# 都バスのリアルタイム運行データを用いた渋滞検知

青柳 宏紀† 岡田 一洸‡ 山名 早人§

†早稲田大学 基幹理工学部 〒169-8555 東京都新宿区大久保 3-4-1
‡早稲田大学大学院 基幹理工学研究科 〒169-8555 東京都新宿区大久保 3-4-1
§早稲田大学 理工学術院 〒169-8555 東京都新宿区大久保 3-4-1
E-mail: †‡§{aoyagih, k-okada, yamana}@yama.info.waseda.ac.jp

**あらまし** 交通渋滞は，われわれの日常生活に悪影響をもたらしており，車が渋滞を回避し走行できるよう，渋滞検知が必要とされている．従来の渋滞検知手法は，車の走行位置情報や道路上のセンサーに依存しており，実現コストが大きい．本稿では，公共交通機関からオープンデータとして公開されるデータをもとに渋滞検知する手法に取り組む．オープンデータからの渋滞検知が可能となれば実現コストを下げることができる．具体的には，バスの運行データと機械学習を組み合わせた渋滞の自動検知手法を提案する．提案手法では，バスの各停留所の発車時刻データを特徴量とし渋滞検知を行う．評価実験では，都バスの運行データを用い，明治通り(渋谷～池袋間)における渋滞検知を行った．全停留所に対して学習器を一つ用意した場合，F1 スコアは 0.306 となり，停留所区間毎に個別の学習器で学習をした場合，F1 スコアは平均 0.399，特定の区間においては F1 スコア 0.675 を得た．また，単一バスの運行データのみで渋滞検知する場合を除くと，特定の区間において F1 スコアを 0.742 へ向上することができた．

**キーワード** 渋滞，オープンデータ，バス，公共交通機関，機械学習

## 1. はじめに

交通渋滞は，われわれの日常生活に対し悪影響をもたらしている．国土交通省によると，日本国内における渋滞による経済的損失は年間 12 兆円と試算され，1 人あたり年間 30 時間の時間損失がある[1]．他にも，渋滞による悪影響として，渋滞中の車両から排出される排気ガスによる環境被害や，救急車や消防車のような緊急車両の遅延がある．そこで，運転手が渋滞を回避できるように，渋滞をリアルタイムで検知することが求められる．しかし，従来の渋滞検知の方法は，車の走行位置情報や道路上のセンサーに依存しており，実現コストが大きい．

一方，近年では，公共交通機関に関するオープンデータの利活用が進んでおり，発着時刻やロケーション情報といった動的データが公開されている．

そこで，本研究では，従来の渋滞検知の方法にかかるコストの削減を目的として，バスの運行データと機械学習を組み合わせた自動渋滞検知の手法を考える．バスは，渋滞が深刻な都市部に多く走っている上，規則的なダイヤで運行しているため，渋滞を検知しやすい．しかし，乗客の乗降やダイヤ調整などの停車により，バスの動きと実際の交通の流れには差異が生じ，渋滞検知における精度の低さが課題となっている[1]．本稿では，オープンデータである都バスの運行データを用い，バスが定期的に走行する渋谷から池袋までの明治通りの区間を対象に，一定の時間間隔毎に「渋滞」「非渋滞」の 2 値分類を行い，渋滞検知の精度向上を目指す．

従来の研究としては，交通状況の調査を目的として走行する車（プローブ車両）を用いた交通予測に関する研究があり，交通速度の推定に関する研究[2][3][4]と渋滞検知に関する研究[5][6][7]に分類できる．これらの研究では，主に車両の GPS 軌道データを用いている．これらの従来研究に対して，本稿ではオープンデータとして公開されているバスの停留所発車時刻からの渋滞検知を試みる．筆者が調べた限り，発車時刻のデータを利用した渋滞検知の研究は存在せず，初めての試みとなる．

以下，2 節で関連研究を紹介し，3 節で提案手法について説明し，4 節で評価実験を行い，5 節でまとめる．

## 2. 関連研究

本節では，関連研究について述べる．

### 2.1 プローブ車両を用いた交通速度推定

Samal ら[2]は，2017 年，バスのリアルタイム GPS 軌道データを利用した交通速度を推定する手法を提案した．一般的にバスの運行が少ない時間帯は推定精度が低くなるため，Samal らは，従来注目されていなかったバスの運転手情報や気象データなどの外部要因に着目し，k-means 法を用いた類似の運行データのクラスタリングを行い，予測精度を改善した．交通速度の RMSE(平均二乗偏差)は，クラスタリングを適用しない場合は 4.0～4.5 マイル/時であったのに対し，適用後は，

[1] https://www.mlit.go.jp/road/ir/ir-perform/h18/07.pdf

2.9～3.3 マイル/時となることを示した．

Kyaw ら[3]は，2018 年，バスの GPS 軌道データに加え，道路上にあるレストラン，学校，病院などの POI(Place of Interest)の数を特徴量として追加し，交通速度を推定する手法を提案した．ただし，論文内で，交通速度推定の精度の定量的な評価はされていない．

Gu ら[4]は，2020 年，バスの軌道データと交通 IC カードから得られるデータを利用した交通速度を推定する手法を提案した．Gu らは，(i)バスの停留所での停車時間，(ii)バスの平均速度，(iii)乗客がバス停間を移動するのに経過した時間の 3 つを入力としたニューラルネットワークを構築し，交通速度を推定した．本手法により推定した交通速度と，実際の交通速度に近いタクシーの平均速度を比較した結果，相関係数 0.94 の強い相関が得られたことを示した．

### 2.2 プローブ車両を用いた渋滞検知

Xu ら[5]は，2012 年，複数台のバスの平均移動時間を用いた渋滞検知の手法を提案した．Xu らは，渋滞検知の対象となる道路区間に対して「T 分以内に通過した全てのバスの平均移動時間(T-window average)」と「道路区間を最も直近に通過した N 台のバスの平均移動時間(N-window average)」を定義し，T-window average と N-window average それぞれが予め個別に設定した閾値を超えた場合，渋滞が発生することをシミュレーションによって示した．

