

RNNによるネットワークトラフィック変動の予測

小山内 遥香¹ 中尾 彰宏² 山本周² 山口 実靖³ 小口 正人¹

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 Variation in Network Traffic by RNN

HARUKA OSANAI¹ AKIHIRO NAKAO² SHU YAMAMOTO² SANEYASU YAMAGUCHI³
MASATO OGUCHI¹

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.

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

¹ お茶の水女子大学

² 東京大学

³ 工学院大学

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では扱っていた長期依存を扱えるようになってきている(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。

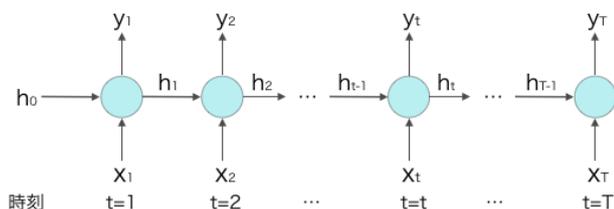


図1 RNNのネットワーク図

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

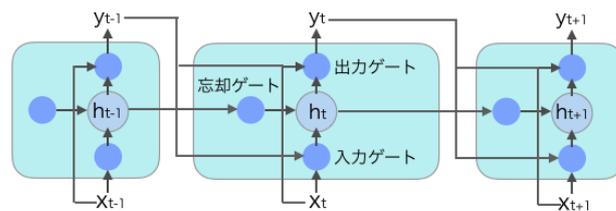


図2 LSTMのネットワーク図

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Convolutional Neural Network(CNN)を用いて日経平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、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。

5. 予備実験

本研究では予備実験として、ブラウザで動画サイトを開きキャプチャしたパケットを用いて学習、予測を行う。

5.1 実験概要

学習を行う際の入力データと正解データの作成方法について説明する。正解データは時刻 $t+1$ 秒の前 4 秒と後 5 秒の間に届いたパケット数の平均を正規化したものである。つまり、時刻 $t-3$ 秒～ $t+6$ 秒の 10 秒間に届いたパケット数の平均を $0\sim 1$ に正規化した値である。10 秒間のパケット数の平均とした理由としては、パケット数の詳細な予測を行うことは困難であるが、ネットワークトラフィックのおおまかな変動が予測できれば十分役に立つためである。また、その値を正規化したものを正解データとしたのは、学習効率を上げるためである。

入力データにはパケット情報から導出した 18 種の特徴量を用いており、詳細は以下である。

- パケット送受信時刻
- プロトコル名 (TCP, UDP, ARP など)
- パケットサイズ
- 送信元 IP アドレス
- 宛先 IP アドレス

- 送信元ポート番号
 - 宛先ポート番号
 - TCP ヘッダのフラグ 10 個
- res, ns, cwr, ecn, urg, ack, push, reset, syn, fin
- ウィンドウサイズ
- $t-9\sim t$ 秒の 10 秒間に届いたパケットそれぞれの上記 18 種の特徴量と正解データを組にして学習を行う。

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

学習に用いた計算機の性能を表 1 に示す。

5.2 学習

Wireshark を用い、研究室 LAN 内においてブラウザで動画サイトをひとつだけ開き、異なる 30 秒間の動画 4 つを自動で連続再生したときのパケットをキャプチャした。図 3 はそのパケット数を 1 秒ごとにプロットしたグラフであり、青いグラフが元データ、オレンジのグラフが移動平均をとった正解データである。今回は予備実験として、このデータを用いて学習を行い予測モデルを作成した。

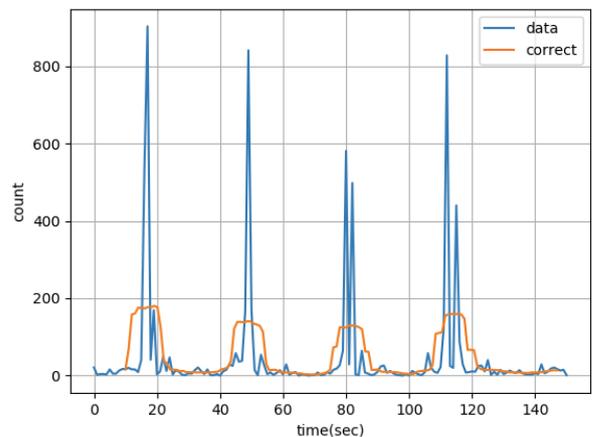


図 3 学習データ

図 3 のデータを用いて学習を行う。epoch 数は 700、隠れ層の数は 49 としている。

予測モデルに学習データを入力し予測した結果、正解データと予測結果の平均二乗誤差が 0.03169 となり、正解データと近い値が予測されており、学習データの特徴をうまく学習できていることがわかる (図 4)。

