

無線 LAN 通信時における深層学習の LSTM を用いた 無線通信端末のパケット解析

山本 葵[†] 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

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

E-mail: [†]aoi@ogl.is.ocha.ac.jp, oguchi@is.ocha.ac.jp, ^{††}sane@cc.kogakuin.ac.jp

あらまし 近年, 世界中に増え続けているスマートフォン, タブレット端末は機能や性能も強化されている. 気軽にネットワークにアクセスし, 動画やゲームなどのデータ通信を楽しむことが出来るようになり, 大容量かつ高速な通信に対する需要は増大している. しかし有線接続に比べ脆弱な無線接続においては, 膨大なパケットが通信中に無線 LAN アクセスポイントに蓄積され, その結果輻輳が発生してしまうという問題も生じている. 本研究では Android 端末を用いて無線 LAN 通信を行い, アクセスポイント周りのパケットをキャプチャした. そのパケットを深層学習の LSTM モデルを用いて解析し無線 LAN 通信時のトラフィックの予測性能を評価した.

キーワード 深層学習, LSTM, 無線通信

Analysis of Packet of Wireless Communication Using LSTM of Deep Learning in Wireless LAN

Aoi YAMAMOTO[†], Saneyasu YAMAGUCHI^{††}, and Masato OGUCHI[†]

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

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

E-mail: [†]aoi@ogl.is.ocha.ac.jp, oguchi@is.ocha.ac.jp, ^{††}sane@cc.kogakuin.ac.jp

1. はじめに

モバイルネットワークにアクセスするスマートフォンやタブレット端末などのワイヤレスデバイスは世界中で増加し続けている. 毎年多様な形状のワイヤレスデバイスが市場に登場し, これらの機能や性能も強化され進化し続けている. 全世界のワイヤレスデバイス数は, 2020 年までには 116 億にまで増加し, 1 人あたりのデバイス数は 1.5 台になるという予想もなされている [1].

端末自体の高機能化, 高性能化は気軽にホームページの閲覧や音楽, 動画やゲームなどのデータ通信を行うことを容易にしている. このようなデータ通信は大容量になることも多くあり得る. 全世界のモバイルトラフィック量は 2015 年から 2020 年の間に約 8 倍も増加すると予想されている [1]. このことから今後無線 LAN へアクセスする端末数やトラフィック量は増加すると考えられる. それに伴いアクセスポイントの負荷も増大し, トラフィックの輻輳が起こる頻度も多くなると考えられる.

TCP プロトコルは, 通信において標準的に利用されているプ

ロトコルの 1 つであり, ネットワークの帯域の公平かつ効率的な利用には TCP の輻輳制御アルゴリズムが大きな影響を与えている. これまでより高いパフォーマンスを得るために改善がなされてきた. Linux や Android が採用しているロススペースアルゴリズムはパケットのロスを観測し, ロスが増加すると輻輳が発生したとして送信量を抑えるというアルゴリズムである. しかし, より高いスループットを得るためにアグレッシブにパケットを送信するため, 有線接続に比べて脆弱な無線接続環境においてはその手法によって膨大パケットが蓄積されてしまうという問題が生じることがある. この問題の解決法としては, 高速通信の規格化があり, 使用可能な周波数帯の増加や伝送速度の向上があれば輻輳は起きにくくなる. 現に 2018 年には新たな規格がリリースされる予定である. しかし規格が広く普及するには時間がかかり, 実際街中では狭い帯域を取り合っているのが現状である.

本研究では, 無線通信のトラフィックを解析し, 輻輳の極めて早期における検出, 予兆の発見をすることを目的とする. 具体的には, Android 端末を用いてデータ通信を行い, 無線 LAN ア

クセスポイント周りのパケットをキャプチャデバイスを用いて取得し、入力セットとする。また通信中、アクセスポイント宛に ping コマンドを用いて端末-アクセスポイント間の RTT 値を測定、記録する。これらのデータを用いて深層学習を行い、トラフィックの予測が可能であるか検証した。

2. 関連研究

2.1 カーネルモニタ

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、通常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。そこで先行研究 [2] により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。これにより輻輳ウィンドウや各種エラーイベントの発生タイミング、RTT などの TCP パラメータを見ることが可能となる。

