

# 有線通信におけるネットワークパラメータに基づく 深層学習を用いた帯域予測手法

小山内 遥香<sup>1</sup> 中尾 彰宏<sup>2</sup> 山本 周<sup>2</sup> 山口 実靖<sup>3</sup> 小口 正人<sup>1</sup>

概要: 通信障害は, 大規模災害による被災地内外の通信過多による輻輳, 同時に起こる OS アップデートや DDoS 攻撃など, 様々な原因で引き起こされる. これらの通信障害は起こってからでは, 対応が手遅れである場合が多く, 確度の高い予測をし, 事前に輻輳を抑制することが重要である. 網内における機械学習により, トラフィック集中を早期に検知し, 効率的に対応するための技術に期待が集まっている. 本論文では, 深層学習のモデルの一種である Recurrent Neural Network(RNN) を用いてトラフィック異状の情報を抽出し, トラフィック変動の兆候を掴むための手法を提案する.

## Prediction of Network Traffic Variations Based on Network Parameters in Wired Network by Deep Learning

HARUKA OSANAI<sup>1</sup> AKIHIRO NAKAO<sup>2</sup> SHU YAMAMOTO<sup>2</sup> SANEYASU YAMAGUCHI<sup>3</sup>  
MASATO OGUCHI<sup>1</sup>

### 1. はじめに

現在, スマートフォンや自動車, 家電など身の回りのあらゆるものがインターネットに接続されており, 通信は, 電気やガス, 水道と並ぶ生活を送るために必要不可欠なものとなっている. そのため, 突然発生する通信障害により通信が利用できなくなると人々を混乱に陥れる可能性が大きい.

通信障害は, 大規模災害による被災地内外からの通信過多による輻輳, DDoS 攻撃や同時に起こる OS アップデートなど, 様々な原因で引き起こされる. 通信障害が発生すると, メールや通話などの連絡手段の利用が不可能になったり, 金融取引や公共交通機関利用時の電子決済ができなくなるなど社会全体が大きな影響を受けてしまう. 特に, 災害時においては, 被災状況や安否の確認, 避難経路などの情報を得ることが難しくなり, 避難や救助活動を適切に行うことができなくなる可能性がある.

従来は, 通信障害の発生を検知したら経路やシステムの切り替えを行うなど, 通信障害が起こってから対処する仕

組みがとられてきた. しかし, 通信障害が起きてからの対処ではサーバや基地局の故障などの可能性があり, 復旧に時間やコストがかかる場合があるため, 対応が手遅れであることが多い. そのため, 確度の高いトラフィック変動予測をし, 事前に輻輳を抑制することが必要である.

また, 近年深層学習の技術が発達してきている. 深層学習は, 画像や音声の認識, 自動車の自動運転など様々な分野に応用され, 非常に高い解析性能により社会に多大な利益をもたらしている.

そのため, 網内における深層学習を用いたトラフィック変動の学習により, トラフィック集中を早期に検知し, 効率的に対応するための技術に期待が集まっている.

本論文では, 深層学習のモデルの一種である Recurrent Neural Network(RNN) を用いてトラフィック異状の情報を抽出し, トラフィック変動の兆候を掴むための手法の提案と作成した予測モデルの性能評価を行う.

### 2. 深層学習

深層学習とは, 機械学習の一種であるニューラルネットワークの階層を深めたアルゴリズムである. これを用いることにより, データの特徴をより深いレベルで学習するこ

<sup>1</sup> お茶の水女子大学

<sup>2</sup> 東京大学

<sup>3</sup> 工学院大学

とができる。

ニューラルネットワークとは、生物の脳の神経回路をモデルとしたアルゴリズムである。入力層、隠れ層、出力層を持ち、各層は複数のノードがエッジで結ばれる構造となっており、各エッジはそれぞれ重みを持っている。出力層からの出力と正解データの誤差を計算し、それを出力層から入力層へ逆に伝播させることにより重みを更新して学習を行う。

ニューラルネットワークは、画像データを扱うのに有効なもの、時系列データを扱うのに有効なもの、クラスタリングや次元削減を行うものなどがあり、それぞれ扱うデータの特徴によって学習結果の精度が変わることがある。