Wang ら[6]は，2013 年，機械学習を用い，1 台のプローブ車両から渋滞を検知する手法を提案した．Wang らは，プローブ車両の平均速度を位置毎に離散化し，特徴ベクトルとして利用した．ランダムフォレスト，AdaBoost，サポートベクターマシンを用い，それぞれ 91.59%，89.43%，87.86%の Accuracy を達成し，閾値で判別する従来手法より優れていることを示した．

Carli ら[7]は，2015 年，バスの GPS 軌跡データを用いた交通渋滞の自動検知手法を提案した．Carli らは，バスの平均速度，バスの加速度の分散の 2 乗，バスが閾値以上の速度で走行している時間の割合などの指標に注目し，これらの指標の異常を検知することにより，信号機故障による交通渋滞を検知できることを示した．

### 2.3 関連研究のまとめ

本節では，プローブ車両データを用いた交通予測に関する研究を紹介した．これらの関連研究では，車両の GPS 軌道データを交通予測に用いている．しかし，一般に公開されるオープンデータには，リアルタイムの GPS 軌道データは存在せず，オープンデータを用いた渋滞検知を行うためには，より簡便なリアルタイムデータを用いた手法の研究が必要となる．

## 3. 提案手法

### 3.1 概要

本節では，バスの各停留所の発車時刻のデータを用いた渋滞検知の手法を提案する．通常の渋滞検知では，渋滞の定義に基づき，渋滞検知対象区間を走行する車両が「時速 10km 以下で継続的に走行している状態」であるかどうかで渋滞を判断する．提案手法においても，連続する 2 つの停留所の発車時刻の差をバスの移動時間とし，移動時間を距離で割ることによりバスの速度を求めることができる．しかし，個々のバスの平均速度に差があり，同じ道路を走行してもバスによる分散が大きいことや，停留所での停車時間にも分散があるため，単純にバスの速度を計算するだけでは，精度が低くなる[1]．そこで，速度だけでなく，平時と比べたときのバスの移動時間や，近傍の時間帯や停留所区間でのデータも検知に用いることで，分類精度の向上を目指す．

### 3.2 特徴量抽出

提案手法の説明で用いる記号を表 3.1 と図 3.1 に示す．

表 3.1　特徴量抽出に用いる記号の定義表

| 記号 | 定義 |
|---|---|
| $b_i$ | $i$ 番目の停留所 |
| $s_i$ | $i$ 番目の停留所区間( $b_i$ ～ $b_{(i+1)}$ 間) |
| $l_i$ | $s_i$ の距離 |
| $N_{t,i}$ | 時間帯 $t$ に $s_i$ を走行するバスの台数 |
| $B_{t,i,j}$ | 時間帯 $t$ に $s_i$ を走行する $j$ 台目のバス |
| $td_{t,i,j}$ | 時間帯 $t$ に $b_i$ を $B_{t,i,j}$ が出発した時刻 |

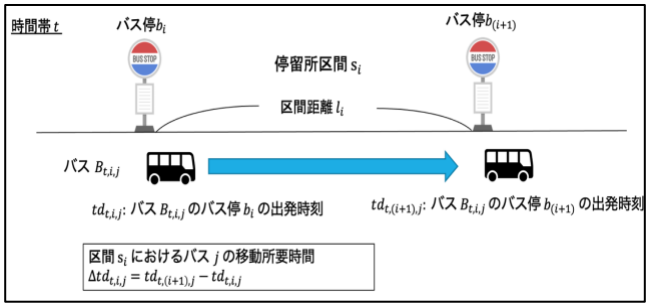


図 3.1　特徴量抽出に用いる記号の図解

渋滞検知は，一定時間間隔で分割した時間帯毎に，2 つの隣接する停留所区間単位で行うとする．今，時間帯の総数を $T$ とし，時間帯 $t\ (1 \le t \le T)$ における，2 つの隣接する停留所$b_i, b_{(i+1)}$間の渋滞検知を考える．停留所 $b_i, b_{(i+1)}$ 間を停留所区間 $s_i$ と定義する．時間帯 $t$ において，停留所区間 $s_i$ を走行するバスの総数を $N_{t,i}$ 台としたとき，そのうちの $j$ 番目のバス $B_{t,i,j}$ (ただし，$1 \le j \le N_{t,i}$) が停留所 $b_i$ を発車した時刻を $td_{t,i,j}$ とおく．このとき，バス $B_{t,i,j}$ が時間帯 $t$ に停留所区間 $s_i$ を走行するのに要した時間を $\Delta td_{t,i,j}$ とおくと，$\Delta td_{t,i,j} = td_{t,(i+1),j} - td_{t,i,j}$ である．ここで，$N_{t,i}$ 台のバス

が時間帯 $t$ に停留所区間 $s_i$ を走行するのに要した平均時間を $\Delta td_{t,i}$ とおくと，$\Delta td_{t,i}$ は式(1)で表される．

$$\Delta td_{t,i} = \frac{\sum_{j=1}^{N_{t,i}} \Delta td_{t,i,j}}{N_{t,i}} \quad (1)$$

ここで，停留所区間 $s_i$ の距離を $l_i$ とおく．$N_{t,i}$ 台のバスが時間帯 $t$ に停留所区間 $s_i$ を走行するのに要した平均速度を $v_{t,i}$ とおくと，$v_{t,i}$ は式(2)で表される．

$$v_{t,i} = \frac{\Delta td_{t,i}}{l_i} \quad (2)$$

また，$\Delta td_{t,i}$ の Z スコアを $z_{t,i}$ とすると，$z_{t,i}$ は式(3)~(5)で計算される．まず，式(3)で $s_i$ における $\Delta td_{t,i}$ の平均値 $\mu_i$ を計算する．次に，式(4)で $s_i$ における $\Delta td_{t,i}$ の標準偏差 $\sigma_i$ を計算する．最後に，式(5)で $\mu_i$, $\sigma_i$ を用いて $z_{t,i}$ を計算する．

$$\mu_i = \frac{\sum_{t=1}^{T} \Delta td_{t,i}}{T} \quad (3)$$

$$\sigma_i = \sqrt{\frac{\sum_{t=1}^{T} \left(\Delta td_{t,i} - \mu_i\right)^2}{T}} \quad (4)$$

