

# 登山の到着時刻推定タスクにおける 深層学習手法の適用と評価

## Deep Learning Approaches for Estimating the Time of Arrival in Hiking

浅古 瑞穂\*  
Mizuho Asako

田原 康之  
Yasuyuki Tahara

大須賀 昭彦  
Akihiko Ohsuga

清 雄一  
Yuichi Sei

電気通信大学

The University of Electro-Communications

### 1. はじめに

登山は世界的に人気のアクティビティである。レジャー白書2023によると、2022年の日本における登山者数は500万人であり、日本国内でも多くの人々が登山を楽しんでいる[1]。一方で山岳事故発生件数は増えており、警察庁によると、令和4年における山岳遭難発生件数は、過去最多の3015件であった[2]。主な山岳遭難の要因として、道迷い、滑落、転倒、病気、疲労が挙げられる[2]。これらの山岳遭難を未然に防ぐためにも、事前に無理のない計画を立てることが重要である。

最近では、登山者がスマートフォンを携帯し、YAMAP[3]やYamareco[4]等のアプリケーションを用いて位置情報を記録しながら登山を行うことが一般的となりつつある。また、登山中に記録した自分の位置情報ログと共に、山行記録を投稿することで、他の登山者と情報を共有することができるようになった。その結果、各サービスには多くの登山者の位置情報ログが蓄積されており、これらのデータを活用することで、登山者の行動を分析することが可能となった。

登山の計画立案には、登山地図に掲載されているコースタイムがよく用いられる。コースタイムとは、山中の各地点間の所要時間を示したものであり、登山者が目的地に到着する時刻を見積もるために用いられる。しかし、従来のコースタイムは登山地図の著者の経験則に基づいて決められており、必ずしも登山者の歩行ペースと一致しているとは限らなかった。そこで、斎藤ら[5]は登山者の位置情報ログを用いて、斜度と歩行ペースの関係を分析し、それに基づいてコースタイムを標準化した。この研究の成果は実際のアプリケーションに適用されており、登山者にデータに基づいたコースタイムを提供することで、より正確な計画立案に貢献している[6]。

コースタイムは登山計画を立てる際に有用な指標であるが、登山者個人の特性によって補正が必要である。補正には、登山者の経験値や体力、装備、パーティ人数、天候、季節等、様々な要因が影響する。また、疲労による速度低下や、休憩による所要時間の増加も考慮する必要がある。これらの要素を総合的に考慮し、正確な所要時間を見積もるには、経験と知識が必要であり、特に登山初心者にとっては難しい作業である。

そこで本研究では、登山の位置情報ログを用いた到着時刻推定手法を提案する。具体的には、交通分野にて発展した深層学習による到着時刻推定技術を、登山における到着時刻推定タスクにも応用し、登山者の位置情報ログをもとに、登山の所要時間を予測するモデルを構築する。これにより、モデルを用いることで正確な所要時間を推定することを可能とし、より安全な登山計画の立案を支援することを目指す。

実験の結果、提案手法はMAPEが14.39%であり、従来の登山所要時間推定手法と比較して、MAPEが14.18ポイント改善した。また、既存の深層学習手法と比較して、提案手法はMAPEが2.14ポイント改善した。

### 2. 関連研究

#### 2.1. 既存の歩行速度推定手法

##### 2.1.1. Naismithの法則

Naismithの法則[7]は、登山の所要時間を見積もるための手法の一つである。Naismithは歩行ペースの推定方法として、“1時間ごとに3マイル(約4.8km)、ただし2000フィート(約610m)登るごとに所要時間は1時間増加する”という経験則を述べた。計算式が単純であり、登山中に容易に計算することができるという利点があるが、登り坂のみを考慮しており、下り坂を考慮していないという欠点がある。

##### 2.1.2. Toblerの歩行モデル

Toblerのモデル[8]は、歩行速度と斜度の関係を指数関数で表したモデルである。具体的には、以下の数式で表される。

$$W = 6e^{-3.5|S+0.05|}$$
$$S = \frac{dx}{dh} \quad (1)$$

ここで、 $W$ は歩行速度(km/h)、 $S$ は斜度(%)を表す。斜度が-0.05(-5%)の時、 $W$ は最大値である6km/hとなる。このモデルは下り坂においても適用可能だが、指数関数モデルであるため、斜度が大きい場合に速度を過小評価しすぎる欠点がある。

##### 2.1.3. Campbellの歩行モデル

Campbellのモデル[9]は、歩行速度と斜度の関係をローレンツ関数で表したモデルである。具体的には、以下の数式で表される。

$$r = c \left( \frac{1}{\pi b \left( 1 + \left( \frac{s+a}{b} \right)^2 \right)} \right) + d + es \quad (2)$$

