明治大学総合数理学部

2020 年度

卒　業　研　究

ウイルスバスター **for Home Network** の調査研究

学位請求者　先端メディアサイエンス学科

平山夏輝

目次

第 1 章

はじめに

近年のネットワークの普及に伴い利用者の多様化がすすみ, 小さい子供から年配の方まで, 誰でも簡単にネットワークを活用できるようになっている.

しかしながら, 誰でも簡単にネットワークを活用できるようになっている反面, 子供のアクセス管理や, 危険なサイトへのアクセスなどのセキュリティ面での問題がある. 企業のネットワークであれば専用のファイアウォール (FW) を導入して, すべてのパケットを検査すればよいが, 商用のプロバイダから借りているルータを用いて接続している家庭内ネットワークで, それを FW に置き換えて自分で管理することは技術的に困難である.

そこで, トレンドマイクロ社は, 家庭内ネットワークの単一のノードとして接続するだけで, 全ネットワークのノード管理を実現する製品であるウイルスバスター home Security[1] (以下, VBHS とする) を開発している.

本稿では, VBHS がどの様な仕組みで専用の FW なしで家庭内ネットワークのパケットを中断検査しているのか, いくつかの実験を行い, その仕組みを解析し, その課題を検討する.

第 2 章

ウイルスバスター for Home Network

2.1　概要

VBHS は, トレンドマイクロ株式会社が販売している家庭内ネットワーク保護用デバイスである. 保護対象となるネットワーク端末に VBHS の 1 台を接続し, 所有しているスマートフォンに管理アプリをインストールするだけでそのネットワーク端末につながっている全ての端末を管理する. 次の様な様々な機能を有しており, 家庭毎にカスタマイズすることができる.

1. 家庭内ネットワークに新しく参入してきたデバイスに対して, ネットワークに接続させるか否かを確認する機能 ［アクセス管理機能］
2. 家庭内ネットワークの中にある不審なデバイスをネットワークから遮断する機能 ［デバイス遮断機能］
3. ネットワークブラウザのフィルタリング機能 ［ペアレンタルコントロール］

2.2　ARP Spoofing

VBHS の原理は ARP Spoofing[2] により実現されていると考えられる. 全ての IP アドレスはデータリンク層で ARP(Address Resolution Protocol) によって動的に構成される ARP テーブルに従って MAC アドレスに変換される. VBHS は, これを利用して各種フィルタリング機能を実現していると考えられる.

2.3　機能の動作確認

本稿では, VBHS のフィルタリング機能について動作を確認する.

2.3.1　URL フィルタリング

ネットワークブラウザを閲覧している時の機能として特定のサイトへの通信を拒否する URL フィルタリングがある.

管理アプリに接続を拒否したいサイトの URL を入力すると, そのサイトへの通信が遮断される. URL の入力には, 文字, 数字および −(ハイフン) なので, https 通信はフィルタリングできない欠点がある.

実際にフィルタリングを指定して, http://windy.mind.meiji.ac.jp/ をブロックした結果を図 1 に示す.

フィルタリングを行った web サイトの画面が「このページは見られません」という特定の記述に変わって

図 2.1　URL フィルタリングの適用結果

いる. この機能は, 後述するカテゴリフィルタリングのようにカテゴリ毎ではなく, 任意の URL を自由に指定できるという点で自由度は高い.

2.3.2　カテゴリフィルタリング

ある特定のカテゴリに属する web サイトの表示をなくす機能にカテゴリフィルタリングがある. 「アダルト」, 「薬物」, 「不適切な広告」などのカテゴリが用意されており, フィルタリングしたいカテゴリを選択するとそのカテゴリに属する web サイトがブロックされる. 実際に「お酒」というカテゴリを選択し, 朝日ビールのサイトである https://www.asahibeer.co.jp/ へアクセスした結果を図 2 に示す.

図 2.2　カテゴリフィルタリング「お酒」の適用結果

第 3 章

調査

3.1　実験環境

本研究には Windows10 が搭載された PC *C* を有線ネットワーク内に設置し, VBHS *B* を接続した. 実験環境図を図 3 に示す.

