

修士論文

運行情報と気象情報の畳み込みによる バス到着時刻予測手法の提案と評価

石長 篤人

奈良先端科学技術大学院大学

先端科学技術研究科

情報理工学プログラム

主指導教員: 藤川 和利 教授

情報基盤システム学研究室（情報科学領域）

2021 年 1 月 25 日

本論文は奈良先端科学技術大学院大学先端科学技術研究科に
修士(工学) 授与の要件として提出した修士論文である。

石長 篤人

審査委員：

藤川 和利 教授	(主指導教員)
安本 慶一 教授	(副指導教員)
新井 イスマイル 准教授	(副指導教員)

運行情報と気象情報の畳み込みによる バス到着時刻予測手法の提案と評価*

石長 篤人

内容梗概

公共交通機関における利便性は利用者の利用意欲に直接的な影響を与えている。利便性を向上させる方法の1つとして、利用者への到着時刻予測の提供が挙げられる。特に路線バスにおいては、天候、乗降者数、交通状況など様々な要素が複雑に関係しており、正確な予測を行うことは難しい。路線バスにおける到着時刻予測の既存研究の中で、現状、Convolutional LSTMをベースにバス路線の過去の運行データを特徴量として用いた手法が最も有効である。しかしながら、この手法には大きく2つの問題点がある。まず時刻表が存在せず、一定間隔の頻度で走行する路線を対象に評価を行っているため早着によるバス停での意図的な停車など、時刻表ベース特有の特徴が考慮されていない点、次に雨などの天候の影響によって運行が乱れた際に予測精度が大きく低下する点である。本研究ではこれらの問題点を解決するために気象データと過去の運行データをConvolutional LSTMを用いて同時に畳み込み、走行時間と停車時間を別々のモデルで予測する手法を提案する。兵庫県神戸市内のバス路線の9ヶ月間の実運行データを利用し、評価実験を行った。データセットの最後の1週間を対象にConvolutional LSTMを用いた既存研究との予測精度の比較を行った結果、1週間全体で見ると、1.36%の平均絶対パーセント誤差の減少が見られた。特に夕方の時間帯や、雨天時に運行が大きく乱れた場合には、約3%と1週間全体と比べて平均絶対パーセント誤差を大きく減少できた。

*奈良先端科学技術大学院大学 先端科学技術研究科 修士論文, 2021年1月25日.

キーワード

IoT, 高度道路交通システム, 機械学習, 公共交通

Proposal and Evaluation of Bus Arrival Time Prediction Method by Convolution of Operation and Weather Information*

Atsuto Ishinaga

Abstract

The convenience of public transportation has a direct impact on the willingness of users to use it. One of the ways to improve convenience is to provide users with arrival time predictions. In particular, it is difficult to make accurate predictions for route-bus due to the complex relationship between various factors such as weather, ridership, and traffic conditions. Among the existing studies on route-bus arrival time prediction, the most effective method is currently the one based on convolutional LSTM and using the past route-bus operation data as features. However, there are two major problems with this method. The first is that it does not take into account the characteristics unique to timetable-based methods, such as intentional stops at bus stops due to early arrivals, because it evaluates routes that run at regular intervals without timetables, and the second is that the prediction accuracy drops significantly when the operation is disrupted by weather conditions such as rain. In this research, we propose a method to solve these problems by simultaneously convolving weather data and past operation data using Convolutional LSTM to predict travel time and stop time with separate models. We conducted an evaluation experiment using 9 months of actual data from a route-bus in Kobe City, Hyogo Prefecture. We compared the prediction

*Master's Thesis, Graduate School of Science and Technology, Nara Institute of Science and Technology, January 25, 2021.

accuracy with that of existing studies using Convolutional LSTM for the last week of the data set, and found an decrease in the mean absolute percent error (MAPE) of 1.36% for the entire week. Especially in the evening hours and when the operation was severely disrupted by rain, the average absolute percentage error of about 3% decreased significantly compared to the entire week.

Keywords:

IoT, Intelligent Transport Systems, Machine learning, Public Transportation

目次

1. はじめに	1
2. 関連研究	5
2.1 路線バスにおける到着時刻予測について	5
2.2 回帰分析ベースの予測手法	8
2.3 LSTM を用いた予測	9
2.4 Convolutional LSTM を用いた予測	10
2.5 Bidirectional LSTM を用いた時系列予測	11
2.6 関連研究のまとめ	12
3. 運行情報と気象情報の畳み込みによるバス到着時刻予測手法	15
3.1 本研究で用いる基礎技術について	15
3.1.1 Long Short Term Memory (LSTM)	15
3.1.2 Bidirectional LSTM	18
3.2 提案手法のアプローチ	18
3.3 提案手法の概要	20
3.4 データ形式	23
3.4.1 過去運行データ	23
3.4.2 気象データ	25
3.5 特徴量	26
3.6 前処理	28
3.6.1 データ分割	28
3.6.2 LinkDuration, DiffTime に対する処理	28
3.6.3 Weather, Temp, Precipitation に対する処理	30
3.7 バス到着時刻予測ネットワークアーキテクチャ	30
4. 評価実験	35
4.1 評価対象路線について	35
4.2 データセットについて	37

4.2.1	過去運行データ	37
4.2.2	気象データ	39
4.3	評価方法	39
4.4	比較対象	41
4.5	パラメータチューニング	43
4.5.1	比較手法のパラメータチューニング	43
4.5.2	提案手法のハイパーパラメータチューニング	46
4.6	評価結果	47
4.6.1	1 週間の評価結果	47
4.6.2	ピークの時間帯の予測精度の比較	49
4.6.3	雨によって運行が乱れた日の評価	51
4.6.4	天候は安定しているが、運行が一部乱れた日における評価	53
4.6.5	天候と運行が安定した日における評価	54
5.	考察	57
5.1	予測対象区間の長さに基づく比較	57
5.2	提案手法の比較	60
5.3	既存手法との比較	61
5.3.1	午後のピーク、9/25(金) に対する考察	61
5.3.2	9/26(土)、9/29(火) に対する考察	63
5.4	今後の展望	63
6.	おわりに	65
	謝辞	66
	参考文献	67

目 次

1	GTFS Realtime によるリアルタイム情報の提示例	6
2	到着時刻予測の誤差に対する意識調査	7
3	バス停 A から B までの到着時刻の定義	8
4	timestep = t における RNN のダイアグラム	16
5	timestep = t における LSTM のダイアグラム	17
6	Bidirectional LSTM による双方向伝搬の概略図	19
7	提案手法の流れ	21
8	過去運行データにおけるパラメータ	22
9	走行時間と停車時間の各特徴量	28
10	学習データセットの分割処理 ($input_timesteps = 8, output_timesteps = 3$)	29
11	Bidirectional Convolutional LSTM 到着時刻予測モデル	32
12	提案手法: エンコーダにおける処理 (走行時間)	33
13	提案手法: デコーダにおける処理	34
14	評価対象路線の経路図	36
15	時間帯ごとの平均所要時間と標準偏差	37
16	過去運行データの作成に用いたデータベースの概略図	39
17	1 週間 (2020 年 9 月 24 日-9 月 30 日) の RMSE と MAE の精度比較 ($t + 1$)	49
18	2020/9/25 の所要時間と過去の平均所要時間	51
19	2020/9/25(金) の到着時刻予測の比較	53
20	2020/9/26(土) の比較	55
21	2020/9/29(火) の比較	56
22	経路の長さに基づく平均絶対誤差の比較 (2020 年 9 月 24 日-9 月 30 日)	58
23	経路の長さに基づく平均二乗誤差の比較 (2020 年 9 月 24 日-9 月 30 日)	59

表 目 次

1	過去運行データの形式	23
2	過去運行データのサンプル	24
3	気象データの形式	25
4	気象データのサンプル	25
5	評価対象路線における予定所要時間	37
6	取得した天候情報の内訳	40
7	PureLSTM: パラメータチューニング結果	44
8	ConvLSTM: パラメータチューニング結果	45
9	提案手法 走行時間: パラメータチューニング結果	46
10	提案手法 停車時間: パラメータチューニング結果	46
11	1 週間の到着時刻予測精度の比較 (2020 年 9 月 24 日-9 月 30 日) . .	48
12	午前ピーク 09:00-11:00(平日のみ)	50
13	午後ピーク 18:00(平日のみ)	50
14	2020/9/25(金) の精度比較	52
15	2020/9/26(土) の精度比較	54
16	2020/9/29(火) の精度比較	56
17	1 週間の到着時刻予測精度の比較 (神戸国際大学前-神戸ベイシェラ トンホテル)	60
18	1 週間の到着時刻予測精度の比較 (神戸ベイシェラトンホテル-新神 戸駅)	60
19	提案手法の平均絶対パーセント誤差の比較	61
20	既存手法との平均絶対パーセント誤差 (MAPE) の比較	62
21	日毎の天候状態、運行の乱れ	62

1. はじめに

現代の生活において公共交通機関は毎日の通勤や通学、観光など様々な用途で利用されており、都市部、地方に拘らず、重要な役割を果たしている。近年の研究で公共交通機関においては利便性が利用者の利用意欲に直接的な影響を与える [1] ことが示されており、利便性を向上させる方法の 1 つとして、利用者への到着時刻予測の提供が挙げられる。例えば複数の公共交通機関を連続して利用する場合に遅延や早着を予測することで、より待ち時間の少なく、到着の早いルート提示が可能となる。このようなリアルタイム情報の提供は既に様々な方法で行われている。例えば Google 社が提供している世界共通の公共交通機関向けのオープンフォーマットである General Transit Feed Specification Realtime (GTFS Realtime)[2] を利用して各公共交通機関はリアルタイム情報を提供できる。他にも各公共交通機関はバス停のサイネージや独自の Web アプリケーションなど様々な形で利用者に対して遅延情報などのリアルタイムな運行情報を提示している。Cats らはこのような公共交通機関における到着時刻予測などの運行情報が利用者の経路選択、乗車時間、降車場所、移動距離などに影響を与えることを明らかにした [3]。また、Gooze らは精度の低い到着時刻予測の提示はかえって利用者を混乱させる一因となり、一定の利用者の利用頻度が低下することを報告している [4]。以上のことから、正確な到着時刻の予測を提示することは非常に重要であると言える。

公共交通機関の中でも特に路線バスにおいては、天候、乗降者数、交通状況など様々な要素が複雑に関係しており、正確な予測は難しい。これまで様々な手法で路線バスの到着時刻を予測する手法が提案されている。Patnaik らは、過去のバス停間の走行時間、バス停での停車時間、乗降者数、ドアの開閉時間などの特徴量を用いた回帰分析による予測手法 [5] を提案した。以降、バス停間の走行時間、バス停での停車時間を含む情報を過去運行データと呼ぶ。Zeng らは回帰分析では扱うことのできない時系列の特徴量を扱うために LSTM を用いて、直前の便における全ての過去運行データを入力として次の便の到着時刻を予測する手法 [6] を提案した。次に Petersen らは LSTM では捉えきれない複数の時間帯に跨がる細かい運行パターンを学習するために Convolutional LSTM を用いた到着時刻

予測手法 [7] を提案している。過去の複数の時間帯の過去運行データを入力として、始点から終点までの所要時間を予測するモデルを構築した。過去運行データのみを特徴量として使用している。これまでに提案されてきた様々な既存研究の中で、現状、Petersen らの Convolutional LSTM をベースにした手法 [7] が最も有用である。それは Convolutional LSTM では重み演算に畳み込み演算を用いることで、従来の LSTM よりも複数の運行データに含まれるパターンを効果的に学習できるためである。しかしながら、この手法には大きく 2 つの問題点がある。まず時刻表が存在せず、一定間隔の頻度で走行する路線を対象に評価しているため早着によるバス停での意図的な停車など、時刻表ベース特有の特徴が考慮されていない点、次に雨などの天候の影響によって運行が乱れた際に予測精度が大きく低下する点である。Convolutional LSTM を用いた手法でこれらの問題点に対するアプローチは行われていない。本研究ではこれらの問題点を解決するために気象情報と過去運行データを Convolutional LSTM を用いて同時に畳み込み、走行時間と停車時間を別々のモデルで予測する手法を提案する。

気象情報を使用する理由は 2 点ある。まず、これまでの研究で気象情報が路線バスの運行に様々な影響を及ぼすことが示されている点である。Stover らは天候、降雨量、気温などが路線バスの乗降客数に影響を与えることを明らかにした [8]。また Hofmann らは、気象情報が乗降者数だけでなく、バスの移動時間にも直接影響を及ぼすことを示した [9]。これらの研究は気象情報が路線バスの走行時間と停車時間の両方に直接影響を与えることを示した。2 点目に気象データの取得の容易さとデータ取得に要するコスト面での優位性が挙げられる。その一方で乗降者数や、ドアの開閉時間を取得する場合には、各バスにそれぞれ専用のセンサーなどを取り付ける必要があり、取得に要するコストが非常に高い。しかし、気象情報は Application Programming Interface (API) などで比較的容易に取得できる。例としては OpenWeatherMap API[10] や Dark Sky API[11] などが挙げられる。以上のことから気象情報を特徴量として含めることで交通状況や乗降者数など路線バスの走行時間や停車時間に直接影響を与える特徴量を低コストで補完できると考えた。

また走行時間と停車時間を別々のモデルで予測を行うのは、特に時刻表の基

いた路線バスにおいて停車時間が走行時間よりも不安定であると考えられるためである。走行時間に関しては時刻表に対する早着や遅延といった情報が与える影響は少ないが、停車時間はバスがバス停に到着した時刻と時刻表との差が停車時間に大きな影響を与える。それだけでなく、乗降者数にも大きな影響を受ける。一方で走行時間は、地域の交通状況に大きく依存する。地域の交通状況は路線バスの早着と遅延、乗降者数と比べると安定しているため、走行時間と停車時間はそれぞれ独立していると考ええる。そこで本研究では走行時間と停車時間でそれぞれ別々の特徴量を用いる。

本研究では、さらなる精度の向上を実現するために、順方向と逆方向の2つの中間層の出力を用いる Bidirectional LSTM[12] に着目した。従来の LSTM は順方向の伝搬のみを行う。これまでに様々な時系列予測に対して Bidirectional LSTM を用いることで、従来の LSTM よりも精度が向上したという結果 [13, 14] が示されている。本研究もこれらと同じ時系列予測であるため、Bidirectional LSTM を用いることでさらに精度が向上する可能性があると考えた。

以上のアプローチによる既存手法の2つの問題点の解決と到着時刻予測の精度向上を本研究の目的とする。兵庫県神戸市内のバス路線の9ヶ月間の実運行データを利用し、評価実験を行った。データセットの最後の1週間を対象に Convolutional LSTM を用いた Petersen らの手法 [7] と予測精度の比較を行った。その結果、Bidirectional LSTM を用いた提案手法の精度が最も優れていた。1週間全体で見ると、1.36% の平均絶対パーセント誤差の減少が見られた。特に夕方の時間帯や、雨天時に運行が大きく乱れた場合には、約 3% と平均絶対パーセント誤差が1週間全体と比べて大きく減少することが確認できた。

本研究による貢献を以下にまとめる。

1. Convolutional LSTM を用いた従来手法では考慮されていなかった気象情報を考慮したモデルを提案し、路線バスの到着時刻予測における気象情報の有効性を示す。
2. 別々のモデルでの走行時間と停車時間の予測結果を結合して最終的な出力とするモデルを提案する。また時刻表ベースの路線においては走行時間と

停車時間を1つのモデルで予測するよりも別々で予測する方が高い精度で予測できることを示す。

3. 未来への伝搬と過去への逆伝搬の双方向で伝搬するネットワークである Bidirectional LSTM が過去の運行データと気象情報を特徴量とした到着時刻予測手法に対して従来の LSTM よりも有用であることを示す。

本稿の構成は以下の通りである。第2章では、研究背景と関連研究についてまとめ、既存研究の問題点を述べる。第3章では、提案手法の基礎となる技術を紹介したあと、既存手法の問題点を解決する手法を提案する。第4章では、提案手法の評価実験の詳細と結果について述べる。第5章では、評価実験の結果を元に考察を行い、今後の課題について述べる。第6章では本稿のまとめを行う。

2. 関連研究

本章では、路線バスにおいて正確な到着時刻の予測が必要な背景とこれまでに提案されてきたバス到着時刻予測手法について述べる。また、提案手法で用いる Bidirectional LSTM を用いた研究についても紹介する。そして最後に関連研究についてまとめる。

2.1 路線バスにおける到着時刻予測について

Mugion らは路線バスなどの公共交通機関において利便性が利用者の利用意欲に直接的な影響を与えることを示した [1]。利用者における公共交通機関の利便性を向上させる方法の 1 つとして、利用者への到着時刻予測の提供が挙げられる。例えば複数の公共交通機関を連続して利用する場合に遅延や早着を予測することで、より待ち時間が少なく、到着の早いルート提示が可能となる。このようなリアルタイム情報の提供は既に行われており、例えば Google 社が提供している世界共通の公共交通機関向けのオープンフォーマットである GTFS Realtime[2] を利用して各公共交通機関はリアルタイム情報を提供できる。GTFS Realtime を利用すると車両の現在地、遅延情報や混雑度などが提供できる。図 1 に GTFS Realtime による路線バスのリアルタイムな遅延情報の提示例を示す。GTFS Realtime 以外にも、各公共交通機関はバス停のサイネージや独自のアプリケーションなど様々な形で利用者に対してリアルタイム情報を提示している。

Cats らはこのような公共交通機関における到着時刻予測などのリアルタイム情報が利用者の経路選択、乗車時間、降車場所、移動距離などに影響を与えることを明らかにした [3]。また、Gooze らは路線バスにおける精度の低い到着時刻予測の提示はかえって利用者を混乱させる一因となり、一定数の利用者の利用頻度が低下することを報告している [4]。Gooze らはオンラインのアンケートにて調査を行った。まず調査対象の 5074 人の 77% にあたる 3866 人から過去 6 ヶ月の間に誤った到着時刻予測の提示を経験したという回答結果を得た。また、どの程度の誤差を正確でないと判断するかについても調査を行っている。1 分未満から 10 分以上までの 6 つの範囲で回答を求めた結果を図 2 に示す。回答者の割合が最も

