

深層学習を用いたネットワークトラフィック変動の予測と評価

明石季利子[†] 中尾 彰宏^{††} 山本 周^{††} 山口 実靖^{†††} 小口 正人[†]

[†] お茶の水女子大学 〒112-8610 東京都文京区大塚 2-1-1

^{††} 東京大学 〒113-8654 東京都文京区本郷 7-3-1

^{†††} 工学院大学 〒163-8677 東京都新宿区西新宿 1-24-2

E-mail: [†]kiriko@ogl.is.ocha.ac.jp, ^{††}nakao@nakao-lab.org, ^{†††}shu@iii.u-tokyo.ac.jp,

^{††††}sane@cc.kogakuin.ac.jp, ^{†††††}oguchi@is.ocha.ac.jp

あらまし 突然発生する通信障害は、大規模災害時による通信過多や DDos 攻撃、同時に起こる OS アップデートなど、さまざまな原因で引き起こされる。これらの通信障害が起こってからでは、原因究明や復旧作業による通信不能な状態が続くため、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。近年機械学習により、トラフィック集中の早期検知や、輻輳を抑制するための効率化の技術に期待が集まっている。本研究では、深層学習のモデルの一種である LSTM を用いて時系列データとなるネットワークのパラメータを使用したトラフィックの輻輳の予測を行う。有線通信時のトラフィック異常の情報を抽出し、トラフィック変動の兆候をつかむための予測手法の提案と評価を行う。

キーワード 深層学習, LSTM, ネットワークトラフィック, 変動予測

Prediction and evaluation of network traffic fluctuations using deep learning

Kiriko AKASHI[†], Akihiro NAKAO^{††}, Shu YAMAMOTO^{††}, Saneyasu YAMAGUCHI^{†††}, and
Masato OGUCHI[†]

[†] Ochanomizu University 2-1-1 Otsuka, Bunkyo-ku, Tokyo 112-8610 Japan

^{††} University of Tokyo 7-3-1-1 Hongo, Bunkyo-ku, Tokyo 113-8654, Japan

^{†††} Kogakuin University 1-24-2 Nishi-shinjuku, Shinjuku-ku, Tokyo 163-8677, Japan

E-mail: [†]kiriko@ogl.is.ocha.ac.jp, ^{††}nakao@nakao-lab.org, ^{†††}shu@iii.u-tokyo.ac.jp,

^{††††}sane@cc.kogakuin.ac.jp, ^{†††††}oguchi@is.ocha.ac.jp

1. はじめに

突然発生する通信障害は、大規模災害時による通信過多や DDos 攻撃、同時に起こる OS アップデートなど、さまざまな原因で引き起こされる。これらの通信障害が起こってからでは対応が手遅れである場合が多い。特に災害時には、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。また、近年深層学習による時系列データの解析技術が急速に発展しており、株価予測や音声認識といった自然言語処理に多く活用されている。そこで本研究では、深層学習モデルの長短期記憶 (LSTM) ネットワークを用いてネットワークパラメータを特徴量とした時系列データの予測を行った。有線通信時のトラフィック異常の情報から、トラフィック変動の兆候を掴むための手法を提案する。

2. 深層学習

2.1 LSTM (Long short-term memory)

RNN は、自己回帰型の構造を持つネットワークであり、以前に計算された情報を記憶することができる。シーケンスデータや時間データの長距離依存性を捉えるため、時系列予測によく用いられている。LSTM ネットワークは RNN の中で代表的なモデルであり、音声認識、自然言語処理などの様々なシーケンス解析・予測アプリケーションで大きな成功を収めている。LSTM の特徴は、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。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