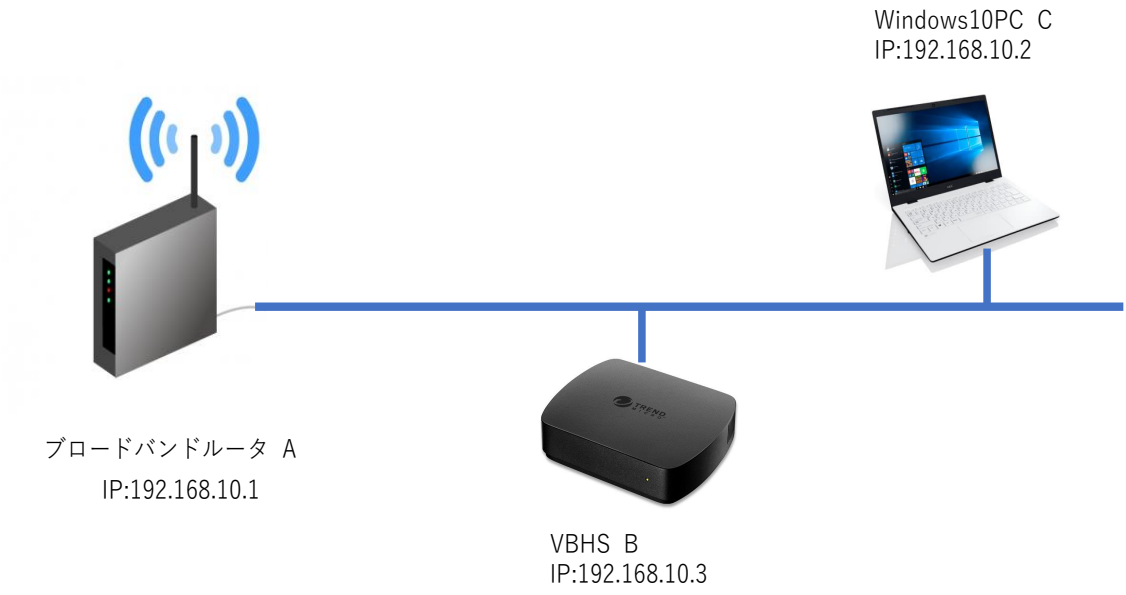

図 3.1　実験環境図

3.2　実験目的と方法

VBHS がいかにしてデバイスの遮断を行っているのかを調べるために, フィルタリングされる対象である PC *C* が属するサブネットにおける ARP パケットを Wireshark を用いて観測し, VBHS の接続前後の ARP

テーブルの変化を調べる. もしも, VBHS *B* がルータ *A* への MAC アドレスをスプーフィングする ARP を送信していれば, ARP Spoofing を用いている証拠である.

3.3 実験結果

VBHS *B* にて PC *C* を遮断対象に指定したところ, 任意に開いたウェブサイトである https://www.google.com/ が図 4 のような表示になり, 対象 PC *C* はネットワーク接続が行われていない挙動を見せた.

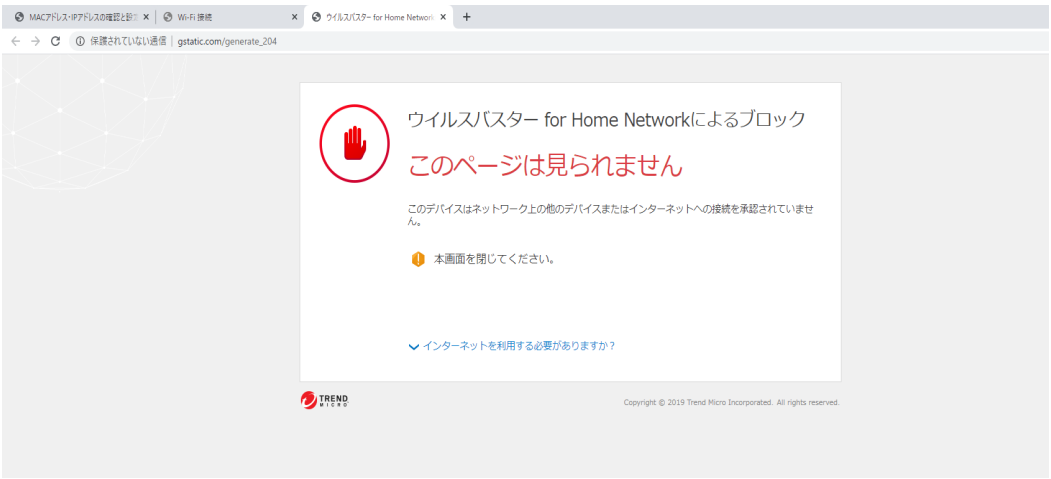

図 3.2 遮断された PC *C* のウェブ画面

まず, ネットワーク内に VBHS *B* が接続されていない通常時の PC *C* から観測した ARP パケットを図 5 に示す.

図 5 の時間の単位は秒であり, 初めのパケットが観測されてからの相対時間を表している. 図 5 から約 30 秒毎にブロードバンドルータから PC *C* に ARP パケットが送信されていることがわかる.

次に, VBHS がを接続した時の ARP パケットの観測結果を図 6 に, ARP テーブルの変化を表 1 に示す.

ARP Spoofing が攻撃対象に行われると, 不正な ARP パケットがブロードキャストされ, その対象の ARP テーブルが書き換えられる [3].