近年では、深層学習を用いて心疾患患者の医療情報を解析を行い、再入院のリスクを予測し、従来基準の2倍以上の患者の再入院の防止や医療費の低減を実現したり、リアルタイムな人口データを用いて30分先のタクシーの需要を92.9%の精度で予測し、売り上げの増加に貢献するなど、社会に多大な利益をもたらしている。

## 2.1 RNN

RNNは、時系列データの学習に用いられるネットワークで、以前に計算された情報を記憶することができる(図1)。しかし、系列が長くなり深いネットワークになると、誤差逆伝播のアルゴリズムでは勾配の消失、発散などの問題が生じ、実際には2, 3ステップ前までの記憶しか保持することができない。

この問題を解決するためにRNNを改良したモデルがLong Short-Term Memory(LSTM)であり、LSTMはRNNの一種である。LSTMはRNNに入力ゲート、出力ゲート、忘却ゲートを導入することにより、RNNでは扱うことができなかった長期依存を扱えるようになっている(図2)。

入力ゲートと出力ゲートは、必要な誤差信号だけが適切に伝播するようにゲートの開閉を行うために導入され、忘却ゲートは、入力系列パターンが変わったときに、一度学習した内容を忘れてノードの状態を一気に初期化するために導入された。

本研究で扱うパケットデータは時系列データであり、比較的長いパターンを学習する必要があるため、長期の時系列データの学習を行うのに有効であるLSTMを用いてネットワークトラフィックの予測を行なっている。

## 3. 関連研究

深層学習を用いて、株価変動や降水量の予測を行う手法が存在する[1][2][3]。

[1]では、LSTMを用いて株価の予測を行う手法が提案されている。この研究はLSTMを用いて時系列データの学習、予測を行なっている点で、本研究と類似している。

[2]では、深層学習のモデルであるCNNを用いて日経平

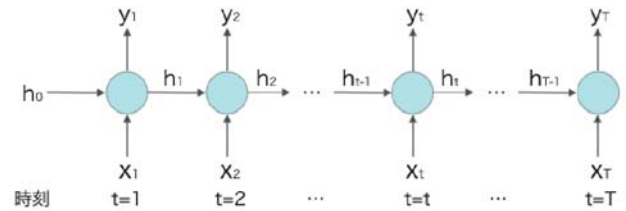


図1 RNNのネットワーク図

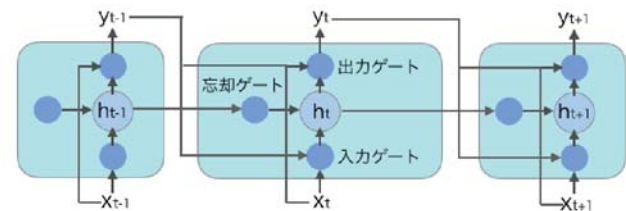


図2 LSTMのネットワーク図

均株価の騰落の予測を行う手法を提案している。CNNは、視覚野の特徴抽出の仕組みをモデル化したもので、画像識別を行うのに有効である。畳み込み層によって特徴量を抽出し、プーリング層によってノイズ処理を行うことによって局所ごとに特徴をまとめ、画像の特徴を学習する。株価データは時系列データであるため、株価データの予測においては時系列データを扱うのに有効なRNNやLSTMを使った予測手法が盛んであり、それと比較してCNNを用いた予測は少ない。

[3]では、CNNとLSTMを組み合わせたConvolutional LSTM(ConvLSTM)を用いて降水量の予測を行なっている。ConvLSTMは、時系列性と空間性の両方を持つ時空間データの予測に用いられるモデルである。降水量は、位置情報を持った時系列データで時空間データであるため、ConvLSTMを用いた手法は精度の高い予測を実現できる手法である。

本研究は、予測対象としてネットワークトラフィックを扱っており、また、将来的には予測結果を用いてネットワークの制御を行うことを目指している。

また、ネットワーク障害を検知するために、SNSによる集合知を用いた研究が存在する。[4]では、災害時に発生する通信障害を早期に検知する手法としてTwitterのツイート解析を用い、解析結果に基づいて自動的にネットワーク制御を行うシステムを構築している。ネットワーク障害検知システムのリアルタイム処理を行うことにより、通信障害の早期検知を実現している。この研究は、通信障害を検知する手法としてSNS情報を用いている点で、深層学習を使用して障害検知を行うことを目指す本研究とは異なる。

