

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

石長 篤人^{1,a)} 新井 イスマイル² 垣内 正年² 藤川 和利²

概要：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。

キーワード：高度道路交通システム、機械学習、公共交通

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。しかし、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[2]。そのため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が、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。Petersenらは回帰ベースの手法[3]や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[4]を提案している。Convolutional LSTMは重み演算に畳み込み演算を用いることで、従来のLSTMよりも複数の運行に含まれるパターンを効果的に学習できることを示した。しかし、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、Convolutional LSTMの畳み込みフィルターとストライドを特徴量の数に合わせることで気象情報と過去運行データを同時に畳み込みを行う。これによって天候の乱れに対してロバストな予測を可能とする。次に走行時間と停車時間を別々のモデルで予測することで、走行時間

¹ 奈良先端科学技術大学院大学先端科学技術研究科
Graduate School of Science and Technology, Nara Institute
of Science and Technology, Ikoma, Nara 630-0192, Japan

² 奈良先端科学技術大学院大学総合情報基盤センター
Information Initiative Center, Nara Institute of Science and
Technology, Ikoma, Nara 630-0192, Japan

a) ishinaga.atsuto.ht1@is.naist.jp

と停車時間で時系列の特徴が異なるという時刻表ベースの路線の特徴を考慮する。停車時間は時刻表から大きな影響を受けると考え、時刻表に基づく特徴量を用いている。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。特に夕方や、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。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。さらに提案手法を3種類に分けて比較することで、気象情報と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を確認した。さらに精度向上のために使用したBidirectional LSTMの有効性についても確認した。

2. 関連研究

2.1 LSTM ベースのバス到着時刻予測

路線バスの到着時刻予測において、線形回帰などの回帰ベースの手法 [3] 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。Zeng らは LSTM を用いて予測直前の便の過去運行データを元に次の便の到着時刻を予測する手法 [5] を提案した。カルマンフィルタを用いた手法と比較した結果、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 LSTM をベースに過去運行データ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 [6] 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。

また、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 を用いて過去の複数の時間帯の運行情報の特徴量として到着時刻を予測する手法 [4] を提案した。また、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ている。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約1分の平均絶対誤差が減少できることを示した。さらに、運行が乱れやすいピークの時間帯では一週間全体と比べて大きく誤差が減少した。

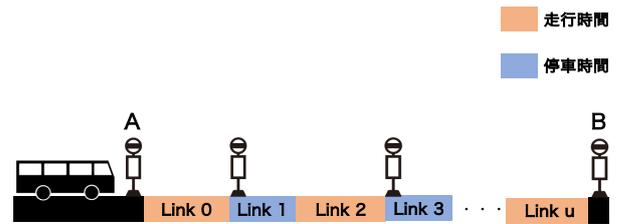


図1 バス停 A から B までの到着時刻の定義

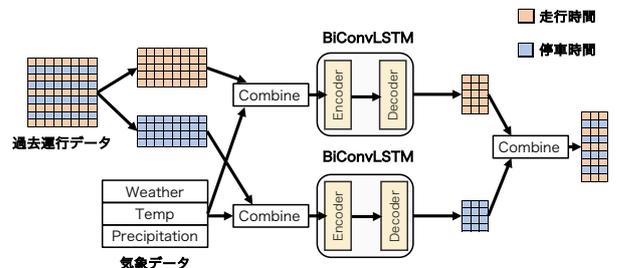


図2 提案手法の概要

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

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

3.1 提案手法の概要

本研究における到着時刻の定義を図1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 $LinkDuration$ 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。

$$Y = \sum_{ln=0}^u LinkDuration_{ln} \quad (1)$$

まず始めに提案手法の全体の流れを図2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に加えて、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。これは天候の乱れに起因する運行の乱れに対してロバストな予測を行うためである。これまでの研究で気象情報が路線バスの運行に直接影響を及ぼすことが示されている [7], [8] が、既存手法では特徴量として考慮されていない。天候情報を用いる利点としては、データ取得に

