, each containing a single element. It is worth noting that the GRU bidirectional model has 63,041 trainable parameters.
- **Stacked-GRU:** The “Stacked-GRU” model comprises one dense layer and four GRU layers. It has 406,113 trainable parameters.

4.3. Attention Model

Attention mechanisms play an important role in enhancing the performance of time series models by allowing them to dynamically focus on relevant temporal information. In the context of neural networks, attention mechanisms were popularized by seminal works such as [15,16]. These mechanisms assign weights to different elements of the input sequence based on their relevance to the current step.

The scoring function computes a set of scores, e_i , for each element in the sequence, given by:

$$e_i = \text{score}(h_t, h_i) \quad (8)$$

where h_t represents the current hidden state and h_i are the hidden states of the sequence. The attention weights, denoted by a_i , are then calculated using a softmax function to normalize the scores:

$$a_i = \frac{\exp(e_i)}{\sum_{j=1}^n \exp(e_j)} \quad (9)$$

These attention weights are then used to calculate the context vector, C_t , by applying a weighted sum over the sequence:

$$C_t = \sum_{i=1}^n a_i \cdot x_i \quad (10)$$

The context vector, C_t , is then combined with the current hidden state for further processing.

4.4. Support Vector Regression

Support Vector Regression (SVR) is a powerful machine learning model that extends the principles of Support Vector Machines (SVM) to regression problems [41]. In the context of time series modeling, SVR is particularly valuable for predicting continuous values based on historical data.

4.5. Our Proposed Hybrid Temporal Forecasting Network (HTF-Net) Model

The Hybrid Temporal Forecasting Network (HTF-NET) represents an advanced neural network architecture engineered for precise temporal forecasting, with a specific focus on predicting bus travel times. This model leverages an integration of LSTM and GRU layers, augmented by an attention mechanism. Notably, the model’s predictive capabilities are further refined through integration with an SVR for travel time predictions.

The attention mechanism, manifested as a custom 3D attention block, assumes an important role in elevating the model’s discernment of pertinent temporal patterns within input sequences. The mechanism involves a sequence of operations, including permutation, a dense layer with softmax activation, and element-wise multiplication. These operations collectively shape the input sequence, compute dynamic attention weights, and apply them to the original input sequence, thereby enhancing the model’s temporal representation.

The HTF-NET architecture begins with an input layer tailored for sequences, each representing a single time step. Successively, an LSTM layer with 512 units captures initial temporal dependencies, followed by the application of the custom attention mechanism. A GRU layer with 256 units further refines temporal features, and the outputs of the GRU and attention layers are intelligently concatenated along the last axis.

Subsequent layers involve additional LSTM units with decreasing capacities (128, 64, 32) strategically employed to capture hierarchical temporal representations. The architecture culminates in a dense output layer generating a singular output, representing the model’s precise prediction. The model undergoes training with the mean absolute error loss function and the Adam optimizer, with early stopping mechanisms in place to mitigate overfitting. Predictions are flattened, and an SVR model is subsequently trained to utilize these flattened predictions. Evaluation against ground truth values is conducted using metrics such as RMSE, MAE, and MSE, providing a robust assessment of the model’s predictive prowess. Figure 6 shows our proposed approach and a brief overview of the proposed HTF-NET model.

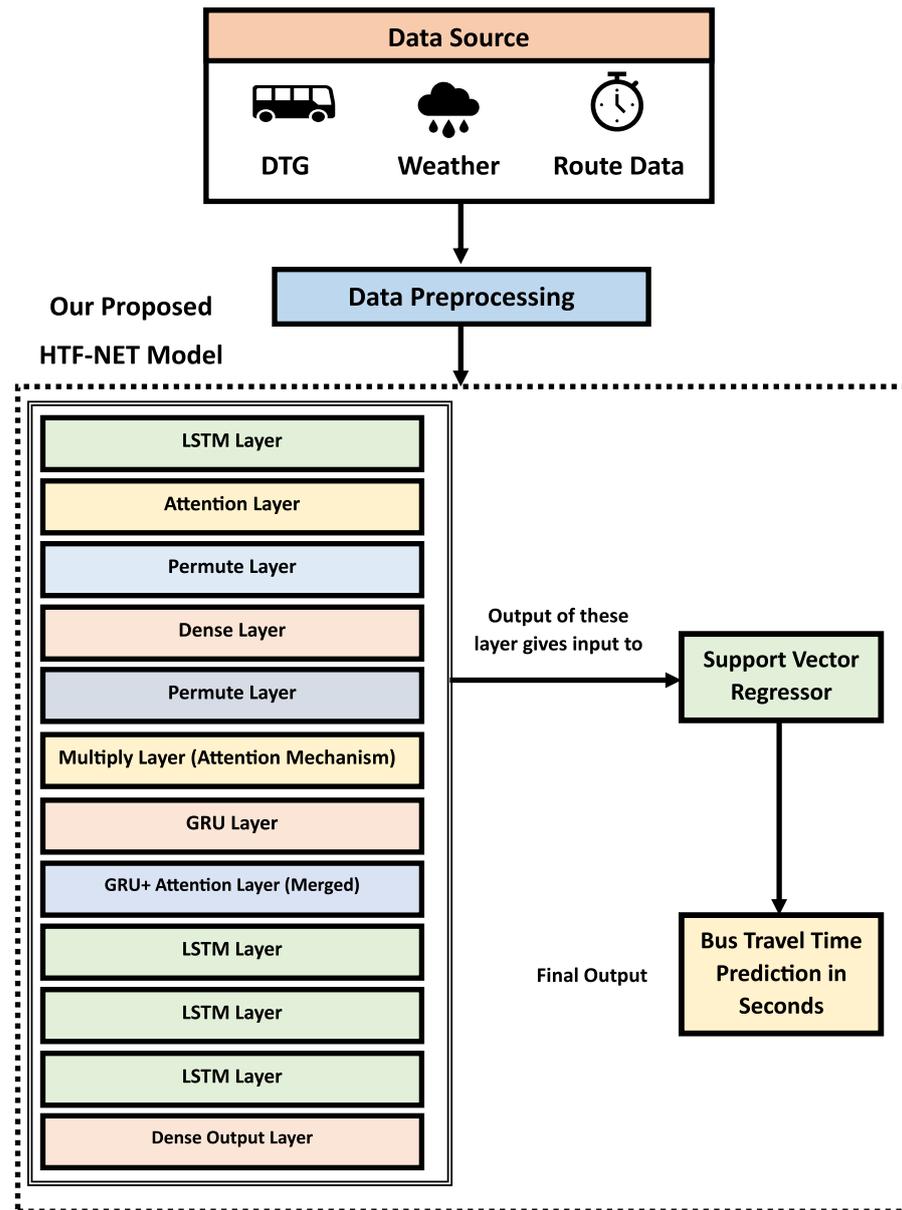


Figure 6. Overview of the proposed HTF-NET model.

