

## 予測可能性を示す多変量時系列 GAN モデルの構築 Multivariate time series GAN model for various futures

川島 優輝<sup>†</sup> 六井 淳<sup>†</sup>  
Yuki Kawashima Jun Rokui

### 1 はじめに

IoT (Internet of Things) の進展により、様々な時系列データが取得可能となった。時系列データとは、時間の経過によって値が変化するデータのことであり、過去のデータから未来を予測する時系列予測の分野では、不確実な未来を予測するために、確率のばらつきを考慮した確率的予測 [1] の研究が盛んに行われている。

確率のばらつきは不確実性と呼ばれ、不確実性を考慮した予測を行うことで、将来の複数の状況を想定して意思決定を行うことができる。確率的予測手法の一種である Generative Adversarial Network (GAN) [2] は、従来手法と比較して、計算コストやパラメータの問題が少ない手法として注目を集めている。

Koochali らによる ForGAN [3] は、GAN を用いて複数の予測値を生成し、未来の様々な予測可能性を示した。ForGAN による予測は、平均回帰手法と比較して、優れた予測精度を示している。ForGAN を構成する生成器と識別器には、Recurrent Neural Network (RNN) [4] の一種である Gated Recurrent Unit (GRU) [5] が利用されている。GRU は、過去の情報を記憶するゲート機構を持ち、過去の情報の保持や忘却が可能であるが、データの局所的な特徴抽出や周期性の考慮ができない [6]。

Long- and Short-term Time-series Network (LSTNet) [7] は、データの局所的な特徴抽出や周期性の考慮が可能な多変量時系列予測モデルであり、GRU の問題点を解決するモデルとして挙げられる。ForGAN と LSTNet を組み合わせることで、従来の ForGAN では難しかったデータ間の依存関係や周期性を考慮した確率的多変量時系列予測が可能になると考えられる。本研究では、ForGAN の生成器と識別器に LSTNet を採用することで、データ間の依存関係や周期性を考慮できる多変量時系列予測モデルを構築する。構築したモデルが、実世界の様々なデータに対して、複数の予測可能性を提示できることを明らかにする。

## 2 関連研究

### 2.1 ForGAN

ForGAN は、Conditional GAN (CGAN) [8] をフレームワークとした時系列予測のための GAN である。本研究に関連する GAN, CGAN, ForGAN, Least Squares GAN (LSGAN) [9] についてそれぞれ述べる。

#### 2.1.1 Generative Adversarial Network (GAN)

GAN は、学習データの確率分布を学習し、その分布から合成データを生成することが可能な深層生成モデルである。GAN のアーキテクチャは、生成器と識別器という二つのニューラルネットワークから成る。生成器  $G$

はガウス分布  $p_z$  から取得されるノイズ  $\mathbf{z}$  を入力とし、学習データの分布  $p_{data}$  に従う合成データ  $G(\mathbf{z})$  を生成する。識別器  $D$  は、入力データが  $G(\mathbf{z})$  または学習データ  $\mathbf{x}$  なのかを最適に判別できるように学習を行う。 $G$  は、 $D$  を騙せる本物に近いデータを生成できるように学習を行う。これらを表した GAN の目的関数  $V(D, G)$  を式 (1) に示す。

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (1)$$

GAN の学習は、 $G$  と  $D$  のミニマックス問題であり、 $D$  が本物と偽物の判別がつかなくなったときに学習を終了する。

#### 2.1.2 Conditional GAN (CGAN)

CGAN は、 $G$  と  $D$  の入力に条件データ  $\mathbf{y}$  を加えることで、条件に従ったデータの生成が可能な GAN である。CGAN の目的関数  $V(D, G)$  を式 (2) に示す。

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [\log D(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})|\mathbf{y}))] \quad (2)$$

#### 2.1.3 ForGAN

ForGAN は CGAN に基づく時系列予測モデルである。 $G$  と  $D$  の条件データとして、過去の時系列情報  $\mathbf{c} = x_0, \dots, x_t$  を入力する。 $G$  からは、予測値  $x_{t+1}$  が生成される。 $D$  は  $x_{t+1}$  が  $\mathbf{c}$  に従っているかを判別する。ForGAN の目的関数  $V(D, G)$  を式 (3) に示す。

$$\min_G \max_D V(D, G) = \mathbb{E}_{x_{t+1} \sim p_{data}(x_{t+1})} [\log D(\mathbf{x}|\mathbf{c})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})|\mathbf{c}))] \quad (3)$$