おけるコスト面での優位性が挙げられる。たとえば乗降者数はバス一台一台に専用機器が必要となるが、天気情報は Application Programming Interface(API) などで容易に取得できる。次に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 LSTM への入力としている。これは、時刻表に基づいた路線の特徴として、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。そのため、時刻表ベースの路線で走行時間と停車時間を同一の LSTM で学習すると、互いが持つ時系列の特徴量が損なわれてしまう可能性がある。これらを考慮するために、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て学習および予測を行う。また、停車時間には天気情報に加えて時刻表に基づく特徴量も用いている。さらに予測精度を向上させるアプローチとして、Bidirectional LSTM を使用している。Schuster らは、順伝搬のみを中間層を持つ従来の LSTM に対して順伝播と逆伝播の両方向の中間層を持つ Bidirectional LSTM[9] を提案した。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 [10], [11] が示されている。本研究も同様に時系列予測であるため、有効である可能性があると考えた。最後に Bidirectional LSTM 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。

3.2 提案手法におけるパラメータ

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3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。

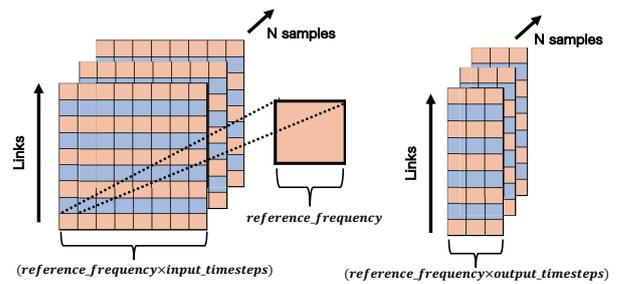


図3 過去運行データにおけるパラメータ

3.3 使用する特徴量

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。図4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。停車時間のみ、時刻表に基づいた特徴量である *DiffTime* を使用している。*DiffTime* は式2で導出される。*TimeTable_i* は対象バス停の時刻表における出発予定時刻、*StopStart_i* は対象バス停への到着時刻をそれぞれ示す。

$$DiffTime_i = TimeTable_i - StopStart_i \quad (2)$$

これは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、時刻表から大きな影響を受け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。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。

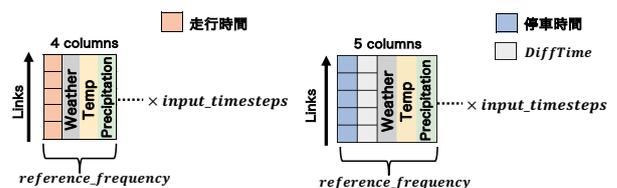


図4 走行時間と停車時間の各特徴量

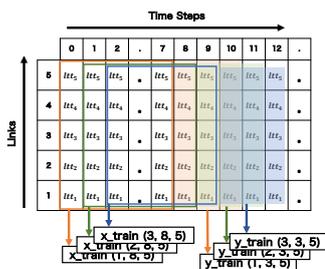


図 5 学習データセットの分割処理 ($input_timesteps = 8$, $output_timesteps = 3$)

3.4 前処理

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。 $input_timesteps = 8$, $output_timesteps = 3$ の場合におけるデータセットの分割処理の例を図 5 に示す。縦軸がリンク、横軸が時系列、 $l_{t,i}$ は各リンク所要時間を示している。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、各入力とそれに対応する出力のデータ構造を得る。

3.4.1 LinkDuration, DiffTime に対する処理

$LinkDuration$ と $DiffTime$ に関しては、それぞれを曜日、 $reference_frequency$ に基づく時間帯毎の統計値を元に式 3 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。

$$x'_{ln,t} = \frac{x_{ln,t} - \bar{x}_{ln,dow,tod}}{\sigma_{ln}} \quad (3)$$

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)[12] 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既存手法 [4] でも同様の方法で行われている。

3.4.2 Weather, Temp, Precipitation に対する処理

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 4 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。

$$x'_t = \frac{x_t - Q_2}{Q_3 - Q_1} \quad (4)$$

3.5 バス到着時刻予測ネットワークアーキテクチャ

図 6 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 らが提案したアーキテクチャ [4] を元に設計した。実装には Tensorflow 2.1.0 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 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 Stacked 2-Layer Convolutional LSTM で構成されている。また、過学習を防ぐために 4 層の各 LSTM 層間に Dropout 層 [13] と Batch Normalization 層 [14] を使用している。変更を加えた点は以下の 2 点である。まず、エンコーダの 1 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、アーキテクチャ全体の構造について述べる。まず、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。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。その後、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。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, ロス関数には平均二乗誤差 (MSE) を用いる。