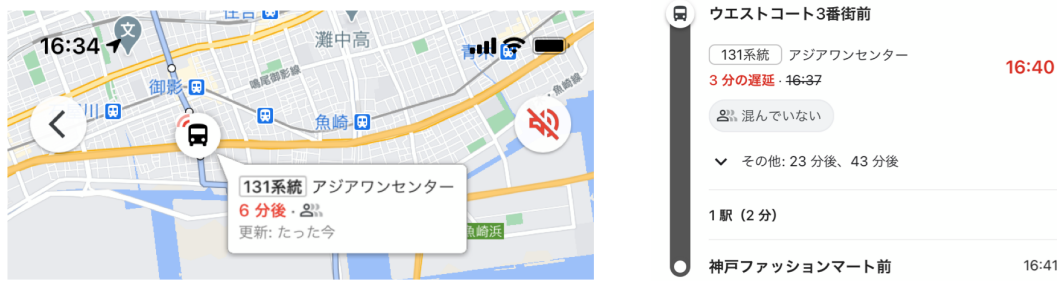


図 1: GTFS Realtime によるリアルタイム情報の提示例

多かったのは4～5分で全体の37%を占めている。次点が2～3分で全体の27%となっている。最後に誤った到着時刻予測が利用頻度に影響したかどうかを調べた。その結果、提示された正確でない予測情報が原因で以前よりも利用頻度が大幅に下がった、もしくは減少したという回答が全体の約9%であった。このように到着時刻の予測が普及している反面、誤った予測が原因で利用者が減少する可能性を示した。また、利用者が求める予測精度も2分～5分が半数以上を占めており、要求精度が高いことが分かる。以上のことから路線バスにおいて利用者の利便性を高めるには、少しでも精度の高い到着時刻予測手法が必要である。

しかしながら、路線バスにおける到着時刻には様々な要因が複雑に関与しているため、正確な予測は非常に難しい。到着時刻に影響を与える要因としては、天候、交通状況、事故、交差点、乗降者数などが挙げられる。これらの要因の変化が起こりやすい都市部や午前、午後のピーク時などは特に予測が難しいことがMugionらの研究[1]で明らかになっている。このような問題にアプローチするためにこれまで様々な路線バスにおける到着時刻の予測に関する研究が、バスから得られる位置情報や乗降者数情報 (APC: Automatic Passenger Counting) などのデータの二次利用の可能性を探ると共に行われてきた。

本研究における到着時刻の定義を図3に示す。このようにバス停Aからバス停Bまでは複数の走行時間と停車時間で成り立つ。また、式2.1に示すようにバス停Aからバス停Bまでの走行時間と各バス停での停車時間を示す *LinkDuration*

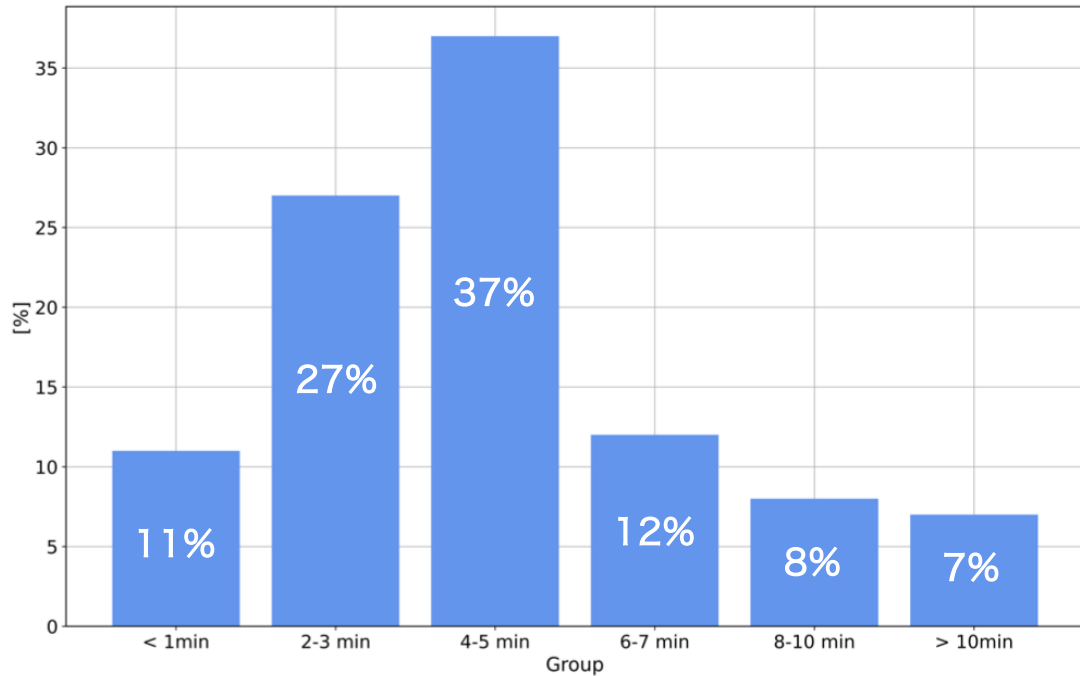


図 2: 到着時刻予測の誤差に対する意識調査 [4]

を足し合わせたものとして到着時刻を定義する。以降、走行時間及び、停車時間をリンク、それぞれのリンクでの所要時間をリンク所要時間と表現する。 ln はリンク、 u は最後のリンクを示している。基本的に到着時刻予測に用いられる特徴量としては過去のバス停間の走行時間や停車時間などの過去運行データに限定されるが、いくつかの限られた手法では気象情報、乗降者数などのデータも副次的に利用されている。本章では以降、到着時刻予測に用いるおおまかな手法毎に節を分けてそれぞれ述べる。

$$Y = \sum_{ln=0}^u LinkDuration_{ln} \quad (2.1)$$

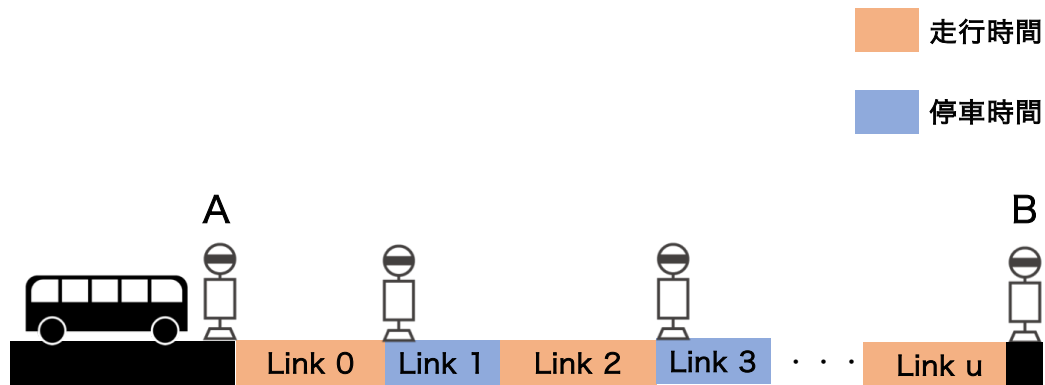


図 3: バス停 A から B までの到着時刻の定義

2.2 回帰分析ベースの予測手法

路線バスにおける走行時間や停車時間を予測する手法として最初に取りられたアプローチは線形回帰をベースにした予測手法である。回帰分析を用いた予測手法の出力は1つのリンクにおける所要時間である。この予測をバスがバス停に到着するたびに再帰的に行うことによって複数のリンク所要時間を予測する。Patnaikらは、予測を行うリンクの過去のリンク所要時間、乗降者数、ドアの開閉時間などの特徴量を用いた回帰分析による予測手法 [5] を提案した。このような特徴量を用いることで、移動距離、路線の需要特性、時間帯などに基づく予測が可能であることを示した。

Chen らは過去運行データを用いた回帰分析では予測できない予測直前のリアルタイムに近い遅延などの情報を含めるためにカルマンフィルタを用いた予測手法 [15] を提案している。過去の運行情報に基づいた回帰分析による予測をカルマンフィルタの初期入力とし、その結果を予測直前の走行時間によって補正することを目指した。このカルマンフィルタを用いた予測手法では回帰分析による静的な予測結果を直前のリアルタイム情報を用いて補正することで精度が上がるこ

を示した。

Gong らも同様に到着時刻が過去の運行情報だけでなく直前の運行情報に依存しているという問題を解決することを目指した。リアルタイムにバスから取得される位置情報を用いて取得した運行情報を用いて回帰分析による予測結果を補正する手法[16]を提案した。直前のバス停までの到着に要した時間とそのバス停での停車時間のみを用いたモデルと予測対象となる時間帯、曜日における各バス停間の平均所要時間と各バス停での停車時間の過去データに基づいた統計モデルをそれぞれ構築した。そして、この2つの回帰ベースモデルを組み合わせたハイブリッドモデルを提案手法として評価した。その結果、それぞれのモデル単体で予測した精度よりも、提案手法であるハイブリッドモデルが最も優れた精度で予測できることを示した。

これらの回帰分析をベースにした到着時刻予測手法は渋滞などが少なく、環境的な要因が安定している地域ではある程度高い精度での到着時刻予測が可能である。しかしながら、これらの手法は過去の運行情報に基づく静的な予測手法であるため、到着時刻の変化に対する予測値の収束がゆるやかである。そのため、雨や交通事故などの短期間で影響の大きい環境要因がある場合に特に予測精度が低下する。また、カルマンフィルタを用いて回帰分析の結果を補正する手法に関してもバス停間の走行時間とバス停での停車時間など複雑な非線形の特徴を捉え、予測に適用することが難しい。例えば、カルマンフィルタの状態は直前の1つの走行時間のみしか考慮することができないため、複数前の走行時間はもちろん複数の時間帯に跨がる運行情報が持つパターンの変化を捉えて回帰分析による予測結果を補正することはできない。さらに、これらの手法は始点から終点までバスが進む度に予測を行うため、始点の出発前に終点までの到着時刻を予測できない。そのため、出力できる予測結果は限られた区間のものとなる。

2.3 LSTMを用いた予測

回帰ベースの予測手法では含めることができない時系列の特徴を扱うために、時系列のデータを学習できる Recurrent Neural Network(RNN) が注目されてきた。RNN の中でも長期的な時系列データを学習することができる Long short-term

memory (LSTM) を用いた路線バスにおける到着時刻の予測手法が有効であるとされており、これまでに LSTM を用いた手法がいくつか提案されている。ここでの複数の時系列データというのは始点から終点までの各リンクのリンク所要時間を指す。2.2 節で紹介した手法では 1 つ前のリンク所要時間しか予測に用いることができなかった。その問題を解決するために Zeng らは LSTM を用いて予測直前の便の全てのリンク所要時間を元に次の便の到着時刻を予測する手法 [6] を提案した。特徴量としては各リンクの所要時間のみを用いている。この手法では、データセットをピーク時とオフピーク時の 2 つに分けて別々のモデルを構築し、予測している。カルマンフィルタベースの手法と比較した結果、ピーク時、オフピーク時に拘らず LSTM を用いた提案手法がより優れた予測を行えることを示した。

Pang らも同様の問題を解決するために LSTM をベースに過去の運行情報と交差点数などの静的な経路の情報の 2 種類のデータを用いて到着時刻を予測する手法 [17] を提案している。線形回帰や k 近傍法、Support Vector Machine (SVM) など他の機械学習をベースとした 6 つの手法と比較した結果、LSTM を用いた提案手法が最も高い精度で予測できることを示した。

以上のことから、長期間の時系列データを学習することができる LSTM は様々な機械学習アルゴリズムの中で最も路線バスの到着時刻予測に適していると言える。

2.4 Convolutional LSTM を用いた予測

2.3 節で紹介した LSTM を用いた手法の拡張として Convolutional LSTM を用いた手法も提案されている。Convolutional LSTM は LSTM の重み演算に全結合の代わりに畳み込み演算を用いることによって、空間的な時系列の特徴量を捉えることができる。そのため時系列予測において一般的な全結合を用いる LSTM よりも高い性能を発揮する可能性がある。Petersen らは Convolutional LSTM を用いて複数の便の始点から終点までの到着時刻を予測する手法 [7] を提案した。過去の複数の時間帯の始点から終点までの路線の所要時間を入力として、未来の複数の時間帯の始点から終点までの所要時間を予測するモデルを提案している。特

微量としては過去の始点から終点までのリンク所要時間を用いた。デンマーク都市部の一路線を対象に6ヶ月間の実運行データを用いて評価実験を行った。最後の1週間を対象に予測精度を評価した結果、従来のLSTMを用いた手法よりも高い精度で予測ができることを示した。LSTMを用いた手法では1週間の平均絶対誤差が3.48分であるのに対し、Convolutional LSTMを用いたモデルでは2.66分と約1分の誤差を減少させることを示した。平均絶対パーセント誤差は0.83%減少した。また、特に予測が難しいとされているピーク時のみを対象に評価を行った。その結果、1週間での評価よりも、LSTMとConvolutional LSTMの誤差の減少幅は大きく、午前(7-9)、午後(14-18)のピーク時の評価では平均二乗誤差、平均絶対誤差が共に1分以上減少した。この結果から、Convolutional LSTMを用いた手法が従来のLSTMを用いた手法よりも有効であると共に、予測の難しいピークの時間帯で特に有効であることを示した。

2.5 Bidirectional LSTM を用いた時系列予測

2.3節、2.4節で紹介したように、路線バスにおける到着時刻予測には、時系列の特徴を学習できるLSTMを用いた手法が有効である。それ以外にも、LSTMは幅広い分野の時系列予測に用いられている。Alahiらは、LSTMを用いて歩行者の軌道を予測する手法[18]を提案している。同じ空間内の複数の歩行者の相関関係のパターンを学習している。従来の線形回帰やRNNを用いた手法よりも優れた予測精度であることを示した。QingらはLSTMを用いて1時間毎の日射量を予測する手法[19]を提案している。RNNをベースにした手法と比較して、平均二乗誤差が42.9%と大幅に減少したことを報告している。

近年、LSTMの中でもBidirectional LSTMを用いた時系列予測が注目されている。従来のLSTMは、ある状態の中間層の出力値を次の状態に順伝播するネットワークである。これに対して、Schusterらは中間層の出力を未来への順伝播と過去への逆伝播の両方向で伝播するBidirectional LSTM[12]を提案した。このBidirectional LSTMはタスクによっては従来のLSTM、Convolutional LSTMよりも優れた性能を発揮することが報告されている。AlthelayaらはBidirectional LSTMを用いて株価を予測する手法[13]を提案した。1日先の終値、30日先の終

値の2種類を予測対象として、それぞれモデルを構築した。過去10日間の株価の推移を特徴量として、それぞれ学習と予測を行った。予測結果の精度を比較したところ、従来のLSTMを用いた手法よりもBidirectional LSTMを用いた手法が双方の予測対象で予測精度が高かった。特に、30日先の終値の予測値における精度の上昇幅が大きかった。Bidirectional LSTMを用いることで時系列の株価変動を従来の順伝搬のみのLSTMよりも効果的に学習できたためである。またJeyakumarらは、スマートフォンから取得できる加速度センサやジャイロセンサなど7種類のセンサデータを特徴量にBidirectional LSTMを用いてユーザーの移動手段を予測する手法[14]を提案している。自動車、バス、徒歩、静止などを含む8つの状態の分類問題として学習、予測を行った。LSTM、Convolutional LSTMなど複数のLSTMベースの手法で分類精度を比較した結果、Bidirectional LSTMが最も高い精度であった。特に予測に用いる時系列データを長くした時に精度の上昇幅が大きくなった。この結果は、Bidirectional LSTMが長期間の時系列データの学習において特に有効であることを示している。このように、Bidirectional LSTMが従来のLSTMよりも高い性能であったことを示す結果がいくつか報告されている。

2.6 関連研究のまとめ

上記のように、様々な手法で到着時刻を予測する手法が提案されている。また、双方向の伝搬を用いたBidirectional LSTMを用いた研究についても紹介した。現段階で路線バスの到着時刻予測に対して最も有効である手法はConvolutional LSTMを用いたPetersenらの手法[7]であると考えられる。しかし、この手法には以下のような問題点が存在する。

1. 時刻表のない路線(一定間隔の頻度で走行する路線)を対象に評価を行っているため早着によるバス停での意図的な停車など、時刻表ベース特有の特徴が考慮されていない
2. 運行データのみを特徴量としているため天候の影響によって運行が乱れた際に予測精度が低下する

まず、1つ目の問題点について述べる。時刻表をベースとした路線バスで既存手法のように走行時間と停車時間を1つのLSTMで学習すると、走行時間と停車時間がそれぞれ持つ時系列の特徴が損なわれてしまうと考えられる。それは、一定間隔の頻度で走行するバス路線と時刻表に基づいて走行するバス路線では、交通状況や乗降者数、天候状態などが走行時間、停車時間にそれぞれ与える影響が大きく異なるためである。走行時間に関しては時刻表に対する早着や遅延といった情報が与える影響は少ないが、停車時間に関してはバス停に到着した時刻と時刻表との差が停車時間に大きな影響を与える。例えば、バス停に到着した時刻が時刻表の出発予定時刻よりも3分余裕があれば、そのバスは乗降者数に拘らず3分間停車する。また、時刻表との差だけでなく、乗降者数の影響も受けるため非常に複雑である。一方で時刻表のない路線ではこのような状況は起こらず、停車時間は乗降者数のみに依存する。それに対して走行時間は、地域の交通状況に大きく依存する。以上のことから、時刻表の基づいた路線バスにおいて停車時間は走行時間よりも不安定であると考えられる。走行時間に大きな影響を与える地域の交通状況は路線バスの早着と遅延、乗降者数と比べると安定しているため、走行時間と停車時間はそれぞれ独立していると考えられる。そのため時刻表をベースにした路線において、走行時間と停車時間を同一のLSTMで学習すると、予測精度が低下すると考えられる。

次に2つ目の問題点について述べる。これまでの研究の中で気象情報を特徴量として利用した手法は回帰分析ベースのものしか存在せず、Convolutional LSTMをベースにした手法で気象情報を特徴量として利用した手法は提案されていない。しかし、これまでの様々な研究で気象情報は路線バスの運行に影響を及ぼすことが示されている。Stoverらはアメリカ合衆国ワシントン州のピュージェット湾周辺地域の路線バスを対象に天候、降雨量、降雪量、気温が路線バスの乗降客数に影響を与えることを明らかにした[8]。またHofmannらは、天候状態が乗降者数だけでなく、バスの移動時間にも直接影響を及ぼすことを示した[9]。特に、雨天時には平均よりも移動時間が長くなる傾向にあると結論づけた。これは、天候状態が地域の交通状況に影響を及ぼすためである。これらの研究は天候状態が路線バスの走行時間と停車時間の両方に直接影響を与えることを示した。そのため、

天候が不安定であれば、路線バスの運行も不安定になると考えられる。既存手法では、気象情報を特徴量として考慮していないため、天候の影響によって運行が乱れた際に、予測精度が大きく低下すると考えられる。