## 4. 関連技術

### 4.1 Chainer

Chainer[5]とは、深層学習を行うためのライブラリである。Chainerは、Pythonの制御構文をそのまま使用できるため単純なネットワーク構成から複雑なネットワーク構成まで直感的に記述できる。ネットワーク構築と学習を同時に行う方式がとられており、幅広い種類のニューラルネットワークに柔軟に対応できるため、Convolutional Neural Network(CNN)、RNNなど、様々なタイプのニューラルネットワークをシンプルに記述できる。また、GPUを利用して高速な計算を行うことが可能であり、計算が長時間になりがちな深層学習を効率良く行えるようになっている。

本研究ではChainerを使用して深層学習を行っている。

### 4.2 Wireshark

Wireshark[6]とは、ネットワークアナライザソフトウェアであり、ネットワークを流れるパケットをキャプチャして解析、表示するソフトウェアである。ネットワークを流れるパケットをリアルタイムに解析でき、解析可能なプロトコルは800以上と解析性能が非常に高い。PCやサーバが送受信するパケットをすべてキャプチャして可視化できるため、ネットワークにおける問題が発生した場合、問題を解決するのに役立つソフトウェアである。Wiresharkはオープンソースで開発されている。

### 4.3 iPerf

iPerf[7]は、ネットワークのスループットを測定するためのフリーソフトウェアである。サーバ、クライアント間で帯域などの条件を設定し、TCP、UDPのネットワーク転送性能を測定することができる。帯域を指定してトラフィックを発生させることができるため、ネットワーク負荷試験にも用いられるツールである。

### 4.4 カーネルモニタ

TCPにおいては、カーネル内部で定義された輻輳ウィンドウサイズ(cwnd)などの輻輳制御パラメータを用いて輻輳制御を行っている。しかし、カーネル内部の処理は通常バックグラウンドで進められているため、ユーザ空間からその処理の様子を監視することはできない。そこで、カーネル内部の情報を見るためにカーネルモニタというツールが開発された[8]。

## 5. 実験

本章では、iPerfを用いてトラフィックを発生させ取得したデータを用いた学習、予測性能評価の実験について説明する。

### 5.1 実験概要

本実験では、3台の実験用マシンを使用し、1台をサーバ、2台をクライアントとしてiPerfを用いてトラフィック発生を行う。サーバとクライアントはTCPで通信し、送受信されたパケットをサーバ側でキャプチャする。その際、カーネルモニタを用いて通信時のクライアントそれぞれのcwndを取得した。取得したデータを用いて、cwndを加えた場合と加えなかった場合でそれぞれ学習と性能評価を行なった。

表1 実験で用いた計算機の性能

OS	Ubuntu 16.04.2 LTS
CPU	SKL-SP 5115 10C/20T 2.4GHz x 2
GPU	GeForce GTX 1080TI x 4
Memory	DDR4-2666 ECC Register 192GB
HDD	1TB SATA x 2

表2 トラフィック発生に用いた計算機の性能

OS	Ubuntu 14.04.5LTS
CPU	Intel Xeon CPU E3-1270 V2 @ 3.50GHz
Memory	16Gbyte

学習に用いた計算機の性能を表1、トラフィック発生に用いた計算機の性能を表2に示す。

本実験では、サーバ・クライアント間にダミーネットを挟み、遅延を100ms、サーバ・ダミーネット間の帯域を70Mbpsとした(図3)。サーバ・クライアント間でのTCP輻輳制御アルゴリズムはRenoを用いた。

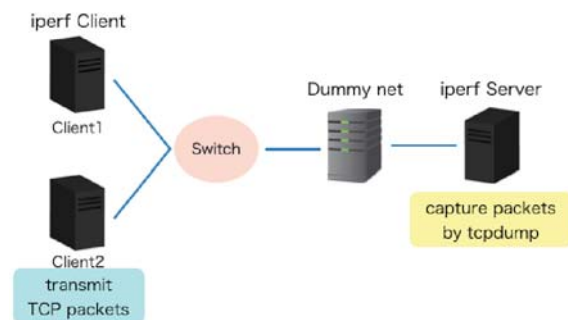


図3 実験環境

入力データにはt-9秒～t秒の10秒間に送受信したパケットから1秒おきに以下の5種の特徴量を導出したものとしており、詳細は以下である。

- パケット送受信時刻t
- パケットサイズの平均
- 時刻tに送受信したパケット数
- クライアント1のcwnd値(cwndありの場合)
- クライアント2のcwnd値(cwndありの場合)