4. 評価実験

4.1 評価環境

兵庫県神戸市中心部の神戸国際大学前から新神戸駅までの約 15 km を経路を持つバス路線を評価対象として選定し

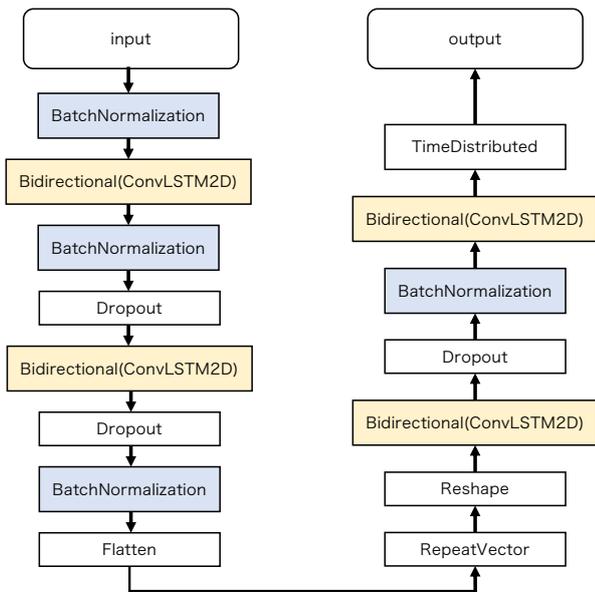


図 6 Bidirectional Convolutional LSTM 到着時刻予測モデル

た。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。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
- (3) 終点から他の公共交通機関への乗り換えが可能

次に評価実験における提案手法のパラメータについて述べる。この路線は 6:00 から 23:00 まで 30 分もしくは 1 時間ごとに便を持つ路線であるため、最低でも 1 運行のサンプルが存在するように *reference-frequency* は 60 分とした。また残り 2 つのパラメータ *input_timesteps*, *output_timesteps* は既存手法と比較評価するために既存手法に合わせてそれぞれ 8 と 3 に設定した。既存手法の評価実験においても直前 8 時間の情報を元に学習および予測を行っている。図 7 に今回の評価で使用した 2020 年 1 月から 9 月末までの各時間帯毎の始点から終点までの平均所要時間とその標準偏差を示す。赤の補助線はこの路線の予定所要時間である 35 分、黒点が平均所要時間、その上下のエラーバーが標準偏差を表している。9:00~11:00 はその前後よりも平均所要時間が長く、前後の時間帯と比べて標準偏差も大きいので午前のピークとして捉えることができる。同様の理由で 18:00 を午後のピークとして定義する。特にこれらの時間帯は他の時間帯よりも運行が不安定であるため、予測が難しいと考えられる。

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。気象データは気象庁が公開している過去の気象データ [15] を使用した。評価対象路線

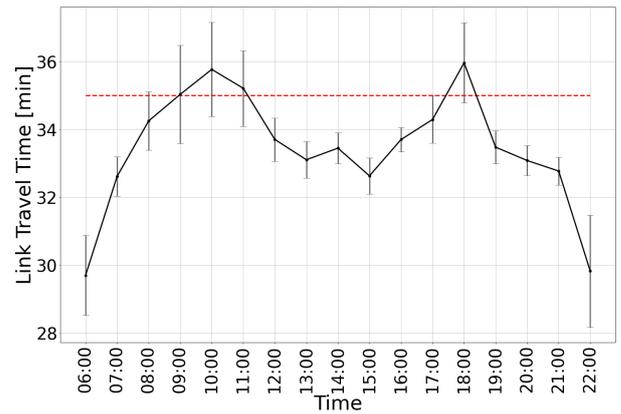


図 7 時間帯ごとの平均所要時間と標準偏差

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。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 種類に整形した。快晴、晴れ、薄曇を晴れ、雨、雷、雪、みぞれ、にわか雨を雨としてそれぞれまとめた。

