

深層学習を用いた有線通信における ネットワークトラフィック変動の予測手法

明石 季利子¹ 中尾 彰宏² 山本 周² 山口 実靖³ 小口 正人¹

概要: 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. 従来は通信障害が起こってから対処しており, 対応が手遅れである場合が多い. 近年, 機械学習を用いたトラフィック集中の早期検知や, 輻輳を事前に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.

A Method for Predicting Network Traffic Variability in Wired Communications Using Deep Learning

KIRIKO AKASHI¹ AKIHIRO NAKAO² SHU YAMAMOTO² SANEYASU YAMAGUCHI³
MASATO OGUCHI¹

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, さまざまな原因で引き起こされる. 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, 通信障害が起こってから対処する仕組みがとられていたが, 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ため, 対応が手遅れである場合が多い. 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, 通信の負荷試験として用いられる iPerf とサーバを用意し, TCP 通信時に取得したネットワークパラメータを特徴量とした帯域予測を行う. 深層学習モ

デルの長短期記憶 (LSTM) ネットワークを扱い, 予測誤差の観点からモデルの有効性を評価した. 有線通信時のモニタリングデータから, 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. 関連研究

2.1 TCP 輻輳制御アルゴリズム

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

¹ お茶の水女子大学
Ochanomizu University

² 東京大学
Tokyo University

³ 工学院大学
Kogakuin University

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。

2.2 ネットワークトラフィックの予測

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。

3. 深層学習

3.1 Long Short Term Memory (LSTM)

Recurrent Neural Network (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 では扱えなかった長期依存性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、入力の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。

3.2 ライブラリ

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. 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。

4.1 実験環境

実験に使用したマシンの性能を表 1, 2 に示す。

表 1: 学習に用いた計算機の性能

OS	Ubuntu 16.04.2 LTS
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 16.04.2 LTS
CPU	Intel Xeon CPU E3-1270 V2 @3.50GHz
Memory	16Gbyte

4.2 実験概要

6 台の実験用マシンを使用し、1 台をサーバ、5 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テスト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 TCP Reno を用いた。

4.3 学習データの作成

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。

- 時刻 t に送受信したパケット数

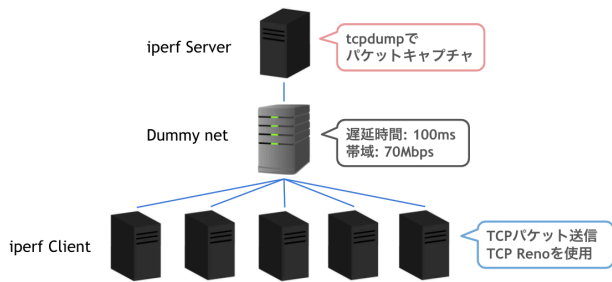


図 1: パケットデータの取得環境

- 各クライアントの送受信したパケット数（クライアント識別あり場合）

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、1 秒、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。サーバと接続している各クライアントのパケット数を特徴量として含める場合と含めない場合のモデルを作成する。学習モデルの性能測定には、平均平方二乗誤差 (RMSE) と一致率 (MATCH) を用い、どの条件で学習したモデルが予測に最も適しているかの評価を行う。

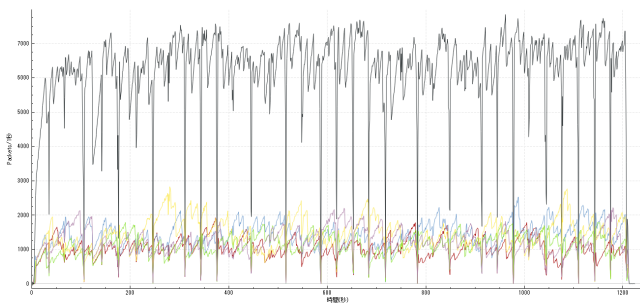


図 2: 学習データ

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。クライアント 5 台それぞれのパケット数を取得し、iperf で TCP による双方向通信を 1200 秒間行い、輻輳を発生させやすくした。

4.4 実験結果

作成した学習モデルを用いて、クライアント台数別によるテスト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の予測結果の図を載せている。テストデータはいずれも 300 秒間 iperf で TCP 双方向通信を行ったデータである。