ここで、 $r$ は歩行速度(m/s)、 $s$ は斜度(°)を表す。 $a, b, c, d, e$ はモデルのパラメータである。モデル作成には、ソルトレイクシティとロサンゼルス近郊の20本のトレイルで記録された1999件の登山者の位置情報ログを用いた。また、歩行速度の個人差を考慮するため、2.5パーセンタイルごとに異なる39個のモデルを作成した。例えば、50パーセンタイルのモデルの場合、 $a = -1.4579, b = 22.0787, c = 76.3271, d = 0.0525, e = 3.2002 \times 10^{-4}$ である。これは斜度が0°の時の歩行速度は4.133km/h、斜度が30°の時の歩行速度は1.707km/hになる。ローレンツ関数による近似は、指数関数による近似と比較して、斜度が大きい場合でも速度を過小評価しすぎないという利点がある。ただし、どのパーセンタイルのモデルを用いるかを決定するためには、登山者の個人差を事前に把握する必要がある。

\*連絡先: 浅古 瑞穂, 電気通信大学大学院 情報理工学系研究科, 〒182-8585 東京都調布市調布ヶ丘 1-5-1, asako.mizuho@ohsuga.lab.ucc.ac.jp

### 2.1.4. Wood の歩行モデル

Wood のモデル[10] は、歩行方向の斜度(歩行斜度)の他に、地形斜度と障害物レベルの影響を考慮したモデルである。歩行斜度は歩行者が歩く方向に対する斜度である。一方、地形斜度は地形そのものの斜度で、地形の特定のある地点に対して一意に定まる。モデル作成には、hikr.org[11]に掲載されている 7636 件の登山者の位置情報ログを用いた。また、数値標高モデル(DEM)から標高と斜度を算出した。さらに、数値表層モデル(DSM)と数値地形モデル(DTM)の標高差から障害物レベルを計算した。これらのデータを元に、一般化線形モデル(GLM)を用いて歩行速度を予測するモデルを作成した。具体的には、以下の数式で表される。

$$v = \exp(a + b\varphi + c\theta + d\theta^2) \quad (3)$$

ここで、 $v$ は歩行速度(km/h)、 $\varphi$ は地形斜度( $^\circ$ )、 $\theta$ は歩行斜度( $^\circ$ )を表す。 $a, b, c, d$ は障害物レベルによって異なる値を持つパラメータである。地形斜度と障害物を加えることによって、歩行速度の予測精度向上したと報告している。一方で、Tobler のモデルと同様に、指数関数モデルであるため、斜度が大きい場合に速度を過小評価しすぎる欠点がある。

これらの既存手法は、すべて斜度から歩行速度を推定するモデルである。そのため、疲労による速度低下や、休憩による所要時間の増加を適切に考慮することができない。Campbell のモデルと Wood のモデルは、データの前処理で休憩時間を除外しており、登山者が実際にこれらのモデルを使うには、休憩時間について考慮する必要がある。

### 2.2. 深層学習による移動時間予測技術

近年、交通分野において、深層学習を用いた到着時刻予測が研究されている。Wang ら[12]は、到着時刻予測のための深層学習モデルである DeepTTE を提案した。DeepTTE は、位置情報ログと属性情報を入力として、部分区間の所要時間推定と全体の所要時間推定を同時に学習するモデルである。最終的な目標は全体の所要時間を予測することであるが、部分区間の所要時間も同時に学習することで、全体の所要時間の予測精度を向上させることができる。具体的なモデルのアーキテクチャは図 1 に示す。

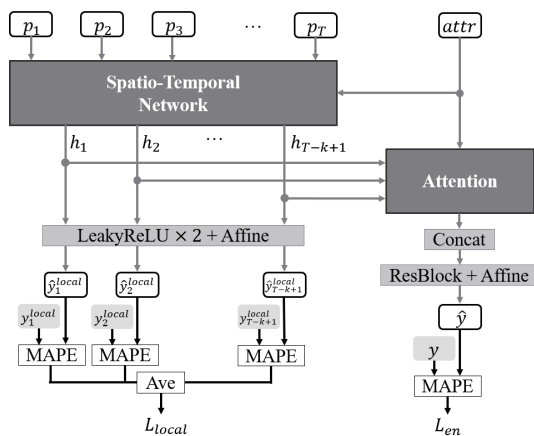


図 1 DeepTTE の概要図

また、DeepTTE の損失関数は以下の式で表される。

$$L = \alpha L_{local} + (1 - \alpha) L_{en} \quad (4)$$

ここで、 $L_{local}$  は部分区間の所要時間の損失(MAPE: Mean Absolute Percentage Error)の平均値、 $L_{en}$  は全体の所要時間の損失(MAPE)の平均値を表す。