4.6. Hyperparameter Settings

The hyperparameters for our deep learning models, outlined in Table 5, were fine-tuned through a series of experimental runs. Key parameters, including the learning rate, hidden layer quantity, number of neurons per hidden layer, and batch size, were optimized. The models consistently employed the ‘adam’ optimizer, ‘relu’ activation function, a learning rate of ‘0.001’, and a batch size of ‘32’, as summarized in Table 5.

Table 5. Hyperparameters of the models.

Model	Parameters	Value
Pure LSTM	layers	2
	LSTM Layer	64 neurons
	dense layer	1 neuron
Pure GRU	layers	2
	GRU Layer	64 neurons
	dense layer	1 neuron
LSTM bidirectional	layers	3
	Bidirectional LSTM Layer 1	64 neurons
	Bidirectional LSTM Layer 2	32 neurons
	dense layer	1 neuron
GRU bidirectional	layers	2
	Bidirectional GRU Layer 1	64 neurons
	Bidirectional GRU Layer 2	32 neurons
	dense layer	1 neuron
Stacked-LSTM	layers	5
	LSTM Layer 1	256 neurons
	LSTM Layer 2	128 neurons
	LSTM Layer 3	64 neurons
	LSTM Layer 4	32 neurons
	Dense Layer	1 neuron
Stacked-GRU	layers	5
	GRU Layer 1	256 neurons
	GRU Layer 2	128 neurons
	GRU Layer 3	64 neurons
	GRU Layer 4	32 neurons
	Dense Layer	1 neuron
HTF-NET	layers	7
	LSTM Layer 1	512 neurons
	Attention Layer	1
	GRU Layer 1	256 neurons
	LSTM Layer 2	128 neurons
	LSTM Layer 3	64 neurons
	LSTM Layer 4	32 neurons
Dense Layer	1 neuron	

4.7. Performance Metrics

To evaluate the accuracy of our deep learning model in predicting bus travel time, we utilized three widely recognized performance metrics: RMSE, MAE, and MSE. In these calculations, t_i represents the actual travel time for the i th trip on the route, while \hat{t}_i represents the predicted travel time for the i th trip.

- **Root mean squared error (RMSE):** RMSE quantifies the average distance between the predicted and actual travel times, measuring the overall prediction error. RMSE is a widely used metric and emphasizes larger errors. It is calculated using Equation (8):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (t_i - \hat{t}_i)^2} \quad (11)$$

- **Mean absolute error (MAE):** MAE provides the average absolute difference between predicted and actual travel times. MAE is a robust metric that offers a straightforward interpretation of the average prediction error magnitude. Unlike RMSE, it does not involve squaring the errors, making it less sensitive to outliers. By incorporating MAE into our analysis, we can gain insights into the average deviation of our predictions from the true values. It is calculated using Equation (9):

$$MAE = \frac{1}{n} \sum_{i=1}^n |t_i - \hat{t}_i| \quad (12)$$

- **Mean squared error (MSE):** MSE computes the average squared difference between predicted and actual travel times. MSE, similar to RMSE, emphasizes larger errors due to the squared differences. It is calculated using Equation (10):

$$MSE = \frac{1}{n} \sum_{i=1}^n (t_i - \hat{t}_i)^2 \quad (13)$$

For detailed derivations, see Appendix A.

5. Results and Discussion

5.1. Experimental Settings

Our study focuses on predicting bus travel time using deep learning models, and for this purpose, we conducted experiments on a Windows 10 Pro machine. The machine specifications were as follows: 12th Gen Intel (R) Core-TM i7-12700 processor, 32.0 GB of RAM, and a 500 GB WD Blue SN570 hard disk. The graphics card used was the NVIDIA GeForce RTX 3060, with 8 GB of RAM. To build and execute deep learning models, we used Python version 3.11.0 in combination with the TensorFlow framework. Additionally, we used Keras version 2.7.0 as the high-level API for model construction and training. Our dataset for bus travel time prediction consisted of 6100 trips over five months, encompassing rush and non-rush hours, weekdays, and weekends. We divided the dataset into three subsets: 70% for training, 20% for validation, and 10% for testing purposes.

5.2. Performance Evaluation of All Models Using the Overall Test Data

According to the experimental results presented in Table 6, our proposed HTF-NET model outperformed all other models in terms of predicting bus travel time. The effectiveness of the HTF-NET model becomes especially pronounced when we employ it to predict an entire bus journey, exemplified in Figure 7, starting from Namchang Village bus stop and concluding at Dongnam-gu Public Health Center. This superiority is further evident when comparing actual and predicted travel times across the test dataset, as illustrated in Figure 8.

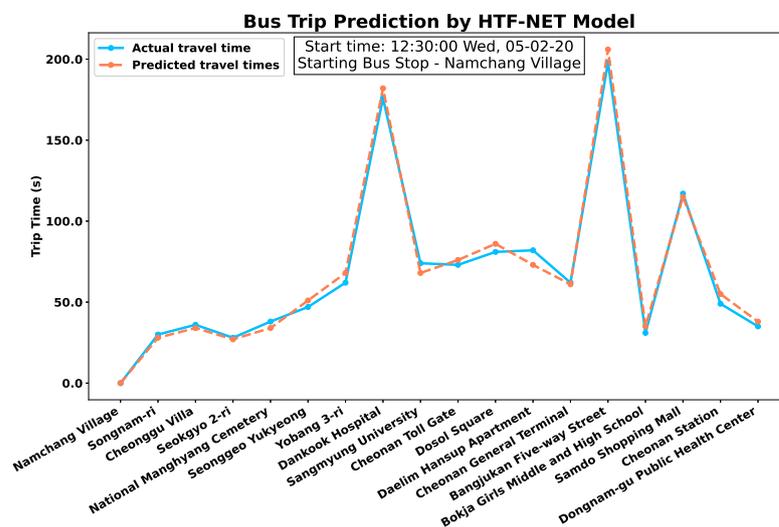


Figure 7. Comparison of actual travel time with HTF-NET model predictions on the entire bus trip from Namchang Village bus stop to Dongnam-gu Public Health Center.

In order to evaluate the effectiveness of our bus travel time prediction model, we present examples of its performance for various origin and destination pairs in Table 7. The table shows the predicted and actual travel times for different trips, allowing us to assess the model's accuracy in capturing real-world travel patterns. Upon examination of the table, it becomes evident that our model performs admirably in most cases, particularly during rush hour periods. The predicted travel times closely align with the actual travel times, indicating a high degree of accuracy and reliability. This is demonstrated by the minimal

difference between predicted and actual times for the majority of the trips. However, it is crucial to acknowledge that there are specific instances in which the model exhibits some inconsistencies in its predictions. Notably, in rows 18 and 19, we can observe a significant disparity between the predicted and actual travel times. These disparities can be attributed to a range of factors, including traffic congestion, the influence of traffic lights, and the prevailing road conditions at the time of travel. These complex scenarios underscore the pressing need for further refinement and enhancement of the model to bolster its overall accuracy and resilience. Through a comprehensive analysis of our bus travel time prediction model's performance using these examples, we glean valuable insights into both its strengths and limitations. This evaluation establishes a solid foundation for future research and improvements, empowering us to devise strategies that effectively address the identified challenges and bolster the model's predictive capabilities.

