

LSTMを用いたネットワークパラメータに基づく トラフィックの変動予測手法の提案と評価

小山内遥香[†] 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: [†]{g1420514,oguchi}@is.ocha.ac.jp, ^{††}nakao@nakao-lab.org, shu@iii.u-tokyo.ac.jp,
^{†††}sane@cc.kogakuin.ac.jp

あらまし 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 Long short-term memory(LSTM) を用いて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掴むための手法を提案し、その評価を行う。

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

Proposal and Evaluation of Variation Prediction Method Based on Network Parameters in Network Traffic by LSTM

Haruka OSANAI[†], 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: [†]{g1420514,oguchi}@is.ocha.ac.jp, ^{††}nakao@nakao-lab.org, shu@iii.u-tokyo.ac.jp,
^{†††}sane@cc.kogakuin.ac.jp

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。

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場合が多い。そのため、トラフィック変動を予測し、通信障害が発生する前に対処することができれば最も効率が良いと考えられる。

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。

トラフィック変動を予測する技術として、実世界に設置されたセンサから得た情報を用いて予測や制御をする技術の検討はすでに行われてきた。センサ情報を用いた技術では、大きなイベントが開催されるとネットワークの混雑が起こるといった予測はできるが、DDoS 攻撃などセンサからは検知が難しい通信障害の場合には、他の対処をとる必要がある。

これらの背景に対し、本研究では、ネットワークの輻輳を検知するための前段階の取り組みとして、深層学習のモデルの一種である LSTM を用いて、トラフィックのモニタリングデータのみからトラフィック異状の情報を抽出し、トラフィック変動を予測することを目的とする。次に、時系列データの増減予測に関する評価指標を用いて作成した学習モデルの性能評価を行い、本研究の提案手法がパケット数変動の予測に関して有用であることを示す。

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 Recurrent Neural Network(RNN)

RNN は、自己回帰型の構造を持つネットワークで、以前に計算された情報を記憶することができる(図 1)。そのため、時系列データの学習に用いられる。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という特徴がある。

2.2 LSTM

LSTM は、RNN が持つ上記の問題を解決するために RNN を改良したネットワークである。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

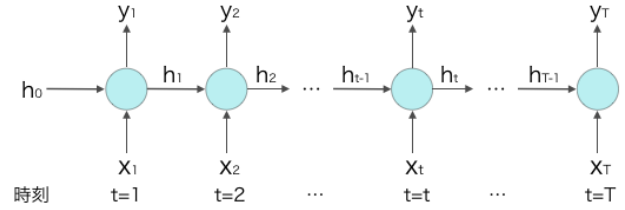


図 1 RNN のネットワーク図

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。

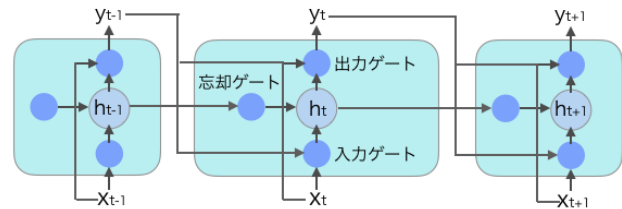


図 2 LSTM のネットワーク図

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 Chainer

Chainer [12] とは、Preferred Networks 社が開発した深層学習を行うためのオープンソースで提供されたライブラリである。Chainer は、Python のライブラリとして提供されており、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。また、GPU 向けの汎用並列コンピューティングプラットフォームである CUDA をサポートしており、GPU を利用して高速な計算を行うことが可能であるため、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。

Chainer では、計算グラフは Python プログラムに基づいて動的に構築され、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。そのため、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、また、デバッグも容易であるという特徴を持っている。

本研究では Chainer を使用して実験を行う。

4.2 Wireshark

Wireshark [13] 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 [14] は、NLANR/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。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。

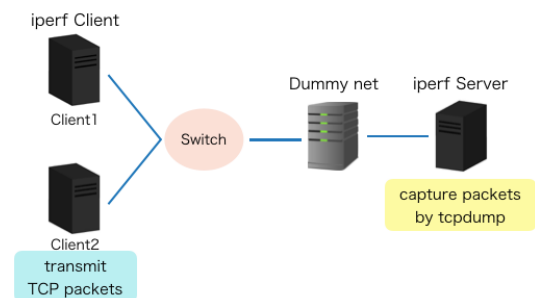


図 3 実験環境

本実験では、輻輳ウィンドウ (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のふたつの場合で学習モデルを作成する。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)
- 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 (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。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のふたつの場合でそれぞれ 3 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において性能を比較し、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 GTX 1080
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。