実験では約 1289 万件のタクシーの位置情報ログをもとに、全体の所要時間を予測するタスクを行った。結果として、

DeepTTE は決定木ベースの手法(GBDT, 勾配ブースティング決定木)や、多層パーセプトロン(MLP)よりも高い予測精度を達成した。

DeepTTE[12] は、データセットの前処理として、GPS の連続する 2 点間の間隔が 200m-400m になるようにリサンプリングを行っている。これは、タクシーのデータセットでは、データが一定間隔で記録されており、モデルが“データの個数を数えることで、所要時間を予測する”という単純なパターンを学習することを防ぐためであると述べている。このようなリサンプリングを行っても問題ない理由として、車は道路上を走っており、道路は直線的であるため、リサンプリングを行っても、データセットの性質が変わりにくいことが挙げられる。

一方で、登山道は道路と比較すると直線的ではなく、曲がりくねった道が多い。そのため、リサンプリングを行うと、直線的な道を比較してデータセットの性質が変わりやすい。また、登山では標高の情報も重要な要素の 1 つであるが、リサンプリングを行うことにより、標高差を過小評価・過大評価する可能性がある。そのため、データの性質を変えずに登山のログデータをリサンプリングすることは難しい。

ところが、リサンプリングを行わない場合、 $L_{local}$  のばらつきが大きくなるという問題がある。MAPE は相対誤差であるため、所要時間予測の場合には、実際の所要時間が短ければ短いほど、MAPE の値が大きくなる。学習は、損失が小さくなるように行われるため、所要時間が短い部分区間ばかりに注目した学習を行い、所要時間が長い部分区間に対しては、予測精度が低くなる可能性がある。最終的な目標は、全体の所要時間を予測することであるため、所要時間が長い部分の予測精度が低いと、全体の所要時間の予測精度低下につながる。DeepTTE は、データ点間の所要時間のばらつきが小さい場合に適用している手法であり、登山データのようなばらつきが大きいデータに対しては、そのまま適用するのは難しい。

そこで本研究では、DeepTTE のアーキテクチャをベースとしつつ、全体の所要時間推定のみを行うモデルを構築し、登山における到着時刻推定タスクに適用することを目指す。

### 3. 提案手法

本研究では、登山者の位置情報ログと属性情報を用いた、深層学習による到着時刻推定手法を提案する。具体的には、DeepTTE[12] のアーキテクチャをベースとして、登山者の位置情報ログと登山に関する属性情報を入力とし、ログ全体の所要時間を出力としたモデルを構築する。概要図を図 2 に示す。

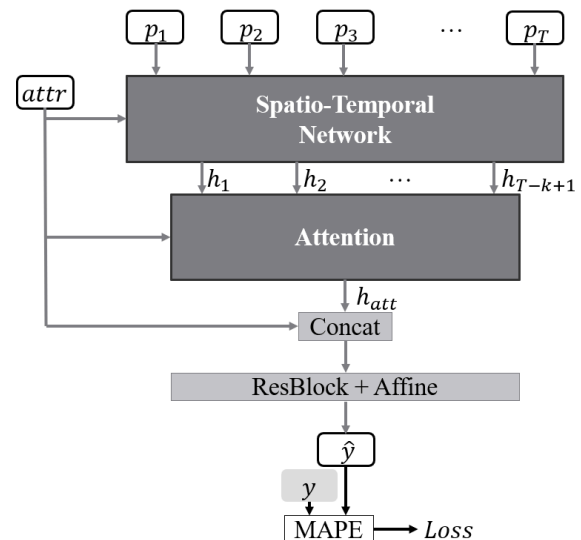


図 2 提案手法の概要図

### 3.1. 属性コンポーネント

属性コンポーネントは、位置情報ログの属性情報を扱う機構である。属性情報として、1.標高の最高点( $ele_{max}$ )、2.標高の最低点( $ele_{min}$ )、3.累計標高登り( $D+$ )、4.累計標高下り( $D-$ )、5.ログの総距離( $dist_{total}$ )の5つの情報を用いる。これらを正規化した後、結合することで5次元の属性表現 $attr$ を生成する。

$$attr = [ele_{max}, ele_{min}, D+, D-, dist_{total}] \quad (5)$$

### 3.2. 時空間コンポーネント

時空間コンポーネントは、位置情報ログの位置情報本体を扱う機構である。概要図を図3に示す。

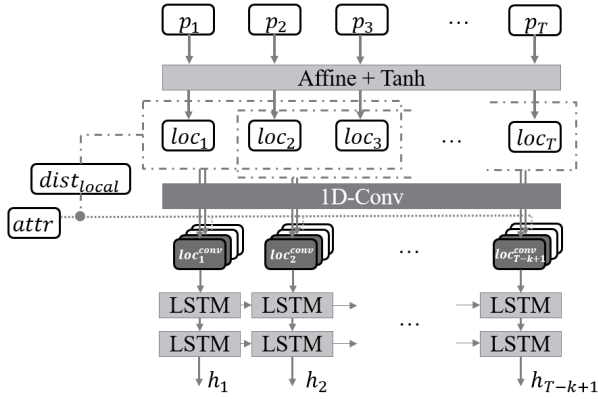


図3 時空間コンポーネントの概要図

入力は長さ $T$ のシーケンスデータであり、各位置情報 $p_i$ は、緯度( $lat$ )、経度( $lng$ )、標高( $ele$ )、累積距離( $dist_{cum}$ )、歩行斜度( $ws$ )、地形斜度( $ts$ )の6つの情報を持つ。これらを正規化した後、非線形変換を行い16次元の位置情報表現 $loc$ を生成する。

