

# LSTMを用いた任意パラメータ指定可能なネットワークトラフィック生成

栗山 海渡<sup>†</sup> 渡部 康平<sup>††</sup>

<sup>††</sup> 長岡技術科学大学 大学院工学研究科 〒940-2188 新潟県長岡市上富岡町 1603-1

E-mail: <sup>†</sup>s193129@stn.nagaokaut.ac.jp, <sup>††</sup>k\_watabe@vos.nagaokaut.ac.jp

**あらまし** 近年、情報化社会に伴って、ユーザの使用する端末の多様化が進んで来ており、多様な端末に応じたネットワークの評価が重要になってきている。ネットワークの評価には、大量のトラフィックデータが必要となる。しかし、既存のトラフィック生成モデルは統計学的手法に基づいたものがほとんどであり、一部のトラフィックの特性しか再現できないといった理由から、その代替手段が求められている。最近の研究では、機械学習を使った生成モデルが注目されてきているが、通信トラフィックを対象とした生成モデルの提案は限定的であり、十分に研究がなされていない。そこで、本稿では、機械学習を用いた任意パラメータ指定可能なトラフィック生成モデルを提案する。このモデルでは、Generative Adversarial Networks (GAN) と Long Short Term Memory (LSTM) を組み合わせることで、トラフィックの特性を多面的に再現し、特徴量を任意に調整可能なトラフィックの生成が可能となる。実際のトラフィックトレースを使用して、提案手法と従来モデルとの比較評価を行った。評価の結果、従来モデルよりもトラフィックの特性の再現性が高いことを示した。さらに、生成するトラフィックの特徴量を任意に調整できることも示した。

**キーワード** 時系列生成, GAN, LSTM, ネットワークトラフィック, 生成モデル

## Machine Learning-Based Network Traffic Prediction with Tunable Parameters

Kaito KURIYAMA<sup>†</sup> and Kohei WATABE<sup>††</sup>

<sup>††</sup> Graduate School of Engineering, Nagaoka University of Technology

Kamitomiokamachi 1603-1, Nagaoka, Niigata 940-2188, Japan

E-mail: <sup>†</sup>s193129@stn.nagaokaut.ac.jp, <sup>††</sup>k\_watabe@vos.nagaokaut.ac.jp

**Abstract** Network evaluation has become increasingly important in recent years. Network evaluation requires large amounts of traffic data. Recent studies have focused on generative models using machine learning. However, few generative models exist for traffic. In this paper, we propose a traffic model using machine learning. Comparative evaluation with a conventional model using actual traffic traces shows that it is more reproducible than the conventional model. Furthermore, we showed that the traffic characteristics can be arbitrarily adjusted.

**Key words** Time series generation, GAN, LSTM, Network traffic, Generative model

### 1. はじめに

近年、ネットワークトラフィックの性質が多様化しており、トラフィックの量も増加傾向にある。これは、情報化社会に伴う通信機器の多様化と、ユーザの増加が原因である。様々なものがインターネットに接続される Internet of Things (IoT) では、膨大な数のデバイスが接続されるため、これらを協調させてネットワークを構築することは難しい課題である。

このことから、ネットワークの評価が重要になってきている。ネットワークの品質を評価することは、ネットワークの信頼性を向上する上で非常に重要である。ネットワークを構築する際、試験用のトラフィックデータを使用して、ネットワーク機器のシミュレーションや負荷テストを事前に行う。

ネットワークの評価には、大量の試験用のトラフィックデータが必要となるが、これを用意するのはコストやプライバシーの観点から難しい。ここでのトラフィックデータとは、ある時間間隔において何バイト通信したかを表すような時系列レベルのトラフィックを想定しており、パケットレベル、フローレベルとは異なる。トラフィックの長期間のトレースは、時間や容量において高いコストがかかる。また、トラフィックには個人情報が多く含まれるため、セキュリティやプライバシーの観点から、測定したトラフィックをそのままシミュレーションやテストに用いるのは、好ましくない。

そこで、一般的にトラフィックジェネレータを用いて試験用のトラフィックデータを生成するが、そのほとんどが統計学的手法に基づいたモデルであり、いくつかの問題がある。例えば、一

部のトラヒックの特性しか再現できないことや、任意の特性を持ったトラヒックが生成できないといった問題が挙げられる。統計学的手法では、いくつかのパラメータを調整することでトラヒックを生成するが、統計学的知識が必要であり、パラメータの設定は非常に難しい。そのため、複雑なネットワークの挙動を多面的に再現するのは難しいとされている。これらの理由から、その代替手段が求められている。

最近では、機械学習に基づいた生成モデルが注目されてきているが、トラヒックを対象とした生成モデルはほとんどない。機械学習は、大規模で複雑な特性を再現したモデルを作成することができる。また、パラメータの設定も学習によって行うため、統計学的知識が必要とならない。このような点で機械学習に基づくモデルは、統計学的手法に基づくモデルと比べて優れている。しかし、時系列レベルのトラヒックにおいては、トラヒックの予測に関して機械学習を用いた研究はあるが、トラヒックの生成タスクに応用した例はない。