ネットワークの輻輳が発生すると、TCP/IP は TCP 輻輳制御を行う。TCP 輻輳制御はスロースタートと輻輳回避に基づいたアルゴリズムであり、TCP Tahoe, TCP Reno, TCP NewReno はこのアルゴリズムを用いて輻輳制御を行なっている [1] [2]。近年リリースされた Linux のバージョンでは、BIC TCP [3] や CUBIC TCP [4] のような改良された TCP 輻輳制御アルゴリズムが実装され、Windows では、ロスベースと遅延ベースを組み合わせた輻輳制御方式である Compound TCP [5] が実装されている。さらに、Google 社が 2016 年にリリースした帯域遅延積に基づいた輻輳制御アルゴリズムである TCP BBR が注目されている [6]。[7] では、Rate-Adaptive TCP (RATCP) というアルゴリズムが開発されている。この TCP 輻輳制御システムは、ボトルネックレートフィードバックに応じて輻輳ウィンドウを変化させるシステムである。ここでは、TCP に対してよりよいフィードバックをするために、様々なネットワーク状況下で RATCP と TCP 比較している。このように、TCP 輻輳制御は長年にわたり異なる観点から改良されてきた。しかしながら、これらのアルゴリズムは、ネットワークの輻輳によるパケットロスのようなイベントが発生してから制御するシステムである。それに対し、輻輳の発生を事前に予測することができれば、非常に効率的なトラフィック制御が実現できることは明らかである。そのため、本研究では深層学習を用いてネットワークトラフィックのふるまいを予測することを目的としている。これまで、ネットワークトラフィックの予測はさまざまな観点から研究されてきた。その典型的な例として、Autoregressive Integrated Moving Average (ARIMA) モデルのような統計モデルを用いる手法があり、また、ネットワークトラフィックのふるまいの予測にニューラルネットワークを用いた研究も存在する [8] [9] [10]。これらのネットワークトラフィック分析・予測手法の多くが批評されている [11]。しかし、これらのほとんどがネットワークの基本的なふるまいやトラフィックのシンプルな変動に注目した予測であり、輻輳状態のような大きな変動に注目した予測手法ではない。本研究ではネットワークトラフィック輻輳の予測に焦点を当てている。

4. 関連技術

4.1 Tensorflow

Tensorflow [12] は、2015 年に Google が開発したオープンソースのライブラリであり、機械学習やニューラルネットワークのさまざまなモデルやアルゴリズムを使うことができる。ニューラルネットワークの構築、訓練ができるシステムの要求に応え、処理に対してテンソルを扱っている。手書きの数字の認識、画像認識、自然言語処理などに利用される。本研究では

Tensorflow と用いて、時系列データ学習モデル LSTM の構築・実験を行う。

4.2 iPerf

iPerf [13] は、NLNR/DAST によって開発された、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。サーバモードとクライアントモードの 2 種類の機能があり、サーバ、クライアント間でテストデータを流し指定した通信を発生させ、片方向または両方向の両端間のスループットの測定を行う。IP アドレス、ポート番号、プロトコル、送信するデータなど様々な条件を設定するためのオプションが用意されており、ネットワーク転送性能を測定することができる。そのため、スループット測定の他に障害発生時の原因調査などにも使用される。また、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。Windows や Linux など各種 OS で使用することができる。

5. 実験

本章では、実験用マシンを用意し、iPerf を用いてトラフィックを発生させデータ取得を行い、このデータを用いた学習とパケット数の変動予測について説明する。

5.1 実験内容

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・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 1)。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した。キャプチャしたデータを用いて学習モデルを作成し、学習データとは異なるバリデーションデータを用いて帯域予測を行なった。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。

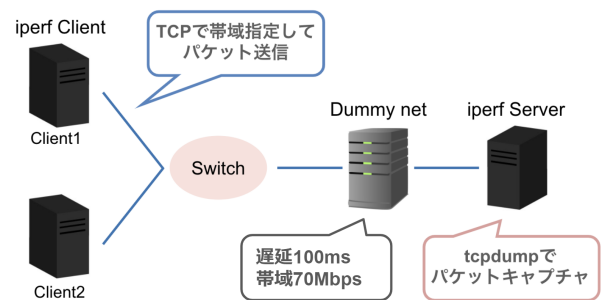


図 1: 実験環境

5.2 提案手法

本実験では、カーネルモニタ [14] を用いて通信時のクライアントそれぞれの輻輳ウィンドウ (cwnd) を取得する。cwnd を学習の特徴量に使用していない場合と使用した場合、増加ラベルを使用した場合の 3 つの場合で学習モデルを作成する。増加ラベルとは、送受信したパケット数が直前の値より増加したかを 0, 1 で表したものである。入力データには $t-9$ 秒 $\sim t$ 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 (移動平均 3,5 秒)
- 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)
- 増加ラベル (増加ラベルありの場合)