第3章では、これらの問題点を解決するために、運行情報と気象情報を特徴量として走行時間と停車時間を別々に予測する Convolutional LSTM ベースの到着時刻予測手法を提案する。

3. 運行情報と気象情報の畳み込みによるバス到着時刻予測手法

本章では、路線バスにおける過去の運行データと気象情報を特徴量として用いた Convolutional LSTM ベースの到着時刻予測手法を提案する。以降の節ではまず、提案手法で用いる基礎技術について述べる。その後に提案手法の概要、提案手法で用いるデータセットと特徴量の定義、及び必要な前処理について述べる。そして最後に構築したモデルのネットワークアーキテクチャに述べる。

3.1 本研究で用いる基礎技術について

本節では、まず、LSTM の元となった RNN について紹介した後、LSTM について概説する。そして最後に提案手法で用いる Convolutional LSTM、Bidirectional LSTM について述べる。

3.1.1 Long Short Term Memory (LSTM)

LSTM は、時間依存関係を学習できる RNN の一種である。 $timestep = t$ における RNN のダイアグラムを図 4 に示す。従来のニューラルネットワークでは、時系列データを正しく学習できなかった。RNN では、前の中間層の出力を再帰的に利用することでその問題を克服した。時系列データとは時間的な変化を持つデータを指す。RNN では入力値 x_t に対し、その 1 つ前の隠れ層 h_{t-1} も考慮しながら出力を行う。出力 h_t は式 3.1 のように算出される。 x_t は入力ベクトル、 h_t は隠れ層ベクトル、 b_h はバイアスベクトル、 U, V はそれぞれパラメータ行列、 σ_h は活性化関数 $\tanh$ をそれぞれ表している。活性化関数には sigmoid や $\tanh$ などが一般的に用いられる。

$$h_t = \sigma_h(U_h x_t + V_h h_{t-1} + b_h) \quad (3.1)$$

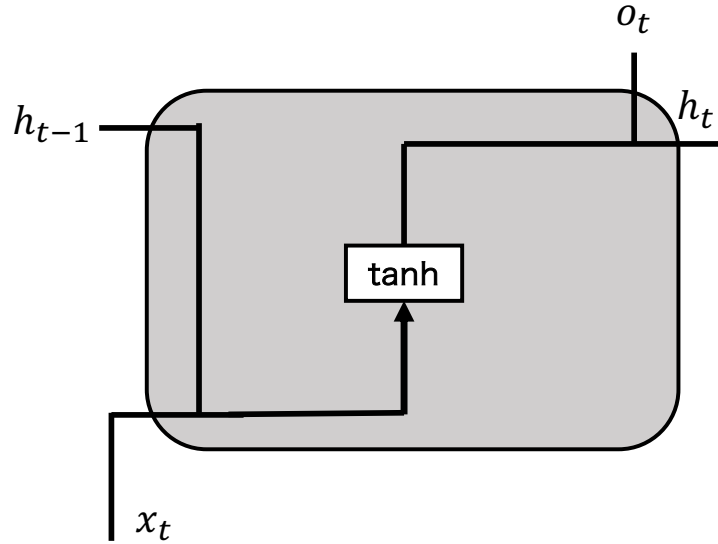


図 4: timestep = t における RNN のダイアグラム

この RNN では短期間の時系列データを扱えるが、長期的な依存の学習は難しいという欠点がある。それを補ったものが LSTM (図 5) である。過去の情報をどれくらい考慮するかを判定する忘却ゲート f_t を追加することで長期的な時系列データを扱うことを可能とした。忘却ゲートでは過去の状態 C_{t-1} をどれくらい残すかを判定するためのゲートである。この判定は式 3.2 のように h_{t-1} と x_t を入力とするシグモイド層によって行われる。 f_t は 0 から 1 の間の数値を出力し、1 は完全維持、0 は完全破棄をそれぞれ示す。次に、セル状態で保存する新たな情報の判定を行う。つまり、過去のセル状態 C_{t-1} から新たなセル状態 C_t への更新を行う。まず、入力ゲート i_t でどの値を更新するかを判定し、セル状態に含む新たなベクトル $\tilde{C}_t$ を生成する。この 2 つと忘却ゲートの値を組み合わせる式 3.5 のようにセル状態を更新する。 O_t は出力ゲートを示しており、セル状態のどの部分を出力するかをシグモイド層を用いて制御する役割を持つ。最後にセル状態に出力ゲートの出力 o_t をかけて、最終的な予測 h_t を算出する。LSTM で行う全ての処理を式 3.2~3.7 に示す。 σ_g は sigmoid、 σ_c は tanh を表している。

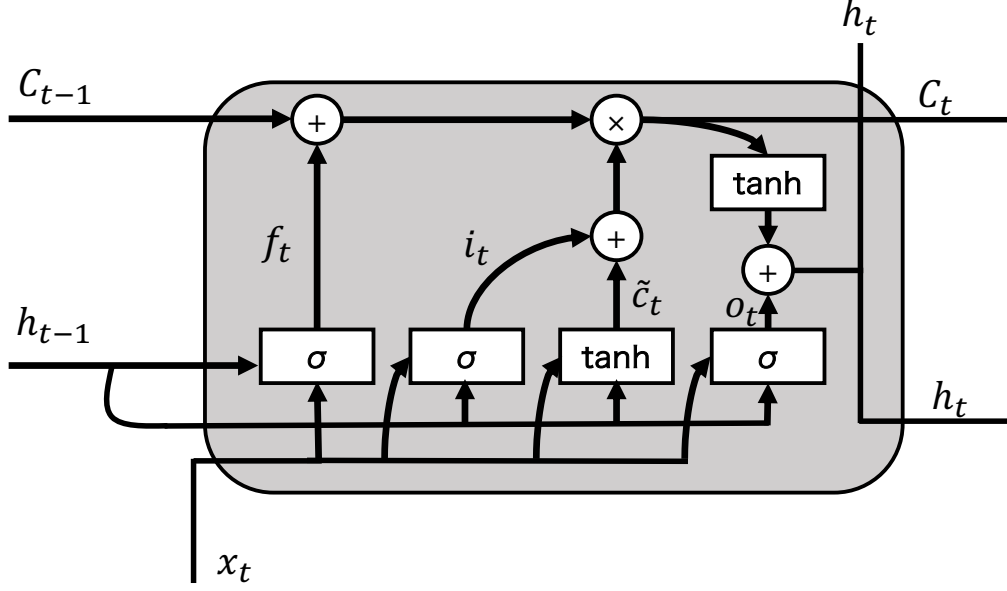


図 5: timestep = t における LSTM のダイアグラム

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (3.2)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (3.3)$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \quad (3.4)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \quad (3.5)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \quad (3.6)$$

$$h_t = o_t \circ \sigma_c(c_t) \quad (3.7)$$

Shi らは畳み込み演算を LSTM に組み込んで単一構造にすることで Convolutional LSTM[20] を実現した。Convolutional LSTM の流れを式 3.8～3.12 に示す。*は畳み込み演算を示している。Shi らは Convolutional LSTM を用いて降水量を時系列予測する手法を提案した。時空間の相関関係をより良い精度で捉え

られるため、従来の重み計算に全結合層を用いた LSTM よりも高い精度で予測できることを示した。

$$f_t = \sigma_g(W_f * x_t + U_f * h_{t-1} + V_f \circ c_{t-1} + b_f) \quad (3.8)$$

$$i_t = \sigma_g(W_i * x_t + U_i * h_{t-1} + V_i \circ c_{t-1} + b_i) \quad (3.9)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \quad (3.10)$$

$$o_t = \sigma_g(W_o * x_t + U_o * h_{t-1} + V_o \circ c_t + b_o) \quad (3.11)$$

$$h_t = o_t \circ \sigma_h(c_t) \quad (3.12)$$

3.1.2 Bidirectional LSTM

LSTM は、ある状態の中間層の出力値を次の状態に順伝播するネットワークである。これに対して、Schuster らは、中間層の出力を、未来への順伝播と過去への逆伝播の両方向で伝播する Bidirectional LSTM[12] を提案した。図 6 にその仕組みを示す。また 2.5 節でいくつか紹介したように、これまでに様々な時系列予測に対して Bidirectional LSTM を用いることで、従来の LSTM よりも精度が向上したという結果 [13, 14] が示されている。これらは順伝搬と逆伝搬の双方向伝搬を行う Bidirectional LSTM を用いることで、時系列の特徴を従来の LSTM よりも効果的に学習できた例である。

3.2 提案手法のアプローチ

2.6 節でも述べたように既存手法には以下のような問題点がある。本節では、これらの問題点に対する提案手法のアプローチについてまとめる。

1. 時刻表のない路線 (一定間隔の頻度で走行する路線) を対象に評価を行っているため早着によるバス停での意図的な停車など、時刻表ベース特有の特徴が考慮されていない
2. 運行データのみを特徴量としているため気象の影響によって運行が乱れた際に予測精度が低下する

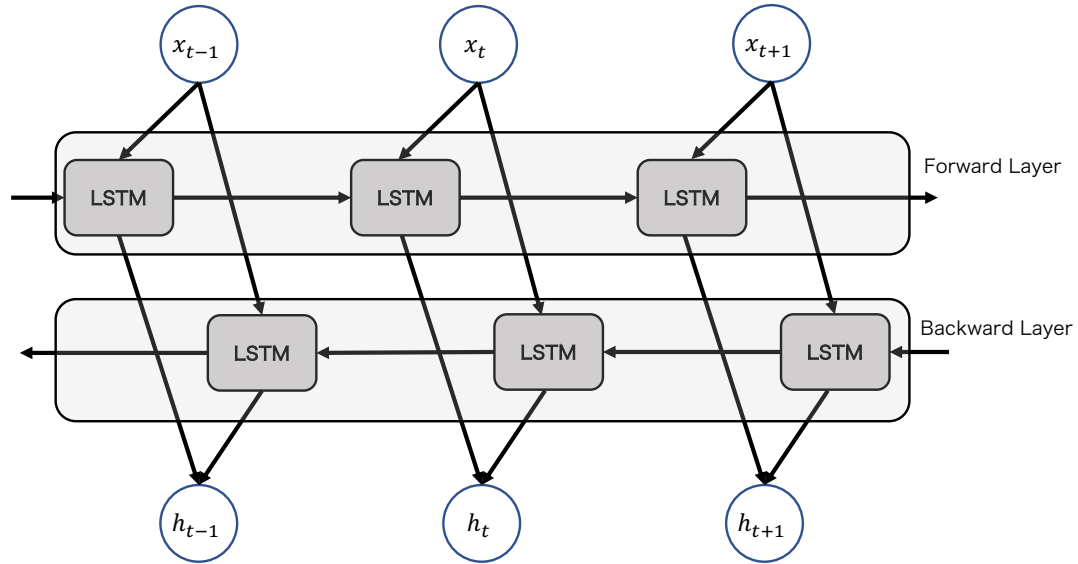


図 6: Bidirectional LSTM による双方向伝搬の概略図

まず1つ目の問題点に対するアプローチについて述べる。時刻表に基づいた路線の大きな特徴として、走行時間と停車時間がそれぞれ独立していることが挙げられる。走行時間に関しては時刻表に対する早着や遅延といった情報が与える影響は少ないが、停車時間に関してはバスがバス停に到着した時刻と時刻表との差が停車時間に大きな影響を与える。これらの特徴を考慮するために、本研究では走行時間と停車時間でそれぞれ別々の特徴量を用いた到着時刻予測手法を提案する。また、停車時間にのみ時刻表に基づいた特徴量を使用する。第4章の評価実験では、走行時間と停車時間を同じ1つのモデルで予測する手法と、走行時間と停車時間を別々のモデルで予測する手法の2つを比較することでこのアプローチの正当性を確かめる。

次に2つ目の問題点に対するアプローチとして、提案手法では、運行情報に加えて、同じ時間帯に観測された気象情報も Convolutional LSTM を用いて同時に畳み込みを行う。それによって気象情報を考慮した学習と予測が可能となるため、

天候に起因して運行が乱れた際に予測精度が大幅に下がるという既存研究の問題点が解決できると考える。また天候状態がバスの運行に影響を与える際、停車時間と走行時間に与える影響は別々であると考え。例えば、停車時間に影響を与えるのは乗降者数であるが、走行時間に影響を与えるのは、走行する地域の交通状況に依存する。そのため1つ目の問題に対するアプローチでも述べたように走行時間と停車時間は別々に予測を行う。第4章の評価実験では、気象情報を特微量に加えた提案手法と気象情報を考慮していない既存手法の予測精度を比較することで、気象情報の有効性を確かめる。特に、天候に起因して運行が乱れた際に精度が向上すると考えられるため、予測対象日の天候ごとに手法間の予測精度の差を評価する。

さらに、予測精度を向上させるアプローチとして、3.1.2節で紹介した Bidirectional LSTM を使用する。2.5節で、Bidirectional LSTM を用いた時系列予測の精度が、従来の LSTM よりも優れた精度であった例を紹介した。本研究のタスクはそれらと同様の時系列予測であるため、Bidirectional LSTM が有効である可能性があると考えた。そこで、提案手法でベースとなる Convolutional LSTM に対して Bidirectional LSTM を適用した。第4章の評価実験では、Bidirectional LSTM を適用したモデルと適用しないモデルを比較することでその有効性を評価する。

3.3 提案手法の概要

本研究では3.2節で述べた3つのアプローチで路線バスの到着時刻予測精度の向上を目指す。提案手法では、予測対象の路線における過去の一定期間のリンク所要時間と気象情報を元に未来の到着時刻を予測する。また、Petersen ら [7] の手法と比較を行うことで本研究の優位性を示す。まず始めに提案手法の全体の流れを図7に示す。過去の走行時間や停車時間などのリンク所要時間を含む過去運行データと天候、気温、降水量を含む気象データの2種類のデータを入力として、走行時間と停車時間の予測を行う。また、図にも示している通り走行時間と停車時間それぞれに気象データを結合させて別々の Bidirectional LSTM への入力としている。その出力として得られる走行時間と停車時間の予測値を結合して、最終的な到着時刻予測の出力として得る。

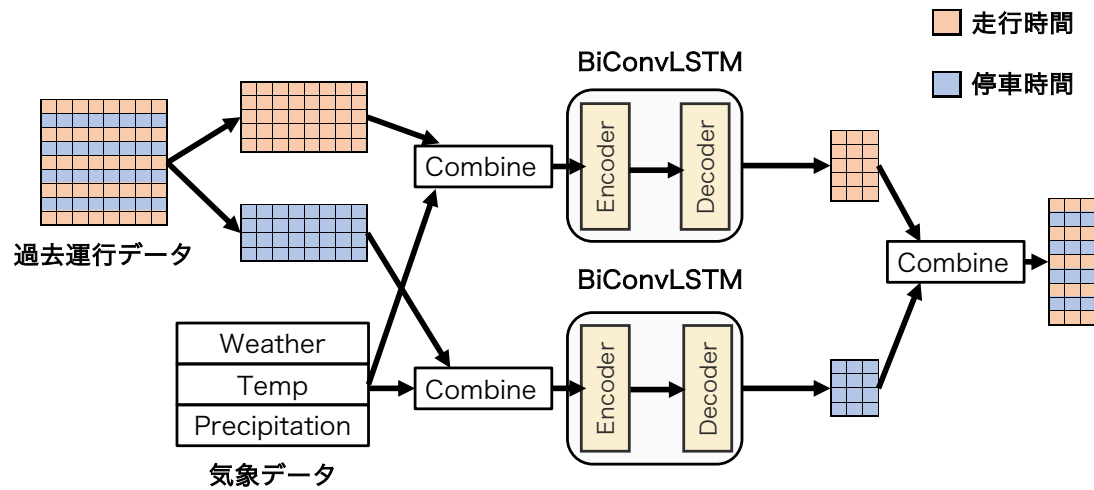


図 7: 提案手法の流れ

提案手法において、以下の3つのパラメータを決定する必要がある。以降それぞれについて述べる。

1. *reference_frequency*
2. *input_timesteps*
3. *output_timesteps*

図8に入出力データに対して上記3つのパラメータがどのように適用されるのかを表したものを示す。縦軸はバス路線における始点から終点までの順方向のリンクを示し、横軸は時間軸を表す。左側が入力、右側が出力を示している。*reference_frequency* は30分や1時間などの時間幅を指定するパラメータである。その時間幅に観測された各リンクのリンク所要時間が過去運行データの各要素となる。そのため正しく学習、予測を行うためには対象となるバス路線の運行頻度によってこの*reference_frequency*を決定する必要がある。これは入力、出力ともに共通のパラメータである。また、もう1つの入力データである気象デー

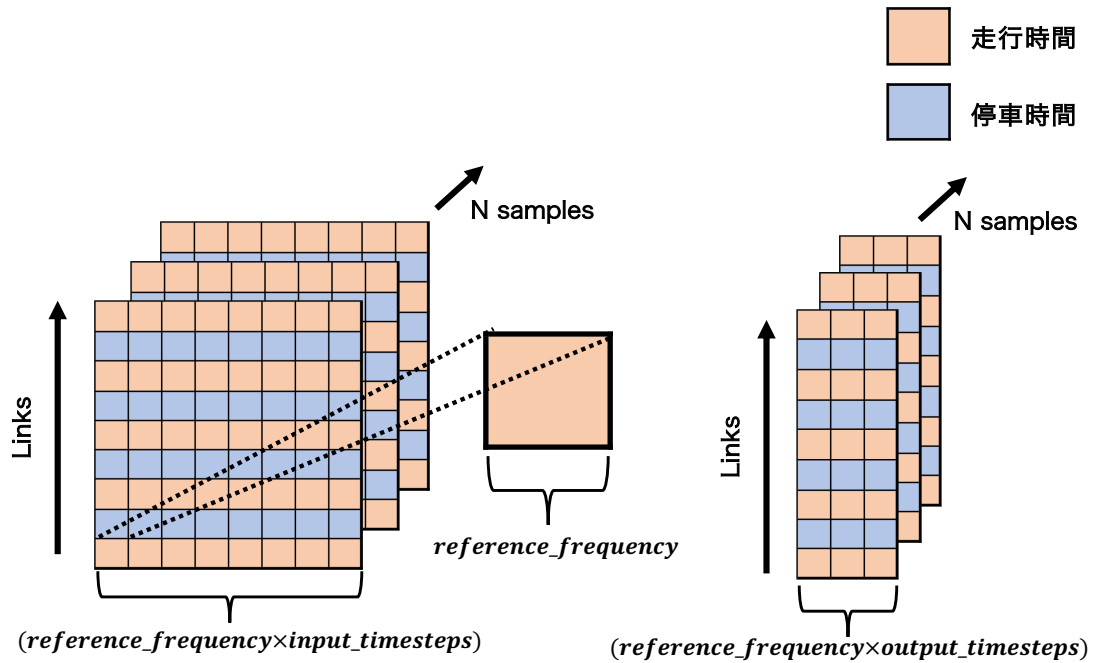


図 8: 過去運行データにおけるパラメータ

タに対しても適用される。 $input_timesteps$ は予測を行うための入力となる時間幅、 $output_timesteps$ は予測を行う時間幅をそれぞれ指定するパラメータである。図中の例では $input_timesteps$ が 8、 $output_timesteps$ が 3 となっている。ここで仮に $reference_frequency$ が 60 min であると仮定すると以下のように入力と出力が決定される。

$$(input_timesteps \times reference_frequency) = 8 \times 60 \text{ min}$$

$$(output_timesteps \times reference_frequency) = 3 \times 60 \text{ min}$$

このように提案手法では $(input_timesteps \times reference_frequency)$ のパターンから $(output_timesteps \times reference_frequency)$ のパターンを Convolutional LSTM を用いて学習および予測を行う。