5.3 性能評価

性能評価では、2 つのテストを行なった。

- テスト 1: 学習データを取得したときと同じ環境で、

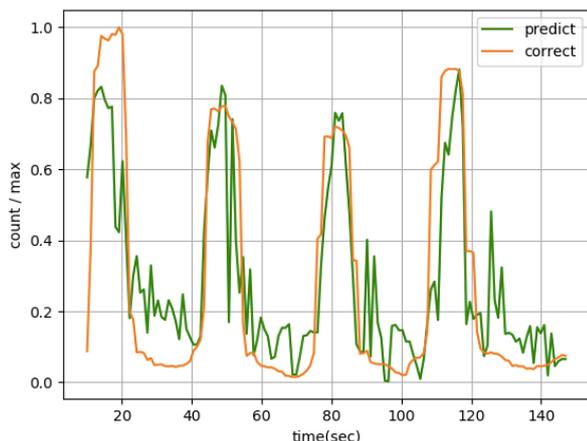


図 4 学習データを用いた予測

学習データとは異なる 4 つの 30 秒間の動画を自動で連続再生しパケットを取得 (図 5)

- テスト 2: 動画サイトを変更し, 30 秒間の動画 4 つを自動で連続再生しパケットを取得 (図 6)

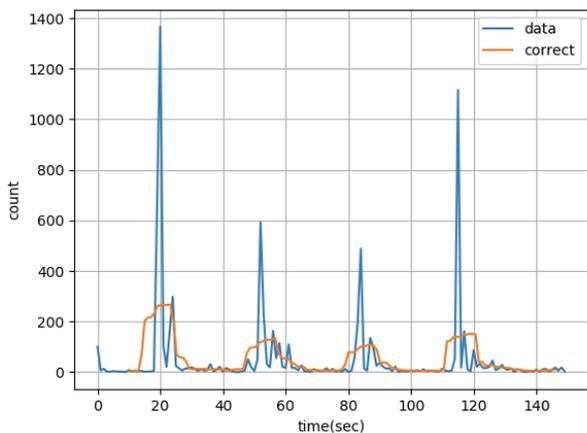


図 5 テストデータ 1

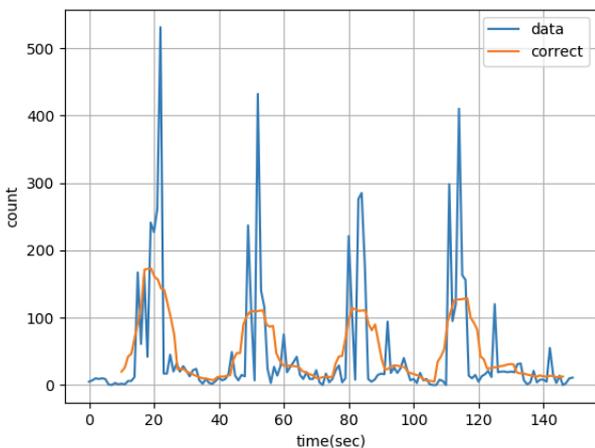


図 6 テストデータ 2

図 7 はテスト 1 の予測結果である. 正解データと予測結果の平均二乗誤差は 0.10636 となり, 学習データと比較す

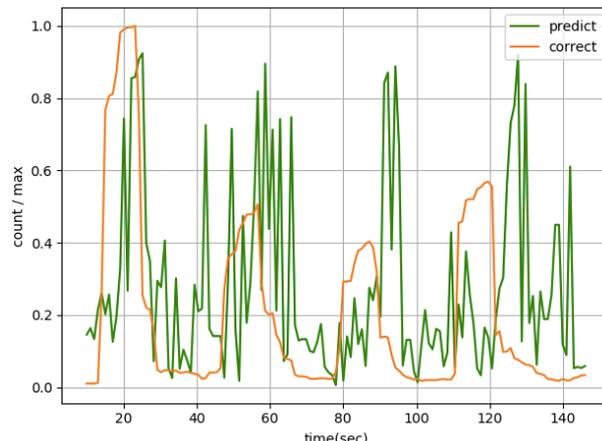


図 7 テスト 1 の予測結果

ると精度が下がるが, 正解データのバケット数が増加する約 10 秒, 45 秒, 80 秒, 110 秒のとき, 予測結果のバケット数も増加しており, バケット数が大幅に変化する箇所は概ね予測できていることがわかる.

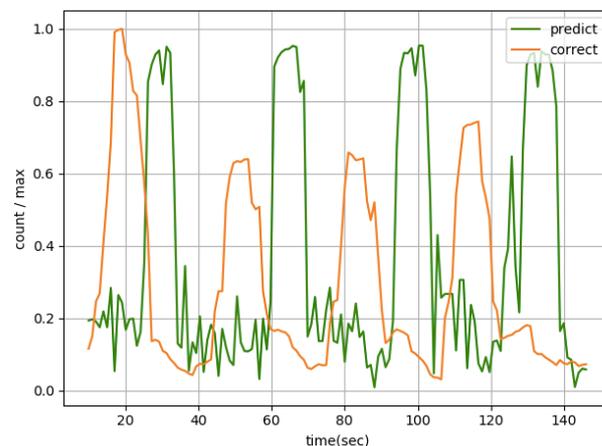


図 8 テスト 2 の予測結果

図 8 はテスト 2 の予測結果である. パケット数の増加, 減少のタイミングがずれてしまい, 平均二乗誤差が 0.24920 と誤差が大きくなっている. しかし, 一度パケット数が増加すると, 大幅に減少するのは約 15~20 秒後で正解データの増減の間隔とほぼ一致しており, 増加や減少の差がテスト 1 の結果よりも明確に予測できている.

また, テスト 1 とテスト 2 の結果より, 確立しているコネクションがひとつの場合は使用するアプリケーションを変更しても予測が可能であることがわかる.