本稿では、機械学習を用いたトラヒックの生成モデルを提案する。機械学習を用いることで、複雑なネットワークの挙動を多面的に再現できると考えられる。ここでの多面的とは、平均や標準偏差、トレンド等の複数の特徴量のことを指し、本稿では、それら全てを再現したトラヒックの生成を目的としている。また、トラヒックの特性を多面的に再現するだけでなく、任意の特徴量の調整を可能にする。具体的には、機械学習の技術を活用した Generative Adversarial Networks (GAN) [1] と Long Short Term Memory (LSTM) [2] を組み合わせることで、特徴量の調整を実現する。

本原稿の貢献は、以下の通りである。

- 機械学習における生成モデルをベースに、GAN と LSTM を組み合わせたトラヒックの生成モデルを提案する。
- トラヒックの特性を多面的に再現及び維持したまま、任意の特徴量を調整可能にするトラヒック生成技術を確立する。
- 実際のネットワークで測定されたトラヒックトレースを使用して、提案モデルと従来モデルとの比較評価を行う。

本稿は、以下のように構成されている。まず、2 章で統計学的手法と機械学習に基づくいくつかの生成モデルについて説明する。次に、3 章で特徴量を指定した場合と指定しない場合について、トラヒックの生成問題を定式化する。次に、4 章で本稿における提案手法の構造や学習、生成の手順について説明する。そして、5 章では、その提案モデルと従来モデルの実験および評価を行う。最後に、6 章で本稿の結論を述べる。

## 2. 生成モデル

生成モデルとは、ある測定データを生成する確率分布を想定し、測定データからその確率分布を推定するモデルである。確率分布を推定するモデルは、統計学的手法に基づくモデルと機械学習に基づくモデルの二つに分類される。

### 2.1. 統計学的手法に基づく生成モデル

統計学的手法に基づく生成モデルは、統計学的知識から確率分布を仮定することで、測定データに似たデータを生成するも

のである。このモデルは、測定データに適した複数のパラメータを調整することで確率分布を推定する。しかし、特徴量の数とパラメータの数が比例するため、複雑な特性を持つ測定データの確率分布を推定することは難しい。

#### 2.1.1 CO 線形予測

Covariation Orthogonal (CO) 線形予測モデルは、急激な変化を捉えることができるという特徴がある。まず、時系列  $\{y_t : t = 1, 2, \dots, n\}$  に対して、係数  $\{a_t : t = 1, 2, \dots, m\}$  を次式で計算する。

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} R_{00} & R_{10} & \dots & R_{i0} & \dots & R_{m0} \\ R_{01} & R_{11} & \dots & R_{i1} & \dots & R_{m1} \\ \vdots & \vdots & & \vdots & & \vdots \\ R_{0j} & R_{1j} & \dots & R_{ij} & \dots & R_{mj} \\ \vdots & \vdots & & \vdots & & \vdots \\ R_{0m} & R_{1m} & \dots & R_{im} & \dots & R_{mm} \end{bmatrix}^{-1} \begin{bmatrix} R_{0k} \\ R_{0(k+1)} \\ \vdots \\ R_{0(k+m)} \end{bmatrix} \quad (1)$$

$$R_{ij} = \sum_{t=1}^n y_{t-i} y_{t-j} \quad (2)$$

次に、 $k$  ステップ先の予測値  $y_{n+k}$  を次式で求める。

$$y_{n+k} = \sum_{t=1}^m a_t y_{n+1-t} \quad (3)$$

次に、 $k$  ステップ先の予測値を含めた時系列  $\{y_t : t = 1+k, 2+k, \dots, n+k\}$  を取得し、(1) 式より、再び係数  $a_t$  を計算する。これを繰り返すことで、予測時系列  $\{\hat{y}_t : t = n+k, n+k+1, \dots\}$  を得る。

### 2.2. 機械学習に基づく生成モデル

機械学習に基づく生成モデルは、機械学習により測定データの持つ複雑な特性を多面的に再現したモデルを作成し、確率分布を推定するものである。このモデルは、測定データを学習データとして使用することで、測定データの持つ複雑な特性を学習により再現することができる。

#### 2.2.1 LSTM

LSTM とは、長期的な依存関係を学習できる Recurrent Neural Network (RNN) の一種である。LSTM では、時系列  $\{x_t : t = 1, 2, \dots, n\}$  を入力とし、時系列を  $\{x_t : t = 1, 2, \dots, n\}$  を出力する。LSTM のモデルは、ブロック  $S$  が鎖状に接続されており、過去の情報を元に未来の出力を得ることができる。LSTM は、セル  $h$  とゲートで構成され、セル  $h$  の情報がブロックを通して伝搬される。このセル  $h$  に対し、ゲートを通して情報を削除または追加していく。ゲートは、忘却ゲート、入力ゲート、出力ゲートの三つがある。これらは、シグモイド層  $\sigma$  と呼ばれるニューラルネットワークにより、 $[0, 1]$  を出力する。この数値は、前の情報をどの程度通すべきかを示しており、0 は何も通さず、1 は全て通す。

#### 2.2.2 GAN

GAN とは、画像などのデータを生成する生成器 (Generator) とデータが本物か生成物かを識別する識別器 (Discriminator) の二つのニューラルネットワークからなる深層学習の一種であ