$$p_i = [p_i.lat, p_i.lng, p_i.ele, p_i.dist_{cum}, p_i.ws, p_i.ts] \quad (6)$$

$$loc = \tanh(W_{loc} \cdot p_i)$$

その後、カーネルサイズ $k$ で、隣接している位置情報表現に対して次元畳み込みを行い、 $c$ 個の特徴マップを生成する。

$$loc_i^{conv} = \sigma_{cnn}(W_{conv} * loc_{i:i+k-1} + b) \quad (7)$$

この $c$ 個の特徴マップに、畳み込んだ区間の部分距離 $dist_{local}$ を結合することで、位置情報表現 $loc^f$ を生成する。

$$loc^f = [loc^{conv}, dist_{local}] \quad (8)$$

位置情報表現 $loc^f$ は、属性表現 $attr$ と結合され、LSTMレイヤに入力される。

$$h_i = \sigma_{lstm}(W_h \cdot [loc_i^f, attr]) \quad (9)$$

LSTMは2層で構成され、最終的な隠れ状態 $h$ を出力する。なお、隠れ状態 $h$ は128次元である。これらの隠れ状態 $h_i$ は、登山ログの部分区間の位置情報表現 $loc_i^f$ と対応している。この機構により、登山ログの位置及び時系列的な関係性を考慮することができる。

### 3.3. 移動時間予測コンポーネント

移動時間予測コンポーネントは、属性コンポーネントの出力 $attr$ と時空間コンポーネントの出力 $h$ から、最終的な所要時間を予測する機構である。概要図を図4に示す。

移動時間予測コンポーネントは、大きく分けて2つの機構から構成される。1つ目は、属性表現 $attr$ をqueryとし、LSTMの

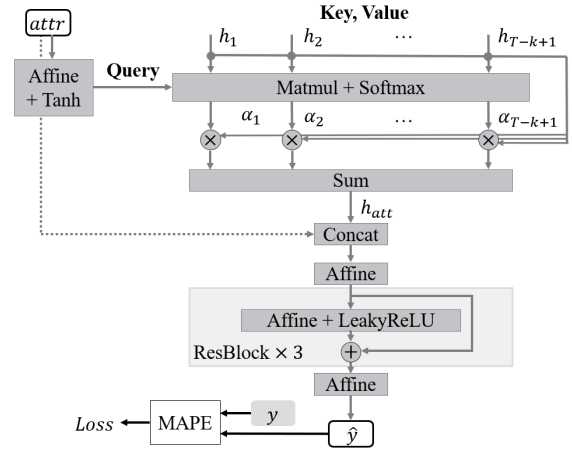


図4 移動時間予測コンポーネントの概要図

出力 $h$ をkey, valueとして、重み付き和を計算するAttention機構である。具体的には、以下の式(10)で表される。

$$z_i = \langle \sigma_{att}(attr), h_i \rangle$$

$$\alpha_i = \frac{\exp(z_i)}{\sum(\exp(z_j))} \quad (10)$$

$$h_{att} = \sum_{i=1}^{|T|-k+1} \alpha_i \cdot h_i$$

ここで、 $\sigma_{att}$ では非線形変換層である。非線形変換と同時に $attr$ の次元数を $h$ と揃えている。 $z_i$ は、 $\sigma_{att}$ と $h_i$ の内積を計算したものである。 $\alpha_i$ は $z_i$ の重みであり、 $h_{att}$ は重み付き和である。この機構により、登山ログの部分区間の重要度を考慮することができる。

2つ目は、全結合層と残差接続層を通じて、最終的な所要時間を予測する機構である。具体的には、以下の式(11)、(12)、(13)で表される。

$$h_l = W_y \cdot [attr, h_{att}] \quad (11)$$

$$h_{l+1} = \text{LeakyReLU}(W_l \cdot h_l) + h_l \quad (12)$$

$$\hat{y} = W_y \cdot h_{l+n} \quad (13)$$

まず、属性表現 $attr$ とAttention機構の出力 $h_{att}$ を結合する。その後、3層の残差接続、全結合層を通じて、今回の目的である全体の所要時間 $\hat{y}$ を出力する。これらの機構により、属性情報と位置情報の両方を考慮した所要時間の予測が可能となる。

## 4. 実験

### 4.1. データセット

本研究では、kaggleに公開されている登山者の位置情報ログデータセット“GPS recorded hikes from hikr.org”[13]を用いた。このデータセットは、hikr.org[11]に掲載されている登山者の位置情報ログを収集したものであり、約12,000件の登山ログからなるデータセットである。この登山ログは、各位置情報 $p_i$ ごとに、緯度( $lat$ )、経度( $lng$ )、標高( $ele$ )、時刻( $time$ )の情報を有している。