それぞれの予測結果の平均二乗誤差を表 2 にまとめる. テスト 2 の誤差が, 学習データを入力として用いたときの約 7.8 倍, テスト 1 の約 2.3 倍と非常に大きくなっているのは正解データと予測にずれがあるためであり, 正解データと予測のずれが解消できれば誤差が減少すると考えられる.

表 2 平均二乗誤差

学習データ	テスト 1	テスト 2
0.03169	0.10636	0.24920

6. 実験

予備実験でのデータ取得方法では、予測モデルのベンチマークテストを行いつらいため、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う。本章では、このデータを用いた学習、性能評価実験について説明する。

6.1 実験概要

本実験では、3 台のマシンを使用し、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。クライアントから UDP で帯域を指定しパケットを送信し、そのパケットをサーバ側でキャプチャする (図 9)。

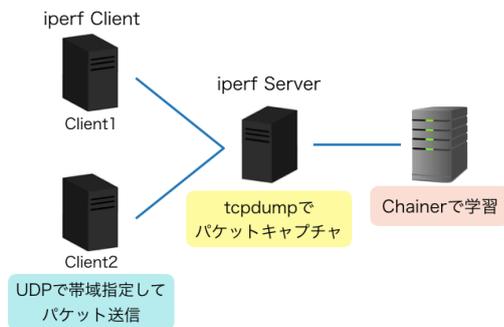


図 9 実験環境

学習を行う際の入力データと正解データの作成方法について説明する。入力データには $t-9$ 秒 $\sim t$ 秒の 10 秒間に届いた全てのパケットそれぞれから 8 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
- プロトコル名 (TCP, UDP, ARP など)
- パケットサイズ
- 送信元 IP アドレス
- 宛先 IP アドレス
- 送信元ポート番号
- 宛先ポート番号
- ウィンドウサイズ

正解データは時刻 $t+1$ 秒に届いたパケット数を 0 \sim 1 に正規化した値である。また、学習時の epoch 数は 1000、隠れ層の数は 49 としている。