る。Discriminator は、入力されたデータが、本物か生成物かを識別するように学習する。Generator は、Discriminator を騙せるほど類似したデータを生成するように学習する。このように、Discriminator と Generator が敵対し交互に学習し合うことで、Generator は本物のデータに類似したデータを生成できるようになる。

GAN には、損失の振動とモード崩壊の二つの課題がある。損失の振動は、Discriminator と Generator の損失が、長期間にわたって安定するのではなく、激しく振動し始める問題であり、学習が不安定になる。モード崩壊は、Generator が Discriminator を騙す少数の解を見つけた時に起こり、この限られた解以外に何も生成できなくなってしまう問題である。

これらの学習の不安定さを解決した GAN の発展形として、Wasserstein GAN (WGAN) [3] があり、最近の拡張として、Wasserstein GAN - Gradient Penalty (WGAN-GP) [4] がある。本稿では、WGAN-GP を採用する。

### 3. 問題の定式化

トラヒックの時系列の測定データから、そのトラヒックを生成する確率分布を推定する問題の定式化を行う。本稿で扱う測定データは、ある時間間隔において流通したトラヒックのバイト数を表す時系列レベルのトラヒックデータであり、データセット  $\mathcal{X}$  における  $i$  番目の測定データは  $X_i = \{x_t | t = 1, 2, \dots, n\} \in \mathcal{X}$  で表される。

#### 3.1. 特徴量を付与しない場合の定式化

特徴量を付与しない場合では、測定データに類似したデータの生成を目的とする。具体的には、測定したトラヒックからそのトラヒックを生成する確率分布を推定し、類似したトラヒックを生成する。ここでの類似とは、測定したトラヒックとの統計的性質が多面的な観点から等しく、かつ、時系列  $X_i$  が測定したトラヒックとは異なる生成を指す。トラヒック  $X_i$  はそれぞれ特徴量ベクトル  $\mathbf{c}_{X_i} = [c_{X_i}^1, c_{X_i}^2, \dots]^T \in \mathcal{C}$  を持つ。特徴量の要素  $c_{X_i}^k$  は、平均、標準偏差、トレンドなどの統計的性質の値を表す。また、生成したトラヒックを  $\hat{X}_i = \{\hat{x}_t | t = 1, 2, \dots, n\}$ 、その特徴量ベクトルを  $\mathbf{c}_{\hat{X}_i} = [c_{\hat{X}_i}^1, c_{\hat{X}_i}^2, \dots]^T$  と表す。

トラヒックの生成では、測定したトラヒック  $X_i$  を特徴量ベクトル  $\mathbf{c}_{X_i}$  に写像する関数  $F(X_i) = \mathbf{c}_{X_i}$  を考える。特徴量ベクトル  $\mathbf{c}_{X_i}$  から  $\hat{X}_i$  に逆写像する関数  $F^{-1}(\mathbf{c}_{X_i}) = \hat{X}_i$  を推定することで、特徴量ベクトル  $\mathbf{c}_{X_i}$  から対応するトラヒックの生成を実現する。つまり、 $X_i \neq \hat{X}_i$ 、かつ、 $\mathbf{c}_{X_i} = \mathbf{c}_{\hat{X}_i}$  となるような関数  $F^{-1}(\mathbf{c}_{X_i})$  を推定することが目的となる。

#### 3.2. 特徴量を付与する場合の定式化

特徴量を付与する場合では、任意の特徴量を調整したデータの生成を目的とする。具体的には、測定したトラヒックの特性を多面的に再現及び維持したまま、特徴量の一部を任意に置き換えたトラヒックを生成する。特徴量の一部とは、特徴量ベクトル  $\mathbf{c}_{X_i} = [c_{X_i}^1, c_{X_i}^2, \dots]^T$  における一部の要素のことを指す。

特徴量を調整した生成では、特徴量を付与しない場合と同様に、測定したトラヒック  $X_i$  を特徴量ベクトル  $\mathbf{c}_{X_i}$  の一部の要

素を任意の値  $\alpha$  に置き換えたベクトル  $\mathbf{c}'_{X_i} = [\alpha, c_{X_i}^2, \dots]^T$  に写像する関数  $F(X_i) = \mathbf{c}'_{X_i}$  を考える。一部の要素を任意の値  $\alpha$  に置き換えた特徴量ベクトル  $\mathbf{c}'_{X_i}$  から  $\hat{X}_i$  に逆写像する関数  $F^{-1}(\mathbf{c}'_{X_i}) = \hat{X}_i$  を推定することで、任意の特徴量  $\alpha$  を備えたトラヒックデータの生成を実現する。つまり、 $X_i \neq \hat{X}_i$ 、かつ、 $\mathbf{c}'_{X_i} = \mathbf{c}_{\hat{X}_i}$  となるような関数  $F^{-1}(\mathbf{c}'_{X_i})$  の推定が目的となる。

### 4. LSTM と GAN を組み合わせたトラヒック生成

本稿では、深層学習の一種である GAN と LSTM を組み合わせたトラヒックの生成モデルを提案する。提案モデルによるトラヒック生成は、主に以下の三つの手順で行われる。