正解データは時刻t+1秒に届いたパケット数を0～1に正規化した値である。また、学習時のepoch数は5000、隠れ層の数は49としている。

## 5.2 学習

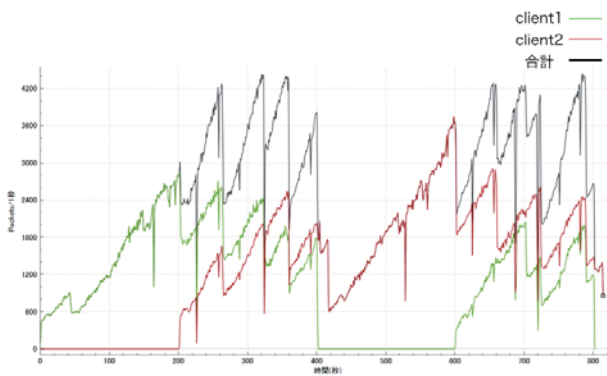


図 4 学習データ

図 4 は、学習データとして使用するトラフィックデータの packets 数の変動を表している。この学習データを用いて、cwnd を特徴量として含める場合と含めない場合で 2 種類のモデルを作成する。

## 5.3 性能評価

作成した学習モデルに 3 種類のバリデーションデータを入力し、性能評価を行う。評価には、平均平方二乗誤差 (Root Mean Square Error, RMSE) を使用する。

$$RMSE = \sqrt{\frac{1}{N} \sum (\hat{y}_{t+1} - y_{t+1})^2}$$

ここで、 $\hat{y}_{t+1}$  は  $t-9 \sim t$  までのデータを元に  $t+1$  での値を予測した値、 $y_{t+1}$  は実際に  $t+1$  でどのような値だったかを示し、RMSE は時刻  $t$  における予測値と時刻  $t$  における正解値の誤差を表す。

また、正解値の増加・減少をどの程度予測できるかを表す一致率も評価の指標として使用する。一致率は、 $\hat{y}_{t+1} - \hat{y}_t$  と  $y_{t+1} - y_t$  の正負が一致する割合として算出する [9]。

### 5.3.1 評価 1

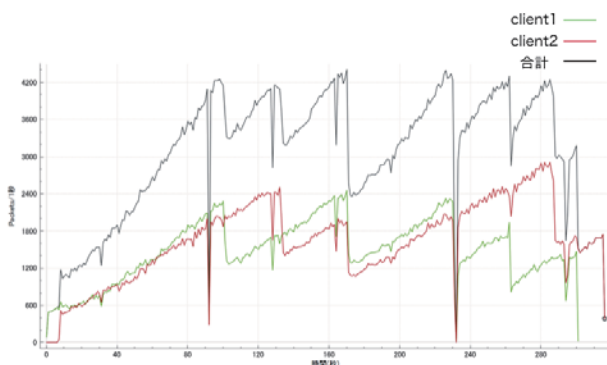


図 5 評価 1 のバリデーションデータ

図 5 は評価 1 で使用するバリデーションデータの packets 数の変動を示すグラフである。図 6 は cwnd なしの場合の学習モデル、図 7 は cwnd ありの場合の学習モデルにこの

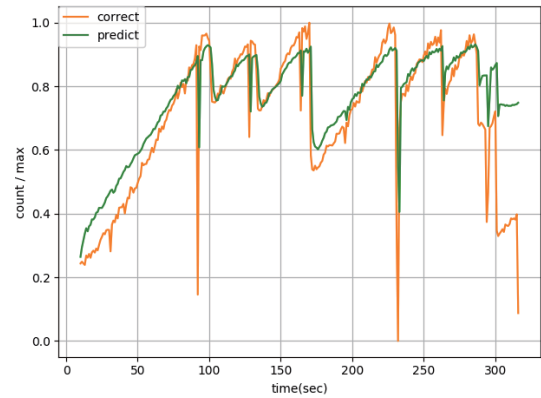


図 6 評価 1 の予測結果 (cwnd なし)