学習データを 3 種類取得し、それぞれのデータを用いて学習と性能評価を行う。

- 学習 1: クライアント 1 は 10MByte の帯域で 150 秒間、クライアント 2 は 100MByte の帯域で 60 秒間パケットを送信 (6.2 章)

- 学習 2: クライアント 1 は 10MByte の帯域で 150 秒間、クライアント 2 は 100MByte の帯域で 20 秒間、50MByte の帯域で 20 秒間、100MByte の帯域で 20 秒間パケットを送信 (6.3 章)
- 学習 3: クライアント 1 は 10MByte の帯域で 150 秒間、クライアント 2 は 10MByte の帯域で 10 秒間送信した後、10 秒間パケットを送信しない時間を設け、その後 100MByte の帯域で 60 秒間パケットを送信 (6.4 章)

表 3 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

トラフィック発生に用いた計算機の性能を表 3 に示す。

6.2 学習 1

クライアント 1 で 10MByte の帯域で 150 秒間、クライアント 2 で 100MByte の帯域で 60 秒間パケットを送信し、キャプチャしたデータを学習データとする。

図 10 は、学習データと、この学習データを用いて作成した予測モデルに再度この学習データを入力した際に得られる予測結果のグラフである。

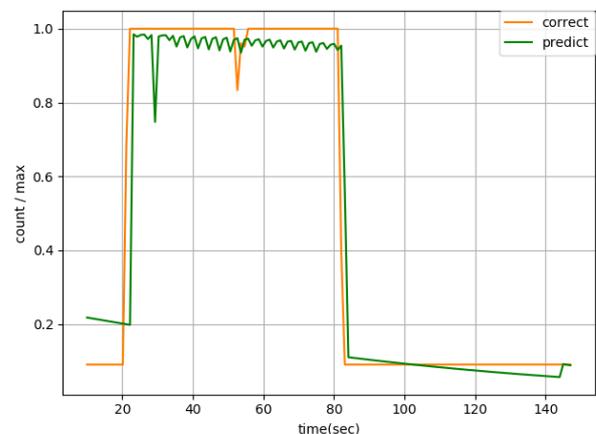


図 10 学習 1 の学習データと予測結果

誤差は 0.01289 となり、学習データの特徴をうまく学習できていることがわかる。

図 11 はクライアント 2 がパケットを送信しはじめるタイミングを変更してキャプチャしたテストデータ 1 と、テストデータ 1 を予測モデルに入力した予測結果のグラフである。誤差は 0.02254 でパケット数の増加、減少のタイミングも予測正確に予測できており、パケット数が増加するタイミングが変わってもうまく予測できることがわかる。

図 12 はクライアント 2 がパケットを送信する時間を 60 秒から 100 秒に変更してキャプチャしたテストデータ 2 と、テストデータ 2 を予測モデルに入力した予測結果のグラフ