2.2 輻輳制御ミドルウェア

先行研究 [3] [4] で開発された輻輳制御ミドルウェアは、カーネルモニタをベースとしたシステムであり、Android 端末間の連携した制御を目的としている。Android 端末間でこれから送信するセグメント数を表す輻輳ウィンドウ値を通知し、周辺端末の通信状況を把握する。さらに周辺端末から受けた情報に基づき、輻輳ウィンドウの上限値を自動で算出し補正することで、端末間で可用帯域を公平に分け合う。これにより無線 LAN アクセスポイントに於ける ACK パケットの蓄積を回避する。

さらに [5] では輻輳制御ミドルウェアの改良を行っている。システムの発動タイミングを調整し多くの端末が同時に通信するときの全体の通信速度と公平性の向上を可能にした。

3. 深層学習

深層学習は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、機械学習を実装するための 1 つの手法である。機械が自分自身で特徴量を抽出できるようになり、また階層を深めることで精度が大幅に向上した。これにより現在は第 3 次 AI ブームとも言われている。代表的な実用例は、郵便局で郵便番号を認識して選別する際に使われる文字認識、Amazon の売り上げを大きくあげたことで有名な商品レコメンドシステム、自動車の運転支援システムなど幅広く使われている。深層学習で用いられるモデルをさらに詳しく説明する。

3.1 ニューラルネットワーク (NN)

人間の脳内にある神経細胞とその繋がり、神経回路網を人工ニューロンという数式的なモデルで表現したもので、一般的なものは図 1 のように多数の層から一方向へ情報が伝搬されるようなモデルがよく使われる。層と層の間にはニューロン同士の繋がり、強さを表す重みがある。NN の発案はコンピュータが普及し始めた時と言われており、当時のコンピュータは非力で膨大な計算を行えるほどの容量や計算能力ではなかったが、近年のコンピュータのスペックの向上により深層学習が容易となった。

3.2 リカレントニューラルネットワーク (RNN)

RNN(図 2) は、入力データは互いに独立であると仮定されていた NN と違い、時系列の流れに意味を持つデータの予測や分類に用いられるモデルである。RNN は以前に計算された情報を

覚えておくための記憶力を持っている。理論的には長いデータを記憶し、利用することが可能だが、実際は 2,3 ステップ前からの記憶しか維持できないという欠点がある。

3.3 ロングショートタームメモリネットワーク (LSTM)

LSTM(図 3) は RNN の拡張として登場した、時系列データに対するモデルである。LSTM は RNN の隠れ層のユニットを LSTM block と呼ばれるメモリと 3 つのゲートをもつブロックに置き換えることで実現された。その最も大きな特徴は RNN ではできなかった長期依存が可能であるということである。

本研究は時系列データであるパケットの解析であるため、この LSTM をモデルとした深層学習を行う。

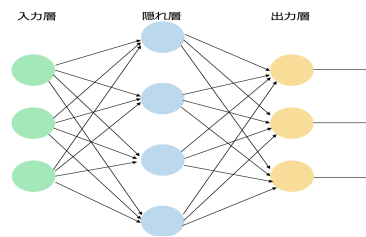


図 1 NN

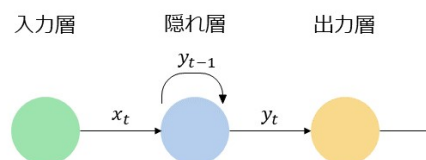


図 2 RNN

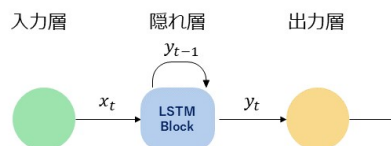


図 3 LSTM

4. 実験

4.1 データセット

深層学習に用いるデータセットについて説明する。

4.1.1 入力データ

入力データとして無線 LAN アクセスポイント周辺のパケットを使用する。用いるキャプチャデバイスは米国、riverbed 社の AirPcap [6] である。AirPcap は wireshark 統合型のワイヤレストラフィックパケットキャプチャデバイスで、制御、管理、データの各フレームを含む、IEEE802.11 a/b/g/n のトラフィックをキャプチャできるデバイスである。

4.1.2 正解データ

正解データとしては、Android 端末でデータ通信中に PC から ping コマンドを用いて測定した pc とアクセスポイント間の RTT 値を用いる。これは RTT 値が増加することはアクセスポイントが混み合い輻輳が発生している可能性があるという考えに基づいたものである。

4.2 深層学習用フレームワーク

今回の実験に用いた深層学習用フレームワークは Chaier である。これは Preferred Networks 社が開発し、2015 年に公開された Python のライブラリである。特徴として”Flexible”, ”Intuitive”, ”Powerful”を掲げている [7].