#### 4.1.1. hikr.org データセットの前処理

まず、データセットには欠損値が含まれているため、欠損値を排除するために各登山ログに対して以下の手順を行った。

1. 欠損値を含むポイントの削除
2. データ点が50点未満の登山ログを除外

3. 不正値(速度が 5m/s = 18km/h を超えるデータと、データ点同士の間隔が 1200 秒以上のデータの確認)
  - a 連続 50 点以上正常値である区間が存在する場合、その区間を切り取り用いる。
  - b 存在しない場合、その登山ログを除外する。

この操作の結果、新たに 8,538 件の登山ログを得られた。また、スタート地点を 0 とした累積距離  $dist_{cum}$  と、スタート地点を 0 とした累積時間  $time_{cum}$  を計算し、各ポイントの属性として追加した。この時、最後のポイントの累積距離をログの総距離  $dist_{total}$ 、最後のポイントの累積時間をログの総時間  $time_{total}$  とした。

次に、ログの属性情報を作成するため、登山ログの標高情報から、標高の最高点  $ele_{max}$ 、標高の最低点  $ele_{min}$  を取得した。また、各ポイント間の標高差から累計標高登り  $D+$ 、累計標高下り  $D-$  を計算した。

#### 4.1.2. 斜度の追加

歩行斜度  $ws$ (単位:度)は、ログの各ポイント間の水平距離と標高差から計算した。

地形斜度の取得  $ts$ (単位:度)には、緯度 60 度以下の地域については NASADEM[14] を用い、緯度 60 度以上の地域については ArcticDEM Mosaic[15] を用いた。

最終的に、各ログは以下の情報を持つデータとなった。

- 属性情報  $attr$ 
  - 標高の最高点  $ele_{max}$  (単位:m)
  - 標高の最低点  $ele_{min}$  (m)
  - 累計標高登り  $D+$  (m)
  - 累計標高下り  $D-$  (m)
  - ログの総距離  $dist_{total}$  (m)
  - ログの総時間  $time_{total}$  (s)
- 位置情報  $p_i$ 
  - 緯度  $lat$  (度)
  - 経度  $lng$  (度)
  - 標高  $ele$  (m)
  - 累積距離  $dist_{cum}$  (m)
  - 歩行斜度  $ws$  (度)
  - 地形斜度  $ts$  (度)
  - 累積時間  $time_{cum}$  (s)

なお、入力に用いられるのは、ログの総時間  $time_{total}$  と累積時間  $time_{cum}$  を除いた情報であり、最終的に予測したい値はログの総時間  $time_{total}$  である。

#### 4.2. モデルの学習

モデルを学習するために、データセットを train データ、validation データ、test データに分割した。全体の 80% を train データ、20% を test データとして分割し、train データの内 5% を validation データとした。損失関数は MAPE(Mean Absolute Percentage Error)を用いた。Optimizer には Adam[16] を用い、学習率は 0.001 として、1000 エポック学習を行った。ハイパーパラメータとして、バッチサイズ 64、畳み込み層のフィルタ数  $c = 32$ 、畳み込み層のカーネルサイズ  $k = 4$  を選択した。

#### 4.3. 評価

評価指標として、MAE(Mean Absolute Error)、MSE(Mean Squared Error)、MAPE(Mean Absolute Percentage Error)を用いた。また、1000 エポックの学習の中で、Validation データに対する損失の推移を確認した。具体的には、1 エポックごとの Validation データに対する損失を記録し、フィルタサイズ 50 エポックで損失の移動平均を取り、その移動平均の値が最も小さくなった時点のモデルを最終的なモデルとして採用した。なお、学習重みを 10 エポックごとに保存したため、移動平均が最小になったエポックに最も近い重みを選択した。test データに対

して、MAPE、MAE、MSE を計算し、提案手法と既存手法との比較を行った。

既存の登山所要時間推定手法として、Naitmith の法則[7]、Tobler の歩行モデル[8]、Campbell の歩行モデル[9]、Wood の歩行モデル[10] の 4 つのモデルと比較した。また、深層学習ベースの既存手法として、LSTM\_seq2vec、DeepTTE を 1000 エポック学習したモデルと比較した。

Naitmith の法則を歩行斜度  $ws$ (度)と速度  $v$ (m/s)の関係に当てはめると、以下の式(14)で表されることから、歩行斜度  $ws$  を用いて部分区間ごとに所要時間を計算し、それらを総和することで全体の所要時間を推定した。

$$v = \frac{4828.02}{3600(1 + \tan(ws) \times 7.92)} \quad (14)$$

ただし、Naitmith の法則は下り坂を扱えないので、 $ws$  が 0 以下の場合、 $ws = 0$  とした。

Tobler の歩行モデルは式(1)で表されることから、歩行斜度  $ws$  から部分区間ごとの所要時間を計算し、全体の所要時間を推定した。ただし、Tobler の歩行モデルは、斜度が大きい場合に速度を過小評価しすぎるという欠点があるため、斜度の絶対値が 80%(38.66 度)を超える場合は、 $ws = 38.66$  とした。なお、この値は、train データに対して、異なる斜度(%)の制限を設けた場合の MAPE を比較し、最も小さな値を選択した。