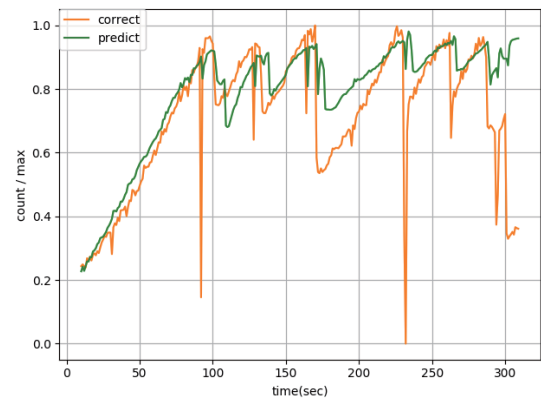


図 7 評価 1 の予測結果 (cwnd あり)

データを入力した予測結果である。RMSE は cwnd なしの場合が 0.1237、cwnd ありの場合が 0.1624 となり、一致率は cwnd なしの場合が 0.5318、cwnd ありの場合が 0.5719 となった。

### 5.3.2 評価 2

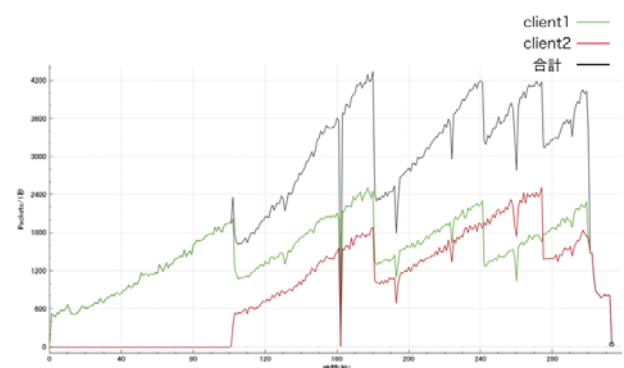


図 8 評価 2 のバリデーションデータ

図 8 は評価 2 で使用するバリデーションデータの packets 数の変動を示すグラフである。図 9 は cwnd なしの場合の学習モデル、図 10 は cwnd ありの場合の学習モデルにこのデータを入力した予測結果である。RMSE は cwnd なしの場合が 0.07456、cwnd ありの場合が 0.1680 となり、一致率は cwnd なしの場合が 0.5251、cwnd ありの場合が



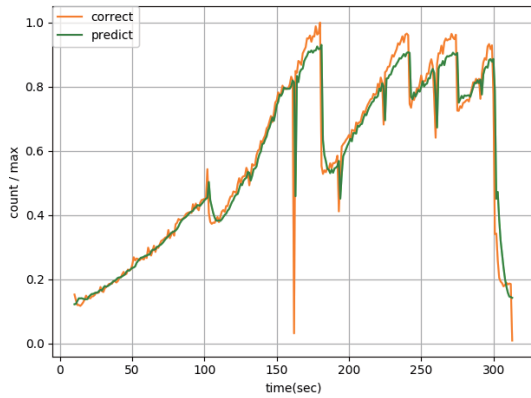


図 9 評価 2 の予測結果 (cwnd なし)

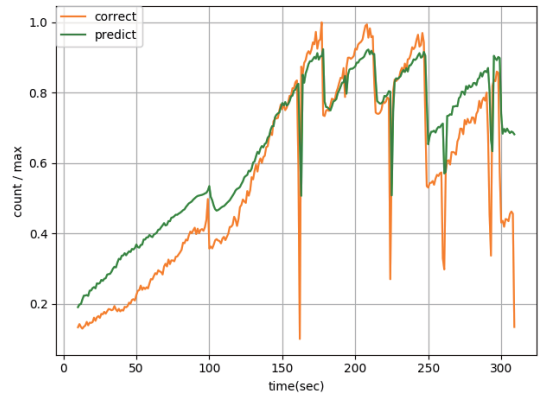


図 12 評価 3 の予測結果 (cwnd なし)