正解データは時刻 $t+1$ 秒に届いたパケット数 (移動平均) を 0.1 に正規化した値である。また、学習に使用したパケット数は、3 秒、5 秒での移動平均をとり、それぞれの移動平均において学習モデルを作成した。つまり、cwnd を学習の特徴量に使用していない場合と使用した場合、増加ラベルを使用した場合の 3 つの場合でそれぞれ 2 種類の移動平均を用いて合計 6 種類の学習モデルを作成した。この 6 種類の学習モデルそれぞれの性能測定には、平均平方二乗誤差 (RMSE) と一致率 (MATCH) を用い、どのモデルが予測に最も適しているかの評価を行う。

表 1: 実験で用いた計算機の性能

OS	Ubuntu 14.04.4LTS
CPU	Intel Core i7-6700K CPU @4.00GHz
GPU	GeForce
Memory	32Gbyte

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。

5.3 学 習

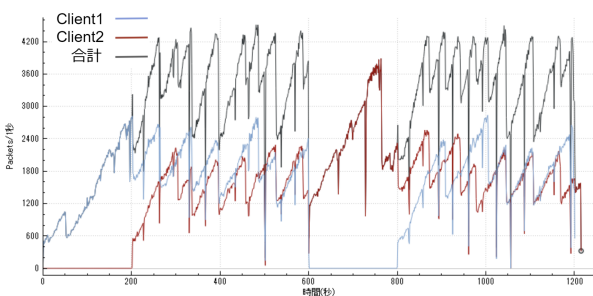


図 2: 学習データ

図 2 は、学習データとして使用するトラフィックデータのパケット数の変動を表している。学習データを取得するにあたり、以下の 3 パターンを含むようなデータを取得した。

- クライアント 2 台が同時にサーバと通信
- クライアント 1 のみがサーバと通信
- クライアント 2 のみがサーバと通信

この学習データを用いて、深層学習によりモデルを作成する。

5.4 学習モデルによる帯域予測

作成した 6 種類の学習モデルを用いて、3 種類のバリデーションデータの帯域予測を行う。予測結果の図に関しては、代表的な予測結果として、3 秒移動平均をとったパケット数を用い cwnd なし、cwnd あり、増加ラベルありの場合の予測結果の図を載せている。

5.4.1 バリデーション 1

図 3 はバリデーション 1 で使用するバリデーションデータのパケット数の変動を表すグラフである。クライアント 2 台が同時にサーバと通信を行うパターンを含むようなデータを用いた。

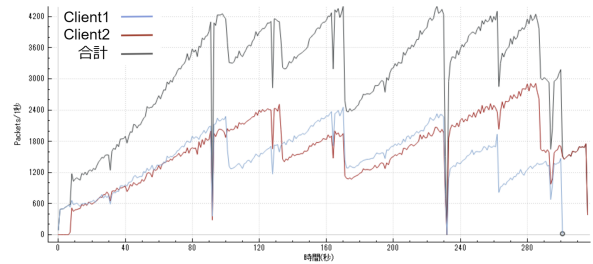


図 3: バリデーションデータ 1

図 4, 5, 6 は 5 秒移動平均をとったパケット数を使用して cwnd を特徴量に含めない場合、cwnd を含む場合、増加ラベルを含む場合に作成した学習モデルにバリデーションデータ 1 を入力した予測結果である。

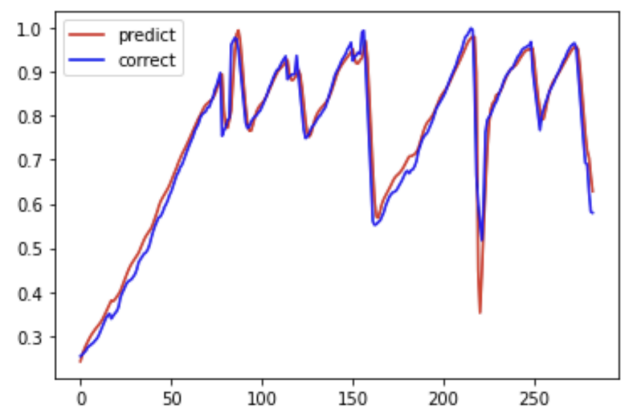


図 4: バリデーションデータ 1 の予測結果 (5 秒平均, cwnd なし)