表 1 から, PC *C* の ARP テーブルのルータ *A* の IP アドレス 192.168.10.1 の MAC アドレスが VBHS *B* の MAC アドレスである 88-6a-e3-5a-81-ee に変化していることがわかる. VBHS *B* の IP アドレスである 192.168.10.3 が追加されており, ARP テーブルが書き換えられている.

図 6 の AlphaNet は VBHS *B* を示している. 図 6 で遮断対象である PC *C* に 10 個以上もの ARP パケットが約 3 秒毎に送信されている. このパケットの送信元は VBHS *B* であり, ルータ *A* の IP アドレスであった

図 3.3　VBHS*B* 接続前の PC *C* から観測した ARP パケット (通常時)

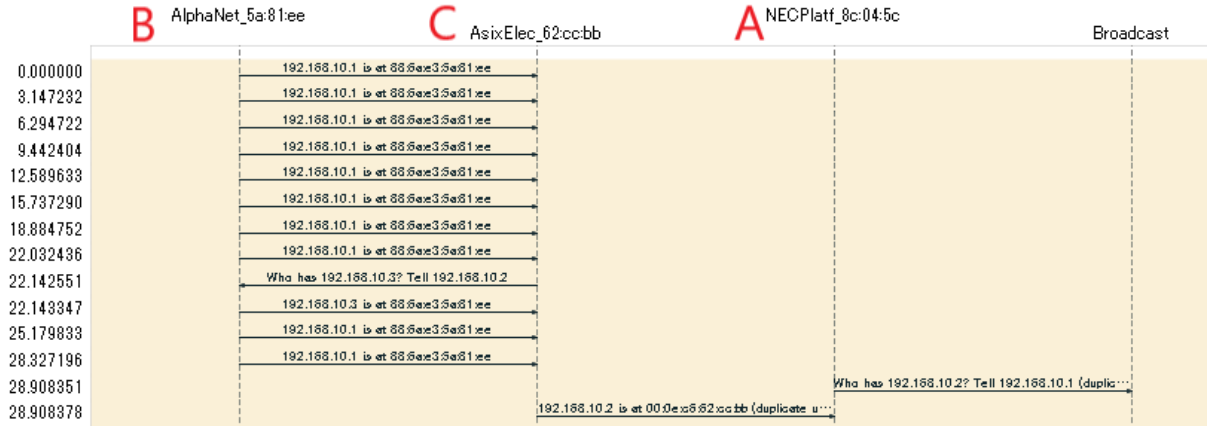

図 3.4　遮断された PC *C* から観測した ARP パケット

表 3.1　ARP テーブルの変化

接続前				接続後		
ホスト	IP アドレス	MAC アドレス	種類	IP アドレス	MAC アドレス	種類
ルータ *A*	192.168.10.1	98-f1-99-8c-04-5c	動的	192.168.10.1	**88-6a-e3-5a-81-ee**	動的
PC *C*	192.168.10.2	00-0e-c6-62-cc-bb	動的	192.168.10.2	00-0e-c6-62-cc-bb	動的
VBHS *B*				192.168.10.3	88-6a-e3-5a-81-ee	動的
ブロードキャスト	192.168.10.255	ff-ff-ff-ff-ff-ff	静的	192.168.10.255	ff-ff-ff-ff-ff-ff	静的

192.168.10.1 の MAC アドレスは VBHS *B* の MAC アドレスである, という ARP パケットを PC *C* に送り続けている. 28.908 秒後にルータ *A* から正しい ARP パケットが送信されている. しかし, 正規は約 30 秒おきなのに対して, VBHS は 3 秒おきであり, ほとんどの場合 B のアドレスに書き換えられてしまう.

以上の結果から VBHS は ARP Spoofing を用いてデバイスの管理を行っていることがわかった.

第 4 章

おわりに

本研究では VBHS を使用し, その機能を調査し, その原理を明らかにした. 本研究では, VBHS の機能である URL フィルタリング機能において, https 通信を遮断できなかったので, その通信の解析を今後の課題とする.

謝辞

本研究を行うにあたり，多くの方より御指導いただきました．特に，多大なる御指導を受け賜りました，指導教官である明治大学総合数理学部先端メディアサイエンス学科の菊池浩明教授に深く感謝申し上げます．また，研究の実験に協力して下さった菊池研究室の松本寛輝さん, 住友孝彰君, 研究室の皆様に深く感謝の意を表するとともに，謝辞とさせていただきます．

参考文献

[1] Trendmicro "https://www.trendmicro.com/ja_jp/forHome/products/vbhn.html" , 2020 年 12 月参照.

[2] 松藤央, 落合秀也, 江崎浩, “無線端末による ARP を用いたセグメント内の通信妨害攻撃とその対策”, マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO), 2018.