$$z_{t,i} = \frac{\Delta td_{t,i} - \mu_i}{\sigma_i} \quad (5)$$

最後に，$\Delta td_{t,i} - \mu_i$ を $l_i$ で正規化した値を $d_{t,i}$ とする．$d_{t,i}$ は式(6)で計算される．

$$d_{t,i} = \frac{\Delta td_{t,i} - \mu_i}{l_i} \quad (6)$$

本手法で用いる特徴量を表 3.2 に示す．

表 3.2　本手法で用いる特徴量

| | | | |
|---|---|---|---|
| ① $v_{t,i}, z_{t,i}, d_{t,i}$ そのもの | $v_{t,i}$ | $z_{t,i}$ | $d_{t,i}$ |
| ②過去 2 つの時間区分での $v_{t,i}, z_{t,i}, d_{t,i}$ | $v_{(t-1),i}$, $v_{(t-2),i}$ | $z_{(t-1),i}$, $z_{(t-2),i}$ | $d_{(t-1),i}$, $d_{(t-2),i}$ |
| ③前，前々停留所区間における $v_{t,i}, z_{t,i}, d_{t,i}$ | $v_{t,(i-1)}$, $v_{t,(i-2)}$ | $z_{t,(i-1)}$, $z_{t,(i-2)}$ | $d_{t,(i-1)}$, $d_{t,(i-2)}$ |

まず，上記で計算した $v_{t,i}, z_{t,i}, d_{t,i}$ の 3 つの変数を特徴量として用いる．$v_{t,i}$ は，バスの速度である．速度は，渋滞の定義にも用いられていることから，渋滞検知に有用であると考えられる．また，$z_{t,i}, d_{t,i}$ は，どちらも当該区間の移動に平時と比べて時間がかかるかを表しており，$z_{t,i}, d_{t,i}$ の値が大きければ，渋滞の可能性が高いと考えられる．

また，渋滞は，時間的または空間的に連続して発生する可能性が高いと考えられる．そこで，$v_{t,i}, z_{t,i}, d_{t,i}$ そのもの以外に，表 3.2 の②，③に示す変数も特徴量として用いることで，精度の向上を目指す．

### 3.3　分類手法

収集したバスの発車時刻データから 3.2 で示した 15 個の特徴量を抽出し，ランダムフォレスト，AdaBoost，XGBoost，サポートベクターマシン(SVM)を用い，各時間帯のバス停留所区間において「渋滞」「非渋滞」の 2 値分類を行う．ここで，「渋滞」とは，車が時速 10km 以下で継続的に走行している状態と定義する．学習器は，全ての停留所区間に対して 1 つだけおく方法と，各停留所区間に 1 つずつおく方法の 2 つの方法を考える．学習器は何れも Python のライブラリである scikit-learn を用いて実装を行う．

## 4. 評価実験

本節では，「全ての停留所区間に対し 1 つの分類器で渋滞検知を行った場合」と「各停留所区間に対し個別の分類器で渋滞検知を行った場合」の実験結果を示す．また，最後にデータの前処理と限定の有効性を示す．

### 4.1　渋滞検知の対象区間

本実験で渋滞検知を行う対象区間を図 4.1 に示す．

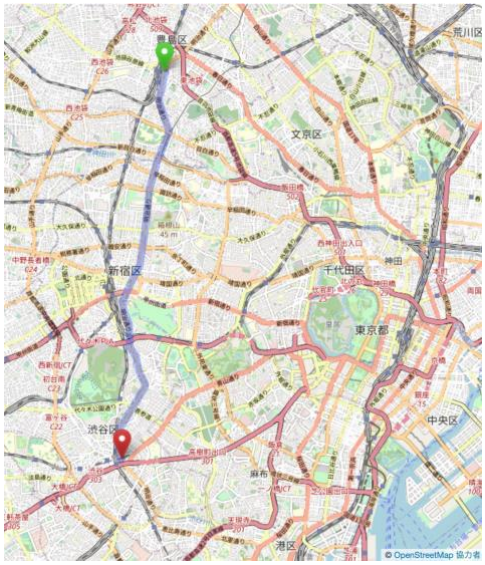


図 4.1　渋滞検知対象区間（OpenStreetMap より）

図 4.1 で示した区間には，東京都交通局が運行する都営バス池 86 系統が，1 時間に 3~5 本の頻度で走行している．本実験の渋滞検知の対象区間は，池 86 系統の停留所のうち，明治通り沿いの「池袋駅東口」～「渋谷駅東口」間とし，それ以外の明治通り沿いに無い停留所区間は渋滞検知の対象から除外する．

また，都営バスには平日，土曜，休日の 3 種類のダイヤが存在するが，本実験では平日ダイヤの運行データのみを使用する．

### 4.2　データセット

#### 4.2.1　バスの運行データの取得方法

バスの運行データは，公共交通オープンデータ協議会が主催する「東京公共交通オープンデータチャレンジ」[2]の API から取得した．本 API は，指定の URL にアクセスすると，公共交通機関に関する JSON 形式の

[2] https://tokyochallenge.odpt.org/index.html

データを返す．表 4.1 に，本研究でアクセスした URL と，返されるデータの内容を示す．

表 4.1　アクセスした URL とデータの内容

| URL | データの内容 |
|---|---|
| https://api-tokyochallenge.odpt.org/api/v4/odpt:Bus?odpt:busroute=odpt.Busroute:Toei.Ike86&acl:consumerKey={MY_CONSUMER_KEY} | 都バス池 86 系統の各停留所の発車時刻データ |

※表中の URL の{MY_CONSUMER_KEY}には，本 API のアクセストークンが入る．ただし，API の利用ガイドラインに従い非開示としている．

本実験では，表 4.1 の URL に定期的にリクエストを行い，継続的にデータを取得した．なお，API は直近のバス通過時間のみを返すため，30 秒毎に API にアクセスを行い，バス通過を見過ごすことがないようにした．取得したデータは整形したのち，CSV ファイルに書き込み保存した．表 4.2 に保存された CSV ファイルの一部を示す．

表 4.2　バスの運行データの CSV ファイル(一部)

| Date | BusNumber | PassPole | PassTime |
|---|---|---|---|
| 2021-11-07 | D360 | Omotesando.297.4 | 09:09:57 |
| 2021-11-07 | D304 | SodaiRiko.1051.1 | 09:09:15 |
| 2021-11-07 | D360 | JingumaeItchome.758.1 | 09:10:48 |
| 2021-11-07 | D304 | GakushuinJoshiDaigaku.856.3 | 09:10:33 |
| 2021-11-07 | D304 | TakadanobabaNichome.857.1 | 09:11:28 |