5.4.2 バリデーション 2

図 7 はバリデーション 2 で使用するバリデーションデータのパケット数の変動を表すグラフである。クライアント 1 のみがサーバと通信するパターンを含むようなデータを用いた。

図 8 は 5 秒移動平均をとったパケット数を使用して cwnd を特徴量に含めない場合、cwnd を含む場合、増加ラベルを含む場合に作成した学習モデルにバリデーションデータ 2 を入力した予測結果である。

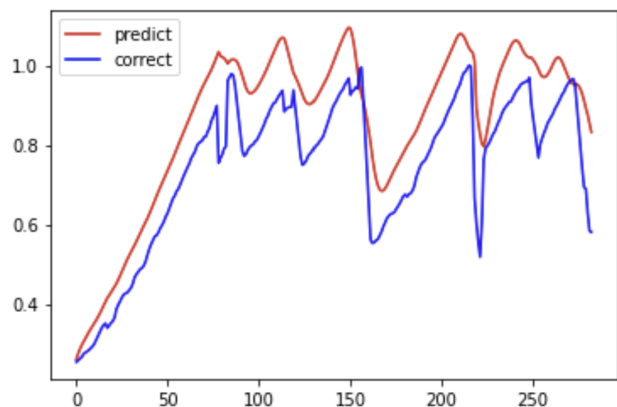


図 5: バリデーションデータ 1 の予測結果 (5 秒平均, cwnd あり)

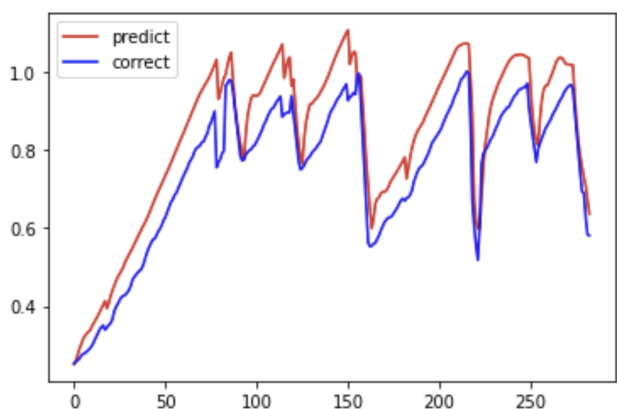


図 6: バリデーションデータ 1 の予測結果 (5 秒平均, 増加ラベルあり)

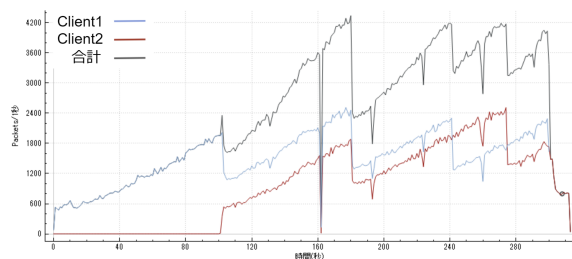


図 7: バリデーションデータ 2

5.4.3 バリデーション 3

図 9 はバリデーション 3 で使用するバリデーションデータの packets 数の変動を表すグラフである。クライアント 2 のみがサーバと通信するパターンを含むようなデータを用いた。

図 10 は 5 秒移動平均をとった packets 数を使用して cwnd を特徴量に含めない場合, cwnd を含む場合, 増加ラベルを含む場合に作成した学習モデルにバリデーションデータ 3 を入力した予測結果である。

5.5 実験結果

本実験では送受信した packets 数の移動平均をとったデータを用いて学習と予測を行なった。作成した学習モデルを用いて 3 種類のバリデーションを行ったところ, すべてのバリデーション

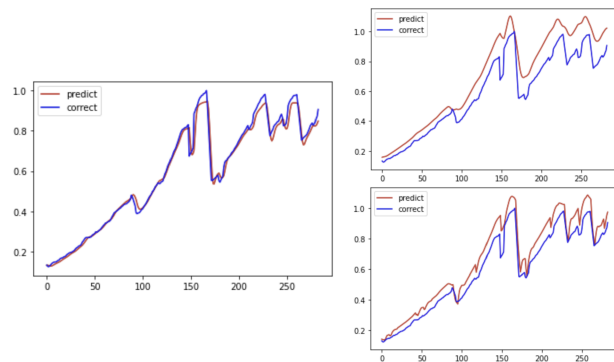


図 8: バリデーションデータ 2 の予測結果 (5 秒平均, 左: cwnd なし 右上: cwnd あり 右下: 増加ラベルあり)