[3] Mahendra Data, “The Defense Against ARP Spoofing Attack Using Semi-Static ARP Cache Table”, Sustainable Information Engineering and Technology(SIET), pp.206-210 , 2018.

付録 A

AR モデルを用いた位置情報の推定

A.1　はじめに

近年のスマートデバイスの発展に伴い, 位置情報の取得が従来よりも容易かつ高精度になった. これらの位置情報から, 人が集まりやすい時間帯や場所を分析し, そこに目的のお店を出店するなど, マーケティングや地域振興などに活用されている.

しかしながら, 人の位置情報は時間軸によって変化するものであり, 予測することが困難であることが分かっている. センシング技術を用いて計測を試みるにはコストが膨大にかかってしまう上, プライバシーの問題もある.

そこで本研究では, 1 人の人がもつ位置情報と時間軸情報を用いて, その人の位置を表す情報を推定することを目的とする. しかし, この目的の為には次にあげる問題点が存在する. (1) 緯度経度情報に基づいて特定の人の経路を大極的にみることが困難である. (2) 緯度経度情報からその人の行動パターンを理解することは難しい. そこで問題点 (1) に対してはオープンソースのライブラリである folium を用いてインタラクティブな可視化を実現し, 行動パターンをより把握しやすくする. (2) に対しては, AR モデルによる推定を試みる.

A.2　提案手法

A.2.1　位置情報の可視化

本研究では, インタラクティブなマップ作成が可能な JavaScript のライブラリである Leaflet.js[1] と Python で Leaflet.js を利用するためのライブラリである folium[2] を組み合わせて位置情報の可視化を試みる. Python で入力・加工した位置情報を地図上にプロットすることによって移動履歴を俯瞰できるようにし, 特定のユーザの行動経路を抽出し, 頻出パターンを獲得するのを容易にする.

本システムを実装し, ユーザ 138 を可視化した例を図 1 に示す. ここで, 赤が始点, 緑が終点である.

また, 単位時間当たりの動きが大きいユーザと小さいユーザの動きの違いがわかる図を図 2 に示す. ユーザ 138 は青, ユーザ 3990 はピンク色のマーカーで示す.

A.2.2　位置情報の推定

本研究では時系列データの取り扱いに優れている自己回帰モデルである, *AR*(*AutoRegressive*) モデル [3] を用いる.