表 4.2 の「Date」は日付，「BusNumber」はバスの車両番号，「PassPole」は停留所名，「PassTime」は停留所の発車時刻をそれぞれ表す．例えば，表 4.2 の最初のレコードでは，車両番号 D360 のバスが，「表参道」の停留所を 2021 年 11 月 7 日午前 9 時 9 分 57 秒に発車したことを示す．

次に，表 4.2 のデータを用い，2 つのバス停留所間の移動時間をまとめたデータベースを作成した(表 4.3)．

表 4.3　2 バス停間の移動時間のデータベース(一部)

| Date | BusNumber | From Pole | To Pole | From Time | To Time | DifSec |
|---|---|---|---|---|---|---|
| 2021-11-07 | D360 | Omotesando.297.4 | JingumaeItchome.758.1 | 09:09:57 | 09:10:48 | 51 |
| 2021-11-07 | D304 | SodaiRiko.1051.1 | GakushuinJoshiDaigaku.856.3 | 09:09:15 | 09:10:33 | 78 |
| 2021-11-07 | D304 | GakushuinJoshiDaigaku.856.3 | TakadanobabaNichome.857.1 | 09:10:33 | 09:11:28 | 55 |

表 4.3 の「FromPole」は 1 つ目の停留所の名前を表し，「ToPole」は 2 つ目の停留所の名前を表す．また，「FromTime」は 1 つ目の停留所を出発した時刻，「ToTime」は 2 つ目の停留所を出発した時刻をそれぞれ表す．「DifSec」の単位は秒であり，「FromTime」と「ToTime」の時刻の差を表す．

### 4.2.2　渋滞ラベル(正解データ)の取得方法

渋滞ラベルの正解データは，NAVITIME[3]の Web ページからスクレイピングすることにより取得した．NAVITIME では，駅やバス停，地名などのスポットの周辺の渋滞情報を表示するサービスを提供しており，本研究では，池 86 系統のすべてのバス停付近の渋滞情報を自動取得し，CSV ファイルに保存した(表 4.4)．

表 4.4　スクレイピングした渋滞情報(一部)

| Date | Time | From Lat | From Lon | ToLat | ToLon |
|---|---|---|---|---|---|
| 2021-11-07 | 09:00 | 35.656833 | 139.704624 | 35.65899 | 139.702670 |
| 2021-11-07 | 09:00 | 35.658481 | 139.702886 | 35.657101 | 139.704283 |

表 4.4 の「Date」と「Time」は渋滞が観測された日付と時刻をそれぞれ表す．「FromLat」と「FromLon」は，渋滞の始点の緯度と経度をそれぞれ表し，「ToLat」と「ToLon」は，渋滞の終点の緯度と経度をそれぞれ表す．

表 4.4 のデータと，池 86 系統のバス停の緯度・経度を照らし合わせることで，渋滞が発生した停留所区間と時間帯を把握することができる．

### 4.2.3　データセットの作成

渋滞検知に必要なデータセット（表 4.5）の作成方法を以下に示す．

表 4.5　作成したデータセット(一部)

| Date | From Time | To Time | From Pole | To Pole | DifSec | IsJam |
|---|---|---|---|---|---|---|
| 2021-11-07 | 09:00:00 | 09:19:59 | Omotesando.297.4 | JingumaeItchome.758.1 | 51 | 0 |
| 2021-11-07 | 09:00:00 | 09:19:59 | SodaiRiko.1051.1 | GakushuinJoshiDaigaku.856.3 | 78 | 1 |

本実験では，時間区間を 20 分毎に分割し，各停留所区間で「渋滞」「非渋滞」の 2 値分類を行う．表 4.5 の「Date」，「FromTime」，「ToTime」は 20 分毎の時間区間を，「FromPole」「ToPole」は停留所区間を示す．「DifSec」は，バスが停留所間を走行するのに経過した秒数であり，4.2.1 のデータから取得される．同一時間区間内に複数の「DifSec」のデータが存在する場合は，複数の「DifSec」のデータの平均値が選ばれる．「IsJam」は，渋滞が発生したかどうかを示す正解ラベルを表しており，4.2.2 のデータから取得される．「IsJam」の値が「0」の場合，渋滞が発生していないことを表し，「1」の場合，渋滞が発生したことを表す．

データの収集は，2021 年 11 月 7 日～2021 年 12 月

[3] https://www.navitime.co.jp/

10 日の約 1 ヶ月間にわたって行われた．表 4.6 に，最終的に得られたデータの個数を,停留所区間別の渋滞・非渋滞データの内訳と共に示す.