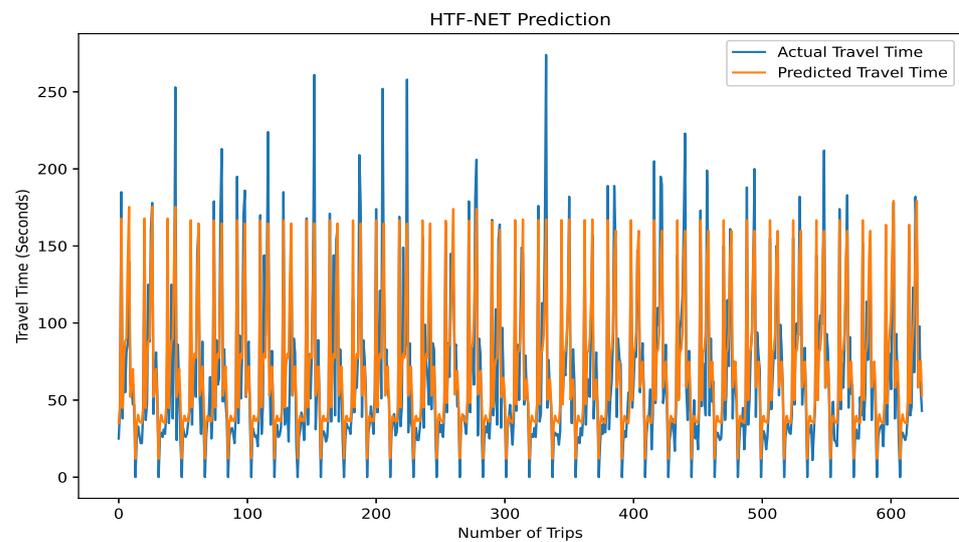


Figure 8. Comparison of actual travel time with HTF-NET model predictions on the entire test dataset.

Table 6. Performance evaluation of all models on the overall test data.

Model Name	RMSE	MAE	MSE
Pure LSTM	33.94	23.17	1152.61
Pure GRU	34.88	24.01	1216.92
LSTM bidirectional	26.65	16.97	710.56
GRU bidirectional	28.06	17.37	787.64
Stacked-LSTM	23.74	15.42	563.73
Stacked-GRU	27.78	17.49	772.09
Our proposed model (HTF-NET)	19.62	13.26	428.64

Furthermore, to establish the robustness and generalizability of our proposed models, we carried out four additional experiments. These experiments aimed to explore the influence of weather-related features on our models and evaluate their performance when trained and tested exclusively on weekday and weekend data. Remarkably, even in these varying conditions, only a marginal decline in model performance was observed.

Table 7. Examples of predictions by our proposed model HTF-NET on the test dataset.

No.	Origin	Destination	Actual Time Travel	Predict Travel Time	Date and Time
1	Namchang Village	Songnam-ri	30	28	5 January 2020 and 09:48 AM
2	Songnam-ri	Cheonggu Villa	36	32	6 January 2020 and 10:52 AM
3	Cheonggu Villa	Seokgyo 2-ri	28	27	12 January 2020 and 12:53 PM
4	Seokgyo 2-ri	National Manghyang Cemetery	38	34	16 January 2020 and 01:43 AM
5	National Manghyang Cemetery	Seonggeo Yukyeong	47	51	18 January 2020 and 03:38 PM
6	Seonggeo Yukyeong	Yobang 3-ri	62	68	22 January 2020 and 04:53 PM
7	Yobang 3-ri	Dankook Hospital	176	185	28 January 2020 and 01:43 PM
8	Dankook Hospital	Sangmyung University	74	68	11 February 2020 and 08:38 AM
9	Sangmyung University	Cheonan Toll Gate	73	76	12 February 2020 and 06:40 PM
10	Cheonan Toll Gate	Dosol Square	81	86	18 February 2020 and 06:43 PM
11	Dosol Square	Daelim Hansup Apartment	82	73	22 March 2020 and 10:45 AM
12	Daelim Hansup Apartment	Cheonan General Terminal	62	61	26 March 2020 and 03:53 PM
13	Cheonan General Terminal	Bangjukan Five-way Street	198	203	27 March 2020 and 08:52 AM
14	Bangjukan Five-way Street	Bokja Girls Middle and High School	31	35	2 April 2020 and 09:52 AM
15	Bokja Girls Middle and High School	Samdo Shopping Mall	117	115	6 April 2020 and 04:14 PM
16	Samdo Shopping Mall	Cheonan Station	49	55	12 April 2020 and 08:37 PM
17	Cheonan Station	Dongnam-gu Public Health Center	35	38	18 April 2020 and 08:38 AM
18	Cheonan terminal	Bangjukan Street	134	186	13 January 2020 and 05:00 PM
19	Samdo Shopping Mall	Cheonan Station	136	80	27 February 2020 and 08:50 AM

5.3. Weather Influence on Travel Time Prediction

To assess the impact of weather conditions on travel time prediction, we evaluated seven deep learning models. Our objective was to demonstrate the significance of incorporating weather features in improving the accuracy of travel time prediction. Initially, we trained and tested our models using the complete dataset, including weather features. The performance of each model was measured in terms of RMSE as an evaluation metric. We then removed the weather features from the dataset and re-evaluated the performance of the models under the same conditions. The results, as summarized in Table 8, indicate that the models performed less effectively when weather data were excluded. This observation suggests that weather conditions indeed play an important role in travel time estimation. This emphasizes the importance of considering weather data for accurate travel time predictions. Figure 9 illustrates the difference between the RMSE calculated with the complete dataset and the RMSE obtained when weather data were omitted. Among the deep learning models, including pure LSTM, pure GRU, LSTM bidirectional, GRU bidirectional, Stacked-LSTM, Stacked-GRU, and HTF-NET, all models exhibited a noticeable decline in performance when weather features were removed.