- (1) LSTM でトラヒックの時系列的な特性を学習する。
- (2) GAN で LSTM が学習した時系列的な特性及び特徴量を学習する。
- (3) GAN の出力を LSTM の初期値として、特徴量の一部を任意に置き換えたトラヒックを再帰的に生成する。

本章ではまず、4.1 で提案モデルの構造について説明する。次に、4.2 で手順 (1)、(2) のモデルの学習について説明する。最後に、4.3 で手順 (3) の再帰的なトラヒックの生成について説明する。

#### 4.1. モデルの構造

提案モデルは、図 1 のような LSTM と GAN を組み合わせた構成である。測定したトラヒックの時系列的な特性を学習するために LSTM を採用している。また、LSTM が学習した時系列的な特性を再現及び維持したまま、特徴量を学習するために、GAN の発展型である WGAN-GP を採用している。これらを組み合わせることにより、測定したトラヒックの特性を多面的に再現及び維持したまま、特徴量の一部を任意に置き換えたトラヒックの生成が可能となる。

前処理として、測定したトラヒック  $X_i$  を分離して 2 次元のベクトルを要素に持つ  $\tilde{X}_i = \{\tilde{x}_t | t = 1, 2, \dots, n\}$  に変換する。変換されたトラヒック  $\tilde{X}_i$  の要素  $\tilde{x}_t = [\tilde{x}_t^{\text{trend}}, \tilde{x}_t^{\text{noise}}]^T$  は、それぞれトレンド成分  $\tilde{x}_t^{\text{trend}}$  とノイズ成分  $\tilde{x}_t^{\text{noise}}$  を表す。トレンド成分  $\tilde{x}_t^{\text{trend}}$  は、測定したトラヒック  $X_i$  の指数平滑移動平均 (Exponential Moving Average; EMA) であり、トラヒックの長期的変動を抽出する。ノイズ成分  $\tilde{x}_t^{\text{noise}}$  は、測定したトラヒック  $X_i$  とトレンド成分  $\tilde{x}_t^{\text{trend}}$  との差であり、トラヒックの短期的変動を抽出する。これにより、トラヒックの長期的及び短期的変動を分離して学習することができるため、トラヒックの特性を効果的に捉えることができる。

LSTM は、まず、測定したトラヒックの特徴量を多面的に学習するため、変換されたトラヒック  $\tilde{X}_i$  の  $j$  番目の要素  $\tilde{x}_j$  と特徴量ベクトル  $\mathbf{c}_{X_i}$  が垂直に連結され、線形層  $f_{\text{in}}$  によって多次元に埋め込まれる。次に、埋め込まれたベクトルが各 LSTM ブロック  $S$  に送られる。最初の隠れ層のベクトル  $\mathbf{h}_0$  は、 $\mathbf{0}$  として初期化される。入力に線形層  $f_{\text{in}}$  を備えた多層 LSTM は、長さ  $n$  の LSTM ブロック  $S$  を隠れ層のベクトル  $\mathbf{h}_j$  に対して再帰的に適用する。各 LSTM ブロックから出力された隠れ層のベクトル  $\mathbf{h}_j$  は、線形層  $f_{\text{out}}$  によって出力トラヒック  $\hat{Y}_i = \{\hat{y}_t | t = 1, 2, \dots, n\}$

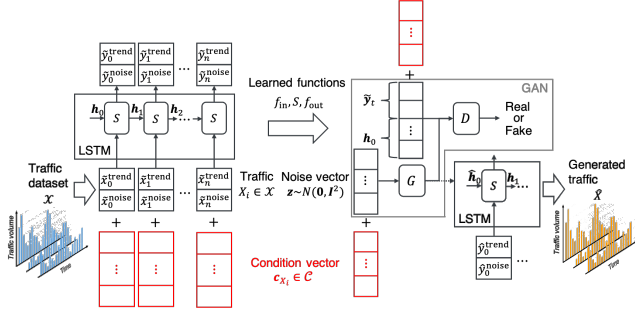


図1 提案モデルの構造

が得られる。上記を要約すると、以下のように表される。

$$h_0 = \mathbf{0}, \quad (4)$$

$$h_{j+1} = A(h_j, f_{in}([\tilde{x}_j^T, c_{X_i}^T]^T)) \quad (j = 0, 1, \dots, n-1), \quad (5)$$

$$\tilde{y}_j = f_{out}(h_j). \quad (6)$$

WGAN-GP は、まず、 $i$  番目の測定データ  $X_i$  を元に本物のデータ  $\zeta_i^{\text{real}}$  を作成する。ただし、本物のデータ  $\zeta_i^{\text{real}}$  は、学習済みの LSTM に変換されたトラフィック  $\tilde{X}_i$  を入力した際の  $i$  番目の出力  $\tilde{y}_i$ 、最初の隠れ層のベクトル  $h_0$  を垂直に連結したものである。次に、本物のデータ  $\zeta_i^{\text{real}}$  の分布  $p_r$  に対する Generator  $G$  の分布  $p_g$  を学習するために、正規乱数ベクトル  $z$  と特徴量ベクトル  $c_{X_i}$  を Generator  $G$  の入力として、データ空間へのマッピングを行う。次に、特徴量ベクトル  $c_{X_i}$  が付与された Generator  $G$  の出力  $\zeta_i^{\text{fake}}$  または本物のデータ  $\zeta_i^{\text{real}}$  が Discriminator  $D$  に入力され、本物か生成物かを識別する単一のスカラー 0, 1 を出力する。上記を要約すると、以下のように表される。