3.4 データ形式

提案手法で用いるデータセットの形式について述べる。本研究では以下の過去運行データと気象データの二種類のデータを用いて路線バスにおける到着時刻予測を行う。

1. 過去運行データ
2. 気象データ

3.4.1 過去運行データ

本研究で到着時刻予測に用いるデータとして過去運行データがある。この過去運行データを用いることで、走行時間や停車時間の変動パターンを学習することが可能となる。表 1 にそのデータ形式を示す。また、過去運行データの一例を表 2 に示す。

表 1: 過去運行データの形式

フィールド	型
DateTime	日付 時刻
DayOfWeek	整数
LineDirectionLinkOrder	整数
LinkRef	文字列
LinkDuration	浮動小数点
DiffTime	浮動小数点

以下に過去運行データに含まれるフィールドについてそれぞれ述べる。

DateTime

各リンクの所要時間の開始時刻を示す。つまり、走行時間であれば出発時刻、停車時間であれば、バス停での停車開始時刻が記録されている。

表 2: 過去運行データのサンプル

DateTime	2020-01-03 15:00:42.5
DayOfWeek	2
LineDirectionLinkOrder	1
LinkRef	1_2
LinkDuration	560
DiffTime	43

DayOfWeek

曜日を 0 から 6 までの識別子で表す。(0 :月曜日、 1 :火曜日、 2 :水曜日 ... 6 :日曜日)

LineDirectionLinkOrder

路線において始点から終点までの順方向で何番目のリンクであるかを示す。

LinkRef

路線における各リンクの識別子

LinkDuration

各リンクの所要時間

走行時間: (バス停への到着時刻 - 前のバス停の出発時刻)

停車時間: (バス停の出発時刻 - バス停の到着時刻)

DiffTime

停車時間のみを対象に、バス停に到着した時刻がそのバス停の出発予定時刻からどれくらい余裕があるかを表す。また到着時刻が既に出発予定時刻を超過していた場合は 0 とする。下記のように算出される。また、 i はそれぞれ過去運行データの行識別子を指す。

$$DiffTime_i = TimeTable_i - StopStart_i$$

$DiffTime_i$: 到着時刻と出発予定時刻の差異 ($0 \leq$)

$TimeTable_i$: 対象バス停の時刻表における出発予定時刻

$StopStart_i$: 対象バス停への到着時刻

3.4.2 気象データ

提案手法で用いる2つ目のデータである気象データについて述べる。到着時刻を予測する地域の3つの気象情報を使用する。表3にそのデータ形式を示す。また、気象データの一例を表4に示す。

表 3: 気象データの形式

フィールド	型
DateTime	日付 時刻
Weather	整数
Temp	浮動小数点
Precipitation	浮動小数点

表 4: 気象データのサンプル

DateTime	2020/1/3 10:00
Weather	10
Temp	12.3
Precipitation	3.5

DateTime

日付と時刻

Weather

天候状態を晴れ (1)、曇り (4)、雨 (10) の3つの分類として示す。

Temp

気温 (摂氏単位 °C)

Precipitation

降水量 (単位:mm)

3.5 特徴量

3.4 節で述べたように本研究では過去運行データと気象データの 2 種類のデータを用いる。走行時間には 4 次元、停車時間には 5 次元の時系列特徴量を Convolutional LSTM を用いて畳み込みを行うことで学習と予測を行う。以下にそれぞれの特徴量を列挙する。

走行時間

1. *LinkDuration*
2. *Weather*
3. *Temp*
4. *Precipitation*

停車時間

1. *LinkDuration*
2. *DiffTime*
3. *Weather*
4. *Temp*
5. *Precipitation*

停車時間にのみ、時刻表に基づいた特徴量である *DiffTime* を使用している。時刻表に基づいて走行している路線バスにおいて停車時間は、時刻表と強い関連性があると考えられるためである。時刻表に記述されている時刻は一般的にそのバス停の出発時刻を表している。つまり、バス停に到着した時刻がその時刻表よりも早かった場合、バス運転手は意図的に停車時間を調整すると考えられる。日本の道路交通法では最大5分間の停車が認められている。一方で走行時間は時刻表から大きな影響を受けることは考えにくい。以上のことから停車時間にのみ時刻表に基づいた特徴量を使用する。バス停に到着した時点で時刻表の出発時刻を超過していた場合に0としているのは、超過していた場合には乗客の乗降が済み次第すぐに出発すると予想されるためである。このような場合は時刻表からの差は停車時間に影響を与えないと考えられる。図9に走行時間と停車時間の予測に用いる特徴量をそれぞれ図示する。このように、走行時間、停車時間に対して同じ行数の Weather, Temp, Precipitation それぞれを *reference_frequency* にしたがってアップサンプリングもしくはダウンサンプリングを行った後に一列として追加する。さらに、畳み込みフィルターとストライドの幅を特徴量の列数にすることで各時間帯の特徴量を同時に畳み込むことができる。そのため走行時間や停車時間の時系列の関係性を保持したまま複数の特徴量を考慮した学習が可能となる。走行時間、停車時間それぞれの入力と出力は以下のようになる。 N はデータ数、 num_links はリンク数を表している。

走行時間

$$input = (N, input_timesteps, num_links, 4, 1),$$

$$output = (N, output_timesteps, num_links, 1, 1)$$

停車時間

$$input = (N, input_timesteps, num_links, 5, 1),$$

$$output = (N, output_timesteps, num_links, 1, 1)$$

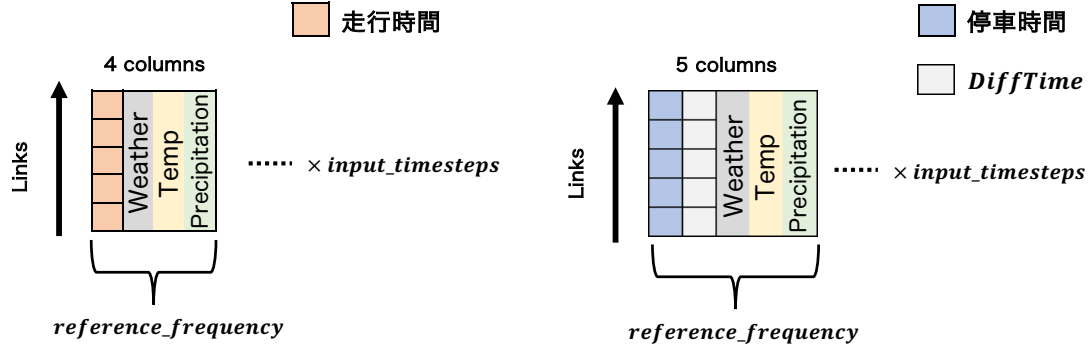


図 9: 走行時間と停車時間の各特徴量

3.6 前処理

本節では、提案手法で用いる 2 つのデータセットに対する前処理について述べる。

3.6.1 データ分割

提案手法の学習を行う前に、過去運行データを指定された input_timesteps と output_timesteps に基づいて分割する処理が必要となる。 $\text{input_timesteps} = 8$, $\text{output_timesteps} = 3$ の場合におけるデータセットの分割処理の例を図 10 に示す。ここで x_{train} が学習時の入力、 y_{train} は学習時の出力をそれぞれ表す。このように、1 timestep ずつスライドさせていくことで、各入力 $(N, \text{input_timesteps}, \text{num_links})$ とそれに対応する出力 $(N, \text{output_timesteps}, \text{num_links})$ のデータ構造を得る。

3.6.2 LinkDuration, DiffTime に対する処理

LinkDuration と DiffTime に関しては、それぞれを曜日、 $\text{reference_frequency}$ に基づく時間帯毎の統計値を元に式 3.13 にしたがって正規化する。 $x_{ln,t}$ は正規化前、 $x'_{ln,t}$ は正規化後の各 LinkDuration もしくは DiffTime をそれぞれ示してい

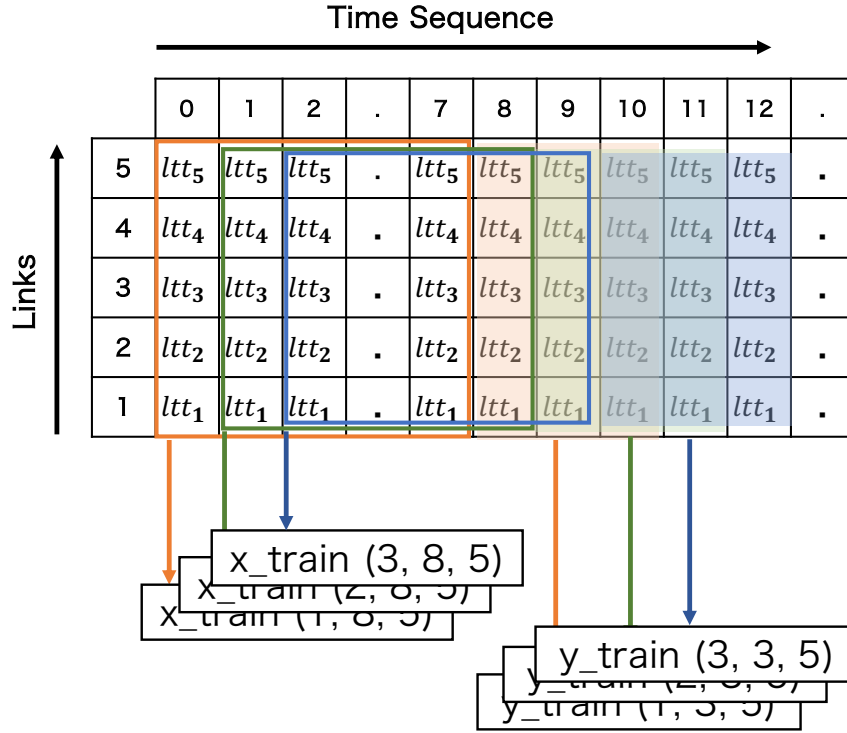


図 10: 学習データセットの分割処理 ($input_timesteps = 8, output_timesteps = 3$)

る。また、 ln はリンクの識別子、 t は観測された時間帯 (timestep) をそれぞれ示す。 $\bar{x}_{ln,dow,tod}$ は各リンク (ln) の曜日 (dow)、時間帯 (tod) に基づく平均値、 σ_{ln} は各リンクの標準偏差を示す。このように、通常の曜日、時間帯が持つパターンからの偏差に基づいて正規化している。

$$x'_{ln,t} = \frac{x_{ln,t} - \bar{x}_{ln,dow,tod}}{\sigma_{ln}} \quad (3.13)$$

また、 $\bar{x}_{ln,dow,tod}$ と σ_{ln} を計算する際に中央絶対誤差 (Median absolute deviation)[21] を用いて極端な外れ値を除いている。これは平均や標準偏差などが外れ値の影響を受けることを防ぐためである。

次に欠損値の処理について述べる。一般的な路線バスは深夜など運行が存在しない時間帯が存在する。このような時間帯では *LinkDuration* と *DiffTime* の欠損値が発生する。このような欠損値が存在すると朝の時間帯の予測を行うこと

ができないため、そのような欠損値に対しては前の値で置換を行う。たとえば、 $reference_frequency = 60$ で 23 時から 5 時までの運行データが存在しない場合はその日の 22 時の運行データで全て置換される。

これらの外れ値の除去と欠損値の補完は Petersen らの手法 [7] でも同様の方法で行われている。

3.6.3 Weather, Temp, Precipitation に対する処理

3.6.2 節では *LinkDuration* と *DiffTime* に対する処理について述べた。ここでは残りの特徴量である *Weather*、*Temp*、*Precipitation* に対する前処理について述べる。これら 3 つの特徴量に対しては、式 3.14 のように四分位点を基準にして標準化を行う。 x_t は標準化前、 x'_t は標準化後の特徴量をそれぞれ示している。 Q_1 、 Q_2 、 Q_3 はそれぞれ x_t の第一四分位点、第二四分位点、第三四分位点を表している。

$$x'_t = \frac{x_t - Q_2}{Q_3 - Q_1} \quad (3.14)$$

3.7 バス到着時刻予測ネットワークアーキテクチャ

図 11 に Convolutional LSTM を用いた到着時刻予測モデルのアーキテクチャを示す。Petersen らが提案した Convolutional LSTM ベースのアーキテクチャ [7] を元に設計した。実装には Tensorflow 2.1.0 [22] 上に Python で構築されたオープンソースニューラルネットワークライブラリである Keras [23] を用いた。以降、図中の左側の処理をエンコーダ、右側の処理をデコーダと呼ぶ。このようにエンコーダ、デコーダそれぞれ 2 層の Convolutional LSTM を持つ。変更を加えた点は以下の 2 点である。まず、エンコーダの一層目の Convolutional LSTM の畳み込みフィルターの幅とストライドを走行時間、停車時間それぞれの特徴量数に合わせることで、過去運行データに加えて気象データも考慮した学習が可能にした点である。2 点目は全 4 層の Convolutional LSTM 層全てに Bidirectional LSTM を適用した点である。

以降、アーキテクチャの全体構造についてエンコーダとデコーダに分けて述べる。まず、エンコーダの処理における *input* と *output* を含めた詳細を図 12 に示す。この図は走行時間の例を示している。まず、過学習を防ぐために各 Convolutional LSTM 層間に Dropout 層 [24] と Batch Normalization 層 [25] を使用している。これは、デコーダに関しても同様である。図中の (1), (2) 層の *input* と *output* に赤字で記述されている 4 は走行時間では 1 *timestep* に対して 4 次元の特徴量があるためであり、停車時間の場合は 5 になる。各 Convolutional LSTM 層の出力形式について述べる。(2) のエンコーダにおける 1 層目の Convolutional LSTM 層では入力の系列データと同じ長さ *input_timesteps* を出力している。(5) の 2 層目の Convolutional LSTM 層では最後の *timestep* の出力のみを取得している。また、(2) の 1 層目 Convolutional LSTM では畳み込みのストライドを (1, 4) にすることで 1 *timestep* に対して 4 列を同時に畳み込みを行う。2 層目の Convolutional LSTM である (5) に関しては (1, 1) である。2 層の Convolutional LSTM と Dropout 層、Batch Normalization 層を経たのちに、Flatten 層で入力を平滑化してからデコーダへと渡す。

次にデコーダでの処理の詳細を図 13 に示す。デコーダでの処理は、走行時間、停車時間に拘らず同じである。エンコーダからの入力に対して *output_timesteps* の数だけ複製を行う Repeat Vector の処理を行うことで、複数先の到着時刻予測を行うことができる。以降はエンコーダと同様に 2 層の Convolutional LSTM 層と間に Dropout 層と Batch Normalization 層を持つ。最後の TimeDistributed では、活性化関数に ReLu(Rectified Linear Unit) を用いることで走行時間や停車時間の最終的な予測を行う。オプティマイザーには RMSprop[26]、ロス関数には平均二乗誤差 (MSE) を用いる。