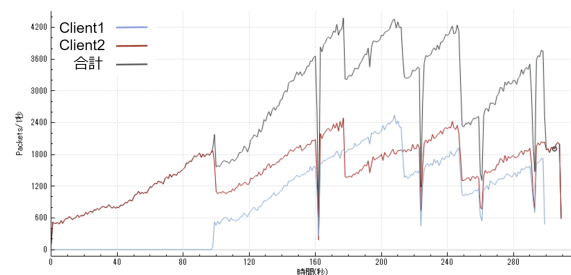


図 9: バリデーションデータ 3

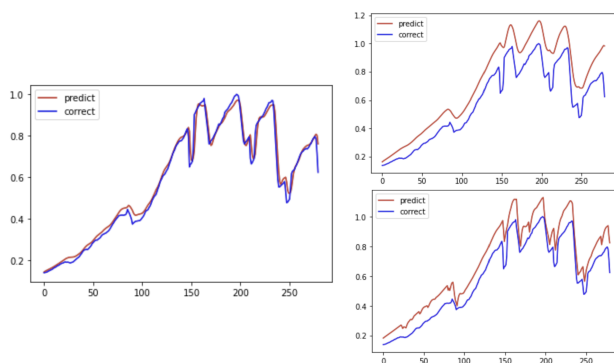


図 10: バリデーションデータ 3 の予測結果 (5 秒平均, 左: cwnd なし 右上: cwnd あり 右下: 増加ラベルあり)

ンデータで目視においてはおおまかな傾向を予測できることがわかる。また, 3 移動平均, 5 移動平均の cwnd なし, cwnd あり, 増加ラベルありのモデルにおいても同様の実験を行い, packets 数変動のおおまかな傾向の学習, 予測ができることを確認した。

5.5.1 提案モデルの検証

図 11 は, 従来の回帰予測に適した評価指標として, 平均平方二乗誤差 (RMSE: Root Mean Square Error) を求め, 正解値と予測値のずれを計算した結果である。

本章で提案した 6 種類の学習モデルを使った予測結果の RMSE の値からは, cwnd を特徴量に含めない学習モデルによる予測精度が高いという結果となった。増加ラベルの有無については, cwnd ありのモデルと比較して, 少し誤差が小さくなっていることが確認できた。packets 数変動のおおまかな傾向の予測ができる一方, 変動の山が訪れるタイミングでは, 変動の

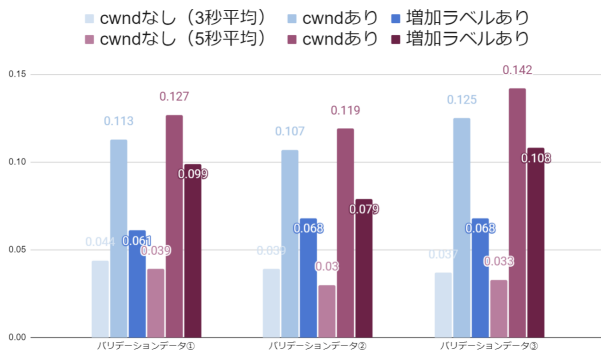


図 11: 提案モデルの RMSE の値

高さに正解値と予測値のずれが生じている。本研究は輻輳の早期検知を目的としており、変動の高さの正確さより、変動が起きたタイミングを正確に捉える事が重要である。しかし RMSE のみによる評価では、変動の高さの誤差が強調されて捉えられてしまうため、次章では変動のタイミングを正確に評価する指標を導入する。

5.6 提案手法の性能評価

本章では、提案手法である前章で作成した学習モデルの予測精度の評価を行う。評価には、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 (MATCH) を指標として使用する [15]。

5.6.1 他手法との比較

送受信パケット数の変動などの時系列データの予測において、増加し続けるなど同じ変動が続く場合、直前の状態を繰り返すことである程度高い精度で予測ができてしまうことが考えられる。本研究では直前の変動を繰り返すモデルを repeat と呼ぶこととする。表 3 は、送受信パケット数の変動で起こる変動の種類を表したものである。repeat は増減が変化する箇所を予測することができないため、変動 2 と変動 3 は予測することができず、変動の方向が変わらない変動 1 と変動 4 のみ予測することができる。repeat と作成した学習モデルを用いて MATCH を算出し、予測できた変動の種類の内訳を用いて、それぞれのモデルの予測精度の比較を行う