Campbell の歩行モデルは式(2)であり、歩行速度のパーセンタイルごとに異なるモデルのパラメータ  $a, b, c, d, e$  を持つため、train データを用いてパラメータを推定した。Campbell[9] は、モデルを作成する際、1. 不正データの除外、2. 個別ログのばらつき修正、3. データセット全体に対して集計を行った。

1. 不正データ除外の操作として、Campbell[9]と同様に、移動速度が 0.2m/s 未満であるポイントは休憩であるため除外し、移動速度が 5m/s 以上であるポイントは歩行ではない可能性があるため除外した。また、斜度が 30° の場合、データ点が少ないため、モデル作成時には除外した。

2. 個別ログのばらつき修正の操作として、Campbell[9]では、個人の歩行速度のばらつきを修正するために、個別のログに対して、斜度 1 度間隔ごとに速度の中央値を計算し、その値をそのログのその斜度における歩行速度とした。一方で、本研究で用いたデータセットは、Campbell のデータセットと比較して、データ点の数が少ないため、斜度 2 度間隔ごとに集計した。

3. データセット全体に対して集計の操作として、Campbell[9]では、先ほど集計した“斜度 2 度間隔ごとの速度の中央値”を歩行速度順に 2.5 パーセンタイルごとに集計して中央値を求め、各パーセンタイルごとに斜度と速度の関係を式(2)にフィッティングした。一方で、本研究では 5 パーセンタイルごとに集計しフィッティングを行った。

5 パーセンタイルごとに 19 個の異なるモデルを作成したが、そのうち 50 パーセンタイルのモデルを全ログの中央値のモデルとして採用した。その結果、式(2)に対して、 $a = -2.730$ 、 $b = 17.33$ 、 $c = 53.96$ 、 $d = 0.1070$ 、 $e = -1.041 \times 10^{-3}$  のパラメータを得た。

Wood の歩行モデルは式(3)であり、障害物レベルによって異なるパラメータ  $a, b, c, d$  を持つ。しかし、本研究では、障害物レベルの計算に用いる DSM と DTM のデータが利用できなかったため、障害物レベルに関わらず、全てのログに対して同じパラメータを用いた。また、本研究は、休憩を含めた全体の所要時間の推定を目的としているため、Wood の歩行モデルの前処理で休憩時間を除外することは行わなかった。そして、train データに対して、一般化線形モデル(GLM)を用いて、Wood の歩行モデルを学習した結果、式(3)に対して、 $a = 1.4806$ 、 $b = -0.0046$ 、 $c = -0.0106$ 、 $d = -0.0015$  のパラメータを得た。さらに、Wood の歩行モデルも Tobler の歩行モデルと同様に、斜度が大きい場合に速度を過小評価しすぎるという欠点があるため、斜度の絶対値が 40 度を超える場合は、 $ws = 40$  とした。な

お、この値は、train データに対して、異なる斜度(度)の制限を設けた場合の MAPE を比較し、最も小さな値を選択した。

LSTM\_seq2vec は、LSTM をもとに作成したモデルであり、位置情報ログ全体を入力とし、最終的な所要時間を出力するモデルである。LSTM は再帰型ニューラルネットワークの一種であり、時系列データの長期的な依存関係を学習することができる[17]。具体的には、図 5 のようなアーキテクチャである。

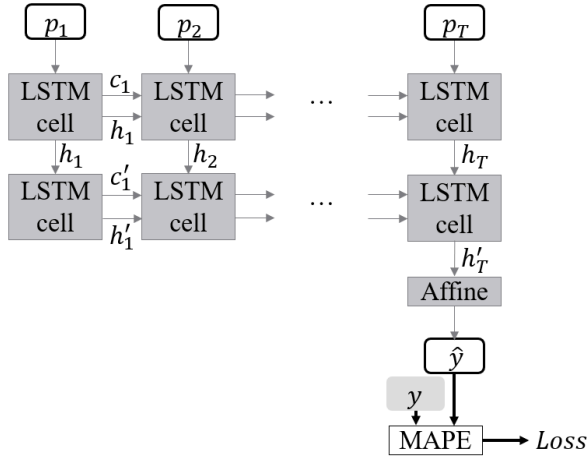


図 5 LSTM\_seq2vec の概要図

LSTM の層は 2 層で構成され、最終的な隠れ状態  $h$  を全体の所要時間の予測結果として出力する。

DeepTTE[12] は、位置情報ログと属性情報を入力として、部分区間の所要時間推定と全体の所要時間推定を同時に学習するモデルである。具体的には、図 1 のようなアーキテクチャであり、損失関数は式(4)で表される。損失関数の重みづけについて、 $\alpha = 0.05, 0.5, 0.95$  に設定し、それぞれに対して実験を行った。