$$\zeta_i^{\text{real}} = [\tilde{y}_i^T, h_0^T]^T, \quad (7)$$

$$G(z, c_{X_i}) = \zeta_i^{\text{fake}}, \quad (8)$$

$$D(\zeta_i^{\text{fake}}, c_{X_i}) = 0 \text{ or } 1. \quad (9)$$

#### 4.2. 学 習

提案モデルは、まず、前処理として、測定したトラフィック  $X_i$  を分離して  $\tilde{X}_i$  に変換した後、Algorithm 1 に従って LSTM の学習を行う。ただし、LSTM の学習における  $j$  番目の正解ラベルを  $\tilde{X}_i$  の 1 ステップ未来のトラフィック  $\tilde{x}_{j+1}$  とするため注意が必要である。ある特定の測定データセット  $\mathcal{X} = \{X_1, X_2, \dots\}$  の  $i$  番目のデータ  $X_i$  は、前処理として、2 次元のベクトルを要素に持つトラフィック  $\tilde{X}_i = \{\tilde{x}_t | t = 1, 2, \dots, n\}$  に変換され、LSTM に入力される。そして、学習済み関数  $f_{in}, S, f_{out}$  が出力される。最初に、総損失が初期化される (2 行目)。次に、Algorithm は全てのデータ  $\mathcal{X}$  について処理を繰り返す (3~10 行目)。LSTM ブロック  $S$  は、再帰的に  $h_j$  を計算する (4~6 行目)。出力された隠れ層のベクトル  $h_j$  は、線形層  $f_{out}$  に入力され、出力トラフィック  $\tilde{y}_j$  が得られる (7 行目)。 $\tilde{y}_j$  と正解ラベル  $\tilde{x}_{j+1}$  から損失が計算され、総損失に追加される (8 行目)。各関数  $f_{in}, S, f_{out}$  は、総損失を逆伝播することによって更新される (11 行目)。上記の手順は、総損失が収束するまで繰り返される (12 行目)。

#### Algorithm 1 LSTM の学習

**Input:** トラフィックのデータセット  $\mathcal{X} = \{X_1, X_2, \dots\}$ , 特徴量ベクトルセット  $\mathcal{C} = \{c_{X_1}, c_{X_2}, \dots\}$

**Output:** 学習済み関数  $f_{in}, S, f_{out}$

```

1: repeat
2:   Loss ← 0
3:   for i from 1 to  $|\mathcal{X}|$  do
4:      $h_0 \leftarrow \mathbf{0}$ 
5:     for j from 0 to  $n-1$  do
6:        $h_{j+1} \leftarrow S(h_j, f_{in}([\tilde{x}_j^T, c_{X_i}^T]^T))$ 
7:        $\tilde{y}_j \leftarrow f_{out}(h_j)$ 
8:       Loss ← Loss +  $\frac{1}{n} \sum_{j=1}^k (\tilde{x}_{j+1} - \tilde{y}_j)^2$ 
9:     end for
10:   end for
11:   Back-propagate Loss and upate weights
12: until stopping criteria

```

次に、Algorithm 2 に従って WGAN-GP の学習を行う。ここで、Generator の重みを  $\theta$ 、Discriminator の重みを  $\omega$  とする。Generator  $G$  には、正規乱数ベクトル  $z$  と特徴量ベクトル  $c_{X_i}$  が入力され、得られた偽物のデータ  $\zeta_i^{\text{fake}}$  または本物のデータ  $\zeta_i^{\text{real}}$ 、特徴量ベクトル  $c_{X_i}$  が Discriminator  $D$  に入力される。ただし、特徴量を付与しない場合の生成のときは、 $c_{X_i}$  に  $\mathbf{0}$  を代入する。Generator と Discriminator が敵対し交互に学習した後、学習済み関数  $G$  を出力する。最初に、Discriminator と Generator の損失が初期化される (2~3 行目)。次に、Algorithm は全てのデータ  $\mathcal{X}$  を繰り返す (4~16 行目)。次に、Discriminator は  $n_{\text{critic}}$  回だけ重み  $\omega$  を更新する (5~11 行目)。次に、正規乱数ベクトル  $z$  を生成し、特徴量ベクトル  $c_{X_i}$  を付与して Generator  $G$  に入力することで偽物のデータ  $\zeta_i^{\text{fake}}$  を生成する (6~7 行目)。 $\zeta_i^{\text{fake}}$  と  $\zeta_i^{\text{real}}$  から損失  $\text{Loss}_d(\zeta_i^{\text{real}}, \zeta_i^{\text{fake}})$  が計算され、Discriminator の損失  $\text{Loss}_d$  に追加される (8~9 行目)。ただし、Discriminator の損失  $\text{Loss}_d(\zeta_i^{\text{real}}, \zeta_i^{\text{fake}})$  は、次式で表される。

$$\text{Loss}_d(\zeta_i^{\text{real}}, \zeta_i^{\text{fake}}) = -(\mathbb{E}_{\zeta_i^{\text{real}} \sim p_r} [D(\zeta_i^{\text{real}})] - \mathbb{E}_{\zeta_i^{\text{fake}} \sim p_g} [D(\zeta_i^{\text{fake}})]) + \text{GP}(\zeta_i'), \quad (10)$$

$$\text{GP}(\zeta_i') = \lambda \mathbb{E}_{\zeta_i^{\text{real}} \sim p_r, \zeta_i^{\text{fake}} \sim p_g} [(\|\nabla_{\zeta_i'} D(\zeta_i')\|_2 - 1)^2], \quad (11)$$

$$\zeta_i' = \epsilon \zeta_i^{\text{real}} + (1 - \epsilon) \zeta_i^{\text{fake}}. \quad (12)$$