表 3: 変動の種類

直前の変動 \ 現在の変動	増加	減少
	変動 1	変動 2
増加	変動 1	変動 2
減少	変動 3	変動 4

図 12,13,14 は、repeat と、3 秒移動平均の cwnd なし、cwnd あり、増加ラベルありのモデルそれぞれの前章のバリデーション 1~3 における MATCH の値と変動の種類の内訳を算出したグラフである。

バリデーション 1~3 全てにおいて、repeat と提案手法である cwnd なし (3 秒移動平均)、cwnd あり (3 秒移動平均) の MATCH の値に大きな差異は見られなかった。また、MATCH の内訳にある変動 1、変動 4 の割合が提案モデルと repeat では

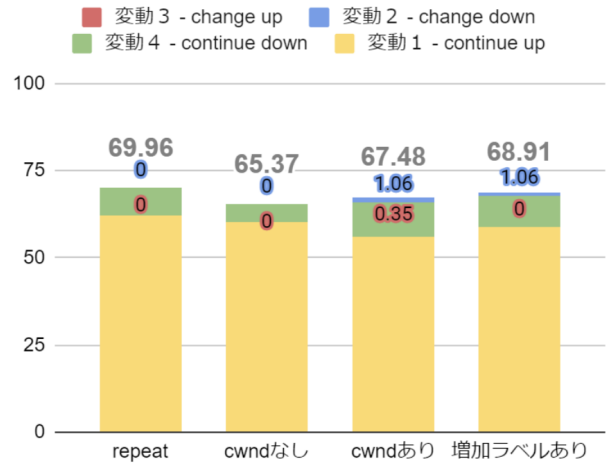


図 12: バリデーションデータ 1 における各モデルの MATCH (3 秒平均)

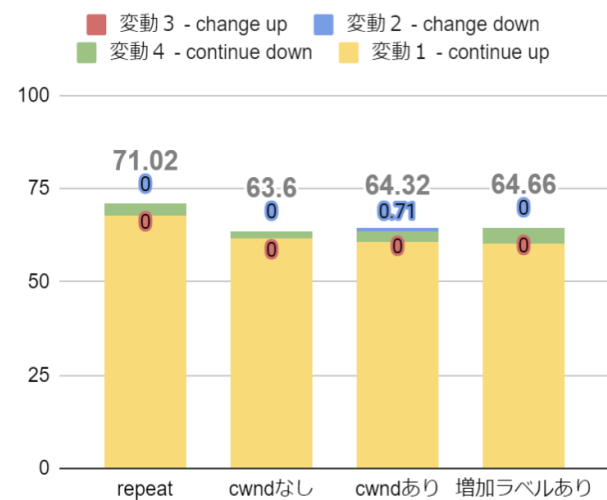


図 13: バリデーションデータ 2 における各モデルの MATCH (3 秒平均)

ば変わらず、提案モデルのうち cwnd あり・増加ラベルありのモデルにおいては、変動 2 と変動 3 の分だけ MATCH の値が 1 割程度増加していることがわかる。

5.7 repeat では予測できない部分に着目した評価

次に、前節の結果を受けて、提案手法が予測できた変動 2、変動 3 の合計が、各バリデーションデータにおける変動 2、変動 3 の合計全体に対してどのくらいの割合かに着目した評価を行う。

図 15,16,17 は、図 12,13,14 それぞれの提案手法 (3 秒移動平均) で予測できた変動 2 と変動 3 が、各バリデーションデータの変動 2、3 の合計の全体に対してどの程度の割合であるかを表すグラフである。repeat は直前の状態を繰り返すモデルであり、パケット数変動の方向が変化する点である変動 2、変動 3 を予測することができないので、全ての場合において 0% となる。つまり、これらのグラフは、repeat で予測できない箇所を提案手法でどの程度予測できるかを表すグラフである。バリデーション 1~3 すべて cwnd ありのモデルに変動 2、変動 3 全