5.2 学 習

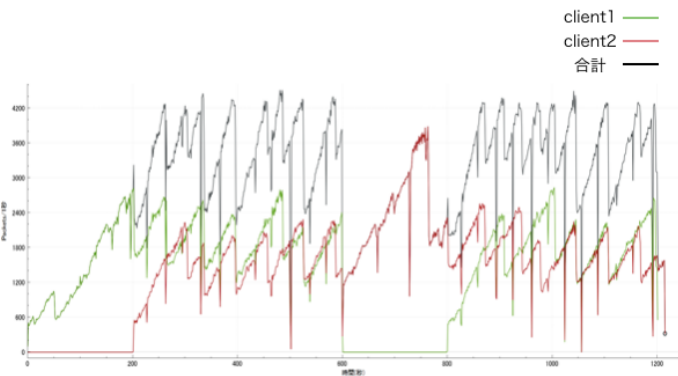


図 4 学習データ

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。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。

5.3 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、5 秒移動平均をとった packets 数を用いた cwnd なし、cwnd ありの場合の予測結果の図を載せている。

5.3.1 バリデーション 1

図 5 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 packets 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。

i). cwnd なし

図 6 は cwnd を特徴量に含まず、5 秒移動平均をとった packets 数を使用して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。

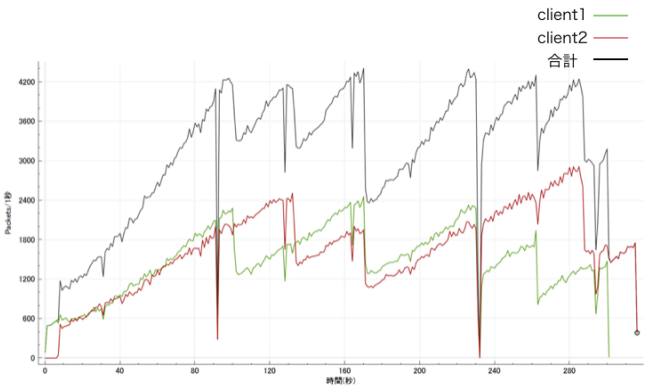


図 5 バリデーションデータ 1

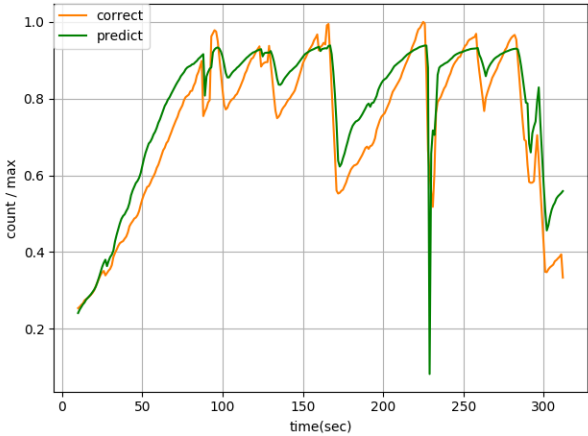


図 6 バリデーション 1 の予測結果 (5 秒移動平均, cwnd なし)

ii). cwnd あり

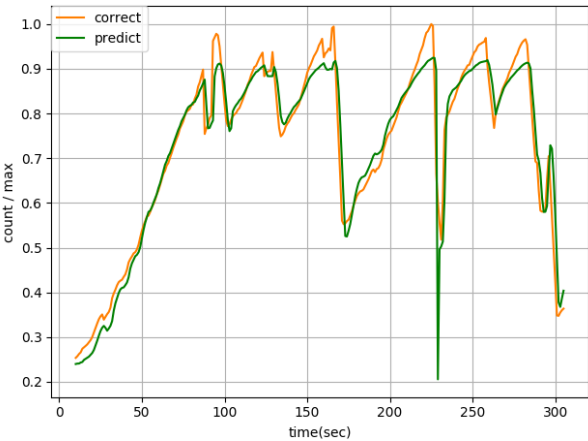


図 7 バリデーション 1 の予測結果 (5 秒移動平均, cwnd あり)

図 7 は cwnd を特徴量に含み、5 秒移動平均をとった packets 数を使用して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。