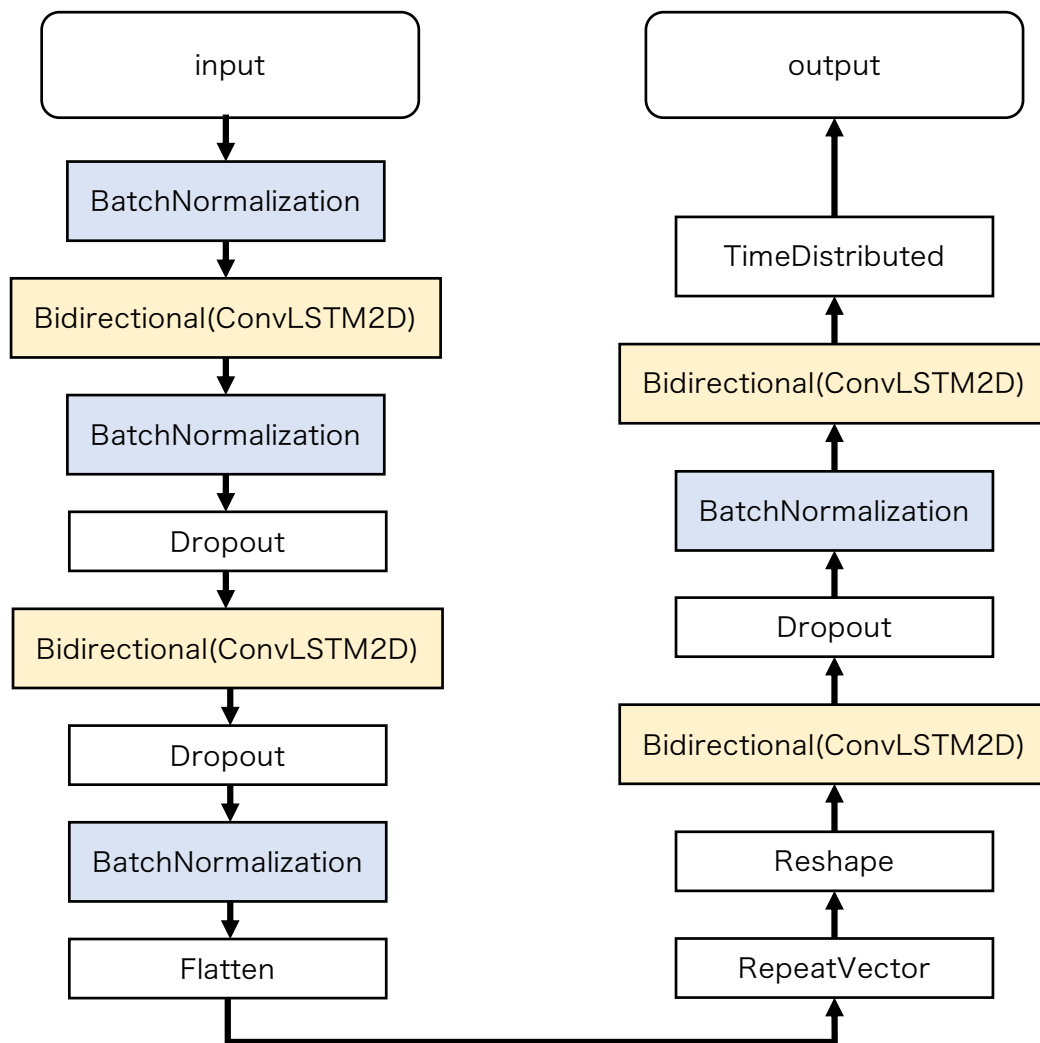


図 11: Bidirectional Convolutional LSTM 到着時刻予測モデル

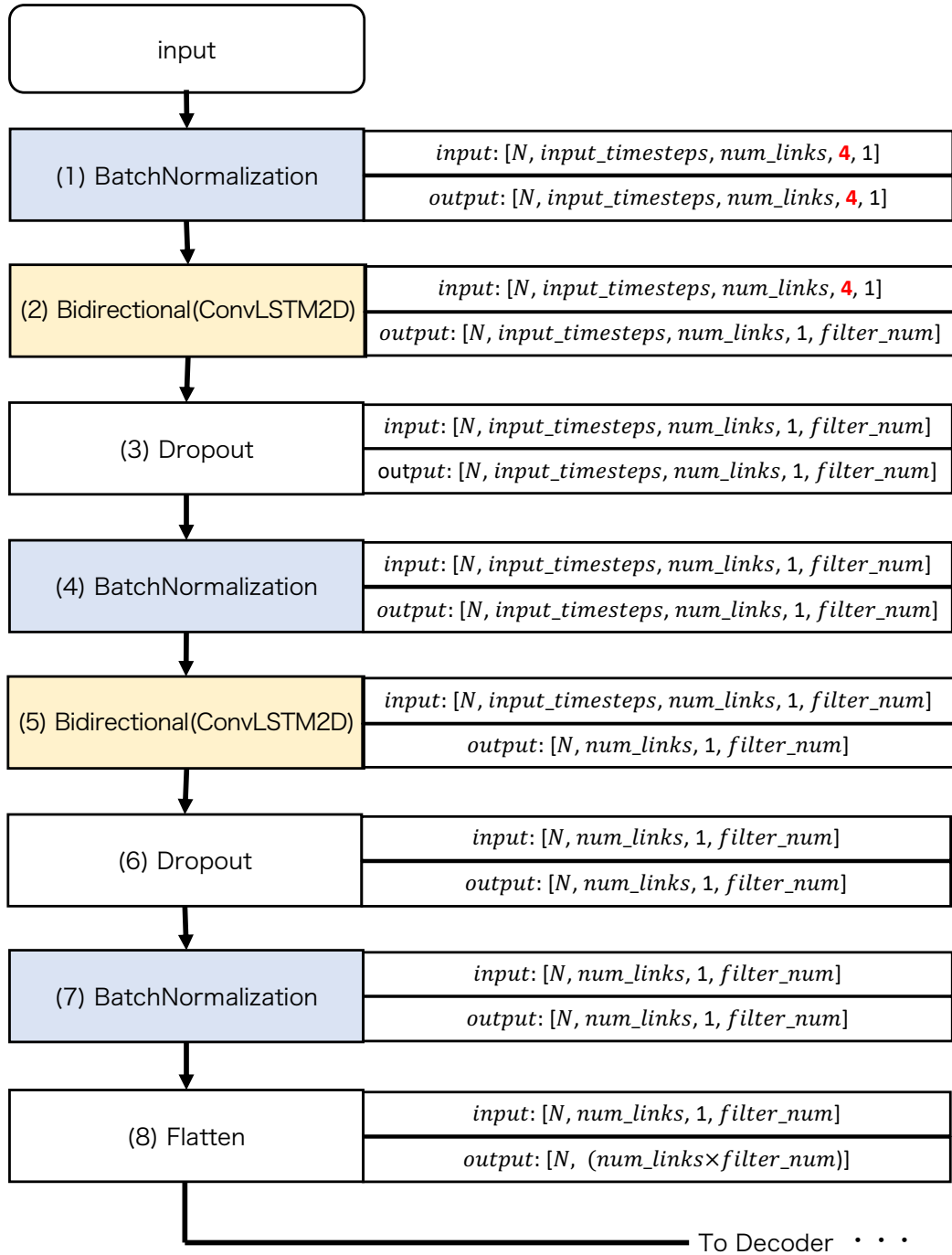


図 12: 提案手法: エンコーダにおける処理 (走行時間)

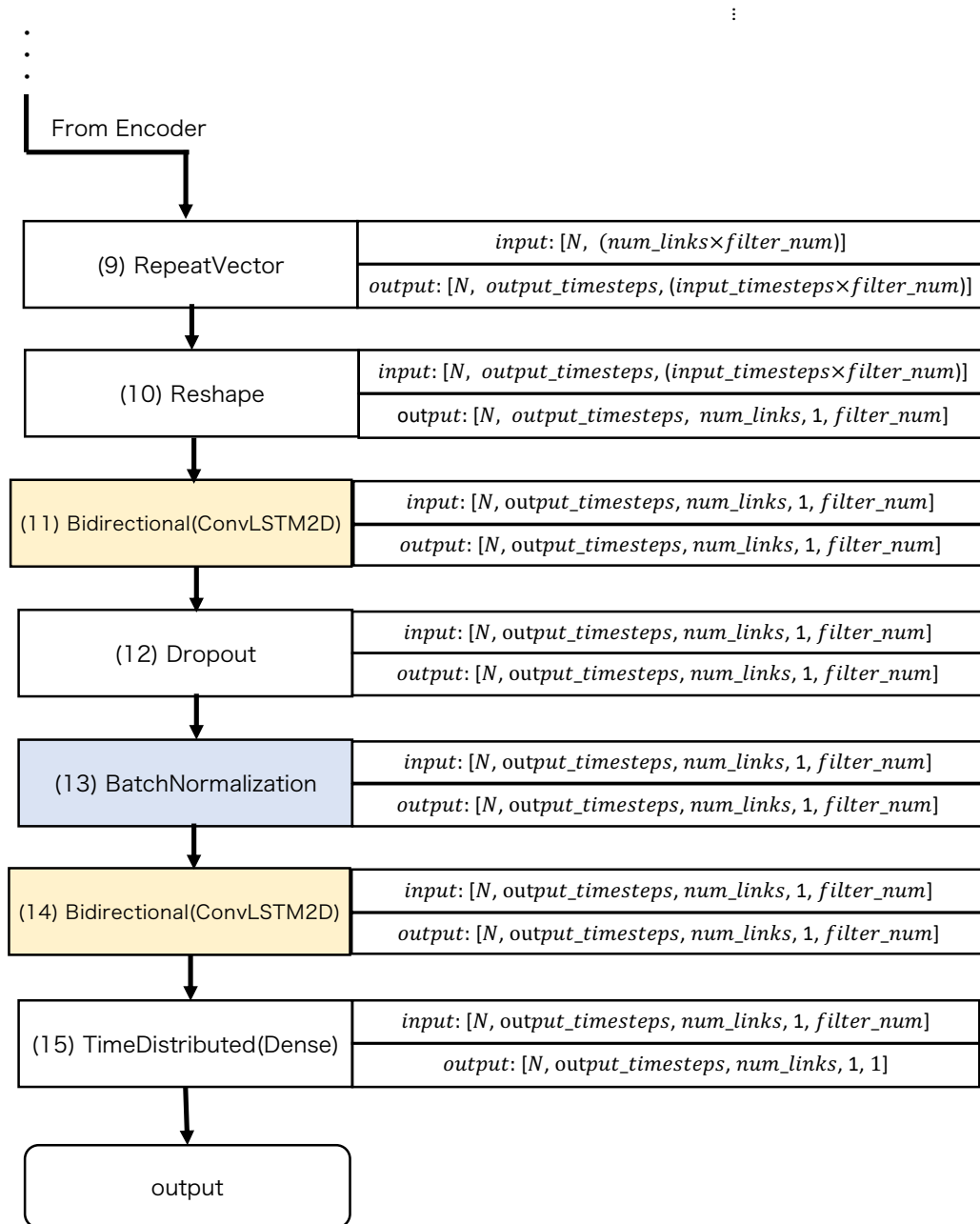


図 13: 提案手法: デコーダにおける処理

4. 評価実験

本章では、兵庫県神戸市の路線バスの実データを用いた提案手法の評価について述べる。まずはじめに評価の対象とした路線とその選定理由について述べる。その後、評価に使用したデータセットについて述べる。次に評価指標と比較対象を説明し、評価結果に期待する仮定についても述べる。次に比較手法と提案手法のパラメータチューニングについて述べる。そして、最後に評価結果についてまとめる。

4.1 評価対象路線について

兵庫県神戸市中心部の約 15 km を経路に持つバス路線を評価対象として選定した。バス停の数は 6 である。選定する際に以下の項目を考慮して決定した。

1. 十分な距離を有した路線
2. 朝、夕の所要時間が乱れやすいピークの時間帯が存在する
3. 終点から他の公共交通機関への乗り換えが可能

本研究において比較対象となる Petersen らは、デンマークの首都であるコペンハーゲン市街地の約 14 km の一路線を対象に評価している [7]。彼らの評価路線は上記の 3 つの条件の内、上 2 つを満たしている。そこで本研究でも同様の 2 つの条件に加えて終点から他の公共交通機関への乗り換えが可能という条件を満たした類似度の高い路線を選定した。3 つ目の追加した条件に関しては、利用者が終点から他の交通機関へ乗り換えるシナリオを想定したためである。評価対象となる路線は始点が神戸国際大学前、終点为新神戸駅の約 15 km の路線である。時刻表では始点での出発から終点への到着まで 35 分間での運行が予定されている。また、バス停数は 6 のため、計 5 区間の走行時間と 4 回の各バス停での停車時間が存在する。評価対象路線の経路図を図 14 に示す。次に、この路線の運行で予定されている各所要時間を表 5 に示す。予定所要時間は、時刻表に記載されている各バス停間の出発から出発、最終区間のみ出発から到着までの所要時間を示して

表 5: 評価対象路線における予定所要時間

出発	到着	予定所要時間
神戸国際大学前	ウエストコート7番街前	2分
ウエストコート7番街前	六甲アイランド病院前	1分
六甲アイランド病院前	神戸ベイシェラトンホテル	2分
神戸ベイシェラトンホテル	神戸三宮	20分
神戸三宮	新神戸駅	10分
		計 35 分

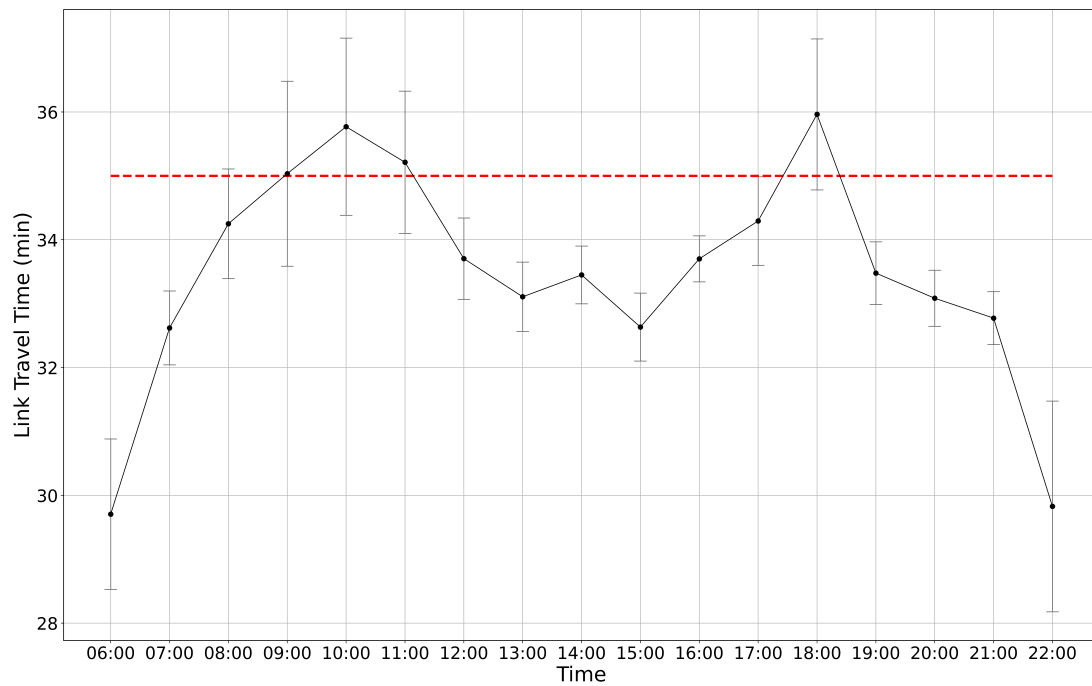


図 15: 時間帯ごとの平均所要時間と標準偏差

4.2 データセットについて

4.2.1 過去運行データ

今回の評価実験では 2020 年 1 月 1 日から 2020 年 9 月 30 日までの 39 週間の過去運行データを使用した。この過去運行データは過去に運行中のバスに取り付け

られている Global Positioning System (GPS) から 0.5 秒毎に取得された位置情報を元に作成した。バスがどの路線のどの便を運行しているかの判定は General Transit Feed Specification (GTFS)[27] と運行指示書のフォーマットを用いて構築したリレーショナルデータベースを元に行った。図 16 に今回、過去運行データを作成するに当たって使用したリレーショナルデータベースの概略図を示す。GTFS とは Google 社が提供している世界共通の公共交通機関向けのオープンフォーマットである。リアルタイム情報を提供するフォーマットである GTFS Realtime に対して静的な情報を扱うため Static GTFS とも呼ばれる。図に示すように、路線バスの場合は複数のテキストファイルを用いて時刻表、経路、バス停などを定義できる。以下、図中に登場する各テキストファイルの役割を簡単に説明する。

stops.txt

緯度、経度を用いてバス停を定義

stop_times.txt

stops.txt に含まれるバス停での全ての到着時刻、出発時刻を定義

shapes.txt

緯度、経度の系列データで経路を定義

routes.txt

shapes.txt を元に路線を定義

trips.txt

routes.txt と stop_times.txt を元に各路線の便を定義

運行指示書とは、乗務員の氏名、車両番号、運行開始、終了時の地点及び日時などが記載されているものである。国土交通省はバス事業者に対してこの運行指示書の作成と 1 年間の保存を義務付けている [28]。この運行指示書に GTFS の trips.txt で定義された複数の便 (trip) を割り当てることで、バスがどの路線のどの便を走行しているというラベル付けが可能となる。

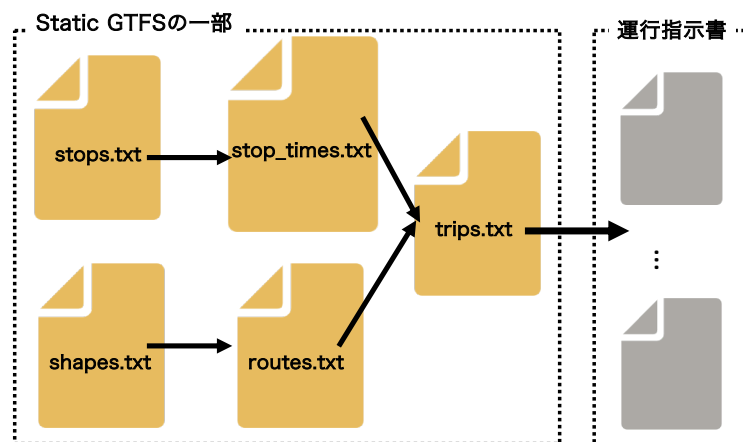


図 16: 過去運行データの作成に用いたデータベースの概略図

4.2.2 気象データ

気象データは気象庁が公開している過去の気象データ [29] を使用した。評価対象路線の場所に基づいて 2020 年 1 月 1 日から 2020 年 9 月 30 日の 1 時間毎の兵庫県神戸市の天気、気温、降水量を取得した。取得した天気情報の内訳を表 6 に示す。提案手法では天気は晴れ、曇り、雨の 3 種類に分類されたデータを用いるため、取得したデータも 3 種類に整形する。快晴、晴れ、薄曇を晴れ、雨、雷、雪、みぞれ、にわか雨をすべて雨としてまとめた。整形後、3 種類の観測回数は晴れが 2987 回、曇り 2661 回、雨 928 回となった。

4.3 評価方法

データセットに含まれる最初の 38 週を学習データ、最後の 1 週間 (2020 年 9 月 24 日 ~ 2020 年 9 月 30 日) をテストデータとして評価実験を行う。また、時間帯 i における $ln = 0 \sim u$ 、始点から終点までの *LinkDuration* の総和を比較するこ

表 6: 取得した天候情報の内訳

天気	観測回数
快晴	84
晴れ	2879
薄曇	24
曇	2661
雨	880
雷	41
雪	4
みぞれ	2
にわか雨	1

とで評価を行う。式 4.1 の Y_i は実際の所要時間、式 4.2 の $\hat{Y}_i$ は予測値を表す。

$$Y_i = \sum_{ln=0}^u Y_{i,ln} \quad (4.1)$$

$$\hat{Y}_i = \sum_{ln=0}^u \hat{Y}_{i,ln} \quad (4.2)$$

評価指標は比較対象である Petersen らの手法 [7] に合わせて以下の誤差をベースにした 3 つの評価指標 (式 4.3~4.5) を用いた。上から平均二乗誤差 (Root Mean Squared Error)、平均絶対誤差 (Mean Absolute Error)、平均絶対パーセント誤差 (Mean Absolute Percentage Error) をそれぞれ表している。RMSE、MAE は共に平均化された誤差に基づく評価指標である。RMSE は MAE と比べて外れ値の影響を受けやすいという特徴がある。単位は共に分である。3 つ目の MAPE は観測値 Y_i に対する相対的な誤差に基づく評価指標である。単位は%である。

$$\text{RMSE}(Y, \hat{Y}) = \sqrt{\frac{\sum_{i=0}^N (Y_i - \hat{Y}_i)^2}{N}} \quad (4.3)$$

$$\text{MAE}(Y, \hat{Y}) = \frac{\sum_{i=0}^N |Y_i - \hat{Y}_i|}{N} \quad (4.4)$$

$$\text{MAPE}(Y, \hat{Y}) = \frac{1}{N} \sum_{i=0}^N \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right| \quad (4.5)$$