4.2 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とで評価を行う。式 5 の Y_i は実際の所要時間、式 6 の \hat{Y}_i は予測値を表す。

$$Y_i = \sum_{ln=0}^u Y_{i,ln} \quad (5)$$

$$\hat{Y}_i = \sum_{ln=0}^u \hat{Y}_{i,ln} \quad (6)$$

評価指標は比較対象である Petersen らの手法 [4] に合わせて以下の 3 つの評価指標 (式 7~9) 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。

$$RMSE(Y, \hat{Y}) = \sqrt{\frac{\sum_{i=0}^N (Y_i - \hat{Y}_i)^2}{N}} \quad (7)$$

$$MAE(Y, \hat{Y}) = \frac{\sum_{i=0}^N |Y_i - \hat{Y}_i|}{N} \quad (8)$$

$$\text{MAPE}(Y, \hat{Y}) = \frac{1}{N} \sum_{i=0}^N \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right| \quad (9)$$

4.3 比較対象

本研究では、以下の3つの既存手法と3つの提案手法、計6手法を比較対象とした。

HA(Historical Average)[3]

曜日、時間帯毎の平均値をそのまま予測値として用いるモデル

PureLSTM[4]

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 [4] である。

ConvLSTM[4]

上記と同様に Petersen らが提案している Convolutional LSTM を用いた手法 [4] である。

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。

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。

4.4 評価結果

4.4.1 1週間の評価結果

2020年9月24日から2020年9月30日までの1週間の *output_timesteps* = 1 における到着時刻予測精度の比較結果を表1に示す。以降、全て *output_timesteps* = 1 における予測結果を対象に比較を行う。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 [4] 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。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 において Model2 の精度の方が高いことは、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手法が有効であることを示している。そして、最後に Model2 と Model2_Bi を比較すると、Model2_Bi の精度の方が高かった。これは提案手法において Bidirectional LSTM が精度向上に有効であることを示している。

表1 1週間の予測精度の比較 (2020年9月24日-9月30日)

Model	RMSE[min]	MAE[min]	MAPE[%]
HA	3.820	2.900	7.987
PureLSTM	3.711	2.730	7.178
ConvLSTM	3.637	2.643	6.944
Model1	3.438	2.454	6.492
Model2	3.093	2.239	6.055
Model2_Bi	2.880	2.054	5.584

4.4.2 ピークの時間帯の予測精度の比較

これまでの研究 [1], [4] で都市部のバス路線の中でも特に朝、夕のピーク時は渋滞などが発生しやすいため、特に予測が難しい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。

平日の午前と午後それぞれのピークの時間帯における予測値の精度比較結果を表2に示す。提案手法はこのような場合に特に有効で、1週間全体と比べると減少幅が大きくなっている。これはピークの時間帯が特に天候などの影響を受けやすく、提案手法ではそのパターンを効果的に学習できていることを示している。

表2 ピーク時間帯の評価: 精度比較

評価対象	Model	RMSE[min]	MAE[min]	MAPE[%]
午前ピーク (9:00-11:00)	ConvLSTM	5.341	4.043	9.453
	Model2_Bi	3.751	2.940	7.035
午後ピーク (18:00)	ConvLSTM	2.298	2.101	5.414
	Model2_Bi	1.209	0.845	2.319

4.4.3 天候状態に基づく評価

次に本研究で特徴量として用いた気象情報が予測精度に及ぼす影響を評価するために、天候と運行の乱れの度合いで場合分けして評価を行う。その評価対象として表3に示す3日を選定した。運行の乱れの度合いとして HA の平均絶対パーセント誤差を使用した。この値は、過去の平均からどのくらい乖離しているかを示しており、運行の乱れの指標として利用できる。その値に基づいて運行の乱れを大中小と3種類に場合分けした。

各評価対象における精度の比較結果を表4に示す。ConvLSTM の予測精度を比較すると、降雨が一因で運行が乱

表 3 天候に基づく評価: 比較対象

評価対象	降雨	HA's MAPE[%]	運行の乱れ
9/25(金)	○	11.701	大
9/26(土)	×	7.657	中
9/29(火)	×	6.535	小