表 4.6　収集したデータ数

| 始点 | 終点 | 渋滞 | 非渋滞 | 計 |
|---|---|---|---|---|
| 渋谷駅東口 | 宮下公園 | 3 | 810 | 813 |
| 宮下公園 | 神宮前六丁目 | 16 | 772 | 788 |
| 神宮前六丁目 | 表参道 | 8 | 867 | 875 |
| 表参道 | 神宮前一丁目 | 0 | 910 | 910 |
| 神宮前一丁目 | 千駄ヶ谷小学校前 | 0 | 980 | 980 |
| 千駄ヶ谷小学校前 | 北参道 | 2 | 986 | 988 |
| 北参道 | 千駄ヶ谷五丁目 | 0 | 967 | 967 |
| 千駄ヶ谷五丁目 | 新宿四丁目 | 143 | 792 | 935 |
| 新宿四丁目 | 新宿伊勢丹前 | 359 | 568 | 927 |
| 新宿伊勢丹前 | 日清食品前 | 12 | 897 | 909 |
| 日清食品前 | 東新宿駅前 | 126 | 807 | 933 |
| 東新宿駅前 | 大久保通り | 35 | 897 | 932 |
| 大久保通り | 新宿コズミックセンター前 | 29 | 844 | 873 |
| 新宿コズミックセンター前 | 早大理工前 | 50 | 813 | 863 |
| 早大理工前 | 学習院女子大学前 | 140 | 754 | 894 |
| 学習院女子大学前 | 高田馬場二丁目 | 124 | 799 | 923 |
| 高田馬場二丁目 | 学習院下 | 0 | 936 | 936 |
| 学習院下 | 千登世橋 | 0 | 912 | 912 |
| 千登世橋 | 東京音楽大学前 | 9 | 881 | 890 |
| 東京音楽大学前 | 南池袋三丁目 | 16 | 184 | 200 |
| 南池袋三丁目 | 池袋駅東口 | 19 | 198 | 217 |
| 池袋駅東口 | 南池袋三丁目 | 5 | 474 | 479 |
| 南池袋三丁目 | 東京音楽大学前 | 7 | 486 | 493 |
| 東京音楽大学前 | 千登世橋 | 3 | 899 | 902 |
| 千登世橋 | 学習院下 | 2 | 945 | 947 |
| 学習院下 | 高田馬場二丁目 | 3 | 958 | 961 |
| 高田馬場二丁目 | 学習院女子大学前 | 14 | 870 | 884 |
| 学習院女子大学前 | 早大理工前 | 22 | 799 | 821 |
| 早大理工前 | 新宿コズミックセンター前 | 28 | 832 | 860 |
| 新宿コズミックセンター前 | 大久保通り | 34 | 892 | 926 |
| 大久保通り | 東新宿駅前 | 4 | 946 | 950 |
| 東新宿駅前 | 日清食品前 | 12 | 963 | 975 |
| 日清食品前 | 新宿伊勢丹前 | 4 | 977 | 981 |
| 新宿伊勢丹前 | 新宿四丁目 | 109 | 792 | 901 |
| 新宿四丁目 | 千駄ヶ谷五丁目 | 9 | 925 | 934 |
| 千駄ヶ谷五丁目 | 北参道 | 0 | 943 | 943 |
| 北参道 | 千駄ヶ谷五丁目 | 1 | 961 | 962 |
| 千駄ヶ谷五丁目 | 神宮前一丁目 | 12 | 944 | 956 |
| 神宮前一丁目 | 表参道 | 178 | 767 | 945 |
| 表参道 | 神宮前六丁目 | 5 | 951 | 956 |
| 神宮前六丁目 | 神南一丁目 | 11 | 936 | 947 |
| 神南一丁目 | 渋谷駅西口 | 46 | 913 | 959 |
| 渋谷駅西口 | 渋谷駅東口 | 213 | 747 | 960 |
| 全区間合計 | | 1,813 | 35,494 | 37,307 |

### 4.3　実験 1: 全ての停留所区間での分類

実験 1 では，全ての停留所区間に対し 1 つの分類器を用いた分類を行う．用いるデータ数は，表 4.6 の全区間合計で示すように，37,307 個(渋滞 1,813 個，非渋滞 35,494 個)である．また，学習データとテストデータの分け方は，時系列順で前半 80%を学習データ，後半 20%をテストデータとする.

#### 4.3.1　ベースライン手法

ベースライン手法として，渋滞の定義で用いられる速度の閾値のみで渋滞検知を行う．速度の閾値は，速度を 0.1[km/h]ごとに変化させ，学習データで渋滞検知をした場合に F1 スコアが最も高くなる値にチューニングする．表 4.7 に本実験の結果の混同行列を，表 4.8 に accuracy，precision，recall，F1 スコアと，その際の速度の閾値を示す.

表 4.7 実験 1 の混同行列（ベースライン手法）

| | | 予測 Positive | 予測 Negative |
|---|---|---|---|
| 実際 | Positive | TP: 147 | FN: 226 |
| | Negative | FP: 392 | TN: 6,697 |

表 4.8 実験 1 の結果（ベースライン手法）

| 速度の閾値[km/h] | accuracy | precision | recall | F1 スコア |
|---|---|---|---|---|
| 5.3 | 0.917 | 0.273 | 0.394 | 0.322 |

#### 4.3.2　提案手法

提案手法では，全ての特徴量を使用して学習した状態を基準とし，ある 1 つの特徴量を除いて学習した状態の F1 スコアと比べることで，各特徴量の重要度を算出し, F1 スコアが最も高くなるように特徴量の選択を行う．特徴量選択後の accuracy，precision，recall，F1 スコアを表 4.9 に，特徴量の順位を表 4.10 に示す.

表 4.9 実験 1 の結果（提案手法)

| アルゴリズム | accuracy | precision | recall | F1 スコア |
|---|---|---|---|---|
| ランダムフォレスト | 0.953 | 0.623 | 0.177 | 0.276 |
| AdaBoost | 0.953 | 0.606 | 0.177 | 0.274 |
| XGBoost | 0.953 | 0.588 | 0.206 | 0.306 |
| SVM | 0.954 | 0.716 | 0.129 | 0.218 |

表 4.10　実験 1 の特徴量順位（提案手法）

| 順位 | ランダムフォレスト | AdaBoost | XGBoost | SVM |
|---|---|---|---|---|
| 1 | $v_{(t-2),i}$ | $d_{t,i}$ | $d_{(t-2),i}$ | $v_{t,(i-2)}$ |
| 2 | $v_{t,(i-1)}$ | $z_{t,(i-1)}$ | $v_{t,(i-2)}$ | $z_{t,(i-2)}$ |
| 3 | $z_{t,(i-1)}$ | $v_{(t-1),i}$ | $v_{(t-2),i}$ | $z_{t,(i-1)}$ |
| 4 | $z_{t,(i-2)}$ | $d_{(t-1),i}$ | $z_{t,i}$ | $d_{t,(i-2)}$ |
| 5 | $v_{t,(i-2)}$ | $d_{(t-2),i}$ | $z_{t,(i-1)}$ | $d_{t,(i-1)}$ |
| 6 | $v_{(t-1),i}$ | $z_{t,(i-2)}$ | $d_{(t-1),i}$ | $d_{(t-2),i}$ |
| 7 | $d_{(t-1),i}$ | $v_{t,i}$ | $d_{t,(i-1)}$ | $d_{(t-1),i}$ |
| 8 | $z_{(t-1),i}$ | $v_{t,(i-2)}$ | $v_{(t-1),i}$ | $d_{t,i}$ |
| 9 | $z_{t,i}$ | $v_{(t-2),i}$ | $z_{t,(i-2)}$ | $v_{t,(i-1)}$ |
| 10 | $v_{t,i}$ | $z_{(t-1),i}$ | $d_{t,(i-2)}$ | $v_{(t-2),i}$ |
| 11 | $z_{(t-2),i}$ | $d_{t,(i-2)}$ | $z_{(t-2),i}$ | $v_{(t-1),i}$ |
| 12 | $d_{t,(i-2)}$ | $v_{t,(i-1)}$ | $z_{(t-1),i}$ | $v_{t,i}$ |
| 13 | $d_{t,(i-1)}$ | $d_{t,(i-1)}$ | $v_{t,i}$ | $z_{(t-2),i}$ |
| 14 | $d_{t,i}$ | $z_{(t-2),i}$ | $v_{t,(i-1)}$ | $z_{(t-1),i}$ |
| 15 | $d_{(t-2),i}$ | $z_{t,i}$ | $d_{t,i}$ | $z_{t,i}$ |