5.3.2 バリデーション 2

図 8 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 packets 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。

i). cwnd なし

図 9 は cwnd を特徴量に含まず、5 秒移動平均をとった packets

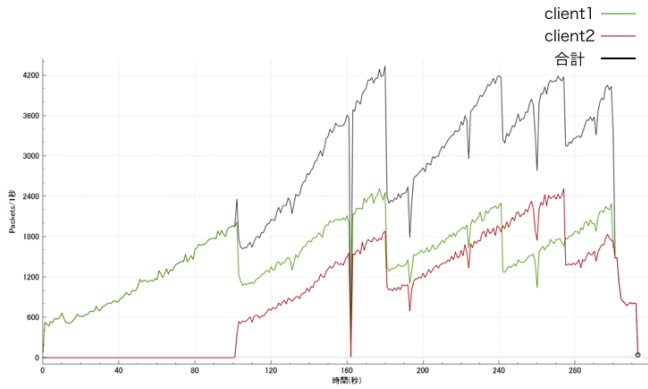


図 8 バリデーションデータ 2

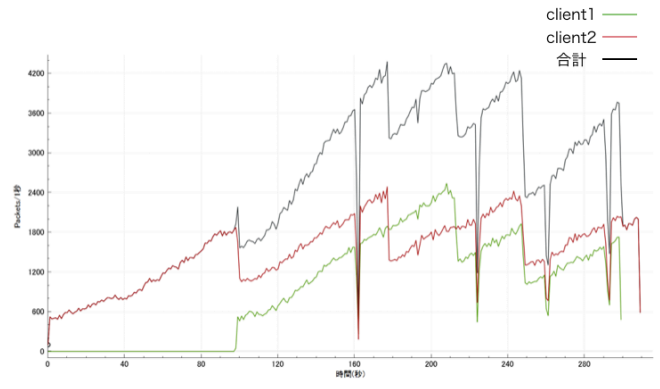


図 11 バリデーションデータ 3

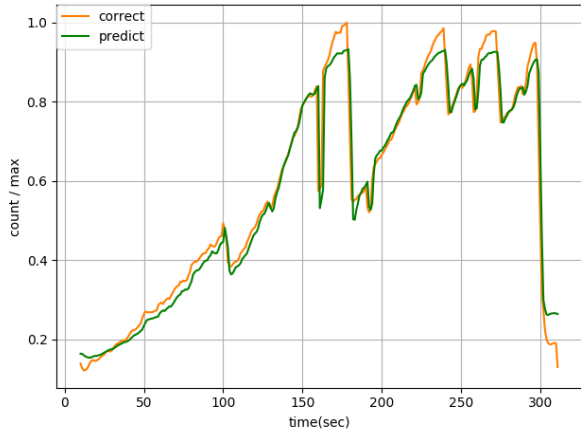


図 9 バリデーション 2 の予測結果 (5 秒移動平均, cwnd なし)

i). cwnd なし

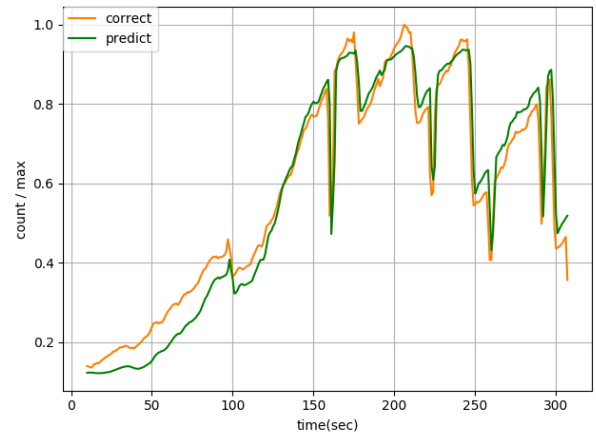


図 12 バリデーション 3 の予測結果 (5 秒移動平均, cwnd なし)