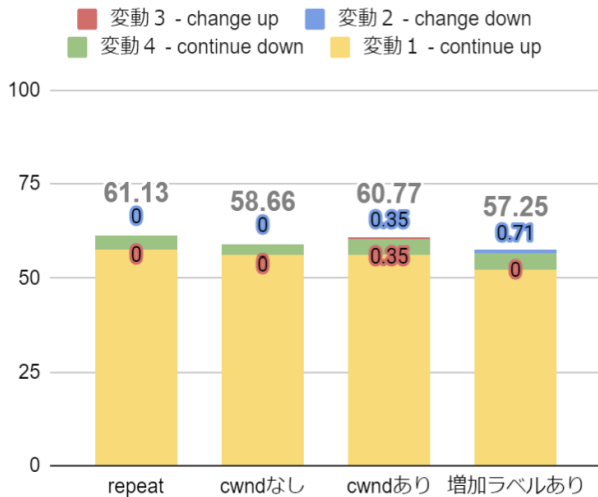


図 14: バリデーションデータ 3 における各モデルの MATCH (3 秒平均)

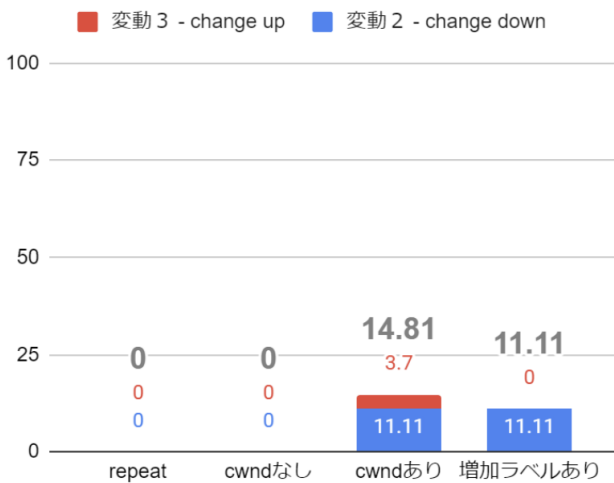


図 15: バリデーションデータ 1 における変動 2, 3 の MATCH (3 秒移動平均)

体の 0~1 割弱の予測が可能であることがわかる。増加ラベルありのモデルも同様の結果となった。一方で cwnd なしのモデルでは、変動 2、変動 3 の予測を当てていないことがわかる。また、3 秒移動平均、5 秒移動平均の cwnd なし、cwnd あり、増加ラベルありのモデルにおいても同様の評価を行なったところ、5 秒移動平均の場合は 3 秒移動平均の場合と同様、提案手法では変動 2、変動 3 全体の 1 割前後の予測が可能であることを確認できた。この結果から、本研究の提案手法は、ただ単に直前の変動を繰り返す repeat の予測できない帯域も予測できることがわかる。また、学習時に直前のパケット数と比べて増加したかどうかを表す増加ラベルを特徴量に含めることによって、変動の方向が変わるタイミングをより捉えることができた。

6. まとめと今後の課題

深層学習モデル LSTM を用いて帯域予測を行う学習モデルの作成と予測性能の評価を行なった。MATCH の観点から他モデルとの性能比較を行うことにより、他モデルでは予測でき

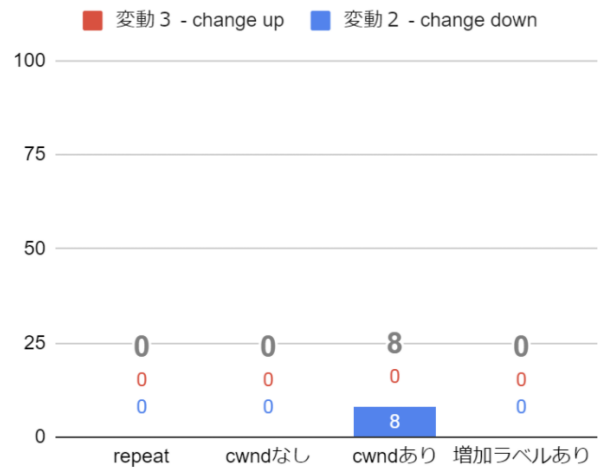


図 16: バリデーションデータ 2 における変動 2, 3 の MATCH (3 秒移動平均)