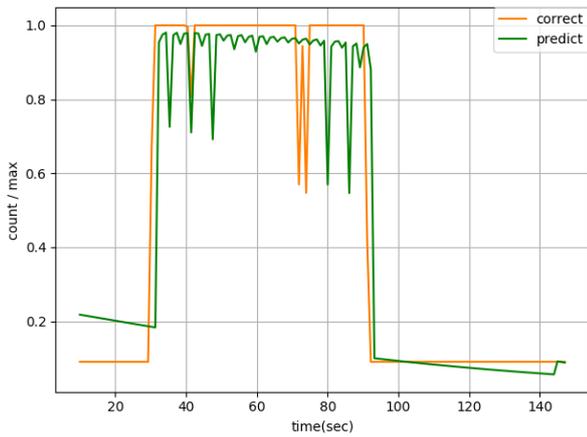


図 11 学習 1 のテストデータ 1 と予測結果

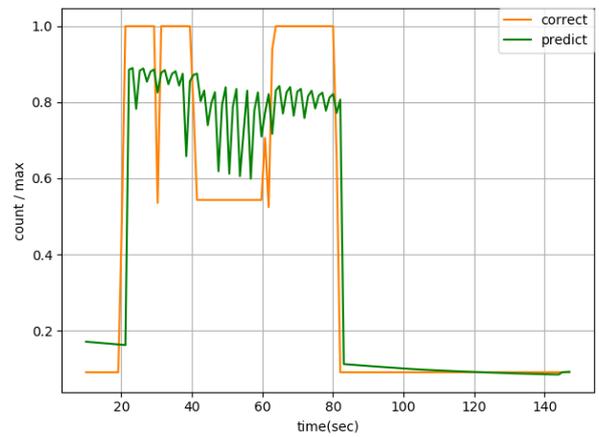


図 13 学習 2 の学習データと予測結果

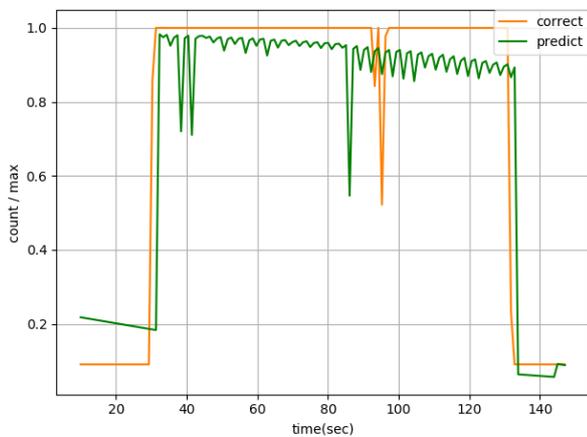


図 12 学習 1 のテストデータ 2 と予測結果

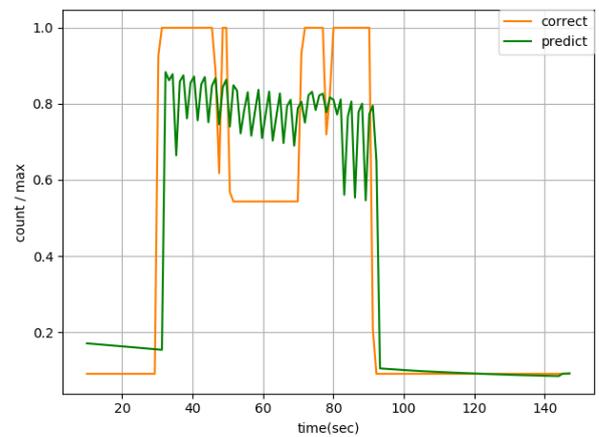


図 14 学習 2 のテストデータと予測結果

である。誤差は 0.02487 となり、クライアント 2 がパケットを送信する時間が長くなってもパケット数の増加、減少のタイミングをうまく予測できることがわかる。

6.3 学習 2

クライアント 1 で 10MByte の帯域指定をして 150 秒間、クライアント 2 で 100MByte の帯域指定をして 20 秒間、50Mbyte で 20 秒間、100MByte で 20 秒間連続でパケットを送信し、キャプチャしたデータを学習データとする。

図 13 は、学習データと、この学習データを用いて作成した予測モデルに再度この学習データを入力した際に得られる予測結果のグラフである。

誤差は 0.02714 となり、40 秒～60 秒でパケット数が減ることを正確に予測することができなかったが、20 秒～80 秒ではパケット数が多いことが予測できているので、トラフィック集中の予測と対応をする場合には十分に役に立つ精度であると考えられる。

図 14 はクライアント 2 がパケットを送信しはじめるタイミングを変更してパケットキャプチャをしたテストデータと、テストデータを予測モデルに入力した予測結果のグラフである。誤差は 0.03621 となり、テストデータを用い

たときもパケット数が減ることを予測することができていないが、パケット数が多いタイミングは予測できており、学習データの場合と同様に輻輳の予測と対応が目的の場合にはこの程度の精度があれば問題ないと考えられる。

6.4 学習 3

クライアント 1 で 10MByte の帯域指定をして 150 秒間パケットを送信、クライアント 2 で 10MByte の帯域指定をして 10 秒間パケットを送信した後、10 秒間パケットを送信しない時間を設け、その後 100MByte の帯域で 60 秒間パケットを送信し、キャプチャしたデータを学習データとする。