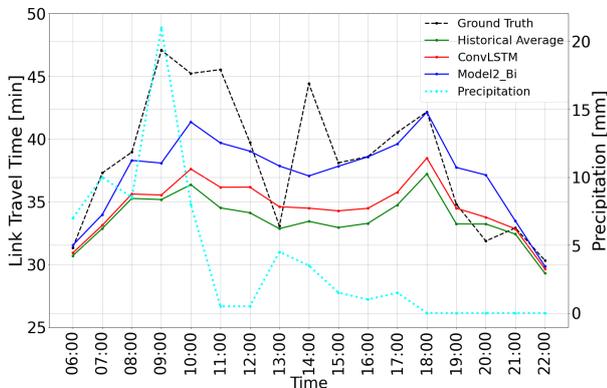


図 8 2020/9/25(金) の到着時刻予測の比較

れた 9 月 25 日は他の日と比べて精度が低下していることが確認できる。図 8 に 9 月 25 日 (金) の予測精度の推移をそれぞれ示す。右の縦軸は降水量の推移、左の縦軸は所要時間の推移を示している。この日は 0:00 から 18:00 まで断続的に雨が降っており、それが一因となって、実測値が Historical Average から大きく外れていることが確認できる。午前のピークを含む 7:00 から 12:00、午後のピークを含む 15:00 から 18:00 の予測値を見ると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、12:00 から 13:00、18:00 から 19:00 では前の時間帯からの急激な所要時間の減少に対して、減少自体は予測できているものの、その減少を過小評価していることが確認できるが、1 日を通して提案手法がより優れていることが分かる。また、天候が安定している 9 月 26 日と 9 月 29 日に関しても、9 月 25 日と比べると減少幅は小さいものの誤差の減少が確認できた。

表 4 天候に基づく評価: 精度比較

評価対象	Model	RMSE[min]	MAE[min]	MAPE[%]
9/25(金)	ConvLSTM	5.389	4.169	9.973
	Model2_Bi	2.158	1.647	6.810
9/26(土)	ConvLSTM	3.428	2.446	6.390
	Model2_Bi	2.158	1.647	4.431
9/29(火)	ConvLSTM	2.408	2.011	5.522
	Model2_Bi	2.020	1.631	4.634

5. 考察と今後の課題

5.1 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を評価

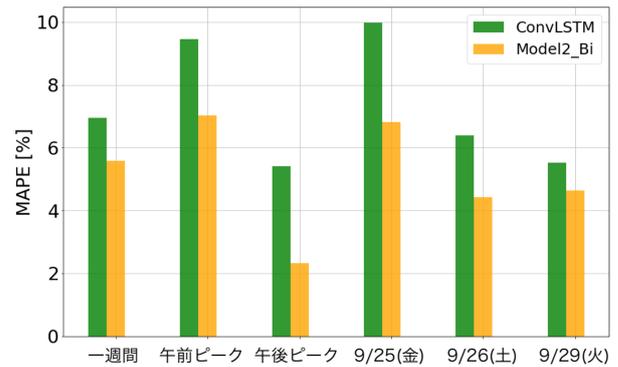


図 9 既存手法との平均絶対パーセント誤差 (MAPE) の比較

対象毎にまとめたものを図 9 に示す。

評価実験では一週間全体に追加して 5 種類の評価項目で比較を行った。既存手法の問題点として述べたように、天候に起因して運行が乱れた際に、既存手法では大きく精度が低下することが確認できた。また、1 週間全体と比べて、9 月 25 日は誤差の減少幅が大きく、平均絶対パーセント誤差で見ると、3%弱の減少を確認できた。このことから提案手法が天候の乱れに起因する運行の乱れに対してロバストな手法であることが確認できた。さらに、Model1 と Model2 の比較から、時刻表ベースのバス路線において走行時間と停車時間を別々のモデルで予測することで同一 LSTM での学習で損なわれていた特徴を学習できたと考えられる。また、午後ピークの時間帯でも 9 月 25 日と同様に 3%弱の平均絶対パーセント誤差の減少が確認できた。これは午前ピークの時間帯の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られ、提案手法ではそのパターンを天候情報を加えることでより効果的に学習できていると考えられる。さらに、天候が安定している 9 月 26 日と 9 月 29 日に関しても、9 月 25 日に比べると減少幅は小さいものの誤差の減少が見られた。9 月 26 日のような天候とは関係なく乱れた運行に対しても既存手法よりも高い精度で予測できることを示した。この結果は提案手法が汎用性の高い手法であることを示している。