図 3 は学習データのパケット数の変動を予測したグラフである。図 4 のクライアント 5 台で通信したテストデータを予測した結果が図 5 である。

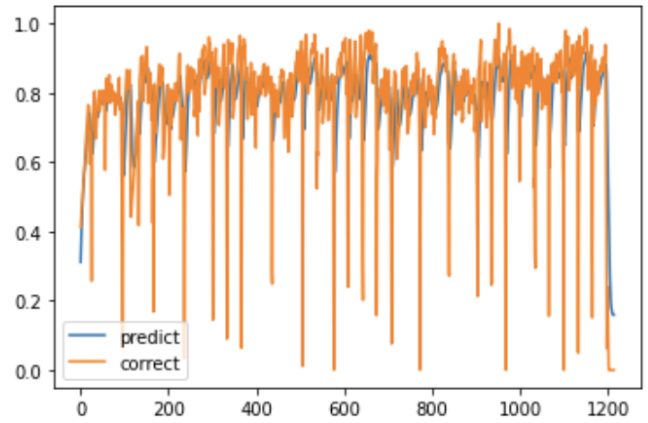


図 3: 学習データによる予測

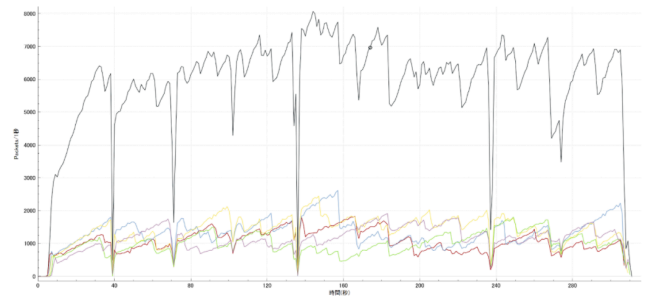


図 4: テストデータ

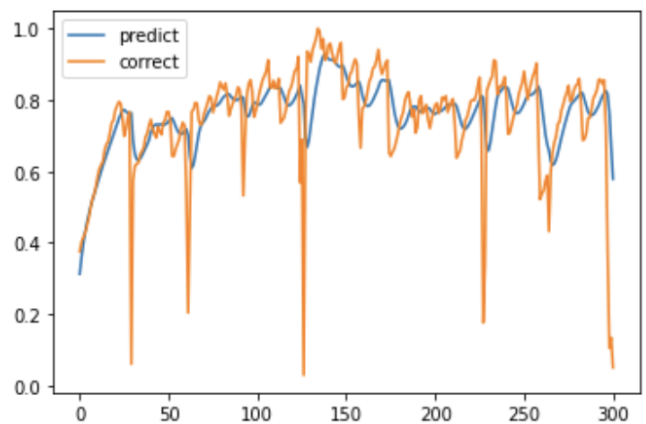


図 5: テストデータによる予測

4.5 実験結果まとめ

図 6 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。クライアント台数の変化による RMSE の値は変わらず、クライアントの識別情報を特徴量として含まないシンプルな学習モデルの予測精度が高いことが示された。

4.6 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

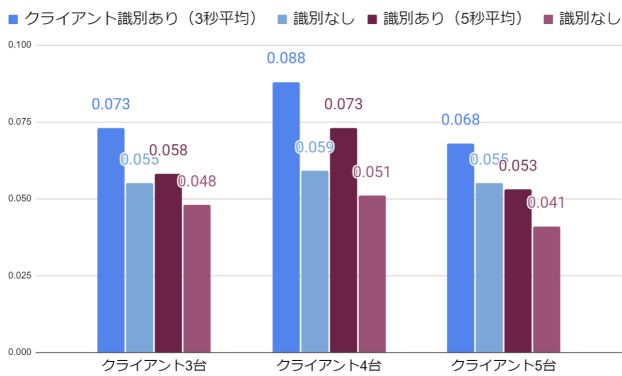


図 6: テストデータの RMSE

して使用する [13].

4.6.1 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
増加	変動 1	変動 2
減少	変動 3	変動 4

図 7 は repeat モデル、提案モデルそれぞれの MATCH の値と変動の種類の内訳を算出したグラフである。

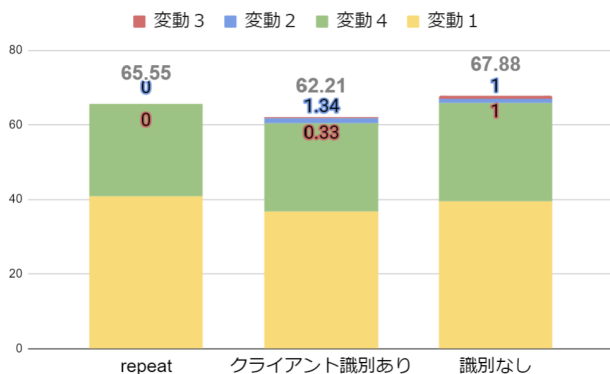


図 7: 予測結果による MATCH

4.7 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、各テスト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。

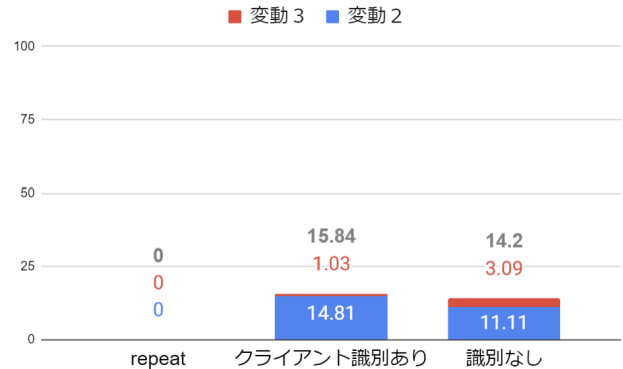


図 8: 変動ラベル 2,3 の MATCH

図 8 は提案手法で予測できた変動 2 と変動 3 が、各テストデータの変動 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。提案モデルでは、変動 2、変動 3 全体の 0~1 割弱の予測が可能であることがわかるこの結果から、本研究の提案手法は、精度にはまだ課題が残るが、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。

5. 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、クライアント別パケット数の有無は RMSE の値を見ると、クライアント識別なしのモデルが良い精度が得られた。MATCH の観点では、特徴量の違いに大きな違いは見られず、repeat では予測できない変動 2,3 において一部増減タイミングを当てられていることがわかった。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

5台使用したが、実験マシンの台数の増減を捉える実験にも取り組みたいと考えている。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。

参考文献

- [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 usingdynamic 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] 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.