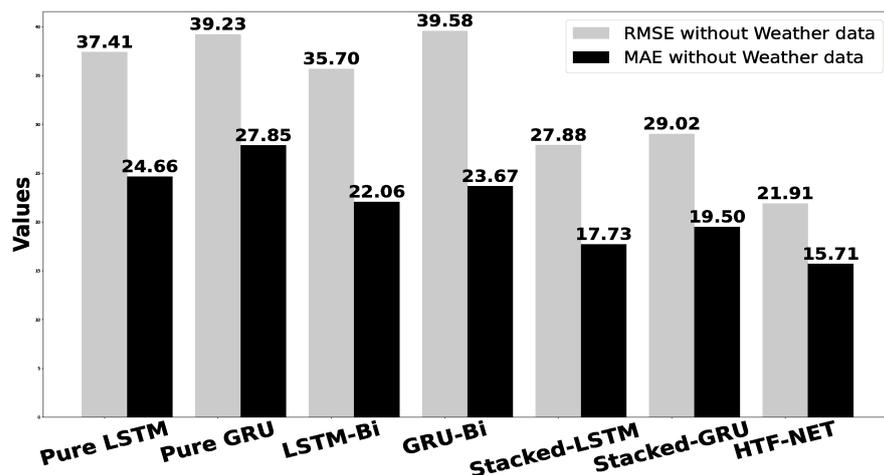


Figure 9. This figure depicts the RMSE and MAE values without incorporating weather data in the prediction models.

Our study reveals that incorporating weather-related features enhances the performance of deep learning models in bus travel time predictions. The results indicate that the HTF-NET with weather data achieves an RMSE of 19.62, compared to an RMSE of 21.91 without weather data. This addition also lowers the MAE from 15.71 to 13.26 and the MSE from 480.06 to 428.64. These improvements underscore that excluding weather data can

negatively impact model accuracy, emphasizing the importance of considering weather conditions for reliable travel time predictions.

Table 8. Experimental outcomes: Investigating weather conditions’ effects on travel time prediction.

Model No.	Model Name	RMSE	MAE	MSE
1	Pure LSTM	37.41	24.66	1434.01
2	Pure GRU	39.23	27.85	1539.06
3	LSTM bidirectional	35.70	22.06	1275.04
4	GRU bidirectional	39.58	23.67	1567.10
5	Stacked-LSTM	27.88	17.73	777.51
6	Stacked-GRU	29.02	19.50	877.23
7	HTF-NET	21.91	15.71	480.06

5.4. Reliability Analysis of Models during Weekdays’ Data

The present study aimed to investigate the effectiveness of using only weekday data to predict bus travel time. This is because weekdays are characterized by varying travel patterns due to school and office schedules, as well as peak and non-peak hours. To investigate this, we conducted a comprehensive experiment and the results are compiled in Table 9. The comparison between the RMSE values obtained using the entire dataset and the subset of weekday data is depicted in Figure 10. The outcomes of our analysis indicate that our proposed model, HTF-NET, yielded the most favorable results in terms of the RMSE, achieving a value of 20.16. Additionally, the MAE was found to be 14.95, while the MSE amounted to 472.41.

Table 9. Weekday-only TTP data: This table presents experimental results derived solely from TTP data collected on weekdays.

Model No.	Model Name	RMSE	MAE	MSE
1	Pure LSTM	30.62	21.41	937.63
2	Pure GRU	34.56	24.10	1194.70
3	LSTM bidirectional	27.53	17.15	758.30
4	GRU bidirectional	33.65	21.83	1132.56
5	Stacked-LSTM	24.76	16.12	613.28
6	Stacked-GRU	25.01	16.30	625.59
7	HTF-NET	20.16	14.95	472.41

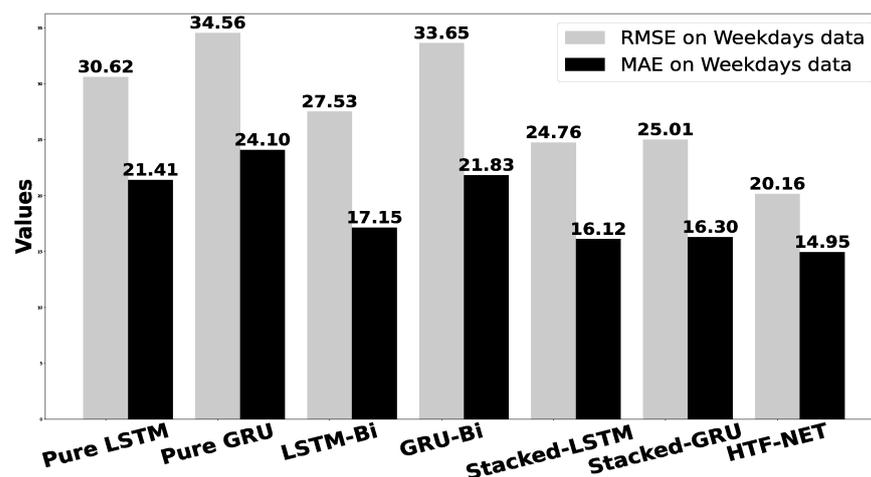


Figure 10. RMSE and MAE of prediction models using weekday-only data: This figure shows the RMSE and MAE values of prediction models using the weekday-only TTP data subset.

5.5. Reliability Analysis of Models during Weekends' Data

This experiment aimed to evaluate the reliability of seven distinct models in predicting bus travel times, specifically during weekends. This investigation is important due to the distinct travel patterns observed on weekends, characterized by the absence of school or office schedules and the lack of peak or non-peak hours. The experimental results are shown in Table 10. Additionally, a comparison between the RMSE values obtained using the entire dataset and the subset comprising only weekend data is illustrated in Figure 11. Notably, the HTF-NET model demonstrated better performance compared to the other models when predicting bus travel times during weekends. These outcomes underscore the effectiveness of the HTF-NET model in capturing the complexities of weekend travel patterns.

Table 10. Weekend-only TTP data: This table presents experimental results derived solely from TTP data collected on weekends.

Model No.	Model Name	RMSE	MAE	MSE
1	Pure LSTM	33.49	22.48	1122.24
2	Pure GRU	35.98	24.42	1294.65
3	LSTM bidirectional	24.83	15.93	616.84
4	GRU bidirectional	37.44	21.56	1133.27
5	Stacked-LSTM	23.20	15.39	538.47
6	Stacked-GRU	27.31	17.24	746.21
7	HTF-NET	20.18	13.11	510.33

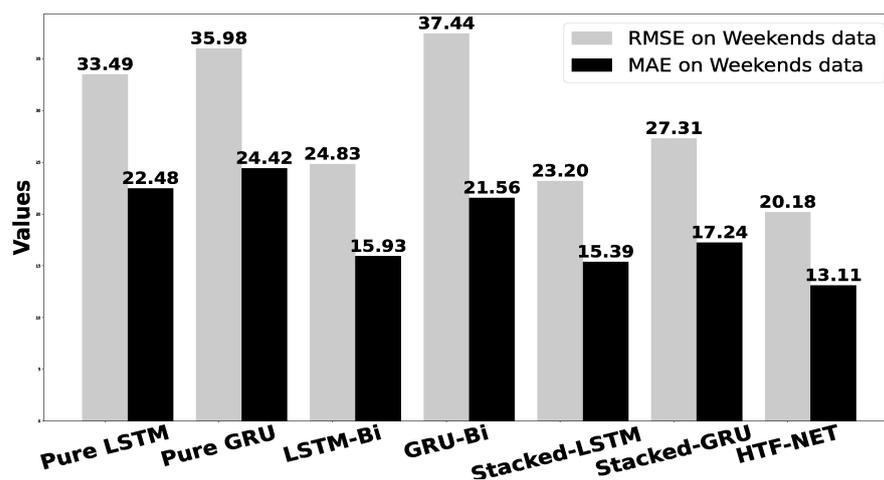


Figure 11. RMSE and MAE of prediction models using weekend-only data: This figure shows the RMSE and MAE values of prediction models using the weekend-only TTP data subset.