ト数を使用して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。

ii). cwnd あり

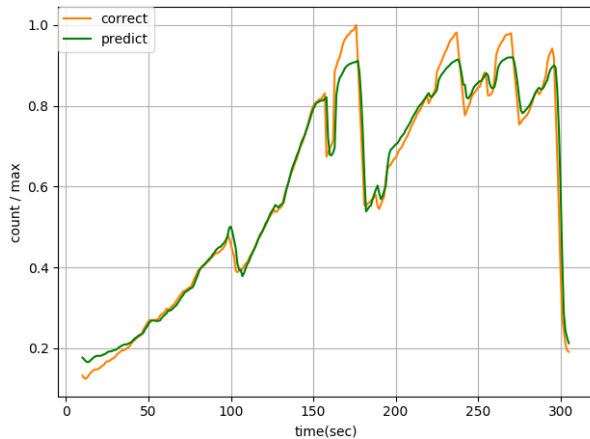


図 10 バリデーション 2 の予測結果 (5 秒移動平均, cwnd あり)

図 10 は cwnd を特徴量に含み, 5 秒移動平均をとったパケット数を使用して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。

5.3.3 バリデーション 3

図 11 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パケット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 12 は cwnd を特徴量に含まず, 5 秒移動平均をとったパケット数を使用して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。

ii). cwnd あり

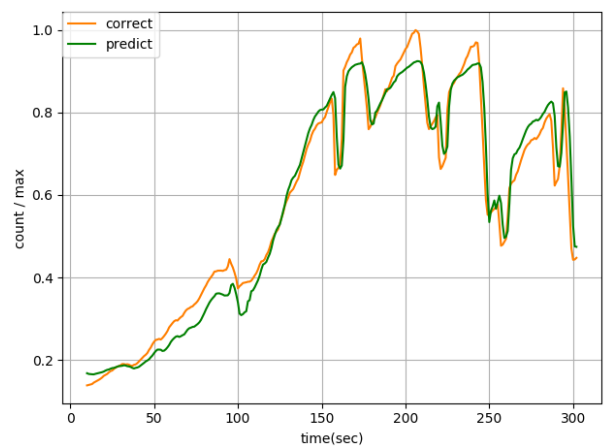


図 13 バリデーション 3 の予測結果 (5 秒移動平均, cwnd あり)

図 13 は cwnd を特徴量に含み, 5 秒移動平均をとったパケット数を使用して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。

5.4 実験結果

本実験では送受信したパケット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なったところ、すべてのバリデーショ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、1秒移動平均、3秒移動平均の cwnd なし、cwnd ありのモデルにおいても同様の実験を行い、パケット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。評価指標を用いた定量的な予測性能の評価については次章で詳しく説明する。

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を指標として使用する [15]。

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一致率 (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

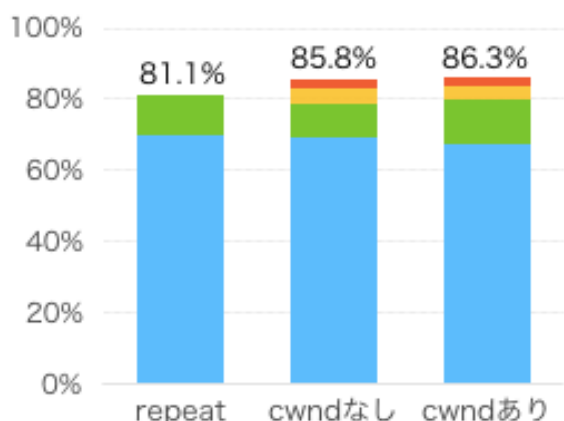


図14 バリデーションデータ1における各モデルの MATCH (5秒移動平均)

図14, 15, 16は、repeat, cwndなし (5秒移動平均), cwnd

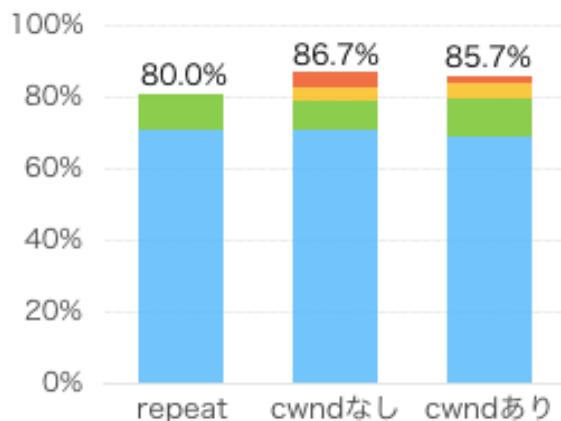


図15 バリデーションデータ2における各モデルの MATCH (5秒移動平均)