ForGAN の学習により、条件データに対する予測値  $x_{t+1}$  の確率分布をモデル化することが可能となる。

#### 2.1.4 Least Squares GAN (LSGAN)

GAN の損失関数にはクロスエントロピー誤差を用いている。LSGAN は、最小二乗誤差に損失関数を変更した GAN である。LSGAN は、出力に活性化関数を通さないため、誤差の飽和を抑え、勾配消失を防ぐことによって、学習を安定化させた。目的関数はパラメータ  $a, b, c$  を用いて式 (4) で表される。

$$\min_D \{ \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [(D(\mathbf{x}) - b)^2], + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [(D(G(\mathbf{z})) - a)^2] \}, \quad (4)$$

$$\min_G \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [(D(G(\mathbf{z})) - c)^2]$$

$a$  と  $b$  はそれぞれ合成データと学習データのラベルを表し、 $c$  は合成データに対して  $G$  が  $D$  に信じさせたい値である。[9] では、 $b - c = 1$ ,  $b - a = 2$  を条件としており、例として  $a = -1$ ,  $b = 1$ ,  $c = 0$  を挙げている。

<sup>†</sup> 静岡県立大学大学院経営情報イノベーション研究科, Graduate School of Management and Information of Innovation, University of Shizuoka

## 2.2 LSTNet

LSTNet は、畳み込み層 (CNN) [10] と RNN 層 & RNNSkip 層 (GRU), 自己回帰層 (AutoRegressive, AR) から成る多変量時系列予測モデルである。畳み込み層で多変量時系列データの局所的な特徴を抽出し, GRU で長期のパターンを捉えることで, 長期と短期のパターンが混在するデータに対して, 高い予測精度を示している。GRU の出力に AR 層の出力を加算することで, より堅牢な予測を実現している。LSTNet の構造を図 1 に示す。

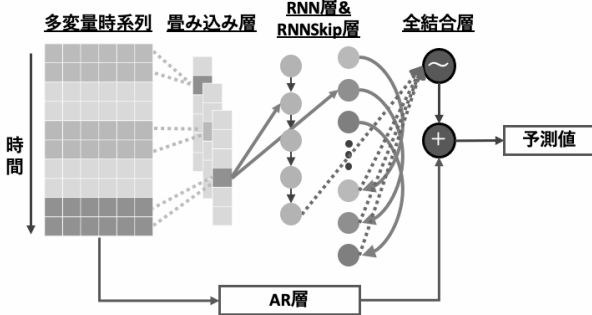


図 1 LSTNet の構造

### 2.2.1 畳み込み層 (CNN)

CNN は、畳み込み層とプーリング層から成るニューラルネットワークであるが、LSTNet の畳み込み処理では、畳み込み層のみでプーリング層は持たない。畳み込み層では、入力である多変量時系列データ  $X$  に対して、局所的な特徴を抽出し、短期のパターンを学習する。窓幅を  $T$ 、次元数を  $D$  とすると、 $X$  は  $T \times D$  の 2 次元行列で表される。  $X$  に対し、カーネルと呼ばれる  $m \times n$  のフィルタを適用することで、特徴の抽出を行う。カーネルが学習重み  $W$  となる。  $k$  番目のカーネルを  $W_k$ 、バイアスを  $b_k$  とすると、畳み込み層での出力  $h_k$  の計算式は式 (5) で表される。

$$h_k = \text{RELU}(W_k * X + b_k) \quad (5)$$

活性化関数には、 $\text{RELU}(x) = \max(0, x)$  を用いる。 (\*) は畳み込み演算を表しており、(\*) の計算式を式 (6) に示す。  $i, j$  をそれぞれ  $i = 0, 1, \dots, (T - m)$ ,  $j = 0, 1, \dots, (D - n)$  とする。  $b$  はバイアスである。