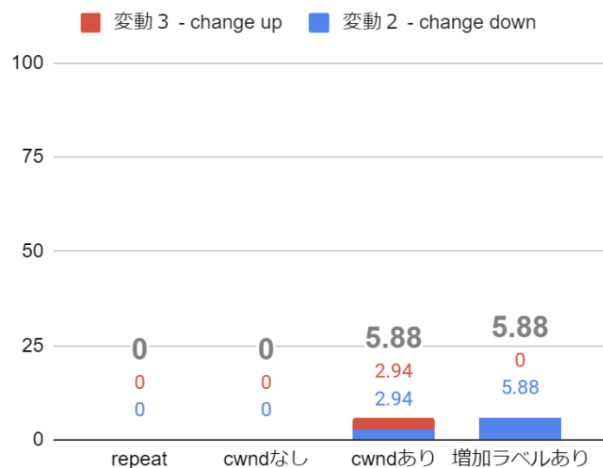


図 17: バリデーションデータ 3 における変動 2, 3 の MATCH (3 秒移動平均)

ないパケット数増減の変動でも、提案手法である深層学習を用いた学習モデルでは予測が可能であることが示された。学習モデル作成時の特徴量の検討について、輻輳ウィンドウの有無は RMSE の値を見ると、cwnd なしのモデルが良い精度が得られた。MATCH の観点では、cwnd ありのモデルが増減タイミングを当てられていることがわかった。増加ラベルの追加については、RMSE の値を見ると追加前のモデルと比べて性能が低下した。MATCH の観点では、他モデルと比べて比較的 repeat モデルに近い値が得られることがわかった。

今後 MATCH の観点から、repeat モデルと比較した学習モデルの精査が必要であることがわかった。学習の入力に用いる特徴量について学習に与える影響の詳しい調査を行い、有用な特徴量についての調査などにも着手し、さらなる予測精度の向上にも取り組みたい。現在は、実験環境でクライアントを 2 台使用したが、実験マシンの台数を増やし、より複雑な通信環境での実験も行いたいと考えている。その上で、災害時や DDoS 攻撃時などに実際に収集されたデータセットを用いた学習モデルを作成することができれば、さらに有効な学習モデルを作成す

ることが可能であると考えられる。

文 献

- [1] M. Allman, V. Paxson, and E. Blanton. Tcp congestion control. In *Internet RFC 5681*, Sept. 2009.
- [2] T. Henderson, S. Floyd, and A. Gurtov. The newreno modification to tcp's fast recovery algorithm. In *Technical Report. IETF*, 2004.
- [3] L. Xu, K. Harfoush, and I Rhee. Binary increase congestion control for fast long-distance networks. In *Proc. INFOCOM*, Mar. 2004.
- [4] S. Ha, I. Rhee, and L. Xu. Cubic: A new tcp-friendly high-speed tcp variant. In http://netsrv.csc.ncsu.edu/export/cubic_a_new_tcp_2008.pdf, year.
- [5] K. Tan, J. Song, Q. Zhang, and M. Sridharan. A compound tcp approach for high-speed and long-distance networks. In *Proc. INFOCOM*, Apr. 2006.
- [6] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. title. In *title*, year.
- [7] Aditya Karnik and Anurag Kumar. Performance of tcp congestion control with explicit rate feedback. In *Proceedings of IEEE/ACM Transactions on Networking (TON) archive Volume 13 Issue 1*, 2005.
- [8] C Park and D-M Woo. Prediction of network traffic by using dynamic bilinear recurrent neural network. In *IEEE*, 2009.
- [9] S. Chabaa, A. Zeroual, and J. Antari. Identification and prediction of internet traffic using artificial neural networks. In *scientific research*, 2010.
- [10] W. Junsong, W. Jiukun, Z. Maohua, and W. Junjie. Prediction of internet traffic based on elman neural network. In *IEEE*, 2009.
- [11] Joshi, Manish, and Theyazn Hassn Hadi. A review of network traffic analysis and prediction techniques. In *title*, 2015.
- [12] Tensorflow. <https://www.tensorflow.org/>. (Accessed on 01/28/2021).
- [13] Nlanr – national laboratory for applied network research. <http://dast.nlanr.net/>. (Accessed on 01/04/2021).
- [14] Kaori Miki, Saneyasu Yamaguchi, and Masato Oguchi. Kernel monitor of transport layer developed for android working on mobile phone terminals. In *Tenth International Conference on Networks (ICN)*, 2011.
- [15] 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.