5.6. Robustness of Models on Short Routes

Furthermore, in order to demonstrate the robustness and generalizability of our proposed models, we conducted an additional experiment to evaluate their performance on shorter routes. Initially, our models were trained on a long route, specifically from the Namchang Village bus stop to the Dongnam-gu Public Health Center. This long route spans approximately 8.5 km with an estimated travel time of 27 min, as depicted in Figure 12 from start to end. To assess the generalization ability of our models on shorter routes, we selected a sub-route from the Dankook University Hospital bus stop to Cheonan Station, which is also illustrated in Figure 12. The sub-route is highlighted by the blue line. This shorter route covers a distance of approximately 4.8 km with a scheduled travel time of 17 min.

We experimented with seven deep learning models for this short route and the results are shown in Table 11, which provides an overview of the results. The comparison between the RMSE values obtained using the entire dataset and the data from the short route is shown in Figure 13.

Notably, among the seven models tested, the HTF-NET model demonstrated superior performance when predicting bus travel times on the short route. These outcomes underline

the effectiveness of the HTF-NET model in capturing the complexities inherent in the short route. Taken together, our findings indicate that our proposed models exhibit good accuracy and robustness not only on long routes but also on shorter ones. By successfully predicting travel times on both types of routes, our models demonstrate their generalizability and suitability for real-world applications in the transportation domain.

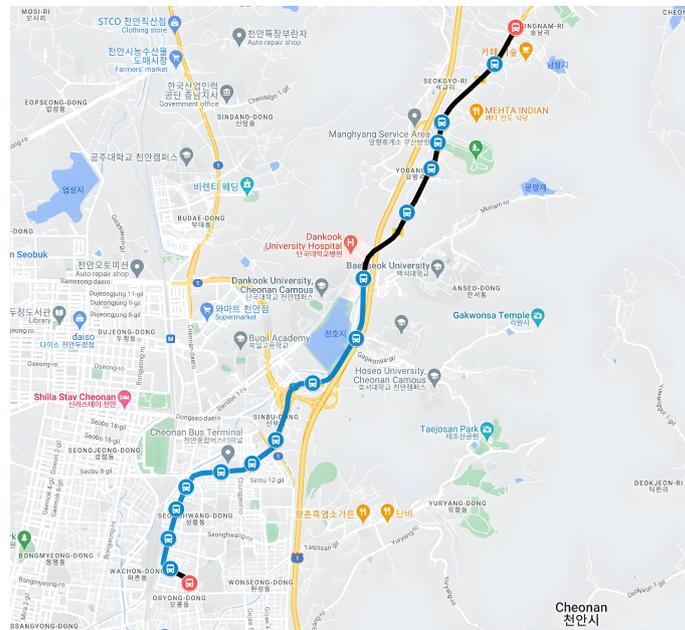


Figure 12. Short route: This figure shows the bus route from Dankook University Hospital bus stop to Cheonan Station.

Table 11. Results from short route data: This table summarizes the findings obtained from analyzing the short route data.

Model No.	Model Name	RMSE	MAE	MSE
1	Pure LSTM	30.22	20.18	1026.22
2	Pure GRU	32.39	22.65	1113.22
3	LSTM bidirectional	21.22	13.16	586.62
4	GRU bidirectional	33.23	19.28	1039.28
5	Stacked-LSTM	19.10	13.29	512.27
6	Stacked-GRU	25.12	15.18	685.21
7	HTF-NET	17.12	11.21	490.27

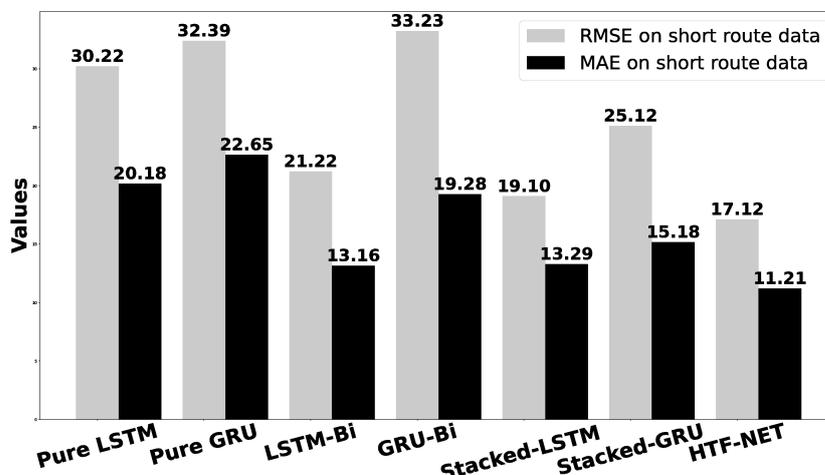


Figure 13. RMSE and MAE of prediction models with short route data: This figure displays the RMSE and MAE values of prediction models using data from the short route in TTP.

5.7. Comparison of All Models with Baseline Model ARIMA

The autoregressive integrated moving average (ARIMA) model is widely used in time series forecasting, combining autoregressive (AR) and moving average (MA) components. ARIMA models are instrumental in predicting future values based on historical observations, making them a valuable tool for time series data analysis and prediction [8]. In the context of bus travel time prediction, several researchers have explored the application of ARIMA models. For instance, Li et al. [9] employed ARIMA and hybrid ARIMA models to forecast bus travel time in a congested urban network in China, with the hybrid ARIMA model demonstrating superior prediction accuracy. Similarly, Liu et al. [10] successfully utilized an ARIMA model to predict bus travel time in Singapore, highlighting its ability to capture data trends and seasonality for precise short-term predictions. In Beijing, China, Hu et al. [11] also leveraged an ARIMA model for bus travel time forecasting, achieving accurate predictions up to 30 min ahead to support real-time bus operations. Table 12 shows the experimental results for bus travel time prediction, and Figure 14 illustrates a comparison between seven deep learning models and the baseline model, ARIMA. Our analysis reveals that ARIMA performed significantly worse than the deep learning models, including LSTM, GRU, and HTF-NET. This gap in performance was reflected in ARIMA’s higher RMSE, MAE, and MSE values, suggesting that it struggles to model complex temporal patterns and nonlinear relationships within the data. The HTF-NET model, on the other hand, achieved superior results, outperforming ARIMA by 63.27% in terms of RMSE. This finding emphasizes the potential of deep learning approaches for accurate bus travel time prediction.