4.3 實驗環境

4.1 章のデータを取得し実験を行う。実験環境を図 4 に示した 1 つのアクセスポイントをに接続した複数台の Android 端末から iperf [8] によって、データをサーバに送信する。それと同時に PC で上記のデータを取得する。その後、適切なデータ形式に加工し、LSTM モデルを用いた深層学習をおこなう。

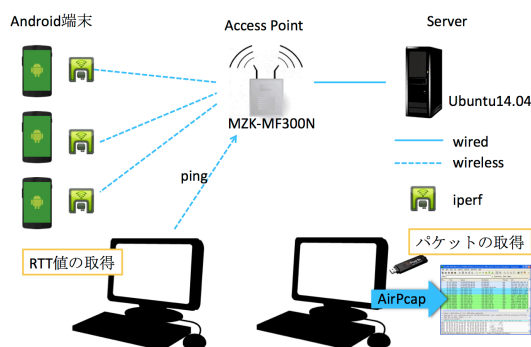


図 4 実験環境

表 1 実験機器の性能

Android	Model number	Nexus S	Nexus 7(2013)
	Firmware version	4.1.1	6.0.0
	CPU	1.0 GHz Cortex-A8	Quad-core 1.5 GHz Krait
	Memory(Internal)	16 GB, 512 MB RAM	16 GB, 2 GB RAM
	WLAN	Wi-Fi 802.11 b/g/n	Wi-Fi 802.11 a/b/g/n
server	OS	Ubuntu 14.04 (64bit) / Linux 3.13.0	
	CPU	Intel(R) Core(TM)2 Quad CPU Q8400	
	Main Memory	8.1GiB	
AP	Model	MZK-MF300N(Planex)	
	Support Format	IEEE 802.11 n/g/b	
	Channel	13	
	Frequency Band	2.4 GHz(2,412-2,472 MHz)	

表 2 解析に用いた PC の性能

PC	OS	14.04.1-Ubuntu
	CPU	Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz
	GPU	GeForce GTX 1080
	Memory	2.4 GHz(32GB)

4.4 実験 1

初めに 2 つの実験を行なった。スマートフォン 2 台、タブレット 2 台の計 4 台の Android 端末を用いて通信を行い、87 秒間のデータを用いてデータセットとした。2 つの実験は入力データに違いがあり、その他の条件は同じである。学習回数を表す epoch 数は 2000 に設定し、どちらの結果のグラフも t から $t+9$ 秒の入力データを用いて $t+10$ 秒の正解データである RTT 値を予測したものである。オレンジの線が正解の RTT 値、青の線が予測したものである。

4.4.1 結果 1-1(一次元入力データ)

1 つ目の実験は第 1 段階として、入力データを表 3 のように 1 秒間の平均データ量のための一次元データを使用して深層学習を行なった。結果は図 5 のように全体的に振れ幅が少なく、平坦なグラフが出力された。70 秒付近の 1 番急激に RTT 値が増加している部分は少し予測もついていっているが、これは十分に予測ができているとはいえず、この程度の予測ではアクセスポイントが混み合っているかの判断材料にすることはできないと考えられる。

表 3 実験 1 データセット

入力データ	正解データ
1 秒間の平均データ量	RTT 値

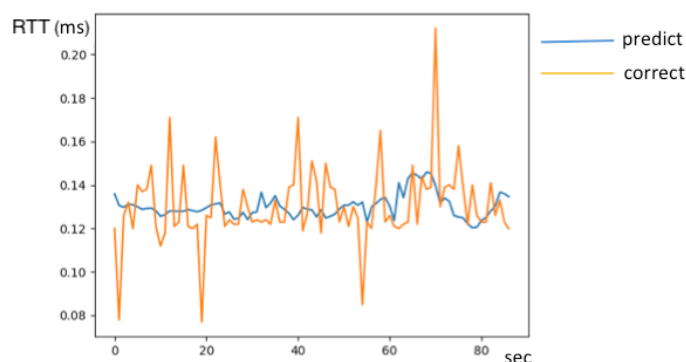


図 5 実験 1 結果 (一次元入力データ)

4.4.2 結果 1-2(五次元入力データ)