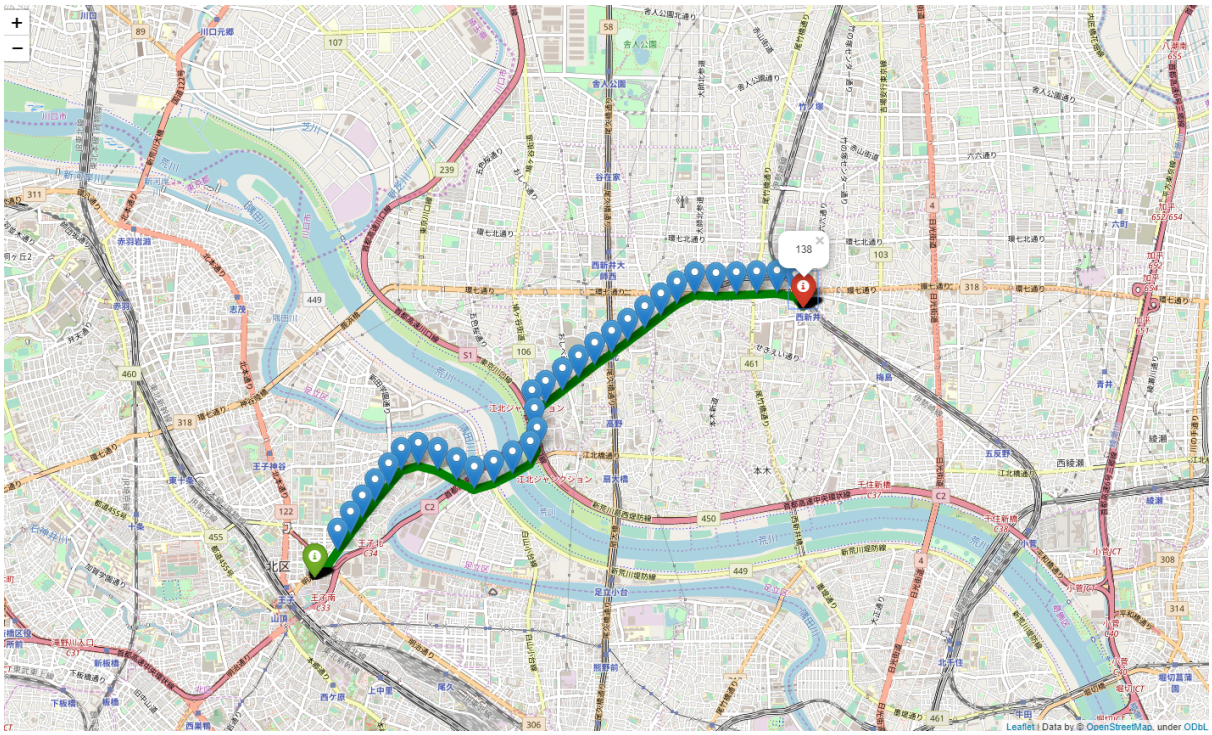

図 A.1　ユーザ 138 の位置経路のプロット例

n 階差分系列を求める計算式を以下に示す. p 時点前までのデータを用いてある時点のデータを表現する p 次 AR モデルは, t 時点におけるデータを y_t, 攪乱項を $\varepsilon_t, y_{t-1}, y_{t-2}, \cdots, y_1$ に対する最小二乗法によって求まる係数を ϕ_i とすると,

$$y_t = \phi_0 + \varepsilon_t + \sum_{i=1}^{p} \phi_i y_{t-i}$$

で与えられる. AR モデルは過去の p 時系列データから $p+1$ 時系列データを推定する.

A.3　実験方法

A.3.1　実験目的

本実験では, 各ユーザが 0 時から 24 時までの完全な位置情報を保持しているわけではなく, 電波状況などをにより一部欠損していることを想定し, 間欠的に取得された位置情報から, 欠損している位置情報を AR モデルで予測する. そして, AR モデルの位置情報の予測精度を明らかにすることを目的とする.

A.3.2　実験方法

24 時間分の位置情報を 7 時から 2 時間毎に 1 時間分 (y_t〜y_{t-12}) 与え, その情報に基づいて 1 時間先の位置情報 (y_{t+1}〜y_{t+12}) を推定する. これを 21 時まで繰り返す.

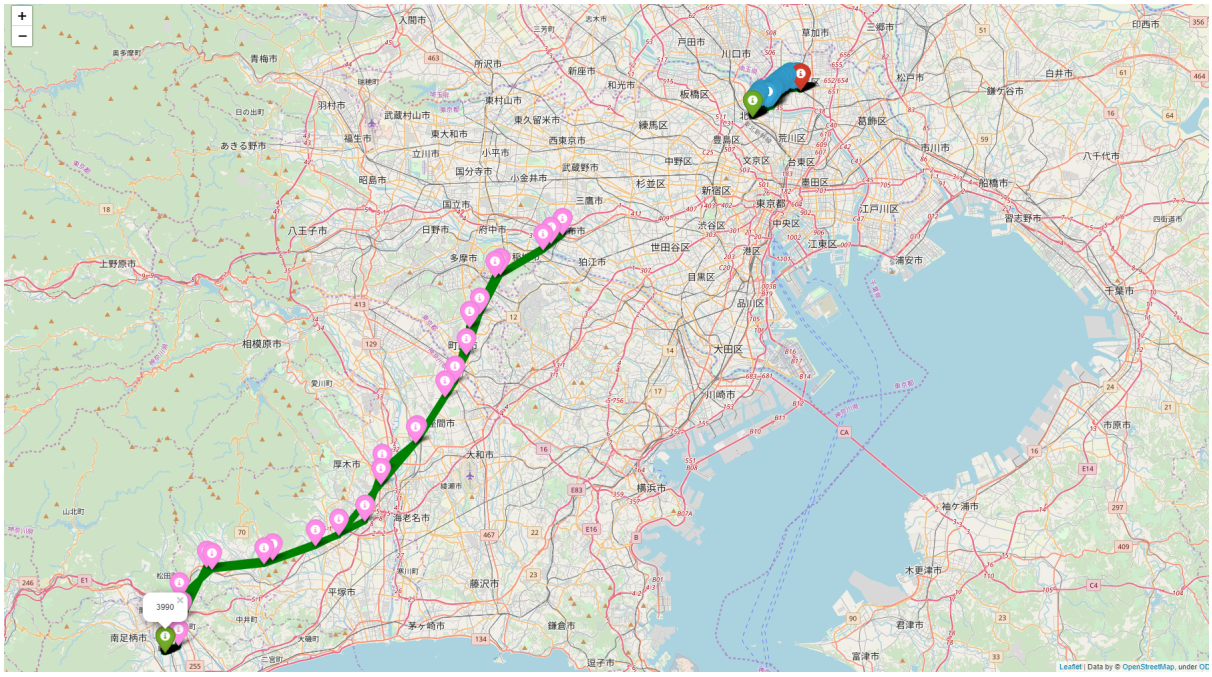

図 A.2　ユーザ 138 とユーザ 3990 の動きの比較

A.3.3　実験環境

本実験では, データの補間, AR モデルへの位置情報の入力・推定, folium を用いたユーザの位置情報の可視化をすべて Python 上で行った.