4.4 比較対象

本研究では、以下の3つの既存手法を比較対象とした。

1. Historical Average[30]
2. PureLSTM[7]
3. ConvLSTM[7]

Historical Average

過去のデータから算出した平均を元に予測を行うモデルである。曜日、時間帯毎の平均値をそのまま予測値として用いる。このような手法は交通パターンが静的な地域での予測では優れた予測を行うことが可能である。しかし、問題点として過去の平均値を用いて予測しているため、データの変化に対する収束速度がLSTMなどの深層学習を用いた手法よりも遅いことが挙げられる。特に、交通状況などの外的要因の変動が激しい都市部を対象とした予測では予測精度の低下が考えられる。この手法は近年の研究において主に比較対象として用いられている。

PureLSTM

Petersen らが提案しているアーキテクチャにおいて Convolutional LSTM を使用していない手法[7]である。Duan らも類似した LSTM を用いた手法[31]を提案している。

ConvLSTM

上記と同様に Petersen らが提案している Convolutional LSTM を用いた手

法 [7] である。彼らの論文では、PureLSTM よりも Convolutional LSTM を用いた手法の方が精度が高くなったことを報告している。これは、都市部のように運行が不安定なバス路線において従来の LSTM では捉えきれない細かいパターンを畳み込みによって学習できるためであると結論づけている。

提案手法に関しても以下の 3 つのモデルに分けて評価を行った。

1. Model1
2. Model2
3. Model2_Bi

Model1

提案手法において、走行時間と停車時間を 1 つのモデルで予測した手法である。特徴量としては *LinkDuration*、*Weather*、*Temp*、*Precipitation* の 4 種類を用いた。走行時間と停車時間を分けて予測することの有用性を示すためにこのモデルを定義した。なお Bidirectional LSTM は使用していない。

Model2

走行時間と停車時間を別々に予測する提案手法において、Bidirectional LSTM を使用していないモデルである。提案手法における Bidirectional LSTM の有効性を確認するために定義した。

Model2_Bi

提案手法をそのまま適用したモデルである。

以上の合計 6 種類のモデルで到着時刻予測精度を比較する。今回の評価対象のような都市部の路線において、既存手法の中では ConvLSTM が最も精度が高くなると考えられる。それは都市部の路線の複雑な運行パターンの変動を Convolutional LSTM を用いることで通常の LSTM を用いる PureLSTM よりも効果的に学習できることが先行研究で示されているからである。提案手法の 3 つの手法はいずれ

も、既存手法で考慮されていなかった気象情報を考慮した学習を行っているため全て既存手法よりも精度が高くなると予想される。また、その中でも走行時間と停車時間を別々のモデルで学習、予測している Model2 と Model2_Bi は Model1 と比べて時刻表ベースの路線に適していると考えられるためいずれも Model1 よりも高い精度を期待する。Model2 と Model2_Bi については、時系列予測のタスクにおいて従来の LSTM と比べて有効である可能性がある Bidirectional LSTM を適用することでさらに精度が上がることを期待する。特に天候に伴い運行が乱れた際に精度の上昇幅が大きくなると考えられる。そのため、予測対象日の天候ごとに精度の上昇幅を評価する。

4.5 パラメータチューニング

今回、Convolutional LSTM を用いた Petersen らの手法 [7] を比較対象として評価を行う。比較手法と提案手法に適している各パラメータは評価対象のバス路線によって異なる。比較対象となる予測結果を算出するにあたって、より予測精度を高くするために PureLSTM と ConvLSTM に対してそれぞれハイパーパラメータチューニングを行った。提案手法に関しても同様である。ハイパーパラメータチューニングとは機械学習や深層学習を行う際に、予めパラメータを決定する作業を指す。パラメータチューニングには HyperBand[32] を用いて、最も有効なパラメータを示す Best Trial のパラメータをそれぞれのモデルのパラメータとして決定した。また、比較手法、提案手法ともに評価対象の 1 週間における平均二乗誤差 (MSE) を Validation Loss としてパラメータ探索を実行した。Maxepoch は Hyperband によるパラメータチューニングにおいて、1 つのモデルを学習するための最大エポック数を指す。

4.5.1 比較手法のパラメータチューニング

表7に PureLSTM、表8に ConvLSTM、それぞれの探索範囲とパラメータチューニングの結果を示す。Batchsize はデータセットをいくつかサブセットに分けた際に 1 つのサブセットに含まれるデータ数のことを指す。Dropout rate は過学

習を防ぐための層である Dropout 層におけるパラメータで、意図的に無効にするノード数を決定する確率を指す。Learning rate は一回の学習でどの程度、重みやバイアスを修正するかの率を示す。この値が大きくなると、一度に修正される重みやバイアスも大きくなる。また、表 7 における Units は LSTM の中間層の次元数を指す。表 8 における Filters は Convolutional LSTM における畳み込みフィルタの数を表している。ConvLSTM における Kernel size 0、Kernel size 1 はエンコーダ、デコーダがそれぞれ持つ 2 層の Convolutional LSTM における畳み込みフィルタの行数を示している。Kernel size 0 はエンコーダ、デコーダそれぞれの一層目のフィルタの行数、Kernel size 1 はエンコーダ、デコーダそれぞれの二層目のフィルタの行数を指す。探索範囲は、評価対象路線の総リンク数である 9 に基づいて決定した。

表 7: PureLSTM: パラメータチューニング結果

Parameter	Value	Best Trial
Maxepoch	100	34
Batchsize	[16,32,64,128,256]	128
Dropout rate	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]	0.3
Units	[16, 32, 64, 128, 256]	128
Learning rate	[0.01, 0.001, 0.0001]	0.001

表 8: ConvLSTM: パラメータチューニング結果

Parameter	Value	Best Trial
Maxepoch	100	89
Batchsize	[16,32,64,128,256]	128
Dropout rate	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]	0.6
Filters	[16, 32, 64, 128, 256]	32
Kernel size 0	[10, 7, 5, 4, 3, 2]	7
Kernel size 1	[7, 5, 3, 2, 1]	5
Learning rate	[0.01, 0.001, 0.0001]	0.001

4.5.2 提案手法のハイパーパラメータチューニング

また、表 9 と表 10 に提案手法に対する探索範囲とパラメータチューニング結果を示す。それぞれの Kernel size の探索範囲は既存手法と同様にそれぞれのリンク数に従って決定している。走行時間のリンク数は 5、停車時間のリンク数は 4 である。

表 9: 提案手法 走行時間: パラメータチューニング結果

Parameter	Value	Best Trial
Maxepoch	100	34
Batchsize	[16,32,64,128,256]	256
Dropout rate	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]	0.2
Filters	[16, 32, 64, 128, 256]	32
Kernel size 0	[5, 4, 3, 2]	5
Kernel size 1	[5, 4, 3, 2]	3
Learning rate	[0.01, 0.001, 0.0001]	0.01

表 10: 提案手法 停車時間: パラメータチューニング結果

Parameter	Value	Best Trial
Maxepoch	100	86
Batchsize	[16,32,64,128,256]	128
Dropout rate	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]	0.7
Filters	[16, 32, 64, 128, 256]	16
Kernel size 0	[4, 3, 2]	4
Kernel size 1	[4, 3, 2]	2
Learning rate	[0.01, 0.001, 0.0001]	0.01

4.6 評価結果

4.6.1 1 週間の評価結果

2020 年 9 月 24 から 2020 年 9 月 30 日までの 1 週間の到着時刻予測精度の比較結果 ($t+1$ 、 $t+2$ 、 $t+3$) を表 11 に示す。また、図 17 に全ての手法の $t+1$ における平均二乗誤差と平均絶対誤差を棒グラフで示す。 $t+1$ を対象に既存手法を見ると Historical Average が最も精度が悪く、次に PureLSTM、ConvLSTM という順序になった。今回の対象路線は Historical Average のような過去の運行情報を利用した単純な回帰予測よりも Petersen ら [7] の LSTM をベースにした手法の方が精度が高くなっていることが分かる。さらに、彼らの主張通り単純な PureLSTM よりも Convolutional LSTM を用いた方がより高い精度で予測できていることが分かる。以上のことから、今回の評価対象路線は都市部のバス路線が持つ複雑性を持っており、回帰をベースとした手法では精度の高い予測が難しいと考えられる。以上のことから本研究における評価対象路線としての妥当性が認められた。

次に提案手法の 3 つの手法では全て既存手法で最も精度が高かった ConvLSTM の精度を上回っていることが分かる。提案手法同士を比較すると、最も精度が高かったのが Model2_Bi、続いて Model2、最後に Model1 という結果となった。これらの結果は 4.4 節で述べた仮説が正しいことを示している。Model1 と ConvLSTM を比べると、Model1 の精度の方が高いことから、気象情報が特徴量として有効であることが分かる。また、Model1 と Model2 では、Model2 の精度の方が高いことから、走行時間と停車時間を別々に予測する手法の有効性を示した。そして、最後に Model2 と Model2_Bi を比較すると、Model2_Bi の精度の方が高いことは、提案手法において、Bidirectional LSTM が有効であることを示している。提案手法である Model2_Bi の ConvLSTM からの平均絶対パーセント誤差の減少幅は $t+1$ から順に 1.36%、0.864%、0.72% と全ての *outtime_steps* で誤差が減少した。特に、 $t+1$ での減少幅が最も大きかった。また、平均二乗誤差は 0.757 分、平均絶対誤差は 0.589 分減少した。

表 11: 1 週間の到着時刻予測精度の比較 (2020 年 9 月 24 日-9 月 30 日)

Model	Time ahead	RMSE [min]	MAE [min]	MAPE [%]
Historical Average	t + 1 (60 min)	3.820	2.900	7.987
PureLSTM	t + 1 (60 min)	3.711	2.730	7.178
PureLSTM	t + 2 (120 min)	3.774	2.816	7.426
PureLSTM	t + 3 (180 min)	3.772	2.824	7.524
ConvLSTM	t + 1 (60 min)	3.637	2.643	6.944
ConvLSTM	t + 2 (120 min)	3.708	2.720	7.176
ConvLSTM	t + 3 (180 min)	3.672	2.706	7.211
Model1	t + 1 (60 min)	3.438	2.454	6.492
Model1	t + 2 (120 min)	3.526	2.587	6.887
Model1	t + 3 (180 min)	3.541	2.567	6.891
Model2	t + 1 (60 min)	3.093	2.239	6.055
Model2	t + 2 (120 min)	3.269	2.452	6.633
Model2	t + 3 (180 min)	3.302	2.452	6.672
Model2_Bi	t + 1 (60 min)	2.880	2.054	5.584
Model2_Bi	t + 2 (120 min)	3.156	2.326	6.312
Model2_Bi	t + 3 (180 min)	3.232	2.381	6.491

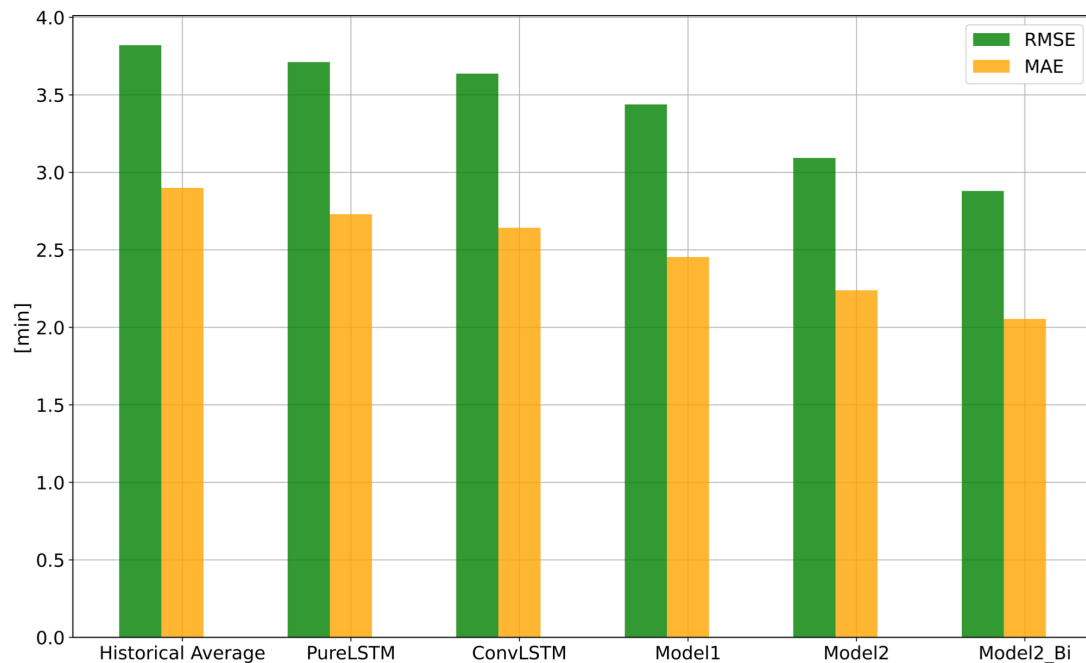


図 17: 1 週間 (2020 年 9 月 24 日-9 月 30 日) の RMSE と MAE の精度比較 ($t + 1$)

4.6.2 ピークの時間帯の予測精度の比較

Mugion らの [1] の研究や Petersen ら [7] の研究などで都市部の路線バスの中でも特に朝、夕のピーク時などは渋滞などが発生しやすいため、特に予測が難しく、精度が落ちることが明らかになっている。

そこで本研究ではピークの時間帯に限定して予測精度の比較を行う。図 15 における標準偏差の大きさから 9:00~11:00 を午前のピーク、18:00 を午後のピークとして定義した。平日の午前と午後それぞれのピークの時間帯に対する $t + 1$ における予測値の精度比較結果を表 12、表 13 に示す。1 週間全体の評価結果と比べると、午前のピークは全ての手法での予測精度が低下しており、午前のピーク

の時間帯における予測が他の時間帯よりも難しいことを示している。一方で午後のピークは全ての手法で1週間全体の予測精度よりも高い精度となった。提案手法である Model2_Bi と ConvLSTM を比較すると午前のピークにおける平均絶対パーセント誤差は 2.418%、午後のピークにおいては 3.095% それぞれ減少した。双方とも 1 週間全体の比較を行った表 11 での $t+1$ の減少幅である 1.36% よりも減少幅が大きいことが分かる。

表 12: 午前ピーク 09:00-11:00(平日のみ)

Model	RMSE [min]	MAE [min]	MAPE [%]
Historical Average	5.872	4.320	10.000
PureLSTM	5.399	4.010	9.324
ConvLSTM	5.341	4.043	9.453
Model1	4.998	3.760	8.804
Model2	3.996	2.982	7.143
Model2_Bi	3.751	2.940	7.035

表 13: 午後ピーク 18:00(平日のみ)

Model	RMSE [min]	MAE [min]	MAPE [%]
Historical Average	2.948	2.640	6.755
PureLSTM	2.825	2.469	6.335
ConvLSTM	2.298	2.101	5.414
Model1	2.234	1.801	4.624
Model2	1.444	1.339	3.545
Model2_Bi	1.209	0.845	2.319

4.6.3 雨によって運行が乱れた日の評価

本研究で特徴量として用いた気象情報が到着時刻の予測精度に及ぼす影響を評価するために、まず天候による運行の乱れが発生した場合での予測精度の比較を行う。その対象として、2020年9月25日(金)を選定した。この日は0:00から18:00ごろまで断続的に降雨があった。それが一因となって運行が大きく乱れていると考えられるため、この日を対象として選定した。図18に当日の観測された1時間ごとの運行所要時間と過去の平均所要時間、降雨量を示す。左の縦軸が所要時間、右の縦軸は降雨量を示す。朝の6:00から11:00まで、大きく遅延が起きていることが分かる。その後、12:00から13:00にかけて通常の運行に回復した後、再び大きな遅延が18:00まで続いている。このように、降雨を1つの要因として、明らかに普段とは異なる運行の乱れが観測されている。

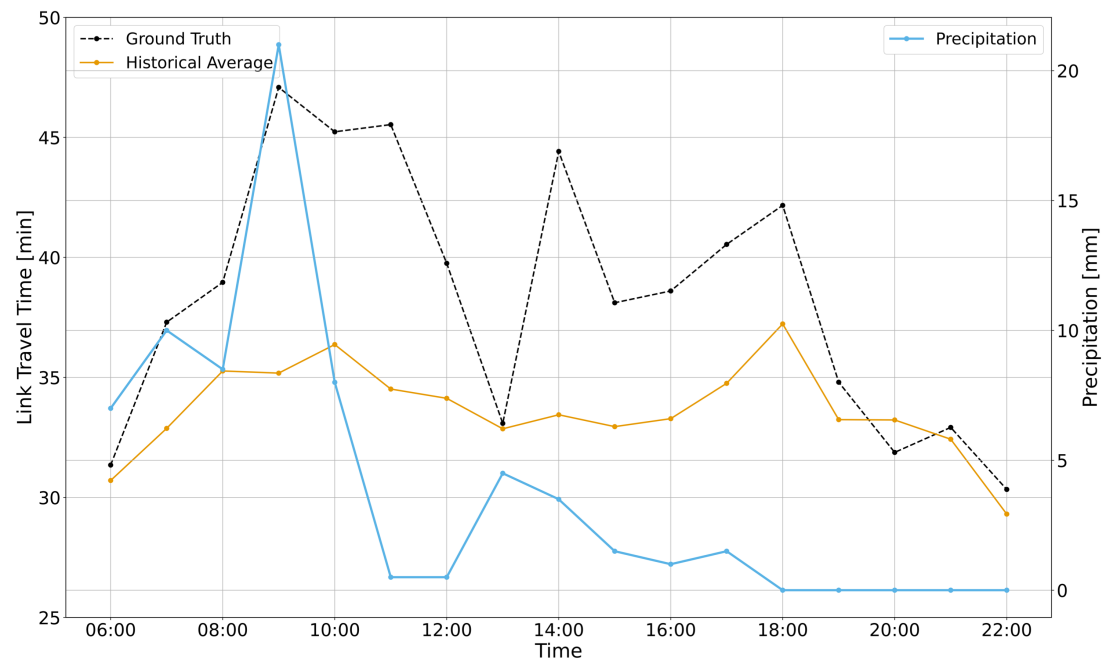


図 18: 2020/9/25 の所要時間と過去の平均所要時間

2020 年 9 月 25 日の $t+1$ の予測結果における 1 日全体の到着時刻予測精度の比較結果を表 14 に、図 19 に 1 日の中での予測精度の推移をそれぞれ示す。1 週間全体との比較はもちろん、4.6.2 節で評価した午前のピークでの $t+1$ の予測精度よりも予測精度が低くなっていることからこの日の予測は非常に難しいと考えられる。1 日全体で見ると、提案手法である Model2_Bi が最も精度が高いという結果となった。ConvLSTM と比べると平均絶対パーセント誤差が 3.163% 減少していることが分かる。これは、これまでの比較の中で最も大きい減少幅である。1 日の流れを見ると、午前のピークの一部である 10:00、11:00、12:00 の予測値に関しては従来手法ではいずれも遅延を過小評価しているが、提案手法では、より実測値に近い値で予測できていることが分かる。その一方で 13:00 で 12:00 の遅延が発生した状態から急激に回復した場合では、回復することは予測できているが、その回復を過小評価してしまい、他の手法より誤差が大きくなってしまっている。午後のピークである 18:00 での予測結果に関しても提案手法での予測値が最も実測値に近くなっている。15:00 から午後のピークである 18:00 までの期間では、実測値に近い予測ができていることが分かる。しかし、ここでも 18:00 から 19:00 の急激な回復を過小評価している。しかしながら 1 日全体で見ると、提案手法は既存手法よりも優れていると言える。