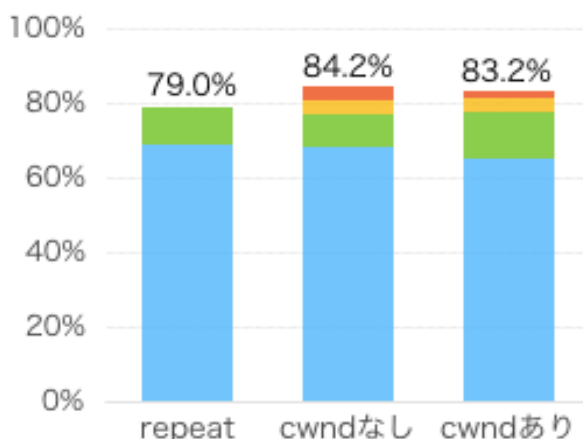


図16 バリデーションデータ3における各モデルの MATCH (5秒移動平均)

あり (5秒移動平均) 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 よりも、提案手法である cwnd なし (5秒移動平均), cwnd あり (5秒移動平均) の MATCH の値が上回っている。また、提案手法の MATCH の値において変動1、変動4の割合が repeat とほぼ変わらず、変動2と変動3の分だけ MATCH の値が増加している。つまり、提案手法では repeat よりも予測精度が向上していることがわかる。

6.2 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17, 18, 19は、図14, 15, 16それぞれの提案手法 (5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

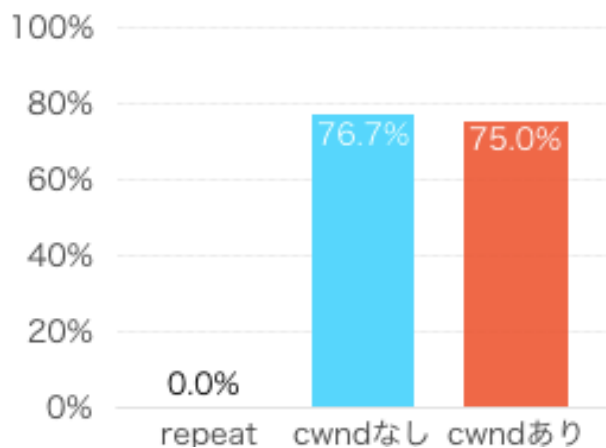


図 17 バリデーション 1 における変動 2 と変動 3 の MATCH (5 秒移動平均)

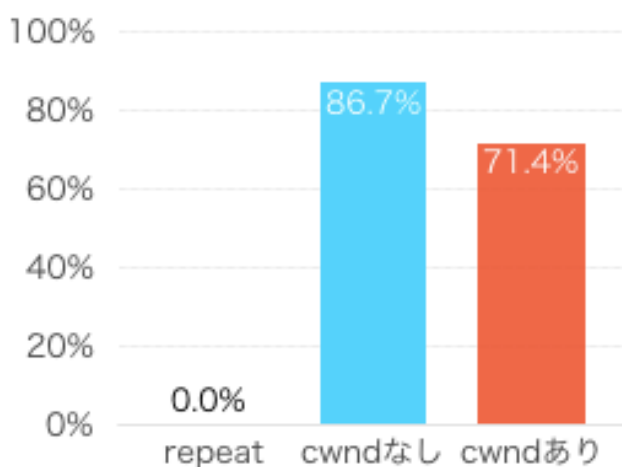


図 18 バリデーション 2 における変動 2 と変動 3 の MATCH (5 秒移動平均)

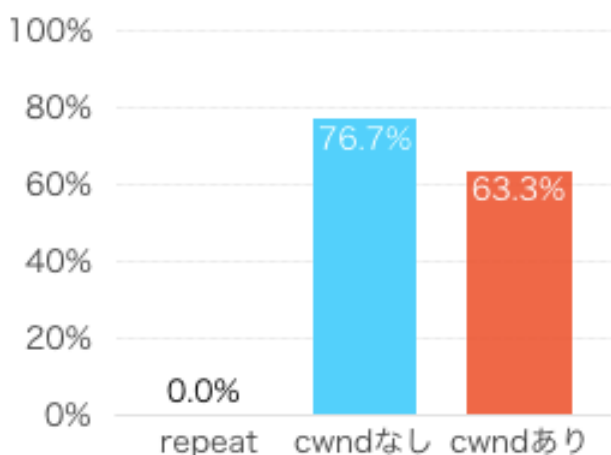


図 19 バリデーション 3 における変動 2 と変動 3 の MATCH (5 秒移動平均)