これらの既存の登山所要時間推定手法、深層学習手法と提案手法との比較結果を以下の表 1 に示す。

表 1 提案手法と既存の登山所要時間推定手法、深層学習手法の比較

モデル	MAPE↓(%)	MAE↓(h)	MSE↓(h <sup>2</sup> )
提案手法	<b>14.39</b>	<b>0.460</b>	<b>0.623</b>
Naismith[7]	30.54	1.005	1.766
Tobler[8]	28.77	0.962	1.607
Campbell[9]	28.57	0.965	1.675
Wood[10]	30.11	1.047	1.824
LSTM_seq2vec	16.53	0.517	0.781
DeepTTE( $\alpha : 0.05$ )	18.42	0.573	0.865
DeepTTE( $\alpha : 0.5$ )	21.98	0.687	1.108
DeepTTE( $\alpha : 0.95$ )	21.77	0.676	1.222

表 1 より、提案手法はすべての評価指標において、既存手法よりも高い予測精度を達成した。既存の登山の歩行モデルである Naismith の法則、Tobler の歩行モデル、Campbell の歩行モデル、Wood の歩行モデルの中で、最も精度の良い Campbell の歩行モデルと比較して、提案手法は 14.18% の改善を達成した。また、深層学習手法である LSTM\_seq2vec、DeepTTE の中で、最も成績が良かった LSTM\_seq2vec と比較して、提案手法は 2.14 ポイントの改善を達成した。

#### 4.4. 最適なカーネルサイズ $k$ の探索

カーネルサイズ  $k$  は、隣接する位置表現に対して一次元畳み込みを行う際の区間の大きさを決定するハイパーパラメタである。登山データに対して、位置情報の空間的な特徴を捉えることによる効果を確かめるために、カーネルサイズ  $k$  を変化させた場合の予測精度を比較した。  $k = 2, 3, 4, 5, 10, 15, 20$  の 7 パターンについて、1000 エポック学習を行い、test データに対して評価を行った。結果を以下の表 2 に示す。

表 2 カーネルサイズ  $k$  に対する結果の比較

$k$	MAPE↓(%)	MAE↓(h)	MSE↓(h <sup>2</sup> )
2	15.14	0.468	0.671
3	14.70	0.465	<b>0.621</b>
4	<b>14.39</b>	<b>0.460</b>	0.623
5	15.80	0.490	0.710
10	16.82	0.500	0.744
15	16.44	0.499	0.823
20	15.83	0.483	0.738

表 2 より、カーネルサイズ  $k = 4$  が最も低い MAPE, MAE を達成した。また、MSE は  $k = 3$  が最も低い値を達成した。全体的な傾向として、 $k = 3, 4$  付近が高精度であることが分かった。また、カーネルサイズを大きくすると、MAPE, MAE, MSE が増加し精度が悪化する傾向がみられた。しかし、カーネルサイズを最小の  $k = 2$  にすると、精度が悪化することが分かった。

このような結果となった理由として、カーネルサイズが大きすぎる場合は、ログの幅広い範囲を畳み込んでしまい、局所的な特徴を捉えることが難しくなり、逆にカーネルサイズが最小値の  $k = 2$  の場合は、2 点間の関係しか畳み込んでおらず、直線的な関係しか捉えられないため、精度が悪化すると考えられる。 $k = 3, 4$  の値は、局所的な特徴を捉えることができるかつ、空間的な特徴も捉えることができるため、高い精度を達成したと考えられる。

#### 5. おわりに

本研究では、登山のスタート地点からゴール地点までの所要時間の推定を目的とし、深層学習手法を用いたモデルを提案した。提案手法のモデルは、交通分野における既存の深層学習手法である DeepTTE をベースとし、登山の位置情報ログの特性に合わせて構築した。結果として、提案手法は既存の登山所要時間推定手法である Naismith の法則、Tobler の歩行モデル、Campbell の歩行モデル、Wood の歩行モデルよりも高い予測精度を達成した。特に、既存の登山所要時間推定手法で最も MAPE スコアが低かった Campbell の歩行モデルと比較して、14.18% の改善を達成した。このことは、単に斜度から速度を推定するモデルの限界を示すとともに、深層学習手法が登山の所要時間推定に対して有効であることを示している。今回の実験結果により、斜度以外の要素が速度に影響を与えていることが示唆されたが、具体的な要因は不明である。そのため、所要時間に影響を与える要因の詳細な分析が、今後の課題である。