5.2 今後の課題

評価実験では、6 種類の手法を比較することで提案手法の有効性と高い汎用性を示すことができた。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パターンが 9 月 25 日 (金) でいくつか見られた。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。

本研究では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て、 $reference_frequency = 60$ の運行データを元に学習及

び予測を行った。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。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。さらに、乗降者数や経路の静的な情報など天候情報以外の外部要因を特徴量として用いることで更なる精度の向上を検討する。

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、精度の高い到着時刻予測手法が必要である。路線バスにおける到着時刻予測手法として Convolutional LSTM を用いた手法 [4] 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。本研究では走行時間と停車時間に対してそれぞれ天候情報と時刻表に基づく特徴量を加えることで解決を試みた。

既存手法と比較した結果、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。特に午後のピーク時間帯や、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、乗降者数や静的な経路情報など路線バスの運行に影響を与える外部要因を特徴量として用いることで精度の向上を目指す。また提案手法における3つのパラメータのチューニングを行うことで更なる精度の向上が見込める。本研究では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て、 $reference_frequency = 60$ の運行データを元に学習及び予測を行ったが、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。

参考文献

- [1] Mugion, R. G., Toni, M., Raharjo, H., Di Pietro, L. and Sebatu, S. P.: Does the service quality of urban public transport enhance sustainable mobility?, *Journal of Cleaner Production*, Vol. 174, pp. 1566–1587 (2018).
- [2] Gooze, A., Watkins, K. and Borning, A.: Benefits of real-time transit information and impacts of data accuracy on rider experience, *Transportation Research Record*, No. 2351, pp. 95–103 (2013).
- [3] Altinkaya, M. and Zontul, M.: Urban Bus Arrival Time Prediction: A Review of Computational Models, *Inter-*

- national Journal of Recent Technology and Engineering*, Vol. 2, No. 4, pp. 1–6 (2013).
- [4] Petersen, N. C., Rodrigues, F. and Pereira, F. C.: Multi-output bus travel time prediction with convolutional LSTM neural network, *Expert Systems with Applications*, Vol. 120, pp. 426–435 (2019).
- [5] Zeng, L., He, G., Han, Q., Ye, L., Li, F. and Chen, L.: A LSTM based bus arrival time prediction method, *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*, pp. 544–549 (2019).
- [6] Pang, J., Huang, J., Du, Y., Yu, H., Huang, Q. and Yin, B.: 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).
- [7] Stover, V. W. and McCormack, E. D.: The impact of weather on bus ridership in Pierce County, Washington, *Journal of Public Transportation*, Vol. 15, No. 1, pp. 95–110 (2012).
- [8] Hofmann, M. and O'Mahony, M.: The impact of adverse weather conditions on urban bus performance measures, *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, Vol. 2005, pp. 84–89 (2005).
- [9] Schuster, M. and Paliwal, K. K.: Bidirectional recurrent neural networks, *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, pp. 2673–2681 (1997).
- [10] Althelaya, K. A., El-Alfy, E. M. and Mohammed, S.: Evaluation of bidirectional LSTM for short-and long-term stock market prediction, *2018 9th International Conference on Information and Communication Systems (ICICS)*, pp. 151–156 (2018).
- [11] Jeyakumar, J. V., Lee, E. S., Xia, Z., Sandha, S. S., Tausik, N. and Srivastava, M.: Deep convolutional bidirectional LSTM based transportation mode recognition, *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pp. 1606–1615 (2018).
- [12] Olewuezi, N. P.: Note on the comparison of some outlier labeling techniques, *Journal of Mathematics and Statistics*, Vol. 7, No. 4, pp. 353–355 (2011).
- [13] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research*, Vol. 15, No. 56, pp. 1929–1958 (2014).
- [14] Ioffe, S. and Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift, *32nd International Conference on Machine Learning, ICML 2015*, Vol. 1, pp. 448–456 (2015).
- [15] 気象庁:過去の気象データ, <https://www.data.jma.go.jp/gmd/risk/obsdl/> (2020-10-23 参照).