Table 12. Comparison of ARIMA with deep learning models on the overall test dataset.

Model No.	Model Name	RMSE	MAE	MSE
1	ARIMA	53.31	41.52	2842.31
2	Pure LSTM	33.94	23.17	1152.61
3	Pure GRU	34.88	24.01	1216.92
4	LSTM bidirectional	26.65	16.97	710.56
5	GRU bidirectional	28.06	17.37	787.64
6	Stacked-LSTM	23.74	15.42	563.73
7	Stacked-GRU	27.78	17.49	772.09
7	HTF-NET	19.62	13.26	428.64

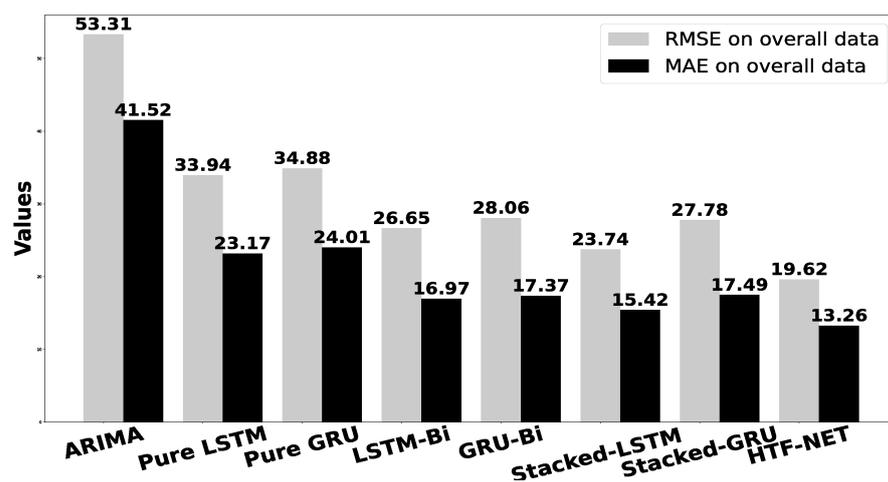


Figure 14. Comparison of ARIMA and deep learning models for RMSE and MAE: This figure shows the performance of the baseline ARIMA model and various deep learning models on the overall test dataset, comparing their RMSE and MAE values.

6. Conclusions and Future Work

In this study, we presented an approach to predicting bus travel time using digital tachograph (DTG) data. Our methodology has the potential to enhance scheduling accuracy and improve passenger's travel experience by providing real-world travel time information. The evaluation involved seven deep learning models tested on a sub-route from Namchang Village bus stop to Dongnam-gu Public Health Center. This route, covering a diverse landscape of universities, hospitals, shopping malls, and public areas over an 8.5 km road length, offered a representative scenario for assessing the models' performance under various traffic conditions. Five experiments were conducted, analyzing the models' performance across different scenarios, including overall test data, weekdays, weekends, with and without weather information, and different route types (long and short). Notably, our proposed Hybrid Temporal Forecasting Network (HTF-NET) model consistently exhibited exceptional performance, with the lowest root mean squared error (RMSE) and mean absolute error (MAE) values. This underscores its strong capacity to predict travel times accurately under diverse traffic patterns on both weekdays and weekends. Our study also highlighted the importance of weather data in travel time prediction. The exclusion of weather information led to a significant drop in prediction accuracy, emphasizing the necessity of integrating weather data into travel time prediction models. Specifically, the HTF-NET model outperformed the baseline ARIMA model by 63.27% in terms of the RMSE, indicating the practicality of this model for real-world applications. However, it is essential to note the limitations of our study. The models were trained on data collected under normal traffic conditions, excluding unexpected events such as accidents or work zone activities. This points to a need for future work to incorporate real-time event data, enhancing the model's robustness and applicability in addressing unforeseen travel disruptions.

In our future work, we plan to integrate additional data sources, such as road conditions and traffic camera feeds, to improve the accuracy of travel time predictions. By expanding our data sources, we aim to make the models more resilient to unexpected situations, such as accidents or roadwork, enhancing the reliability of our predictions. Additionally, we intend to conduct an in-depth analysis using 12 months of weather data to gain a comprehensive understanding of weather's impact on bus travel time. By refining the precision of bus travel time predictions, our methodology could play an important role in assisting transportation planners and policymakers in managing weather-related risks within the transportation system. Furthermore, our research can contribute significantly to smart city mobility applications, fostering more efficient and reliable transportation networks.

Author Contributions: Conceptualization, Y.H.; data curation, Y.H. and G.M.; funding acquisition, S.-J.C.; investigation, G.M.; methodology, G.M.; project administration, Y.H. and S.-J.C.; resources, Y.H. and S.-J.C.; software, G.M.; supervision, Y.H. and S.-J.C.; validation, Y.H.; visualization, G.M.; writing—original draft, G.M.; writing—review and editing, Y.H. and S.-J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. 2021R1A2C2012574); and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MIST) (No. 2022-0-01022, Development of Collection and Integrated Analysis Methods of Automotive Inter/Intra System Artifacts through Construction of Event-based experimental system).

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ITS	Intelligent Transportation System
TTP	Travel time prediction
DTG	Digital tachograph
RNN	Recurrent Neural Network
GPS	Global positioning system
OD	Origin–destination
BMS	Bus Management System
BIS	Bus Information System
STD	Spatio-temporal data
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
HTF-NET	Hybrid Temporal Forecasting Network
ARIMA	Autoregressive integrated moving average
RMSE	Root mean squared error
MSE	Mean squared error
MAE	Mean absolute error

Appendix A. Derivation of Performance Metrics

In this appendix, we provide detailed calculations for deriving the performance metrics RMSE, MAE, and MSE from the attention mechanism.

Appendix A.1. Derivation of RMSE from Attention Mechanism (Equation (11))

From Equation (8), we have:

$$e_i = \text{score}(h_t, h_i) \quad (\text{A1})$$

Let us proceed with the step-by-step derivation:

1. **Definition of Score Function:** The score function is defined as the measure of similarity between the current hidden state h_t and the hidden state h_i at time i .

$$e_i = \text{score}(h_t, h_i) \quad (\text{A2})$$

2. **Substitution of Score Function:** We substitute the score function into the calculation of attention weights a_i in Equation (9).

$$a_i = \frac{\exp(e_i)}{\sum_{j=1}^n \exp(e_j)} \quad (\text{A3})$$

3. **Context Vector Calculation:** Using the attention weights a_i , we compute the context vector C_t in Equation (10) as the weighted sum of the input sequence x_i .

$$C_t = \sum_{i=1}^n a_i \cdot x_i \quad (\text{A4})$$

4. **Combination with Hidden State:** Finally, we combine the context vector C_t with the current hidden state h_t for further processing.
5. **Definition of RMSE:** Equation (11) defines the root mean squared error (RMSE) metric for evaluating the model's performance in predicting travel time.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (t_i - \hat{t}_i)^2} \quad (\text{A5})$$

6. **Substitution of Predictions:** Substituting the predicted travel time \hat{t}_i with the context vector C_t obtained from the attention mechanism.
7. **Further Simplification :** Further simplification and manipulation may be performed to obtain the final form of Equation (11).