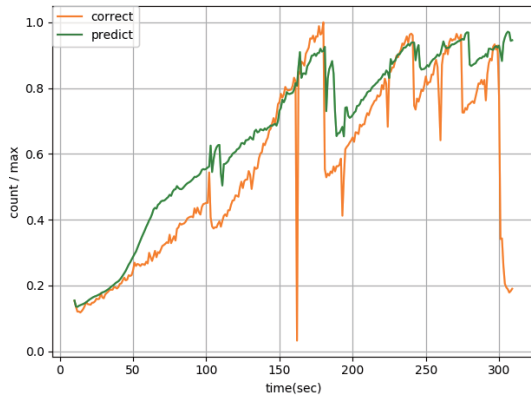


図 10 評価 2 の予測結果 (cwnd あり)

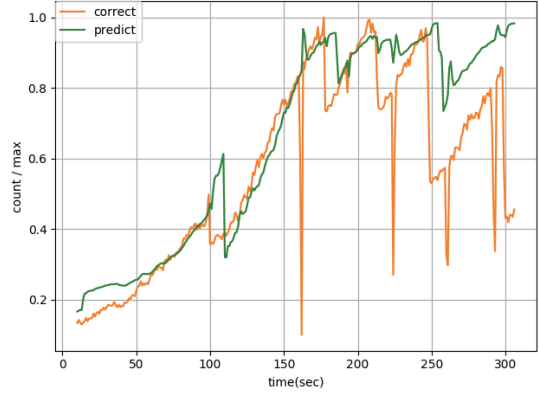


図 13 評価 3 の予測結果 (cwnd あり)

0.5485 となった。

### 5.3.3 評価 3

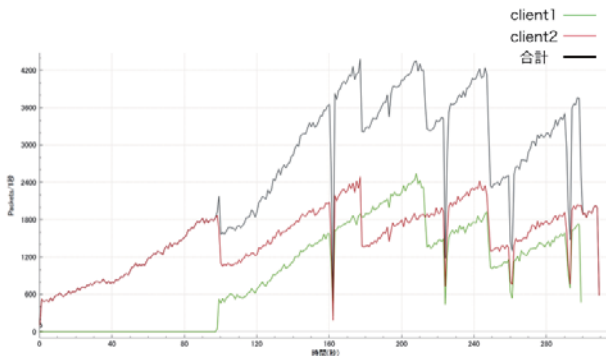


図 11 評価 3 のバリデーションデータ

図 11 は評価 3 で使用するバリデーションデータの packets 数の変動を示すグラフである。図 12 は cwnd なしの場合の学習モデル、図 13 は cwnd ありの場合の学習モデルにこのデータを入力した予測結果である。RMSE は cwnd なしの場合が 0.1224, cwnd ありの場合が 0.1653 となり、一致率は cwnd なしの場合が 0.5518, cwnd ありの場合が 0.6081 となった。

#### 5.3.4 cwnd の有無による誤差と一致率の比較

表 3 は評価 1~3 それぞれの場合の RMSE を、表 4 は評

表 3 平均平方二乗誤差 RMSE

RMSE	cwnd なし	cwnd あり
評価 1	0.1237	0.1624
評価 2	0.07456	0.1680
評価 3	0.1224	0.1653

表 4 一致率 MATCH

一致率	cwnd なし	cwnd あり
評価 1	0.5318	0.5719
評価 2	0.5251	0.5485
評価 3	0.5518	0.6081

価 1~3 それぞれの場合の一致率をまとめたものである。RMSE の観点から見ると、評価 1~3 すべてにおいて cwnd なしの方が誤差が小さくなっていることがわかる。しかし、一致率を見ると RMSE のときとは逆に評価 1~3 すべてで cwnd ありの方が一致率が高くなっている。

表 5 平均平方二乗誤差 RMSE'

RMSE	cwnd なし	cwnd あり
評価 1	0.1002	0.1568
評価 2	0.03659	0.1633
評価 3	0.1047	0.1654

表 5 は、t-9~t までのデータを元に t+1 での値を予測した値と、実際に t でどのような値だったかの誤差であ

表 6 一致率  $MATCH'$

一致率	cwnd なし	cwnd あり
評価 1	0.8054	0.7215
評価 2	0.8054	0.7114
評価 3	0.7483	0.6475

る  $RMSE'$  を評価 1~3 それぞれで算出しまとめたものである。

$$RMSE' = \sqrt{\frac{1}{N} \sum (\hat{y}_{t+1} - y_t)^2}$$

また、表 6 は一致率  $MATCH'$  を算出しまとめたものである。 $MATCH'$  は、 $\hat{y}_{t+1} - \hat{y}_t$  と  $y_t - y_{t-1}$  の正負が一致する割合である。