Discriminator の重み  $\omega$  は、Discriminator の損失  $\text{Loss}_d$  を逆伝播することによって更新される (10 行目)。再び生成した正規乱数ベクトル  $z$ 、特徴量ベクトル  $c_{X_i}$  から得られた Generator の出力  $\zeta_i^{\text{fake}}$  から Generator の損失  $\text{Loss}_g(\zeta_i^{\text{fake}})$  が計算され、Generator の損失  $\text{Loss}_g$  に追加される (12~14 行目)。ただし、Generator の損失  $\text{Loss}_g(\zeta_i^{\text{fake}})$  は、次式で表される。

$$\text{Loss}_g(\zeta_i^{\text{fake}}) = -\mathbb{E}_{\zeta_i^{\text{fake}} \sim p_g} [D(\zeta_i^{\text{fake}}, c_{X_i})]. \quad (13)$$

Generator の重み  $\theta$  は、Generator の損失  $\text{Loss}_g$  を逆伝播することによって更新される (15 行目)。上記の手順は、損失が収束しないため、任意の回数だけ繰り返される (17 行目)。

---

**Algorithm 2** WGAN-GP の学習

---

**Input:** 本物のデータ  $\zeta_i^{\text{real}}$ , 特徴量ベクトルセット  $\mathcal{C} = \{c_{X_1}, c_{X_2} \dots\}$ **Output:** 学習済み関数  $G$ 

```

1: while  $\theta$  has not converged do
2:    $\text{Loss}_d \leftarrow 0$ 
3:    $\text{Loss}_g \leftarrow 0$ 
4:   for  $i$  from 1 to  $|\mathcal{X}|$  do
5:     for  $t$  from 1 to  $n_{\text{critic}}$  do
6:        $z \sim N(0, I^2)$ 
7:        $\zeta_i^{\text{fake}} \leftarrow G(z, c_{X_i})$ 
8:        $\zeta_j' \leftarrow \epsilon \zeta_i^{\text{real}} + (1 - \epsilon) \zeta_i^{\text{fake}}$ 
9:        $\text{Loss}_d \leftarrow \text{Loss}_d + \text{Loss}_d(\zeta_i^{\text{real}}, \zeta_i^{\text{fake}})$ 
10:      Back-propagate Loss and upate discriminator weights  $\omega$ 
11:    end for
12:     $z \sim N(0, I^2)$ 
13:     $\zeta_i^{\text{fake}} \leftarrow G(z, c_{X_i})$ 
14:     $\text{Loss}_g \leftarrow \text{Loss}_g + \text{Loss}_g(\zeta_i^{\text{fake}})$ 
15:    Back-propagate Loss and upate generator weights  $\theta$ 
16:  end for
17: end while

```

---

### 4.3. 生成

提案モデルは、Algorithm 3 に従ってトラヒックを生成する。特徴量ベクトル  $c_{X_i}$  を入力することで、特徴量ベクトル  $c_{X_i}$  を反映したトラヒック  $\hat{X} = \{\hat{x}_t | t = 1, 2, \dots, n\}$  を出力する。ただし、特徴量を付与しない場合の生成のときは、 $c_{X_i}$  に  $\mathbf{0}$  を代入する。最初に、正規乱数ベクトル  $z$  及び特徴量ベクトル  $c_{X_i}$  から、学習済みの Generator  $G$  によって偽物のデータ  $\zeta_i^{\text{fake}} = [\hat{y}_t, \hat{h}_0]$  を出力する (1~2 行目)。次に、生成するトラヒック  $\hat{X}$  が初期化される (3 行目)。また、学習済みの LSTM の最初の 2 次元ベクトルを要素に持つトラヒック  $\hat{x}_0$ 、最初の隠れ層のベクトル  $h_0$  が、偽物のデータ  $\zeta_i^{\text{fake}}$  の各要素  $\hat{y}_t, \hat{h}_0$  でそれぞれ初期化される (4~5 行目)。次に、LSTM ブロック  $S$  は、再帰的に  $h_j$  を計算する (6~7 行目)。出力された隠れ層のベクトル  $h_j$  は、線形層  $f_{\text{out}}$  によって出力トラヒック  $\hat{y}_j$  が得られる (8 行目)。次に、2 次元のベクトル  $\hat{y}_j$  を複合してスカラー  $\hat{x}_j$  に戻す (9 行目)。複合したスカラー  $\hat{x}_j$  は、生成するトラヒック  $\hat{X}$  に水平方向に連結される (10 行目)。最終的に出力される  $\hat{X} = \{\hat{x}_t | t = 1, 2, \dots, n\}$  が生成したトラヒックとなる。また、特徴量を調整した生成では、特徴量ベクトル  $c_{X_i}$  の一部の要素を任意の値に置き換えたベクトル  $c'_{X_i}$  を  $c_{X_i}$  の代わりに入力して Algorithm 3 を適用することで、任意の特徴量を反映したトラヒックが生成できる。