また、提案手法は、他の深層学習手法である LSTM\_seq2vec、DeepTTE と比較しても、すべての評価指標において最も良いスコアを達成した。特に、LSTM\_seq2vec と比較して、提案手法は 2.14 ポイントの改善を達成した。これは、一次元畳み込みと LSTM、Attention 機構を組み合わせた手法が、登山のような位置情報ログと属性情報を含むデータを扱う際に有効であることを示している。また、DeepTTE( $\alpha : 0.05$ ) の精度は、LSTM\_seq2vec と比較して 1.89 ポイント悪い結果となった。こ

のことから、交通予測の分野とは異なり、登山の所要時間予測では、部分区間の所要時間の推定が、全体の所要時間の推定に対して有効であるとは限らないことが示唆された。今後の課題として、交通分野における所要時間予測タスクと登山の所要時間予測のタスクの違いを明らかにし、それに基づいて、より高い予測精度を達成するためのモデルを構築することが挙げられる。

## 謝辞

本研究は JSPS 科研費 JP22K12157, JP23K28377, JP24H00714 の助成を受けたものです。

## 参考文献

- [1] レジャー白書 2023 —余暇の現状と産業・市場の動向—。公益財団法人日本生産性本部, 2023
- [2] “令和 4 年における山岳遭難の概況,” 警察庁生活安全局生活安全企画課, Accessed on 30.05.2023 URL: [https://www.npa.go.jp/publications/statistics/safetylife/r04sangakusounan\\_gaikyou.pdf](https://www.npa.go.jp/publications/statistics/safetylife/r04sangakusounan_gaikyou.pdf).
- [3] “YAMAP,” 株式会社ヤマップ, Accessed on 30.05.2024 URL: <https://yamap.com/>.
- [4] “Yamareco,” 株式会社ヤマレコ, Accessed on 30.05.2024 URL: <https://www.yamareco.com/>.
- [5] 斎藤大輔, 松本英高, 山本正嘉, “大規模 GPS ログデータに基づく一般的な登山道における標準コースタイム計算手法の提案,” 登山医学, vol. 42, pp.37–47, 2022
- [6] “コースタイムの「バラつき」解消へ,” 株式会社ヤマップ, Accessed on 5.06.2024 URL: <https://corporate.yamap.co.jp/news/a5NO7ep9>.
- [7] W. W. Naismith, “Cruach Ardran, Stobinian, and Ben More,” *The Scottish Mountaineering Club Journal*, vol. 2, pp.135–136, 1892
- [8] W. R. Tobler, “Three presentations on geographical analysis and modeling,” *National Center for Geographic Information and Analysis*, 1993
- [9] M. J. Campbell, P. E. Dennison, and M. P. Thompson, “Predicting the variability in pedestrian travel rates and times using crowdsourced GPS data,” *Computers, Environment and Urban Systems*, vol. 97, 2022
- [10] A. Wood, W. Mackaness, T. I. Simpson, and J. D. Armstrong, “Improved prediction of hiking speeds using a data driven approach,” *PLoS One*, 2023
- [11] “Hikr.org,” Accessed on 30.05.2024 URL: <https://www.hikr.org/>.
- [12] Dong Wang, Junbo Zhang, Wei Cao, Jian Li., and Yu Zheng, “When will you arrive? estimating travel time based on deep neural networks,” *Thirty-Second AAAI Conference on Artificial Intelligence*, vol. 32, 2018
- [13] “GPS recorded hikes from hikr.org,” Kaggle, Accessed on 4.06.2024 URL: <https://www.kaggle.com/datasets/roccoli/gpx-hike-tracks>.
- [14] “NASA JPL (2020). NASADEM Merged DEM Global 1 arc second V001 [Data set],” NASA EOSDIS Land Processes DAAC, Accessed on 4.06.2024 URL: [doi:10.5067/MEASURES/NASADEM/NASADEM\\_HGT.001](https://doi.org/10.5067/MEASURES/NASADEM/NASADEM_HGT.001).
- [15] “Porter, Claire; Morin, Paul; Howat, Ian; Noh, Myoung-Jon; Bates, Brian; Peterman, Kenneth; Keesey, Scott; Schlenk, Matthew; Gardiner, Judith; Tomko, Karen; Willis, Michael; Kelleher, Cole; Cloutier, Michael; Husby, Eric; Foga, Steven; Nakamura, Hitomi; Platson, Melisa; Wethington, Michael, Jr.; Williamson, Cathleen; Bauer, Gregory; Enos, Jeremy; Arnold, Galen; Kramer, William; Becker, Peter; Doshi, Abhijit; D'Souza, Cristelle; Cummins, Pat; Laurier, Fabien; Bojesen, Mikkel (2018). ArcticDEM, Version 3 [Data set],” Harvard Dataverse, Accessed on 4.06.2024 URL: [doi:10.7910/DVN/OHHUKH](https://doi.org/10.7910/DVN/OHHUKH).
- [16] Diederik P. Kingma and Jimmy Ba, “Adam: A Method for Stochastic Optimization,” *International Conference on Learning Representations (ICLR)*, 2015
- [17] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, 1997