特徴量選択の結果，F1 スコアが最も高くなったのは，ランダムフォレスト，AdaBoost，XGBoost では上位 14 個，サポートベクターマシンでは上位 12 個の特徴量を使用したときであった．

また，表 4.9 で得られた結果のうち，F1 スコアが最も高かったアルゴリズムである XGBoost の混同行列を表 4.11 に示す．

表 4.11　XGBoost を用いた提案手法の混同行列

| | | 予測 Positive | Negative |
|---|---|---|---|
| 実際 | Positive | TP: 77 | FN: 296 |
| | Negative | FP: 54 | TN: 7,035 |

表 4.8(ベースライン手法)と表 4.9(提案手法)を比較すると，提案手法の全てのアルゴリズムについて，recall，F1 スコアはベースライン手法が提案手法を上回り，accuracy, precision は提案手法がベースライン手法を上回った．

### 4.4　実験 2: 各停留所区間での分類

停留所区間毎に 1 つの分類器を用いた分類を考える．本実験で対象とする停留所区間は，渋滞データ数が学習を行う上で十分な数(100 以上とする)存在する区間とする．実験 2 の対象となる停留所区間と，各停留所区間における渋滞データ数，非渋滞データ数を表 4.12 に示す.また,学習データとテストデータの分け方は，4.3 と同様に，時系列順で前半 80%を学習データ，後半 20%をテストデータとする．

表 4.12　実験 2 の対象区間とデータ数

| 停留所区間 | 渋滞 | 非渋滞 |
|---|---|---|
| 学習院女子大学～高田馬場二丁目 | 124 | 799 |
| 神宮前一丁目～表参道 | 178 | 767 |
| 日清食品前～東新宿駅 | 126 | 807 |
| 千駄ヶ谷五丁目～新宿四丁目 | 143 | 792 |
| 渋谷駅西口～渋谷駅東口 | 213 | 747 |
| 新宿伊勢丹前～新宿四丁目 | 109 | 792 |
| 新宿四丁目～新宿伊勢丹前 | 359 | 568 |
| 早大理工前～学習院女子大学 | 140 | 754 |

#### 4.4.1　ベースライン手法

4.3.1 と同様に，ベースライン手法として，速度の閾値のみで渋滞検知を行う．表 4.13 にベースライン手法を用いた際の accuracy, precision, recall, F1 スコアと，その際の速度の閾値を示す．

表 4.13　実験 2 の結果（ベースライン手法）

| 停留所区間 | 速度の閾値 [km/h] | accuracy | precision | recall | F1 スコア |
|---|---|---|---|---|---|
| 学習院女子大学～高田馬場二丁目 | 8.7 | 0.827 | 0.571 | 0.343 | 0.429 |
| 神宮前一丁目～表参道 | 10.7 | 0.550 | 0.581 | 0.272 | 0.370 |
| 日清食品前～東新宿駅 | 12.3 | 0.711 | 0.217 | 0.122 | 0.156 |
| 千駄ヶ谷五丁目～新宿四丁目 | 8.7 | 0.775 | 0.778 | 0.368 | 0.500 |
| 渋谷駅西口～渋谷駅東口 | 3.0 | 0.609 | 0.796 | 0.375 | 0.510 |
| 新宿伊勢丹前～新宿四丁目 | 6.0 | 0.641 | 0.407 | 0.183 | 0.253 |
| 新宿四丁目～新宿伊勢丹前 | 4.5 | 0.602 | 0.667 | 0.563 | 0.611 |
| 早大理工前～学習院女子大学 | 12.0 | 0.821 | 0.630 | 0.436 | 0.515 |
| 平均 | 8.2 | 0.692 | 0.581 | 0.333 | 0.418 |

#### 4.4.2　提案手法

4.3.2 と同様の方法で特徴量選択を行い，特徴量選択後の F1 スコアが最も高かったアルゴリズムであるランダムフォレストの結果について，特徴量選択後の accuracy, precision, recall, F1 スコアを表 4.14 に示す．

表 4.14　実験 2 の結果（提案手法）

| 停留所区間 | accuracy | precision | recall | F1 スコア |
|---|---|---|---|---|
| 学習院女子大学～高田馬場二丁目 | 0.897 | 0.583 | 0.333 | 0.424 |
| 神宮前一丁目～表参道 | 0.772 | 0.500 | 0.163 | 0.246 |
| 日清食品前～東新宿駅 | 0.882 | 0.571 | 0.174 | 0.267 |
| 千駄ヶ谷五丁目～新宿四丁目 | 0.877 | 0.571 | 0.593 | 0.582 |
| 渋谷駅西口～渋谷駅東口 | 0.740 | 0.474 | 0.184 | 0.265 |
| 新宿伊勢丹前～新宿四丁目 | 0.851 | 0.500 | 0.111 | 0.182 |
| 新宿四丁目～新宿伊勢丹前 | 0.715 | 0.724 | 0.632 | 0.675 |
| 早大理工前～学習院女子大学 | 0.899 | 0.846 | 0.407 | 0.550 |
| 平均 | 0.829 | 0.596 | 0.325 | 0.399 |