表 3 と表 5 を比較すると、cwnd なしの場合は評価 1~3 すべての場合で  $RMSE'$  の方が小さくなっていることがわかる。一方、cwnd ありの場合は  $RMSE$  と  $RMSE'$  では大きな差がない。

表 4 と表 6 を比較すると、cwnd の有無に関わらず  $MATCH'$  が  $MATCH$  を大きく上回っており、予測値が正解値から 1 秒遅れていることがわかる。しかし、cwnd なしの  $MATCH'$  は、評価 1~3 すべてにおいて cwnd ありの場合よりも高く、また、表 4 では評価 1~3 いずれにおいても cwnd ありの場合の  $MATCH$  が cwnd なしの場合よりも高い。これより、cwnd を学習の特徴量に含めることは、ネットワークトラフィックの予測モデルの作成により影響を与えることがわかる。

## 6. まとめと今後の課題

深層学習のモデルで、RNN の一種である LSTM をネットワークトラフィックの予測に応用し、学習と予測性能の評価を行なった。

学習の特徴量に cwnd を含めた場合と含めない場合の 2 種類の予測モデルを作成し、それぞれについて予測値と正解値の平均平方二乗誤差  $RMSE$ 、 $RMSE'$  と、増加・減少の一致率  $MATCH$ 、 $MATCH'$  を求めた。その結果、どちらのモデルも  $MATCH$  より  $MATCH'$  の方が高く、予測精度に改善の余地はあるが、 $MATCH$  の値は行った 3 種類の評価すべてで cwnd ありのモデルの方が高いことがわかった。また、cwnd なしのモデルでは  $RMSE'$  の値が  $RMSE$  の値より小さくなるが、cwnd ありのモデルではそのような傾向は見られなかった。このことから、学習の特徴量に cwnd を追加することは学習により影響を与えると考えられる。

今回の実験では、iPerf 通信にサーバ 1 台、クライアント 2 台の計 3 台のマシンを用いており、比較的シンプルなトラフィックでの学習、予測を行なった。今後は学習データの増加、学習に用いる特徴量の増加、マシンの台数を増やしての通信、実際に輻輳が起きているデータを用いるなど、より複雑なトラフィックでの実験を行いたいと考えて

いる。

## 謝辞

本研究は一部、JST CREST JPMJCR1503 および総務省戦略的情報通信開発推進事業 (SCOPE) 先進的通信アプリケーション開発推進型研究開発によるものである。

## 参考文献

- [1] 松井藤五郎, 汐月智也” LSTM を用いた株価変動予測”, 第 31 回人工知能学会全国大会 (JSAI 2017), 2D3-OS-19a-2 (2017).
- [2] 宮崎邦洋, 松尾豊” 深層学習を用いた株価予測の分析”, 第 31 回人工知能学会全国大会 (JSAI 2017), 2D3-OS-19a-3 (2017).
- [3] 林政行, Rafik Hadfi, 伊藤孝行” 深層学習を用いたマルチモーダル学習による降水量予測”, 情報処理学会全国大会講演論文集, 78rd,p.2.369-2.370, (2016).
- [4] 丸千尋, 榎美紀, 中尾彰宏, 山本周, 山口実靖, 小口正人” 大規模災害時における SNS による集合知に基づいたネットワークの QoS 制御”, マルチメディア, 分散, 協調とモバイル (DICOMO2016) シンポジウム, pp.1418-1426, 2016 年 7 月.
- [5] Tokui, S., Oono, K., Hido, S. and Clayton, J.: Chainer: a Next-Generation Open Source Framework for Deep Learning, In Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS) (2015). 6 pages.
- [6] <https://www.wireshark.org>
- [7] <http://dast.nlanr.net/Projects/Iperf/>
- [8] Kaori Miki, Saneyasu Yamaguchi, and Masato Oguchi: “Kernel Monitor of Transport Layer Developed for Android Working on Mobile Phone Terminals”, Proc. ICN2011, pp.297-302, January 2011.
- [9] 南正太郎” LSTM-RNN を用いたイベント考慮後の株価時系列予測”, 第 19 回人工知能学会金融情報学研究会 (SIG-FIN 2017), 2017 年 10 月.