表 14: 2020/9/25(金) の精度比較

Model	RMSE [min]	MAE [min]	MAPE [%]
Historical Average	6.163	4.883	11.701
PureLSTM	5.544	4.411	10.584
ConvLSTM	5.389	4.169	9.973
Model1	5.183	4.048	9.726
Model2	4.246	3.167	7.859
Model2_Bi	3.885	2.717	6.810

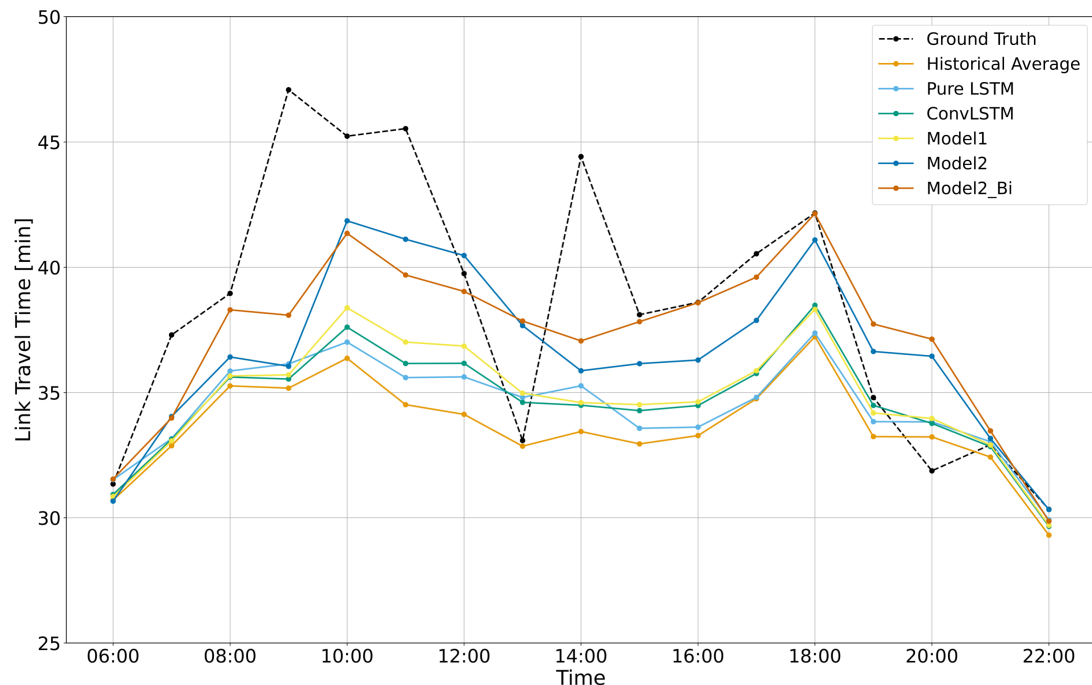


図 19: 2020/9/25(金) の到着時刻予測の比較

4.6.4 天候は安定しているが、運行が一部乱れた日における評価

天候が安定した日における精度の比較を行うことで、提案手法の汎用性を評価する。まず天候とは無関係であるが、一部運行が乱れている日を対象にして評価を行う。その対象として、2020年9月26日(土)を選定した。この日は1日を通して曇りではあるが降雨がなく、天候が安定していた。運行に関しても前日の2020年9月25日(金)と比べると比較的安定した運行が観測されているが、7:00や10:00など午前中は一部、大きな乱れが発生している。表15に1日の予測精度の比較結果、図20に1日における予測精度の推移をそれぞれ示す。表を見ると、従来手法においても前日の9月25日に比べると誤差がかなり小さくなっていることが分かる。また、Historical Averageでも平均絶対パーセント誤差が11.7%から7.657%と減少していることから、9月25日にくらべると過去の平均に近い土

曜日の運行であると言える。既存手法と提案手法の精度を比較すると、提案手法はこのような天候とは無関係ではあるが、運行が乱れているような場合においても既存手法よりも高い精度で予測できていることが分かる。Model2 と Model2 Bi を比較すると、平均絶対パーセント誤差は 0.162% と僅かではあるが Model2 の方が低くなっている。その一方で平均二乗誤差 (RMSE) は Model2 Bi の方が 0.019 分と僅かではあるが精度が高くなっている。

表 15: 2020/9/26(土) の精度比較

Model	RMSE [min]	MAE [min]	MAPE [%]
Historical Average	3.905	2.928	7.657
PureLSTM	3.457	2.588	6.787
ConvLSTM	3.428	2.446	6.390
Model1	2.911	2.026	5.290
Model2	2.177	1.584	4.269
Model2_Bi	2.158	1.647	4.431

4.6.5 天候と運行が安定した日における評価

次に天候と運行ともに安定していた 1 日に対する評価を行う。その対象として、2020 年 9 月 29 日 (火) を選定した。この日は天候も安定しており、運行に関してこれまで評価した 2 日に比べると比較的安定した運行が観測されている。表 16 に 1 日の予測精度の比較結果、図 21 に 1 日における予測精度の推移をそれぞれ示す。表を見ると、従来手法においても 9 月 25 日 (金)、2020/9/26(土) に比べると誤差が小さくなっていることが分かる。また、Historical Average でも平均絶対パーセント誤差が 6.535 とこれまでの 2 日より誤差が少ないことから、それらと比べると平均的な予測のしやすい火曜日の運行であると言える。また、全期間の平均を図示している図 15 と比べると、所要時間の推移に類似性が見られることが分かる。1 日全体での提案手法の精度を比較すると、かなり拮抗していることが分かる。平均絶対パーセント誤差は Model1 が最も良かったが、他の 3 つの

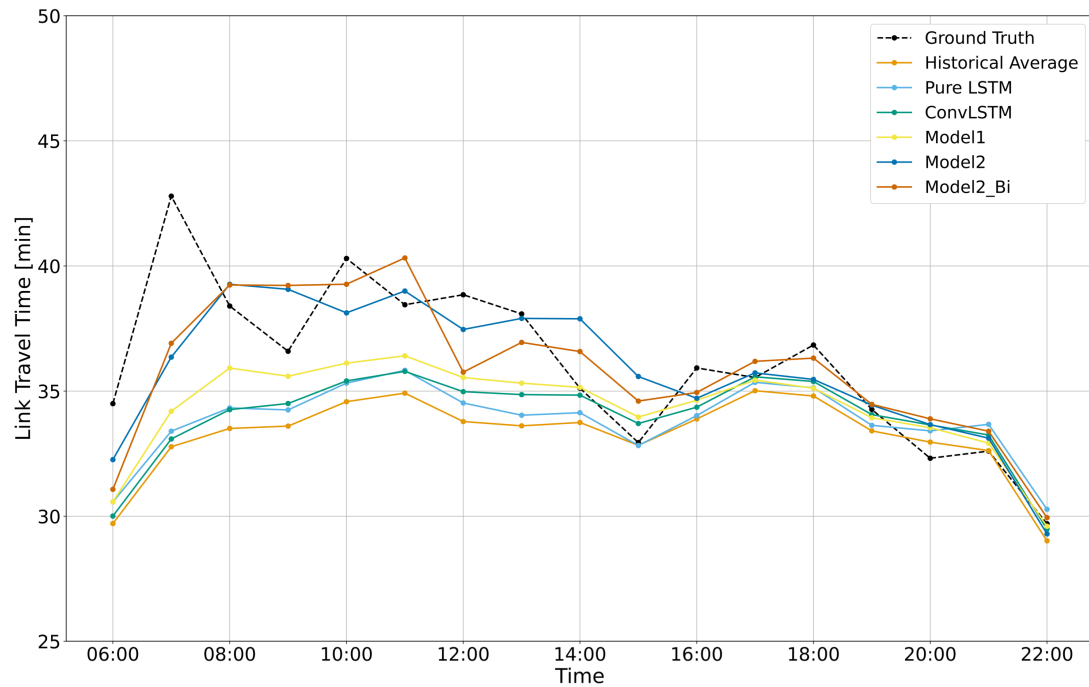


図 20: 2020/9/26(土) の比較

評価指標に関しては提案手法である Model2_Bi が良い精度であった。以上のことから、総合的に評価すると Model2_Bi が最も優れた予測ができていると言える。また、天候、運行ともに安定しているような場合においても提案手法が既存手法よりも有効であると言える。

表 16: 2020/9/29(火) の精度比較

Model	RMSE [min]	MAE [min]	MAPE [%]
Historical Average	2.726	2.380	6.535
PureLSTM	2.428	2.003	5.497
ConvLSTM	2.408	2.011	5.522
Model1	2.147	1.654	4.575
Model2	2.082	1.680	4.799
Model2_Bi	2.020	1.631	4.634

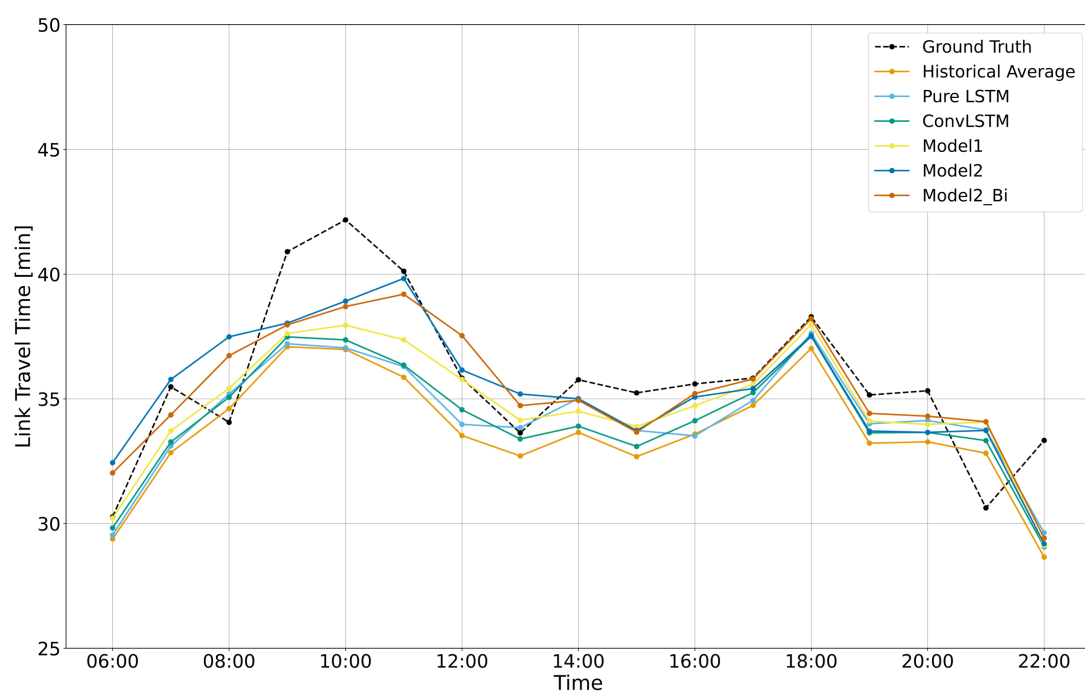


図 21: 2020/9/29(火) の比較

5. 考察

第4章の評価実験では、神戸市の市街地を走るバス路線の9月末の1週間を対象に精度の比較評価を行った。その中で様々な条件別に計6つの評価を行った。本章では、評価実験で得られた結果から考察を述べ、今後の展望についても述べる。

5.1 予測対象区間の長さに基づく比較

今回の評価実験で対象となった路線はバス停数が6の路線であるため、合計5つの区間がある。また、始点から終点の区間において時刻表で予定されている所要時間は35分である。しかし、神戸ベイシェラトンホテル-神戸三宮間が20分、神戸三宮-新神戸駅間が10分と最後の2区間が35分のほとんどを占めている。これまでの評価のように始点から終点までの合計で評価した場合、前半の誤差と後半の誤差で相殺されて精度が高くなっている可能性が考えられる。そのため、本節では経路の長さに基づいて評価を行うことで、提案手法が既存手法よりも有効であることの妥当性を示す。評価した路線には全部で9のリンクが存在する。1週間全体の経路の長さに基づいた平均絶対誤差と平均二乗誤差の比較を図22図23にそれぞれ示す。横軸は最初のリンクから0_1、1_2、...、8_9という表現に基づいてリンクの長さを示している。例えば0_2は始点からの2つのリンクを表しており、0_9は始点から終点までの9つのリンクの長さを表している。図から、平均絶対誤差、平均二乗誤差ともに0.6の区間までほとんど差がないことが分かる。しかし、最後の2区間では既存手法と比べて、誤差が減少していることが分かる。

次に1週間全体を対象に最初の3区間と最後の2区間を分けて精度評価を行う。最初の3区間である神戸国際大学前-神戸シェラトンホテル間における1週間の予測精度の比較を表17に示す。この3区間にはそれぞれ1回ずつ走行時間と停車時間が含まれる。同様に、最後の2区間である神戸シェラトンホテル-新神戸駅間における1週間の予測精度の比較を表18に示す。この2区間には2回の走行時間と1回の停車時間が含まれる。この2つの表を元に考察を行う。まず、神戸国際大学前-神戸ベイシェラトンホテル間では、全ての手法で拮抗した精度となった。この区間は5分の所要時間が予定されているが、その中で、平均二乗誤差が最も小さ

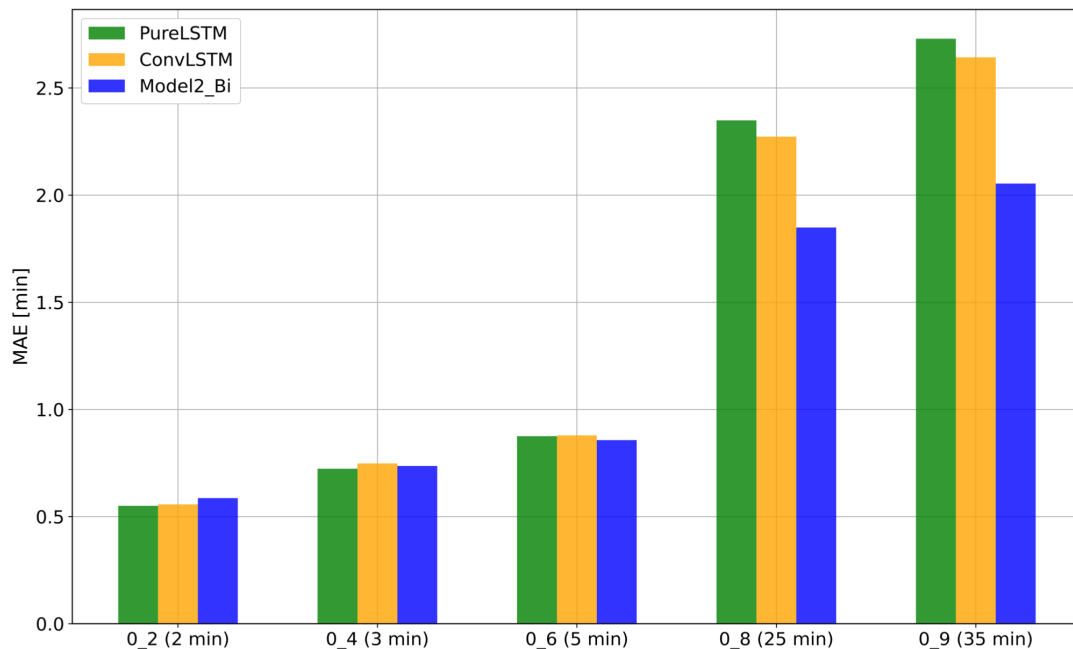


図 22: 経路の長さに基づく平均絶対誤差の比較 (2020 年 9 月 24 日-9 月 30 日)

かったのは Model2_Bi の 1.305 分である。しかし、全ての手法で 1.3 台の為、明確な精度の差はないと言える。その他の評価指標についても同様である。これは、3 区間の予定所要時間がそれぞれ 2 分、1 分、2 分と非常に短いことが原因として挙げられる。それぞれの予定所要時間には、走行時間と停車時間が含まれる。本来であれば、交通状況や乗降者数などが走行時間、停車時間にそれぞれ大小様々な影響を与える。提案手法では、天候情報と過去運行データを用いて、その影響のパターンを学習している。しかし、この 3 区間はそれぞれの走行時間と停車時間が短いため、その影響が極めて小さいと考えられる。そのため、手法によって精度に大きな差が見られなかったと考える。

次の神戸ベイシェラトンホテル-新神戸駅間では、最初の 3 区間よりも明確な

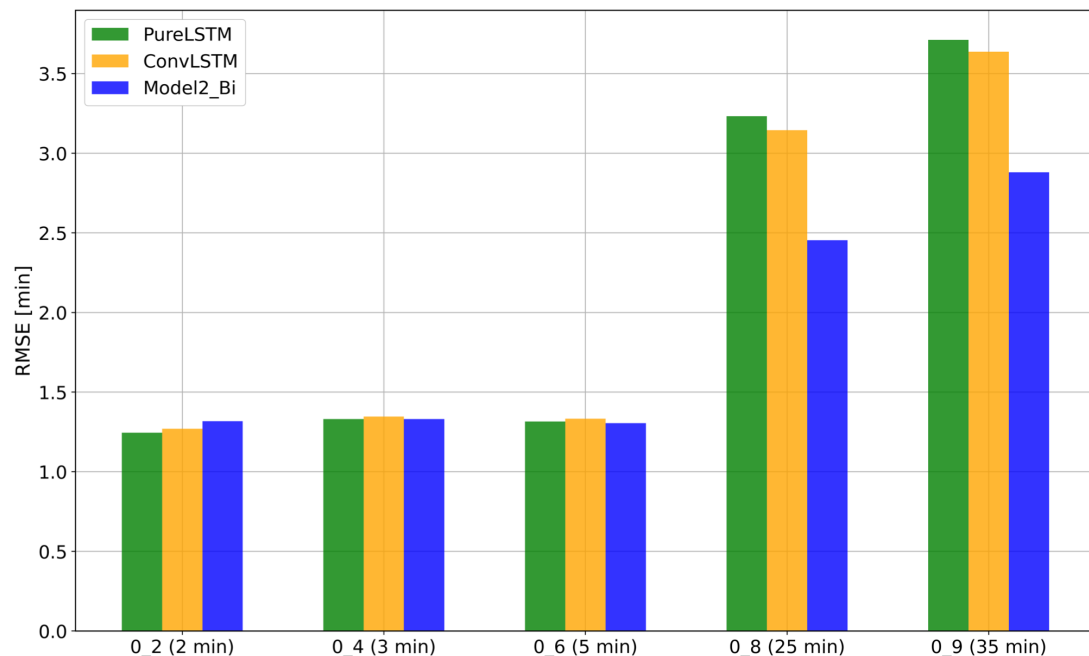


図 23: 経路の長さに基づく平均二乗誤差の比較 (2020 年 9 月 24 日-9 月 30 日)