表 4.13(ベースライン手法)と表 4.14(提案手法)の区間平均を比較すると，recall, F1 スコアはベースライン手法が提案手法を上回り，accuracy, precision は提案手法がベースライン手法を上回った．これは実験 1 と同様の傾向である．また，F1 スコアの最高値は提案手法の「新宿四丁目～新宿伊勢丹前」間で 0.675 であった．

### 4.5　追加実験 1: データの前処理の適用

精度向上のため，学習前にデータの前処理を行った場合の結果を示す．

#### 4.5.1　データの前処理の方法

データの前処理は，以下の 2 つである．

① バス専用レーンの設定時間帯のデータの削除

渋滞検知の対象となるバス路線上でバス専用レーンが設定されている場合，的確に渋滞を判断できないため，専用レーンが設定されている時間帯のデータを削除する．渋滞検知の対象となっている区間(明治通り)では，午前 7 時 30 分から午前 9 時の間，バス専用レーンが設定されており，当該時間帯のデータを削除する．

② 停留所区間特有のノイズの削除

提案手法では，2 つの連続する停留所 $b_i, b_{(i+1)}$ の発車時刻の差分から区間内のバスの移動時間を計算しているが，この移動時間に含まれるノイズの削除を考える．ここで，ノイズとは，信号待ちや乗客の乗降などの要因により，バスが停止している時間を意味し，その時間の長さは停留所区間で特有の値であると考えられる．

また，渋滞とは，「車が 10km/h 以下で継続的に走行している状態」と定義されるが，実際のバス運行データには，計算上のバスの速度が 10km/h 以下であるにも関わらず，上記のノイズが原因で「非渋滞」と判定されているデータが存在する．そこで，「渋滞時の当該区間を走るバスの速度は必ず 10km/h 以下」という前提のもと，時系列順で前半 20%のデータから，「計算上の速度が 10km/h 以下となっている非渋滞データ」を抽出する．そして，上記で抽出したデータに対して，速度を 10km/h にするために移動時間から減算すべき時間を求める．そして，この値の平均値を区間毎に計算し，区間特有のノイズと考える．最後に，ノイズを求めるのに使わなかった時系列順で後半 80%のデータについて，求めたノイズを移動時間から減算することで，ノイズの影響を排除したデータセットを得る．

### 4.5.2 全ての停留所区間での分類

全ての停留所区間に対し 1 つの分類器を用いた分類結果を示す．学習データとテストデータの分け方は，前処理で用いなかった時系列順で後半 80%のデータのうち，前半 80%を学習データ，後半 20%をテストデータとする．4.3.2 と同様の方法で特徴量選択を行い，特徴量選択後の accuracy，precision，recall，F1 スコアを表 4.15 に示す．

表 4.15　追加実験 1 の結果(全区間)

| アルゴリズム | accuracy | precision | recall | F1 スコア |
|---|---|---|---|---|
| ランダムフォレスト | 0.948 | 0.560 | 0.140 | 0.224 |
| AdaBoost | 0.949 | 0.571 | 0.167 | 0.258 |
| XGBoost | 0.950 | 0.610 | 0.190 | 0.290 |
| SVM | 0.949 | 0.689 | 0.092 | 0.163 |

表 4.9(前処理無し)と表 4.15(前処理あり)を比較すると，最も F1 スコアが高いアルゴリズムの F1 スコアは 0.306 から 0.290 に低下した．全ての停留所区間に対し 1 つの分類器を用いた分類を行う場合は，前処理がモデルの精度向上に寄与していないことがわかる．

### 4.5.3 各停留所区間での分類

次に，停留所区間毎に 1 つの分類器を用いた分類結果を示す．対象とする停留所区間は，実験 2 と同様の区間とする．学習データとテストデータの分け方は，4.5.2 と同様とする．4.3.2 と同様の方法で特徴量選択を行い，特徴量選択後の F1 スコアが最も高かったアルゴリズムである XGBoost の結果について，accuracy，precision，recall，F1 スコアを表 4.16 に示す．

表 4.16　追加実験 1 の結果(各区間)

| 停留所区間 | accuracy | precision | recall | F1 スコア |
|---|---|---|---|---|
| 学習院女子大学～高田馬場二丁目 | 0.909 | 0.692 | 0.474 | 0.562 |
| 神宮前一丁目～表参道 | 0.733 | 0.435 | 0.270 | 0.333 |
| 日清食品前～東新宿駅 | 0.895 | 0.455 | 0.333 | 0.385 |
| 千駄ヶ谷五丁目～新宿四丁目 | 0.867 | 0.593 | 0.640 | 0.615 |
| 渋谷駅西口～渋谷駅東口 | 0.773 | 0.833 | 0.238 | 0.370 |
| 新宿伊勢丹前～新宿四丁目 | 0.801 | 0.455 | 0.179 | 0.256 |
| 新宿四丁目～新宿伊勢丹前 | 0.727 | 0.718 | 0.699 | 0.708 |
| 早大理工前～学習院女子大学 | 0.889 | 0.714 | 0.435 | 0.541 |
| 平均 | 0.824 | 0.612 | 0.409 | 0.471 |

表 4.14(前処理無し)と表 4.16(前処理あり)を比較すると，平均 F1 スコアが 0.399 から 0.471 に向上した．停留所区間毎に 1 つの分類器を用いた分類を行う場合，前処理がモデルの精度向上に寄与したことがわかる．

## 4.6 追加実験 2: データ限定の適用

同一時間帯に 1 台分のバスのデータしかない場合，4.5.1 で述べたノイズの影響が大きく，渋滞の推定が困難であると考えられる．そこで，精度向上のため，同一時間帯内に複数台のバスのデータがある場合のみのデータに限定し，学習を行った場合の結果を示す．

### 4.6.1 全ての停留所区間での分類

全ての停留所区間に対し 1 つの分類器を用いた分類結果を示す．使用したデータ数は，4.3 で使用した全データ 37,307 個(渋滞 1,813 個，非渋滞 35,494 個)のうち，18,321 個(渋滞 1,017 個，非渋滞 17,304 個)である．学習データとテストデータの分け方は，4.3 と同様，時系列順で前半 80%を学習データ，後半 20%をテストデータとする．4.3.2 と同様の方法で特徴量選択を行い，特徴量選択後の accuracy，precision，recall，F1 スコアを表 4.17 に示す．