A.3.4　用いるデータセット

本研究では, 株式会社ナイトレイによって無料公開されている疑似人流データ [4] を使用する. このデータは株式会社ナイトレイが保有する SNS ベースの地域解析結果を参考に, 正確な道路ネットワークデータによる移動経路の補完や統計処理・ランダム化処理を行うことで, 大まかな人の流れをオープンな CSV データとして公開したものである. データセットは, ユーザ ID, 性別 (推定値), 時刻, 緯度, 経度, 滞在者カテゴリ (大分類), 滞在者カテゴリ (小分類), 滞在情報の 9 つの属性からなる. データセットの一部を表 1 に示す.

表 A.1　データセットの例 (一部)

uid	gender	timestamp	lat	lon	category1	category2	*STAY, MOVE*
10007	male	2013/7/7 23:00	35.54349	139.4472			STAY
10007	male	2013/7/7 23:05	35.54393	139.4467			MOVE
10007	male	2013/7/7 23:10	35.54441	139.4463			MOVE
10007	male	2013/7/7 23:15	35.54495	139.4463	food	Japanese	STAY

使用するデータは疑似人流データの 2013 年 7 月 1 日における関東圏ユーザ毎の 0 時から 24 時までの 5 分

毎の緯度経度情報であり, 1 日の時系列データである.

A.3.5 前処理

本疑似人流データは, 必ずしもユーザ全員が 0 時から 24 時間まで 5 分ごとの位置情報を所有しているわけではない. ユーザのもつ属性の中に状態情報があり, 位置情報が変化するときは $MOVE$, 位置情報が変化しない時間が始まる場合は $STAY$ を取る. $STAY$ から次の $MOVE$ が始まる時間帯まで補間する. 加工前と加工後のデータを表 2, 表 3 とする.

表 A.2　データ補間前

uid	gender	timestamp	lat	lon	$STAY, MOVE$
10015	male	2013/7/1 0:00	35.669963	139.767	STAY
10015	male	2013/7/1 0:20	35.66994	139.7665	MOVE
10015	male	2013/7/1 0:25	35.67025	139.7661	MOVE
10015	male	2013/7/1 0:30	35.67055	139.7658	MOVE

表 A.3　データ補間後

uid	gender	timestamp	lat	lon	$STAY, MOVE$
10015	male	2013/7/1 0:00	35.669963	139.767	STAY
10015	male	2013/7/1 0:05	35.669963	139.767	$STAY$
10015	male	2013/7/1 0:10	35.669963	139.767	$STAY$
10015	male	2013/7/1 0:15	35.669963	139.767	$STAY$
10015	male	2013/7/1 0:20	35.66994	139.7665	MOVE
10015	male	2013/7/1 0:25	35.67025	139.7661	MOVE
10015	male	2013/7/1 0:30	35.67055	139.7658	MOVE

A.3.6 位置情報の推定

本実験では, ユーザをランダムに 30 人ほどサンプリングし, AR モデルを用いて推定する.

A.3.7 実験結果

本実験ではユーザのもつ元の位置情報と予測した位置情報を比較し, 距離の誤差を算出する. 予測結果が分かるような図を図 3, 4 に示す. ここで, ユーザの元の緯度情報は青, 予測結果は赤で示してある.

7 時間を単位時間とし, その平均誤差が $5km$ を超えたユーザに対しては推定は失敗と定める. 推定結果を表 4 に示す. サンプリングした 30 人ユーザのうち, 9 人が推定成功した.