図 15 は、学習データとこの学習データを用いて作成した予測モデルに再度この学習データを入力した際に得られる予測結果のグラフである。

誤差は 0.01932 となり、学習データの特徴をうまく学習できていることがわかる。

図 16 はクライアント 2 がパケットを送信しはじめる時間を変更し、クライアント 2 がパケットを送信しない時間を 10 秒から 15 秒に変更しキャプチャしたテストデータと、テストデータを予測モデルに入力した予測結果のグラ

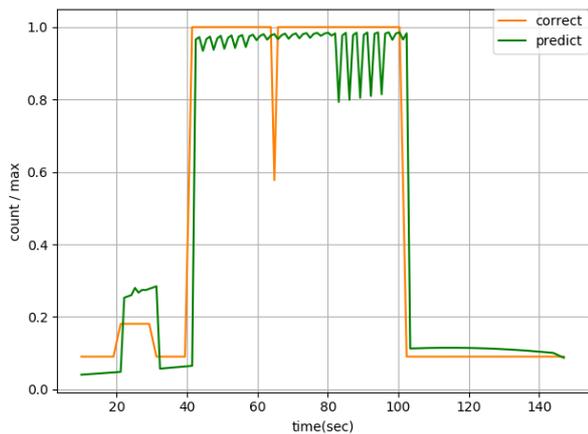


図 15 学習 3 の学習データと予測結果

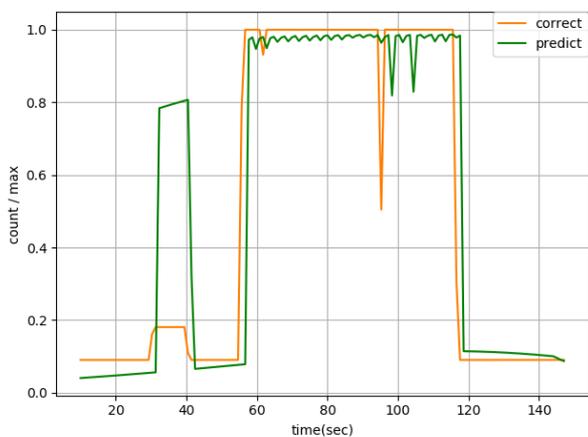


図 16 学習 3 のテストデータと予測結果

フである。誤差は 0.04834 となり、初めにパケット数が増加する 10 秒間の誤差が大きくなってしまいが、パケット増加のタイミングやそれ以降のパケット数の変動はうまく予測できていることがわかる。

7. 考察

実験の結果、深層学習を用いてシンプルなトラフィックのデータを学習させると、トラフィックを流れるパケット数の変動やそのタイミングをほぼ正確に予測でき、トラフィック変動の特徴を学習できることがわかった。この結果から、今後、輻輳発生時のより複雑なトラフィックの学習、予測を行う際にも特徴や兆候の学習が可能であると考えられる。

また、今回は 1 秒先のパケット数の予測を行なったが、今後は予測精度をさらに向上させ、トラフィック集中が発生することをより早期に予測することを目指している。その予測モデルをネットワーク制御装置に組み込み、ネットワーク経路の制御を行うことで、輻輳発生による通信障害を未然に防ぐことができるのではないかと考えている。

8. 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、性能評価を行なった。予備実験では、動画サイトで動画を再生し取得したパケットデータを学習データとして予測を行った結果、トラフィック変動をある程度予測することができ、学習データの特徴をうまく学習できていることがわかった。

iPerf を用いてトラフィックを発生させキャプチャしたデータを用いて学習と予測を行うと、トラフィック変動をほぼ正確に予測することができることが判明した。

今回は輻輳が起きていない場合のデータを用いた学習と予測を行ったため、今後の課題として、iPerf を用いて輻輳を起こしたときのデータを用いた学習と予測実験を行うことを考えている。また、学習に用いる特徴量の再検討と増加による予測精度の向上を検討しており、輻輳ウィンドウ値や iPerf が測定したスループット値などの追加を行いたいと考えている。

輻輳発生時のデータを用いた精度の高い予測が可能になり次第、予測結果を用いた障害検知システムの構築や、ネットワーク制御を行いたい。

謝辞

本研究は一部、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と JST CREST JPMJCR1503 の支援を受けたものである。

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 QoE 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/>