実験結果 1-1 から、さらに精度の向上を目指すため、入力データを先ほどの一次元から五次元に増やし再度実験を行なった。データセットの詳細は表 4 である。結果は先ほどよりも良いものとなった。特に 10 秒付近の RTT 値が急増している箇所や、65 秒付近ではうまく予測が行われていることがわかる。さらに実験結果が正解とどれくらい離れているかを表す loss 値をみていく (図 7)。これは 0 に近づくほど正解に近いということであるが、loss の値は順調にさがっており学習がうまくいっていることがわかる。

表 4 実験 2 データセット

入力データ		正解データ
1 秒間	時間 パケット数 平均データ量 送信機器の台数 受信機器の台数	RTT 値

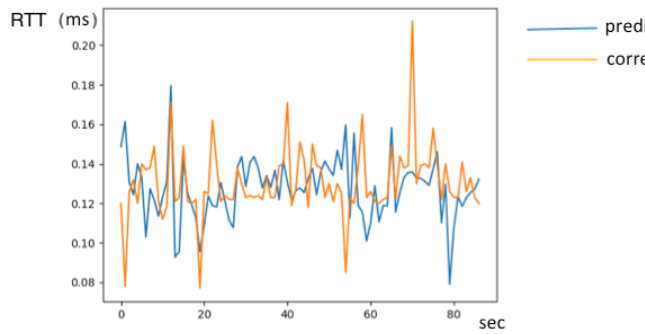


図 6 実験 2 結果 (五次元入力データ)

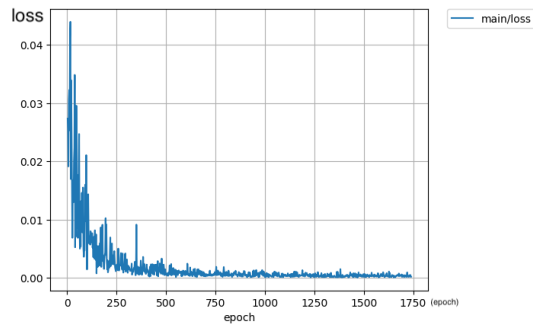


図 7 実験 2 loss

4.5 実験 2

実験結果 1-2 より、五次元入力データによる学習は予測ができる可能性があることがわかった。そこで精度の向上を目指すため、長時間のデータによる学習を行なった。さらにこの学習の汎化能力を検証するためテストデータによる検証も行った。この実験にはスマートフォン 7 台、タブレット端末 2 台の計 9 台の Android 端末を用いた。データセットは全て新規に作成し、学習用データセットとして 609 秒間のパケット情報と RTT 値を、テスト用データセットとして 353 秒のパケット情報と RTT 値を使用した。入力データの詳細は先ほどと同じく表 4 である。全ての結果のグラフも同じく t から $t+9$ 秒の入力データを用いて $t+10$ 秒の正解データである RTT 値を予測したものであり、オレンジの線が正解の RTT 値、青の線が予測したものである。

4.5.1 結果 2-1

結果は、1 枚のグラフで出力するとデータ数が多く見づらいため、学習データによる予測結果は 3 つに、テストデータによる予測結果は 2 つに分割した。学習データによる予測結果は、図 8 の 0-200 秒の範囲では合っている部分もあるが、図 9 の RTT 値が安定して低い 300-375 秒付近は全く予測できていなかった。テストデータによる予測結果は、図 11 の 0-25 秒、図 12 の 175-225 秒付近はうまく予測が行われているが、RTT 値が急増していたり、反対に安定して低い部分は予測できていなかった。

5. まとめと今後の課題

本研究では、アクセスポイントに接続する端末が複数台通信を行い、輻輳がおこるであろう場合において、輻輳の極めて早期の検出、予兆の発見を行うために、深層学習を用いてトラフィッ