$$\sum_{k=0}^{m-1} \sum_{l=0}^{n-1} c_{kl} \cdot x_{(i+k)(j+l)} + b \quad (6)$$

### 2.2.2 RNN 層

畳み込み層の出力  $h_k$  は、RNN 層と RNNSkip 層の二つに入力される。RNN 層では GRU を用いて、長期の依存関係を学習する。GRU は、過去の情報を記憶するゲート機構を持ち、リセットゲート (reset gate) と更新ゲート (update gate) から成る。リセットゲートは、過去の情報に対して情報をどの程度弱めるかを計算する。更新ゲートは、隠れ状態  $h_t$  をどの程度更新するかを計算する。二つのゲートを利用して、過去の情報の保持や忘却を制御することで、GRU は長期記憶を持つことが

可能となる。LSTNet では、GRU の活性化関数に  $\text{RELU}$  を用いる。時刻  $t$  におけるリセットゲートの出力を  $r_t$ 、更新ゲートの出力を  $u_t$ 、学習重みを  $W$ 、バイアス  $b_r$  と  $b_u$ 、時刻  $t-1$  での出力を  $h_{t-1}$  とすると、 $r_t$  と  $u_t$  の計算式は式 (7)(8) で表される。

$$r_t = \sigma(x_t W_{xr} + h_{t-1} W_{hr} + b_r) \quad (7)$$

$$u_t = \sigma(x_t W_{xu} + h_{t-1} W_{hu} + b_u) \quad (8)$$

$\sigma$  は、活性化関数であるシグモイド関数、 $x_t$  は時刻  $t$  における RNN 層への入力を表している。次に、 $h_{t-1}$  を  $r_t$  倍することで、過去の情報をどの程度弱めるかの計算を行う。この計算の出力を  $c_t$ 、バイアスを  $b_c$  とすると、計算式は式 (9) で表される。

$$c_t = \text{RELU}(x_t W_{xc} + r_t \odot (h_{t-1} W_{hc}) + b_c) \quad (9)$$

$\odot$  は要素ごとの積を計算するアダマール積を表す。最終的な RNN 層での出力  $h_t$  は、式 (10) で表される。

$$h_t = (1 - u_t) \odot h_{t-1} + u_t \odot c_t \quad (10)$$

$h_t$  は、時刻  $t$  における隠れ状態である。

### 2.2.3 RNNSkip 層

RNN 層に用いる GRU は、長期の依存関係を捉えることが可能だが、勾配消失によって限界がある。より長期的な依存関係を捉えるために、RNNSkip 層を導入する。RNNSkip 層は、実世界のデータにおける周期的なパターンを利用した時間的スキップ構造を持つ GRU である。スキップ構造を持った GRU の計算式を式 (11)~(14) に示す。

$$r_t = \sigma(x_t W_{xr} + h_{t-p} W_{hr} + b_r) \quad (11)$$

$$u_t = \sigma(x_t W_{xu} + h_{t-p} W_{hu} + b_u) \quad (12)$$

$$c_t = \text{RELU}(x_t W_{xc} + r_t \odot (h_{t-p} W_{hc}) + b_c) \quad (13)$$

$$h_t = (1 - u_t) \odot h_{t-p} + u_t \odot c_t \quad (14)$$

$p$  はスキップした隠れ状態の数を表す。  $p$  は手動で設定するハイパーパラメータであり、自明な周期を持つデータセットに対しては、設定が容易である。自明でない場合は、調整が必要となる。

RNN 層と RNNSkip 層の出力は、全結合層を用いて結合を行う。RNN 層の出力と RNNSkip 層の出力をそれぞれ  $h_t^R$  と  $h_t^S$ 、バイアスを  $b$  とすると、全結合層の出力  $h_t^D$  は、式 (15) で表される。

$$h_t^D = W^R h_t^R + \sum_{i=0}^{p-1} W_i^S h_{t-i}^S + b \quad (15)$$

### 2.2.4 自己回帰層 (AR)

CNN と RNN を組み合わせたモデルの問題点は、非線形性により、スケール変化に対して弱い点である。LSTNet では、AR 層の出力を  $h_t^D$  に加算することで、スケール変化に対しても敏感に対処できるようになる。入