## 5. 実験と評価

### 5.1. ベースライン

トラヒックの生成タスクにおいて、提案モデルの基本的な特性を確認するために、CO 線形予測モデル、LSTM と比較する。CO 線形予測モデルは、統計学的手法に基づくトラヒックを対象とした生成モデルにおいて代表的なモデルである。機械学習に基づくトラヒックを対象とした生成モデルは、まだ発展途上であるため、トラヒック予測タスクにおいて代表的なモデルで

---

**Algorithm 3** 提案モデルによる生成

---

**Input:** 特徴量ベクトル  $c_{X_i}$ **Output:** 特徴量  $c_{X_i}$  を反映したトラヒック  $\hat{X} = \{\hat{x}_t | t = 1, 2, \dots, n\}$ 

```

1:  $z \sim N(0, I^2)$ 
2:  $[\hat{y}_t, \hat{h}_0] \leftarrow G(z, c_{X_i})$ 
3:  $\hat{X} \leftarrow []$ 
4:  $\hat{x}_0 \leftarrow \hat{y}_t$ 
5:  $h_0 \leftarrow \hat{h}_0$ 
6: for  $j$  from 0 to  $n - 1$  do
7:    $h_{j+1} \leftarrow A(h_j, f_{\text{in}}([\hat{x}_j^T, c_{X_i}^T]^T))$ 
8:    $[\hat{y}_j^{\text{trend}}, \hat{y}_j^{\text{noise}}] \leftarrow f_{\text{out}}(h_j)$ 
9:    $\hat{x}_j \leftarrow \hat{y}_j^{\text{trend}} + \hat{y}_j^{\text{noise}}$ 
10:  append  $\hat{x}_j$  to  $\hat{X}$ 
11: end for

```

---

ある LSTM を比較対象として用いる。また、これらの比較対象は、特徴量を指定した生成ができないため、再現精度の観点からのベースラインとしてのみ使用する。

### 5.2. パラメータ設定とデータセット

提案モデルのパラメータは以下のように設定する。LSTM は、4 層の LSTM ブロック  $S$  を使用し、256 次元の隠れ層のベクトル  $h_j$  を持つ。  $f_{\text{in}}$  の次元は、  $[\hat{x}_j^T, c_{X_i}^T]^T$  と同次元、  $f_{\text{out}}$  は 2 次元に設定される。最適化関数として SGD を使用し、バッチサイズを 2、学習率を 0.01 として、学習を 50000 epoch 繰り返す。GAN は、100 次元の正規乱数ベクトル  $z$  が Generator  $G$  に入力され、1026 次元のベクトルが出力される。最適化関数として Adam を使用し、バッチサイズを 2、学習率を 0.01、  $n_{\text{critic}} = 5$  として、学習を 5000 epoch 繰り返す。

データセットは、MAWI ワーキンググループのトラヒックアーカイブ [5] を用いる。MAWI は、WIDE プロジェクトのワーキンググループの一つであり、ネットワークトラヒックの長期的な測定の分析に焦点を当てている。本稿では、サンプリング点 G おける 2018 年、2019 年に記録された 24 時間にわたる二つのトラヒックトレースを用いる。具体的には、図 2 に示すような 30 秒あたりに通信した双方向のトラヒック量である。図 2 に対して、前処理として標準化を行った。また、図 2 から  $t$  を 50 ずつ区切って、学習データを 100 個作成し、ランダムに訓練データ 80 個、テストデータ 20 個に分割した。

### 5.3. 特徴量

特徴量とは、トラヒックが持つ固有の値のことであり、測定したトラヒック  $X_i$  毎に異なる値を持つ。本稿では、特徴量として平均  $\mu$ 、標準偏差  $\sigma$ 、トレンド  $\rho$  の三つから特徴量ベクトル  $c_{X_i} = [\mu_{X_i}, \sigma_{X_i}, \rho_{X_i}]^T$  を作成する。ただし、トレンド  $\rho$  は、最も単純なトレンド推定方法である線形近似の傾きを使用する。従って、  $c_{X_i} = c_{\hat{X}_i}$  となるトラヒック及び一部の要素を任意の値  $\alpha$  に置き換えた特徴量ベクトル  $c'_{X_i} = [\alpha, \sigma_{X_i}, \rho_{X_i}]^T$  において  $c'_{X_i} = c_{\hat{X}_i}$  となるトラヒックの生成が目的となる。尚、  $c'_{X_i}$  はあくまで一例であり、  $\alpha$  に置き換える要素は任意に選択できる。