表 4.17　追加実験 2 の結果(全区間)

| アルゴリズム | accuracy | precision | recall | F1 スコア |
|---|---|---|---|---|
| ランダムフォレスト | 0.947 | 0.698 | 0.202 | 0.313 |
| AdaBoost | 0.943 | 0.560 | 0.193 | 0.287 |
| XGBoost | 0.946 | 0.602 | 0.257 | 0.360 |
| SVM | 0.943 | 0.667 | 0.092 | 0.161 |

表 4.9(限定無し)と表 4.17(限定あり)を比較すると，最も F1 スコアが高いアルゴリズムの F1 スコアは 0.306 から 0.360 に向上した．全ての停留所区間に対し 1 つの分類器を用いた分類を行う場合，データの限定

がモデルの精度向上に寄与したことがわかる．

### 4.6.2 各停留所区間での分類

次に，停留所区間毎に 1 つの分類器を用いた分類結果を示す．対象とする停留所区間は，実験 2 と同様の区間とし，使用したデータ数を表 4.18 に示す．

表 4.18 追加実験 2 のデータ数(各区間)

| 停留所区間 | 渋滞 | 非渋滞 |
|---|---|---|
| 学習院女子大学～高田馬場二丁目 | 78 | 388 |
| 神宮前一丁目～表参道 | 97 | 366 |
| 日清食品前～東新宿駅 | 62 | 392 |
| 千駄ヶ谷五丁目～新宿四丁目 | 74 | 385 |
| 渋谷駅西口～渋谷駅東口 | 118 | 356 |
| 新宿伊勢丹前～新宿四丁目 | 62 | 396 |
| 新宿四丁目～新宿伊勢丹前 | 195 | 254 |
| 早大理工前～学習院女子大学 | 93 | 366 |

学習データとテストデータの分け方は，4.3 と同様，時系列順で前半 80%を学習データ，後半 20%をテストデータとする．4.3.2 と同様の方法で特徴量選択を行い，特徴量選択後の F1 スコアが最も高かったアルゴリズム(ランダムフォレスト)の結果について，accuracy，precision，recall，F1 スコアを表 4.19 に示す．

表 4.19 追加実験 2 の結果(各区間)

| 停留所区間 | accuracy | precision | recall | F1 スコア |
|---|---|---|---|---|
| 学習院女子大学～高田馬場二丁目 | 0.904 | 0.818 | 0.562 | 0.667 |
| 神宮前一丁目～表参道 | 0.710 | 0.286 | 0.190 | 0.229 |
| 日清食品前～東新宿駅 | 0.879 | 0.600 | 0.250 | 0.353 |
| 千駄ヶ谷五丁目～新宿四丁目 | 0.837 | 0.462 | 0.429 | 0.444 |
| 渋谷駅西口～渋谷駅東口 | 0.768 | 0.692 | 0.333 | 0.450 |
| 新宿伊勢丹前～新宿四丁目 | 0.772 | 0.222 | 0.125 | 0.160 |
| 新宿四丁目～新宿伊勢丹前 | 0.722 | 0.766 | 0.710 | 0.742 |
| 早大理工前～学習院女子大学 | 0.826 | 0.611 | 0.550 | 0.579 |
| 平均 | 0.802 | 0.557 | 0.394 | 0.453 |

表 4.14(限定無し)と表 4.19(限定あり)を比較すると，平均 F1 スコアが 0.399 から 0.453 に向上した．停留所区間毎に 1 つの分類器を用いた分類を行う場合においても，データの限定がモデルの精度向上に寄与したことがわかる．

## 5. まとめ

本稿では，バスの各停留所の発車時刻データのみを利用した渋滞検知の手法を提案した．本手法を用いて明治通り沿いの交通状況に対して「渋滞」「非渋滞」の 2 値分類を行った結果，全停留所に対して学習器を一つ用意した場合，F1 スコアは 0.306 となり，停留所区間毎に個別の学習器で学習をした場合，F1 スコアは平均 0.399，特定の区間においては F1 スコア 0.675 を得た．また，単一バスの運行データのみで渋滞検知する場合を除くと，特定の区間において F1 スコアを 0.742 へ向上することができた．

今後の課題としては，一部の停留所区間において高い F1 スコアが得られなかった原因を解明する必要がある．例えば，停留所におけるダイヤ調整の影響を排除すること等が考えられる．

## 参 考 文 献


[1] R. Hall, et al. "Buses as a traffic probe: Demonstration project", Transportation Research Record: Journal of the Transportation Research Board, no. 1731, pp. 96-103, 2000.

[2] C. Samal, F. Sun and A. Dubey, "SpeedPro: A Predictive Multi-Model Approach for Urban Traffic Speed Estimation," 2017 IEEE International Conference on Smart Computing (SMARTCOMP), 2017, pp. 1-6, doi: 10.1109/SMARTCOMP.2017.7947048.

[3] T. Kyaw, N. N. Oo and W. Zaw, "Estimating Travel Speed of Yangon Road Network Using GPS Data and Machine Learning Techniques," 2018 15th International Conference on Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2018, pp. 102-105, doi: 10.1109/ECTICon.2018.8619908.

[4] Y. Gu, Y. Wang, and S. Dong, “Public Traffic Congestion Estimation Using an Artificial Neural Network,” *ISPRS International Journal of Geo-Information*, vol. 9, no. 3, p. 152, 2020.

[5] Y. Xu, Y. Wu, J. Xu and L. Sun, "Efficient Detection Scheme for Urban Traffic Congestion Using Buses," 2012 26th International Conference on Advanced Information Networking and Applications Workshops, 2012, pp. 287-293, doi: 10.1109/WAINA.2012.62.

[6] C. Wang and H. Tsai, "Detecting urban traffic congestion with single vehicle," 2013 International Conference on Connected Vehicles and Expo (ICCVE), 2013, pp. 233-240, doi: 10.1109/ICCVE.2013.6799799.

[7] R. Carli, M. Dotoli, N. Epicoco, B. Angelico and A. Vinciullo, "Automated evaluation of urban traffic congestion using bus as a probe," 2015 IEEE International Conference on Automation Science and Engineering (CASE), 2015, pp. 967-972, doi: 10.1109/CoASE.2015.7294224.