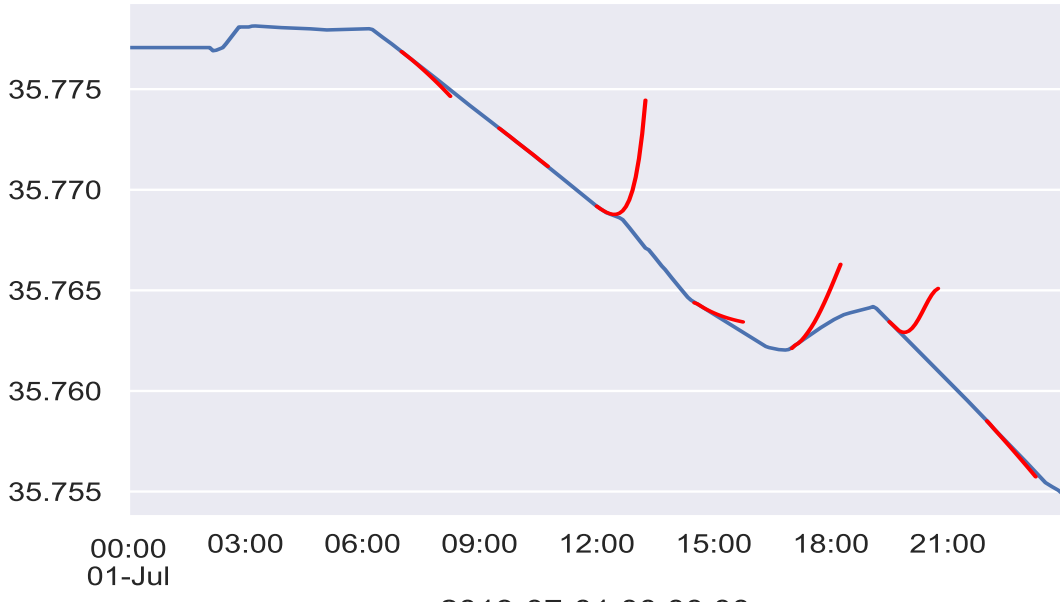

図 A.3　ユーザ 138 の緯度の推定結果

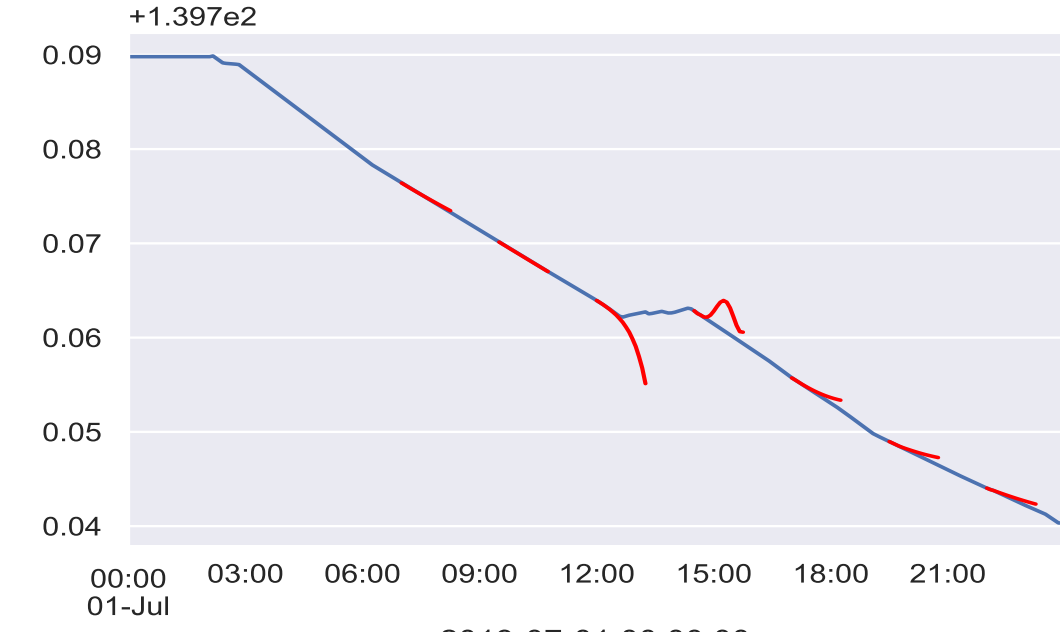

図 A.4　ユーザ 138 の経度の推定結果

表 A.4　各ユーザの推定結果

ユーザ	平均誤差 (*km*)	標準偏差	正誤
41	0.6	0.01	○
91	3	0.52	○
138	1	0.18	○
164	5	2.02	○
194	0.1	0.01	○
205	0.2	0.01	○
306	3	0.61	○
3076	3	0.56	○
4022	4	1.34	○
4	38	16.34	×
115	10	4.51	×
124	9	1.20	×
212	19	6.31	×
235	14	7.66	×
243	40	15.47	×
278	27	9.38	×
348	9	2.57	×
445	27	8.61	×
472	13	2.72	×
482	16	3.08	×
1439	40	14.43	×
3236	26	4.43	×
3369	16	2.40	×
3990	63	13.14	×
4010	6	0.93	×
4061	42	11.53	×
4199	47	15.73	×
4263	35	8.16	×
4343	30	9.06	×
4463	6	0.65	×

A.3.8　結果の可視化

本実験では AR モデルを用いてユーザの緯度・経度の情報二つについてそれぞれ推定を行った. 結果の可視化では, 推定によって得られたユーザの緯度・経度の情報に基づいてそれらをユーザの元の位置情報に加えて可視化を行う. 正しい位置情報と予測情報の差異を示す為に folium を用いて可視化をした結果を図 5 に示す. 正しい位置情報は青色, 推定された位置情報はピンク色のマーカーで示す.