### 5.4. 評価指標

本稿は、トラヒックの特性を多面的に評価する必要があるため、コルモゴロフ・スミルノフ (KS) 検定、周波数特性、自己相

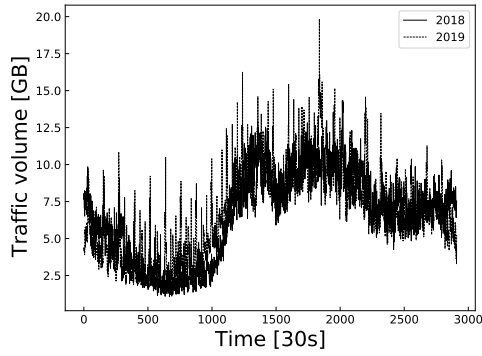


図2 トラヒックトレース (24 時間)

関、特徴量を評価指標として用いる。訓練データの特性を統計的に反映しつつ、訓練データには存在しない、新たなトラヒックを生成することが目的となる。つまり、訓練データとテストデータの特性 (統計量) と同程度の類似度を持つデータが望ましい。KS 検定では、KS 統計量の平均値を使用する。周波数特性、自己相関では、訓練データとの二乗平均平方根誤差 (RMSE) を使用する。特徴量では、各特徴量の平均値を使用する。これらが、訓練データとテストデータの統計量と近ければ、測定したトラヒックとの統計的性質が多面的な観点から等しく、かつ、測定したトラヒックとは異なる生成ができたといえる。

### 5.5. 特徴量を指定しない生成

5.2 で説明した学習データを使用して、CO 線形予測モデル (CO model), LSTM によってトラヒックを生成した。尚、CO 線形予測モデルは、 $k = 1$ ,  $m = 30$  個に設定した。

表 1 に、特徴量を指定しない生成結果の評価を示す。尚、最も良い結果は太字で強調されている。表 1 より、ほぼ全ての評価指標において、LSTM 及び提案モデルが最も良い結果を示していることがわかる。また、平均値において CO 線形予測モデルが最も良い結果となっていることがわかる。しかし、一部の特徴量しか再現できていないため、統計的性質の再現精度が多面的な観点から見て優れているとはいえない。このことから、提案モデルは、測定したトラヒックとの統計的性質の再現精度が多面的な観点から見て統計学的手法よりも優れており、LSTM と同程度の生成精度を持つモデルであるといえる。

### 5.6. 特徴量を指定した生成

表 2 に、特徴量を指定した生成結果の評価を示す。表 2 より、提案モデルは、特徴量を指定することによって生成結果が変化していることがわかる。また、指定した特徴量以外の統計的性質も維持されていることが確認できた。このことから、トラヒックの特性を多面的に再現及び維持したまま、任意の特徴量を調整することができたといえる。また、指定した特徴量と生成した特徴量との値が離れていることがわかる。これは、指定した特徴量の学習データが不足していることが要因として考えられる。しかし、時系列レベルのトラヒックを対象とした生成モデルにおいて、特徴量を任意に調整可能なモデルは他にないため、任意の特徴量によって生成結果が変化したという部分で、研究の第一段階として十分貢献できたのではないかと考え

表 1 特徴量を指定しない生成結果の評価

Evaluation index	CO model	LSTM model	Proposed model	Real data
Average of KS statistics	0.347	0.223	<b>0.229</b>	0.259
RMSE of frequency spectrum	0.00163	<b>0.00141</b>	0.00124	0.00145
RMSE of autocorrelation	0.0899	<b>0.0489</b>	0.0690	0.0313
Average	<b>0.0163</b>	0.168	0.154	0.0719
Standard deviation	0.297	<b>0.348</b>	0.347	0.368
Trend	0.00163	-0.00197	<b>0.00115</b>	-0.000277

表 2 特徴量を指定した生成結果の評価

Evaluation index	condition	Proposed model
Average	-1.5	-0.529
	0	-0.00785
	1.5	0.698
Standard deviation	0.2	0.428
	0.4	0.460
	0.6	0.468
Trend	-0.02	-0.0147
	0	-0.000281
	0.02	0.0123

られる。

## 6. まとめと今後の展望

本稿では、GAN と LSTM を組み合わせたトラヒックの生成モデルを提案した。提案モデルは、任意の特徴量を入力として与えることで、その特徴量を反映したトラヒックが生成できる。実際のトラヒックトレースをデータセットとして使用して、CO 線形予測モデル、LSTM、提案モデルの比較評価を行った。評価の結果、提案モデルは、従来モデルよりも測定したトラヒックの特性を多面的に再現することができ、その特性を維持したまま、任意の特徴量を調整したトラヒックが生成できた。

今後の展望として、指定した特徴量と生成した特徴量との値が離れている問題を解決するため、GAN を使用したデータセットの拡張を行う予定である。また、平均、標準偏差、トレンド以外の特徴量の検討も考えている。

## 文 献

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS 2014), 2014.
- [2] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol.9, no.8, pp.1735-1780, 1997.
- [3] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks," Proceedings of the 34th International Conference on Machine Learning (ICML 2017), pp.214-223, 2017.
- [4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A.C. Courville, "Improved Training of Wasserstein GANs," Advances in Neural Information Processing Systems 30 (NIPS 2017), vol.30, pp.5767-5777, 2017.
- [5] C. Zhang, X. Ouyang, and P. Patras, "Traffic Data Repository at the WIDE Project," Proceedings of 2000 USENIX Annual Technical Conference (USENIX ATC 2000) FREENIX Track, 2000.