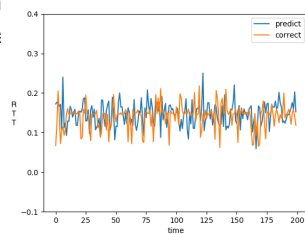


図 8 学習データによる
予測結果 1

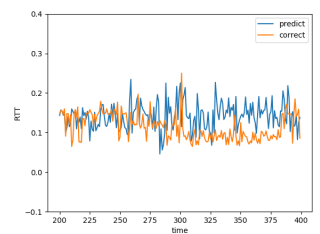


図 9 学習データによる
予測結果 2

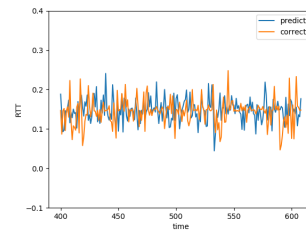


図 10 学習データによる予測結果 3

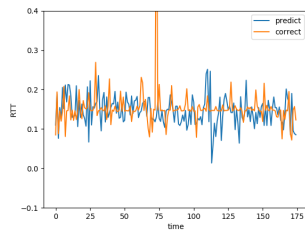


図 11 テストデータによる
予測結果 1

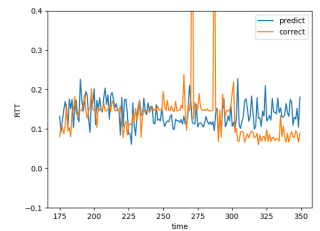


図 12 テストデータによる
予測結果 2

クの予測を行なった。具体的には無線 LAN アクセスポイントに接続した Android 端末を用いてデータ通信を行い、そのパケットをデバイスを用いてキャプチャする。そのパケット情報を入力データとし、またアクセスポイントの混雑を表すであろうアクセスポイント宛の RTT 値を正解データに用いて深層学習を行なった。深層学習はフレームワークに Preferred Networks 社の Chainer を使い、時系列データに適する LSTM モデルを使用してトラフィックの予測が行えるか実験を行なった。その結果から入力データの情報は非常に重要で、多ければより正確な予測を行えることがわかった。

しかし、五次元入力データは一次元入力データとの比較では精度が向上していると言えるが、部分的にであり、全体的にみるとまだまだ十分な精度とは言えない。この学習用データセット、テスト用データセットを作成するにあたって、複数の Android 端末を用いて、全台数が通信をしている状態や反対にほとんど通信していない状態などランダムに状態を変化させ様々な状況でのパケット情報と RTT 値の増減が現れるように作成したつもりだが、まだあらゆる状況下でのデータが十分ではない可能性がある。学習過程において各パラメータのチューニングを行ったとしても 10 分程度のパケット情報では精度には限界があることがわかった。

今後の課題としては精度の向上が1番に挙げられる。具体的には入力データの情報量を増やすことである。今回の実験で入力データを一次元から五次元にただで精度の向上がみられた。よってさらに入力データの情報量を増やせばさらに精度の向上が見込まれる。キャプチャデバイスから得られる情報は非常にたくさんあり、多数のレイヤの情報がとれる。今回はその中から一部の情報を集計して用いたが、得られる情報全て、各パケット1つずつを入力データにしてトラフィックを多面的に捉え学習、予測をしていきたい。また正解データも工夫の余地があるのではないかと考えられる。また、今回は609秒間という時間のデータで実験を行なったが、数十分、数時間、数日とさらに長時間のデータを集めてあらゆる状況においてのデータを用いて十分に学習をさせていきたい。

また今後輻輳制御を行っていくことを見据え、計算時間も考慮に入れて実験を行ってきたい。いくら予測精度が良くても、予測するために計算している間に結果の予測時刻を過ぎてしまつては意味がない。どの程度先の時刻まで精度よく予測できるかという実験も今後行っていく予定である。

文 献

- [1] Cisco Visual Networking Index, https://www.cisco.com/c/ja_jp/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html
- [2] 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.
- [3] Hiromi Hirai, Saneyasu Yamaguchi, and Masato Oguchi: "A Proposal on Cooperative Transmission Control Middleware on a Smartphone in a WLAN Environment," Proc. IEEE WiMob2013, pp.710-717, October 2013.
- [4] Ai Hayakawa, Saneyasu Yamaguchi, Masato Oguchi: "Reducing the TCP ACK Packet Backlog at the WLAN Access Point," Proc. ACM IMCOM2015, 5-4, January 2015.
- [5] Ayumi Shimada and Masato Oguchi: "A Study of Android Tables Performance," Proc.DEIM2017,H2-3,March 2017
- [6] riverbed, <https://www.riverbed.com>
- [7] Chainer,Framework for Neural Networks, <https://chainer.org/>
- [8] Iperf For Android Project in Distributed Systems, <http://www.cs.technion.ac.il/sakogan/DSL/2011/projects/iperf/index.html>