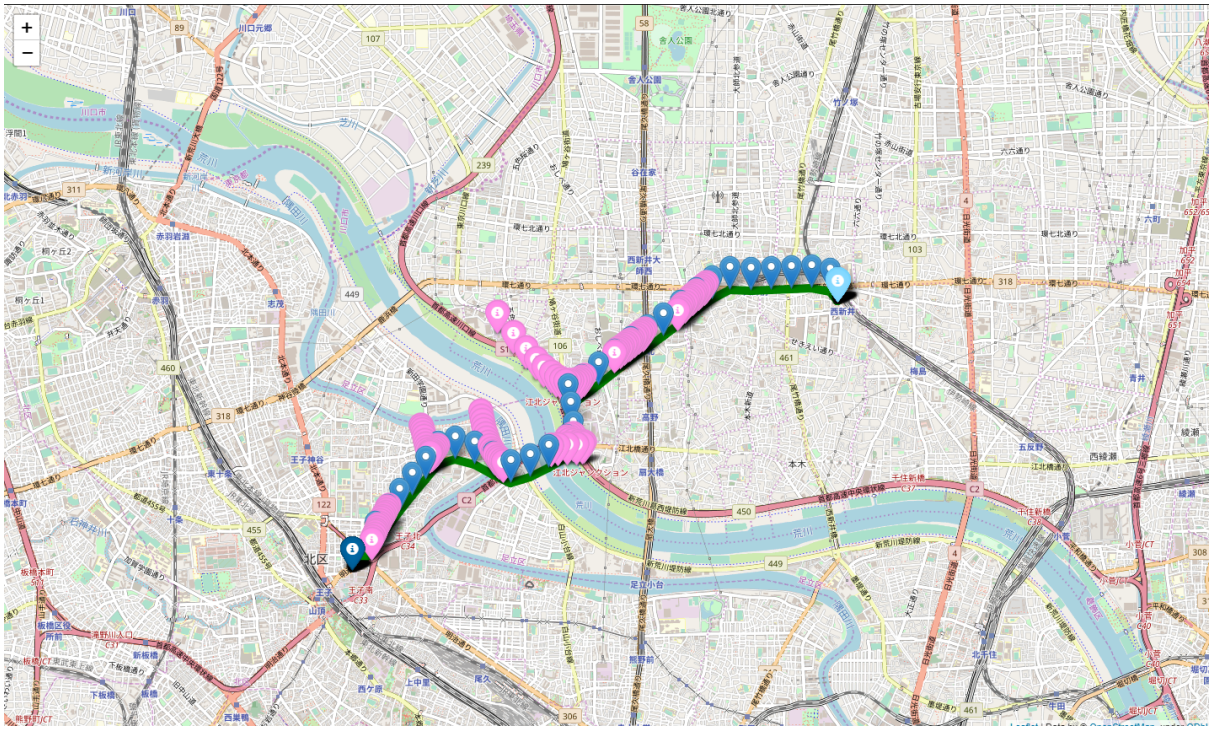

図 A.5　ユーザ 138 の位置情報と推定結果の比較

A.4　おわりに

本実験から, ユーザのもつ位置情報を正確に推定することは, 非常に困難であることが明らかになった. 図 2 と表 3 からわかるように, 推定に失敗したユーザは単位時間当たりの動きが成功したユーザのものより大きい. よって, 単位時間当たりの動きが大きいユーザほど推定が困難であることがわかる.

これを解決するには, 理想であればユーザの位置情報の常時取得が望ましいが, 前述した通りコスト面での問題があるので次のような対策が必要であると考えられる. さらに細かい単位での予測が効果的であると考えられる. 本実験では 1 時間前のデータを用いて 1 時間先のデータを推定したのでその推定数の多さにより誤差が大きくなってしまったと考えられる. よって予測の幅を細かくすることにより, ユーザが単位時間に大きく動いたとしても誤差が少なくなるように推定を行えば解決できるのではないだろうか.

参考文献

[1] Leaflet an open-source JavaScript library for mobile-friendly interactive maps (https://leafletjs.com, 2019 年 11 月参照)

[2] 柴田有基, 篠田広人, 難波英嗣, 石野亜耶, 竹澤寿幸, ” 観光の形態に基づいた旅行ブログエントリの分類と可視化”. 研究報告情報基礎とアクセス技術（2019-IFAT-135), pp.1-8

[3] 石井直宏, 岩田彰, 鈴村宣夫, ” 最尤法を適用した自己回帰モデルによる同定”, 名古屋工業大學學報, pp.421-428, 1978.

[4] 株式会社ナイトレイ, 東京大学 CSIS との研究活動成果として SNS 解析データを元とした「疑似人流データ」を無料公開, http://nightlay.jp/archives/1954, 2019 年 11 月参照