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 なし、cwnd ありのモデル共に変動 2、変動 3 全体の 7 割前後の予測が可能であることがわかる。

また、1 秒移動平均、3 秒移動平均の cwnd なし、cwnd あり

のモデルにおいても同様の評価を行なったところ、3 秒移動平均の場合は 5 秒移動平均の場合と同様に、提案手法では変動 2、変動 3 全体の 7 割前後の予測が可能であることを確認できた。一方、1 秒移動平均の場合は、提案手法においても変動 2、変動 3 全体の 1 割程度しか予測ができず、予測性能が repeat とほぼ変わらないことがわかった。これは、パケット数の増減の細かい変動がノイズとなっていたためであり、3 秒や 5 秒の移動平均をとることによりノイズが抑えられ、変動の方向が変化する点の予測が可能になったのではないかと考えられる。

この結果から、学習、予測を行う際に 3 秒や 5 秒などパケット数の移動平均をとることによって本研究の提案手法は、ただ単に直前の変動を繰り返す repeat よりも予測できる箇所が多く、精度よくパケット数変動の予測をすることができることがわかる。

7. まとめと今後の課題

本稿では、災害時や他者からの攻撃によりネットワークの輻輳が突然発生する際にも、事前に輻輳発生の兆候を発見し対処したいという背景に対して、ネットワークの輻輳を検知するための前段階の取り組みとして、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り予測性能が向上することが示された。しかし、今回の結果では、特徴量における cwnd の有無ですべてのケースにおいてどちらが良いという結果にはなっておらず、今後の精査が必要であることがわかった。

今後は、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。また現在は、過去 10 秒間のデータから 1 秒後に送受信するパケット数を予測しているが、今後はより長期間の入力データから 5 秒や 10 秒先などに送受信するパケット数の予測などに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。今後はこのような学習モデルを作成し、ルーティングのシステムに組み込むことにも取り組んでいきたい。

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。

文 献

- [1] M. Allman, V. Paxson, and E. Blanton, TCP Congestion Control, Internet RFC 5681, Sept. 2009.
- [2] T. Henderson S. Floyd and A. Gurtov. 2004. The NewReno Modification to TCP's Fast Recovery Algorithm. Technical Report. IETF.

- [3] L. Xu, K. Harfoush, and I. Rhee, Binary Increase Congestion Control for Fast Long-Distance Networks, Proc. INFOCOM, Mar. 2004.
- [4] S. Ha, I. Rhee, and L. Xu, CUBIC: A New TCP-Friendly High-Speed TCP Variant,
<http://netsrv.csc.ncsu.edu/export/cubic.a.new.tcp.2008.pdf>
- [5] K. Tan, J. Song, Q. Zhang, and M. Sridharan, A Compound TCP Approach for High-Speed and Long-Distance Networks, Proc. INFOCOM, Apr. 2006.
- [6] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson, BBR: Congestion-Based Congestion Control, Queue 14, 5, pages 50 (October 2016), 34 pages, 2016.
DOI:<https://doi.org/10.1145/3012426.3022184>
- [7] Aditya Karnik and Anurag Kumar. 2005. Performance of TCP congestion control with explicit rate feedback. In Proceedings of IEEE/ACM Transactions on Networking (TON) archive Volume 13 Issue 1, February 2005, Pages 108-120.
<https://ieeexplore.ieee.org/document/1402475/>
- [8] C Park, D-M Woo, Prediction of Network Traffic by Using Dynamic Bilinear Recurrent Neural Network, (IEEE, 2009) 978-0-7695-3736-8.
- [9] S. Chabaa, A. Zeroual, J. Antari, Identification and Prediction of Internet Traffic Using Artificial Neural Networks, (scientific research, 2010) 2, 147-155.
- [10] W. Junsong, W. Jiukun, Z. Maohua, W. Junjie, Prediction of Internet Traffic Based on Elman Neural Network (IEEE, 2009) 978-1-4244-2722-2.
- [11] Joshi, Manish, and Theyazn Hassn Hadi. A review of network traffic analysis and prediction techniques. arXiv preprint arXiv:1507.05722 (2015).
- [12] 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.
- [13] <https://www.wireshark.org>
- [14] <http://dast.nlanr.net/Projects/Iperf/>
- [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), 2017 年 10 月.