Appendix A.2. Derivation of MAE from Attention Mechanism (Equation (12))

From Equation (9), we have:

$$a_i = \frac{\exp(e_i)}{\sum_{j=1}^n \exp(e_j)} \quad (\text{A6})$$

Let us proceed with the step-by-step derivation:

1. **Definition of Attention Weights:** The attention weights a_i are calculated based on the scores e_i obtained from Equation (8).

$$a_i = \frac{\exp(e_i)}{\sum_{j=1}^n \exp(e_j)} \quad (\text{A7})$$

2. **Substitution of Attention Weights:** We substitute the attention weights a_i into the formula for the MAE (Equation (12)).

$$MAE = \frac{1}{n} \sum_{i=1}^n |t_i - \hat{t}_i| \quad (\text{A8})$$

where \hat{t}_i is represented as the weighted sum of the input sequence x_i based on the attention weights a_i .

3. **Substitution of Predictions:** Substituting \hat{t}_i with the weighted sum of input sequence x_i .
4. **Absolute Difference Calculation:** Absolute differences $|t_i - \hat{t}_i|$ are calculated for each trip i .
5. **Average Calculation:** Taking the average of these absolute differences over all n trips to obtain the mean absolute error (MAE).

Appendix A.3. Derivation of MSE from Attention Mechanism (Equation (13))

From Equation (10), we have:

$$C_t = \sum_{i=1}^n a_i \cdot x_i \quad (\text{A9})$$

Let us proceed with the step-by-step derivation:

1. **Definition of Context Vector:** The context vector C_t is computed as the weighted sum of the input sequence x_i based on the attention weights a_i .

$$C_t = \sum_{i=1}^n a_i \cdot x_i \quad (\text{A10})$$

2. **Substitution of Context Vector:** We substitute the context vector C_t into the formula for the MSE (Equation (13)).

$$MSE = \frac{1}{n} \sum_{i=1}^n (t_i - \hat{t}_i)^2 \quad (\text{A11})$$

where \hat{t}_i is represented as the weighted sum of the input sequence x_i based on the attention weights a_i .

3. **Substitution of Predictions:** Substituting \hat{t}_i with the context vector C_t .

4. **Squared Difference Calculation:** Squared differences $(t_i - \hat{t}_i)^2$ are calculated for each trip i .
5. **Average Calculation:** Taking the average of these squared differences over all n trips to obtain the mean squared error (MSE).

By following these steps, we have demonstrated the derivation of the performance metrics RMSE, MAE, and MSE from the attention mechanism.

References

1. Mahmood, A.; Siddiqui, S.A.; Sheng, Q.Z.; Zhang, W.E.; Suzuki, H.; Ni, W. Trust on Wheels: Towards Secure and Resource Efficient IoV Networks. *Computing* **2022**, *104*, 1337–1358. [CrossRef]
2. Hou, Z.; Li, X. Repeatability and Similarity of Freeway Traffic Flow and Long-Term Prediction under Big Data. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1786–1796. [CrossRef]
3. Kim, Y.; Back, J.; Kim, J. A Tamper-Resistant Algorithm Using Blockchain for the Digital Tachograph. *Electronics* **2021**, *10*, 581. [CrossRef]
4. Jeong, M.-H.; Lee, T.-Y.; Jeon, S.-B.; Youm, M. Highway Speed Prediction Using Gated Recurrent Unit Neural Networks. *Appl. Sci.* **2021**, *11*, 3059. [CrossRef]
5. Jeong, H.; Hong, J.; Park, D. A Framework of an Integrated Livestock Vehicle Trajectory Database Using Digital Tachograph Data. *Sustainability* **2021**, *13*, 2694. [CrossRef]
6. Ahn, S.-H.; Shin, Y.-E. Analysis of Taxi Passenger Travel Patterns Based on Busan DTG Data. *KSCE J. Civ. Environ. Eng. Res.* **2018**, *38*, 907–916. [CrossRef]
7. Jeon, S.-B.; Jeong, M.H.; Lee, T.-Y.; Lee, J.-H.; Cho, J.-M. Bus Travel Speed Prediction Using Long Short-term Memory Neural Network. *IEEE Access* **2020**, *32*, 4441. [CrossRef]
8. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 2nd ed.; OTexts: Melbourne, Australia, 2018. Available online: <https://otexts.com/fpp3/> (accessed on 27 March 2023).
9. Li, J.; Zhang, J.; Wang, Y.; Cheng, M.; Liu, X. Hybrid ARIMA Model for Bus Travel Time Prediction in Congested Urban Network. *Adv. Transp.* **2019**, *2019*, 1–11. Available online: <https://www.scribd.com/document/354923010/arima-models-for-bus-travel-time-prediction> (accessed on 5 April 2023).
10. Liu, J.; Sun, Y.; Liu, J. Prediction of Bus Travel Time Based on ARIMA Model in Singapore. *Int. J. Intell. Transp. Syst. Res.* **2021**, *19*, 106–117.
11. Hu, X.; Chen, X.; Li, Z.; Li, Y.; Li, H. Bus Travel Time Prediction Based on ARIMA Model. *J. Adv. Transp.* **2021**, *2021*, 1–14.
12. Qi, X.; Mei, T. A Deep Learning Approach for Long-Term Traffic Flow Prediction with Multifactor Fusion Using Spatiotemporal Graph Convolutional Network. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1105–1116. [CrossRef]
13. Rahmani, M.; Jenelius, L. Route Travel Time Estimation Using Low-Frequency Floating Car Data. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), Hague, The Netherlands, 6–9 October 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 2292–2297. Available online: https://www.researchgate.net/publication/259214549_Route_Travel_Time_Estimation_Using_Low-Frequency_Floating_Car_Data (accessed on 12 April 2023).
14. Rahmani, M.; Jenelius, L. Floating Car and Camera Data Fusion for Non-parametric Route Travel Time Estimation. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC 2014), Qingdao, China, 8–11 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1286–1291. Available online: <https://dokumen.tips/documents/floating-car-and-camera-data-fusion-for-non-route-route-fig-1-an-example-of.html?page=1> (accessed on 12 April 2023).
15. Wang, Y.; Zheng, L. Travel Time Estimation of a Path Using Sparse Trajectories. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 25–34. Available online: <https://dl.acm.org/doi/10.1145/2623330.2623656> (accessed on 18 April 2023).
16. Li, Y.; Fu, L. Multi-Task Representation Learning for Travel Time Estimation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19–23 August 2018; pp. 1695–1704. Available online: <https://dl.acm.org/doi/pdf/10.1145/3219819.3220033> (accessed on 27 April 2023).
17. Wang, H.; Tang, X.; Kuo, Y.-H.; Kifer, D.; Li, Z. A Simple Baseline for Travel Time Estimation Using Large-Scale Trip Data. *Acm Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [CrossRef]
18. Sun, X.; Zhang, H.; Tian, F.; Yang, L.; Huang, Y.; Chen, Y.; Wu, X. The Use of a Machine Learning Method to Predict the Real-Time Link Travel Time of Open-Pit Trucks. *Math. Probl. Eng.* **2018**, *2018*, 4368045. Available online: <https://www.hindawi.com/journals/mpe/2018/4368045/> (accessed on 29 April 2023). [CrossRef]
19. Wang, Z.; Fu, K.; Ye, J. Learning to Estimate the Travel Time. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19–23 August 2018; pp. 858–866. Available online: https://www.researchgate.net/publication/326503244_Learning_to_Estimate_the_Travel_Time (accessed on 29 April 2023).
20. Zheng, F.; Zuylen, H.V. Urban Link Travel Time Estimation Based on Sparse Probe Vehicle Data. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 145–157. [CrossRef]
21. Mendes-Moreira, J.; Jorge, A.M. Improving the Accuracy of Long-Term Travel Time Prediction Using Heterogeneous Ensembles. *Neurocomputing* **2015**, *150*, 428–439. [CrossRef]