精度の差があることが分かる。最も優れた精度なのは提案手法である Model2_Bi であった。ConvLSTM と比較すると、平均二乗誤差は 0.559 分、平均絶対誤差は 0.394 分、平均絶対パーセント誤差は 1.034% 上昇した。また、Model1 も既存手法よりも高い精度であるため、走行時間と停車時間を別々に予測することが有効であることが分かる。Model2、Model2_Bi についても同様に Bidirectional LSTM の有効性を示している。

前半の 3 区間に関しては大きな精度の向上は見られなかったが、後半の 2 区間では明確な精度の向上が確認できた。また、区間の長さに基づいて比較評価を行うことで提案手法の汎用性を確認できた。

表 17: 1 週間の到着時刻予測精度の比較 (神戸国際大学前-神戸ベイシェラトンホテル)

Model	RMSE [min]	MAE [min]	MAPE [%]
PureLSTM	1.315	0.875	12.207
ConvLSTM	1.333	0.879	12.252
Model1	1.372	0.930	13.748
Model2	1.391	0.882	12.670
Model2_Bi	1.305	0.857	12.406

表 18: 1 週間の到着時刻予測精度の比較 (神戸ベイシェラトンホテル-新神戸駅)

Model	RMSE [min]	MAE [min]	MAPE [%]
PureLSTM	3.288	2.374	7.627
ConvLSTM	3.180	2.270	7.287
Model1	2.997	2.137	6.898
Model2	2.721	1.930	6.408
Model2_Bi	2.621	1.876	6.253

5.2 提案手法の比較

まず、提案手法の比較を行う。各評価における 3 つの提案手法それぞれの平均絶対パーセント誤差をまとめたものを表 19 に示す。一番右の列の減少幅は、基本的には Model1 から Model2_Bi の平均絶対パーセント誤差の減少幅を示す。下の 2 行に関しては、最も誤差が小さいのが Model2_Bi ではないため、最も誤差が小さいモデルと Model2_Bi との差を示している。

提案手法の精度を比較すると、下の 2 行の 9/26(土)、9/29(火) 以外は Model2_Bi の誤差が最も小さいことが分かる。9/26(土)、9/29(火) で最も誤差が小さかったのはそれぞれ Model2 と Model1 であった。しかしながら、Model2_Bi との差は 0.162%、0.059% と他の減少幅と比べると極めて小さいことから、全体的に評価

すると、Model2_Bi が最も優れていると言える。

表 19: 提案手法の平均絶対パーセント誤差の比較

評価対象	Model1 [%]	Model2 [%]	Model2_Bi [%]	減少幅 [%]
1 週間	6.492	6.055	5.584	0.908
午前ピーク	8.804	7.143	7.035	1.769
午後ピーク	4.624	3.545	2.319	2.305
9/25(金)	9.726	7.859	6.810	2.916
9/26(土)	5.290	4.269	4.431	0.162
9/29(火)	4.575	4.799	4.634	0.059

5.3 既存手法との比較

既存手法の中で最も精度が優れていた ConvLSTM と提案手法である Model2_Bi の平均絶対パーセント誤差とその減少幅を条件毎にまとめたものを表 20 に示す。また、別々に評価を行った 3 日間それぞれの天候状態、運行の乱れ、HistoricalAverage の平均絶対パーセント誤差をまとめたものを表 21 に示す。HistoricalAverage の平均絶対パーセント誤差は、過去の平均との差を表している。この値が大きければ大きいほど、過去の平均所要時間から大きく外れていると捉えることができる。

5.3.1 午後のピーク、9/25(金) に対する考察

この表から、減少幅が特に大きかったのが午後のピークと 9/25 日 (金) であることが分かる。いずれも 3% ほどの誤差減少が見られた。まずこの結果に対する考察を行う。午後のピークである平日 18:00~18:59 の時間帯の減少幅は 2 番目に大きい 3.095% であった。18:00 の予測を行う際、10:00~17:00 を入力として予測を行っている。特に朝のピークの時間帯の天候と運行状態は午後のピークの時間帯である 18:00 の運行に大きな影響を与えていると考える。提案手法では、その

表 20: 既存手法との平均絶対パーセント誤差 (MAPE) の比較

評価対象	ConvLSTM [%]	Model2_Bi [%]	減少幅 [%]
1 週間	6.944	5.584	1.360
午前ピーク	9.453	7.035	2.418
午後ピーク	5.414	2.319	3.095
9/25(金)	9.973	6.810	3.163
9/26(土)	6.390	4.431	1.959
9/29(火)	5.522	4.634	0.888

表 21: 日毎の天候状態、運行の乱れ

日付	天候	運行の乱れ	Historical Average [%]
9/25(金)	雨	大	11.701
9/26(土)	曇り	中	7.657
9/29(火)	曇り	小	6.535

ような変動パターンを天候情報を加えることで正しく学習できているため、大きな誤差減少につながったと考えられる。

今回の評価実験で減少幅が最も大きかったのが9/25日(金)である。この日は1日を通して降雨により、運行が著しく乱れた特に予測の難しい例である。既存手法の中で最も精度が良かったConvLSTMでも平均絶対パーセント誤差が9.973%と、1週間全体の精度よりも大きく精度が落ちていることが分かる。その他の手法では共に10%を上回っている。提案手法ではこのような場合において特に精度の向上がみられ、平均絶対パーセント誤差が3.163%減少した。これも午後のピークと同様に天候状態に起因する運行の乱れのパターンを、天候情報を含めることで効果的に学習ができているためであると考えられる。

5.3.2 9/26(土)、9/29(火) に対する考察

次に9/26(土)、9/29(火) に対する考察を行う。これらの評価を行った意図としては、天候によって運行が乱れた日だけでなく、天候が安定している日に限定して評価を行うことで、天候情報を用いた提案手法の汎用性を確かめるためである。9/26(土) は天候は安定しているが、一部運行が大きく乱れた例である。既存研究の ConvLSTM でも 6.390% と 9/25(金) に比べると、平均絶対パーセント誤差が 3.583% 低い。この例では Model2_Bi は 1.959% の減少であった。9/25(金) に比べると減少幅は小さいが、誤差の減少が見られた。

次の 9/29(火) は天候も運行も安定している例である。既存研究の ConvLSTM でも 5.522% と 9/25(金) に比べると、平均絶対パーセント誤差が 4.451% 低い。こちらも同様に乱れの大きかった 9/25(金) と比べると減少幅は小さいが、このような例でも、提案手法は 0.888% の減少が見られた。この 3 日間の評価結果から、天候が運行に与える影響の大きさにかかわらず、提案手法では精度が上昇していることが分かる。以上のことから、天候状態が与える影響が特に大きいと考えられる午後のピークの時間帯と、天候状態に起因して運行の乱れが普段よりも大きい場合に特に減少幅が大きくなる傾向があると言える。

5.4 今後の展望

本研究で提案した手法では、過去の運行情報と天候情報を元に路線バスの到着時刻を予測している。評価結果を全体的に見ると提案手法は有効であると言える。しかし、前の時間帯から急激に所要時間が減少する場合において、減少自体は予測できているが、その減少幅を過小評価してしまう場合がいくつか確認できた。このような急激な減少が予測されている場合は他の手法の予測結果を適用することで対応できる可能性があると考ええる。また、今回は 1 つの路線を対象に評価実験を行ったが、対象となる路線と同じ地域の他のバス路線や公共交通機関の過去運行データや乗降者データとは相関関係にある可能性がある。そのようなデータを組み合わせることは、さらに予測精度を安定させる可能性がある。また別のアプローチとして交差点や信号の数などの道路情報、地域の詳細な道路情報を含む

ETC2.0 など他の特徴量を加えることでさらに精度向上の可能性があると考える。提案手法では、過去に観測された天気、降雨量、気温の3つの天候情報を特徴量として用いた。特に、天気、降雨量に関しては予報情報を特徴量として用いることが更なる予測精度向上に有効である可能性がある。本研究では、既存手法と比較するためにある $reference_frequency = 60$ の運行データを元に学習及び予測を行った。また、既存手法と比較するために $input_timesteps = 8$ 、 $output_timesteps = 3$ で評価実験を行ったが、このパラメータをチューニングすることで更なる精度の向上が見込める。評価対象路線として、1日を通して一定の頻度で運行する時刻表を持つ路線を対象とした。今後の課題として便毎に学習、予測できるように、さらに時刻表ベースの路線バスに最適化する必要があると考える。そのためには便の運行間隔によって、どのくらいの数の便を元に予測を行うかを動的に決定する手法が必要であると考ええる。

6. おわりに

現代の生活において公共交通機関は毎日の通勤や通学、観光など様々な用途で利用されており、都市部、地方に拘らず、重要な役割を果たしている。また、予測の難しい路線バスにおいて到着時刻予測の精度は重要である。これまでに提案されたバス到着時刻手法の中で最も有効なのは Convolutional LSTM を用いた手法である。しかし、この手法には時刻表ベースの路線の特徴を考慮していない点や天候に伴って運行が大きく乱れた際に大きく精度が低下するといった問題点が存在する。本稿では天候情報が予測精度の向上に対して最も費用対効果が高いと考え、過去の運行情報に加えて天候情報を特徴量として用いた Convolutional LSTM ベースの到着時刻予測ネットワークアーキテクチャを提案した。また、提案手法の有効性を評価するために兵庫県神戸市の全長約 15 km の路線を対象に評価実験を行い、Convolutional LSTM を用いた既存手法と比較した。その結果、1 週間全体で見ると、1.36% の平均絶対パーセント誤差の減少が見られた。特に夕方時間帯や、雨天時に運行が大きく乱れた場合には、約 3% と平均絶対パーセント誤差が 1 週間全体と比べて大きく減少することが確認できた。また、提案手法を 3 種類に分けて評価することで、時刻表ベースの路線において走行時間と停車時間を別々に予測することの有効性や、Bidirectional LSTM の有効性についても確認できた。今後の課題として、まず同じ地域の他のバス路線や公共交通機関のデータと組み合わせること、天気予報、道路状況を含む ETC2.0 など他の特徴量の利用で更なる精度の向上を目指す。また、本研究では既存手法と比較するために $input_timesteps = 8$ 、 $output_timesteps = 3$ で評価実験を行ったが、このパラメータをチューニングすることで更なる精度の向上が見込める。さらに便毎の学習や予測を行えるように、時刻表ベースのバス路線に対してさらに最適化することを検討する。

謝辞

主指導教員であり、適切な研究指導をしていただくなど様々な側面から研究のサポートをしてくださいました本学情報基盤システム学研究室の藤川和利教授に心から感謝致します。副指導教員であり、研究の方向性についての的確な助言をくださいました本学ユビキタスコンピューティングシステム研究室の安本慶一教授に心から感謝致します。副指導教員であり、研究指導、論文添削、システム開発など様々な面でサポートをしていただき、研究発表の機会も与えてくださいました本学情報基盤システム学研究室の新井イスミル准教授に心から感謝致します。休日や深夜の時間帯でも添削をしてくださったことにも深く感謝致します。研究指導だけでなく学内システムの運用などについても教えてくださいました本学情報基盤システム学研究室の垣内正年助教に心から感謝致します。研究活動や論文執筆に関して指導を戴きました本学情報基盤システム学研究室の油谷暁助教に心から感謝致します。また、研究活動を行うにあたり、必要なデータ提供などに快く協力してくださいましたみなと観光バス株式会社の皆様に心から感謝致します。さまざまな面から研究活動を支援してくださいました本学総合情報基盤センターの辻元理恵女史、中野彩子女史に心から感謝致します。様々な事務手続きだけではなく、研究室での生活の中でたくさんのコミュニケーションをとっていただきました。そして、さまざまな面から研究活動を支え、切磋琢磨しあった本学情報基盤システム学研究室の学生の皆様に心より感謝いたします。最後に、修士前期課程への進学にあたり、私の意思を尊重し、経済面や生活面での援助や励ましと共に6年間見守ってくださいました家族に心から感謝致します。

参考文献

- [1] R. G. Mugion, M. Toni, H. Raharjo, L. Di Pietro, and S. P. Sebatu, “Does the service quality of urban public transport enhance sustainable mobility?” *Journal of Cleaner Production*, vol. 174, pp. 1566–1587, 2018.
- [2] “The GTFS Realtime specification,” <https://developers.google.com/transit/gtfs-realtime/>, (Accessed on 2020-6-21).
- [3] O. Cats, H. N. Koutsopoulos, W. Burghout, and T. Toledo, “Effect of real-time transit information on dynamic path choice of passengers,” *Transportation Research Record*, no. 2217, pp. 46–54, 2011.
- [4] A. Gooze, K. Watkins, and A. Borning, “Benefits of real-time transit information and impacts of data accuracy on rider experience,” *Transportation Research Record*, no. 2351, pp. 95–103, 2013.
- [5] J. Patnaik, S. Chien, and A. Bladikas, “Estimation of bus arrival times using APC data,” *Journal of Public Transportation*, vol. 7, no. 1, pp. 1–20, 2004.
- [6] L. Zeng, G. He, Q. Han, L. Ye, F. Li, and L. Chen, “A LSTM based bus arrival time prediction method,” in *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019*, 2019, pp. 544–549.
- [7] N. C. Petersen, F. Rodrigues, and F. C. Pereira, “Multi-output bus travel time prediction with convolutional lstm neural network,” *Expert Systems with Applications*, vol. 120, pp. 426–435, 2019.
- [8] V. W. Stover and E. D. McCormack, “The impact of weather on bus ridership in Pierce County, Washington,” *Journal of Public Transportation*, vol. 15, no. 1, pp. 95–110, 2012.

- [9] M. Hofmann and M. O'Mahony, "The impact of adverse weather conditions on urban bus performance measures," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 2005, 2005, pp. 84–89.
- [10] "OpenWeatherMap API," <https://openweathermap.org/api>, (Accessed on 2020-10-30).
- [11] "Dark Sky API," <https://darksky.net/dev>, (Accessed on 2020-10-30).
- [12] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.
- [13] K. A. Althelaya, E. M. El-Alfy, and S. Mohammed, "Evaluation of bidirectional lstm for short-and long-term stock market prediction," in *2018 9th International Conference on Information and Communication Systems (ICICS)*, 2018, pp. 151–156.
- [14] J. V. Jeyakumar, E. S. Lee, Z. Xia, S. S. Sandha, N. Tausik, and M. Srivastava, "Deep convolutional bidirectional lstm based transportation mode recognition," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1606–1615.
- [15] M. Chen, X. Liu, and J. Xia, "Dynamic prediction method with schedule recovery impact for bus arrival time," *Transportation Research Record*, no. 1923, pp. 208–217, 2005.
- [16] J. Gong, M. Liu, and S. Zhang, "Hybrid dynamic prediction model of bus arrival time based on weighted of historical and real-time GPS data," in *2013 25th Chinese Control and Decision Conference, CCDC 2013*, 2013, pp. 972–976.

- [17] J. Pang, J. Huang, Y. Du, H. Yu, Q. Huang, and B. Yin, “Learning to Predict Bus Arrival Time from Heterogeneous Measurements via Recurrent Neural Network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3283–3293, 2019.
- [18] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human trajectory prediction in crowded spaces,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.
- [19] X. Qing and Y. Niu, “Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM,” *Energy*, vol. 148, pp. 461–468, 2018.
- [20] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” *Advances in Neural Information Processing Systems*, pp. 802–810, 2015.
- [21] N. P. Olewuezi, “Note on the comparison of some outlier labeling techniques,” *Journal of Mathematics and Statistics*, vol. 7, no. 4, pp. 353–355, 2011.
- [22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.
- [23] C. François, “Keras,” <https://keras.io>, (Accessed on 2020-1-18).
- [24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

- [25] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *32nd International Conference on Machine Learning, ICML 2015*, vol. 1, 2015, pp. 448–456.
- [26] G. Hinton, N. Srivastava, and K. Swersky, “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent,” vol. 14, no. 8, 2012.
- [27] “GTFS Static Overview,” <https://developers.google.com/transit/gtfs>, (Accessed on 2020-1-18).
- [28] 国土交通省, “自動車運送事業者が事業用自動車の運転者に対して行う一般的な指導及び監督の実施マニュアル,” https://www.mlit.go.jp/jidosha/anzen/03safety/resource/data/bus_honpen.pdf, (Accessed on 2020-1-19).
- [29] “国土交通省 気象庁 過去の気象データ,” <https://www.data.jma.go.jp/gmd/risk/obsdl/>, (Accessed on 2020-10-23).
- [30] M. Altinkaya and M. Zontul, “Urban Bus Arrival Time Prediction: A Review of Computational Models,” in *International Journal of Recent Technology and Engineering*, vol. 2, no. 4, 2013, pp. 1–6.
- [31] Y. Duan, Y. Lv, and F. Y. Wang, “Travel time prediction with LSTM neural network,” in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2016, pp. 1053–1058.
- [32] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A novel bandit-based approach to hyperparameter optimization,” *Journal of Machine Learning Research*, vol. 18, pp. 1–52, 2018.

発表リスト

国際会議

Atsuto Ishinaga, Ismail Arai and Kazutoshi Fujikawa, "Route-Bus Driver Evaluation System Using Digital Tachograph Data and Static Route Features," *2020 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT), Dubai, United Arab Emirates*, Dec. 2020.

国内会議

石長 篤人, 新井 イスマイル, 垣内 正年, 藤川 和利, “運行情報と気象情報の畳み込みによるバス到着時刻予測手法.”, 研究報告高度交通システムとスマートコミュニティ (*ITS*) , vol.2021-ITS-84, no.6, pp.1-8, Feb, 2021.