22. Yu, B.; Wang, H.; Shan, W.; Yao, B. Prediction of Bus Travel Time Using Random Forests Based on Near Neighbors. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 333–350. [CrossRef]
23. Gupta, B.; Awasthi, S.; Gupta, R.; Ram, L.; Kumar, P.; Prasad, B.R.; Agarwal, S. Taxi Travel Time Prediction Using Ensemble-Based Random Forest and Gradient Boosting Model. In *Advances in Big Data and Cloud Computing*; Springer: Singapore, 2018; pp. 63–78. Available online: https://www.researchgate.net/publication/324270612_Taxi_Travel_Time_Prediction_Using_Ensemble-Based_Random_Forest_and_Gradient_Boosting_Model (accessed on 2 May 2023).
24. Cristóbal, T.; Padrón, G.; Quesada-Arencibia, A.; Alayón, F.; de Blasio, G.; García, C.R. Bus Travel Time Prediction Model Based on Profile Similarity. *Sensors* **2019**, *19*, 2869. [CrossRef] [PubMed]
25. Zhi-jian, W.; Da-biao, L.; Xia, C. Travel Time Prediction Based on LSTM Neural Network in Precipitation. *J. Transp. Syst. Eng. Inf. Technol.* **2020**, *20*, 137. Available online: <http://www.tseit.org.cn/EN/abstract/abstract19987.shtml> (accessed on 2 May 2023).
26. Chughtai, J.-U.-R.; Haq, I.U.; Shafiq, O.; Muneeb, M. Travel Time Prediction Using Hybridized Deep Feature Space and Machine Learning Based Heterogeneous Ensemble. *IEEE Access* **2022**, *10*, 98127–98139. [CrossRef]
27. Lee, G.; Choo, S.; Choi, S.; Lee, H. Does the Inclusion of Spatio-Temporal Features Improve Bus Travel Time Predictions? A Deep Learning-Based Modelling Approach. *Sustainability* **2022**, *14*, 7431. [CrossRef]
28. Lee, C.; Yoon, Y. A Novel Bus Arrival Time Prediction Method Based on Spatio-Temporal Flow Centrality Analysis and Deep Learning. *Electronics* **2022**, *11*, 1875. [CrossRef]
29. Shaji, H.; Vanajakshi, L.; Tangirala, A. Effects of Data Characteristics on Bus Travel Time Prediction: A Systematic Study. *Sustainability* **2023**, *15*, 4731. [CrossRef]
30. Zhao, L.; Chien, S.I.-J. Analysis of Weather Impact on Travel Speed and Travel Time Reliability. In Proceedings of the 2012 International Conference on Transportation and Development, Reston, VA, USA, 18–21 March 2012. Available online: <https://researchwith.njit.edu/en/publications/analysis-of-weather-impact-on-travel-speed-and-travel-time-reliab> (accessed on 10 March 2023).
31. Tsapakis, I.; Cheng, T.; Bolbol, A. Impact of Weather Conditions on Macroscopic Urban Travel Times. *IEEE J. Transp. Geogr.* **2013**, *28*, 204–211. [CrossRef]
32. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
33. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to Forget: Continual Prediction with LSTM. In Proceedings of the 1999 Ninth International Conference on Artificial Neural Networks ICANN 99, Edinburgh, UK, 7–10 September 1999; Volume 2, pp. 850–855. [CrossRef]
34. Gers, F.; Schraudolph, N.; Schmidhuber, J. Learning Precise Timing with LSTM Recurrent Networks. *J. Mach. Learn. Res.* **2002**, *3*, 115–143. Available online: <https://www.jmlr.org/papers/volume3/gers02a/gers02a.pdf> (accessed on 18 March 2023).
35. Han, Y.; Wang, C.; Ren, Y.; Wang, S.; Zheng, H.; Chen, G. Short-term prediction of bus passenger flow based on a hybrid optimized LSTM network. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 366. [CrossRef]
36. Liu, Y.; Zhang, H.; Jia, J.; Shi, B.; Wang, W. Understanding urban bus travel time: Statistical analysis and a deep learning prediction. *Int. J. Mod. Phys. B* **2023**, *37*, 2150023. [CrossRef]
37. Du, W.; Sun, B.; Kuai, J.; Xie, J.; Yu, J.; Sun, T. Highway travel time prediction of segments based on ANPR data considering traffic diversion. *J. Adv. Transp.* **2021**, *2021*, 6656429. [CrossRef]
38. Qiao, K.; Chen, J.; Wang, L.; Zhang, C.; Zeng, L.; Tong, L.; Yan, B. Category decoding of visual stimuli from human brain activity using a bidirectional recurrent neural network to simulate bidirectional information flows in human visual cortices. *Front. Neurosci.* **2019**, *13*, 692. [CrossRef] [PubMed]
39. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
40. Choi, E.; Schuetz, A.; Stewart, W.F.; Sun, J. Using recurrent neural network models for early detection of heart failure onset. *J. Am. Med. Inform. Assoc.* **2017**, *24*, 361–370. [CrossRef]
41. Drucker, H.; Burges, C.J.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. In Proceedings of the Advances in Neural Information Processing Systems 9 (NIPS 1996), Denver, CO, USA, 2–5 December 1996; pp. 1–7.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.