力を  $x^{ar}$ , 学習重みを  $W^{ar}$ , バイアスを  $b^{ar}$  とすると, 時刻  $t$  における出力を  $h_t^{ar}$  は式 (16) で表される.

$$h_t^{ar} = \sum_{k=0}^{q^{ar}-1} W_k^{ar} x_{t-k}^{ar} + b^{ar} \quad (16)$$

$q^{ar}$  は, 窓幅を表している.  $h_t^D$  と  $h_t^{ar}$  を加算することで, LSTNet の最終的な出力である予測値  $\hat{y}_t$  を得る. この計算式を式 (17) に示す.

$$\hat{y}_t = h_t^D + h_t^{ar} \quad (17)$$

### 3 提案手法

本研究では, ForGAN の生成器と識別器に LSTNet を採用した多変量時系列 GAN モデル (Multivariate Time-series GAN, MTGAN) を構築する.

#### 3.1 MTGAN

MTGAN は, ForGAN の生成器と識別器を GRU から LSTNet に変更し, 学習を安定させるために, 損失関数に LSGAN の仕組みを導入する. 生成器の構造を図 2 に示す. 基本的な構造は LSTNet と同様である. 異なる箇所は, RNN 層と RNNSkip 層の出力にノイズを加算するところである. ForGAN でも RNN の出力にノイズを加え, 全結合層への入力としている. 次に, 識別器の構造を図 3 に示す. 本研究では, LSGAN の仕組みを採用しているため, 全結合層の出力を活性化関数に通さずに, そのまま出力とする. それ以外の構造は LSTNet と同様である. MTGAN の目的関数を式 (18) に示す.

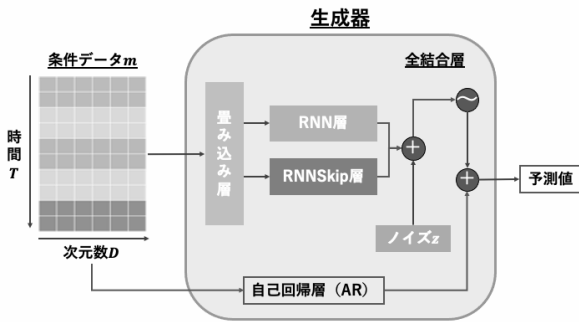


図 2 生成器の構造

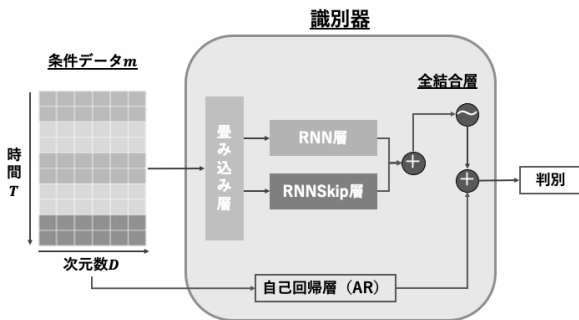


図 3 識別器の構造

$$\min_D L(D) = \left\{ \mathbb{E}_{x_{t+1} \sim p_{\text{data}}(x_{t+1})} [(D(\mathbf{x}|\mathbf{m}) - b)^2] + \mathbb{E}_{z \sim p_z(z)} [(D(G(\mathbf{z})|\mathbf{m}) - a)^2] \right\}, \quad (18)$$

$$\min_G L(G) = \mathbb{E}_{z \sim p_z(z)} [(D(G(\mathbf{z})|\mathbf{m}) - c)^2]$$

$$\mathbf{m} = \begin{pmatrix} x_1^1 & \cdots & x_1^D \\ \vdots & \ddots & \vdots \\ x_T^1 & \cdots & x_T^D \end{pmatrix} \quad (19)$$

$$a = -1, \quad b = 1, \quad c = 0 \quad (20)$$

条件データ  $m$  は, 時間長を  $T$ , 次元数を  $D$  とすると,  $T \times D$  の 2 次元行列である.

#### 3.2 予測方法

MTGAN を用いた時系列予測では, 生成器から複数の予測値を生成することによって, 複数の予測可能性を提示することを目的とする. 複数の予測値を生成する方法について **Algorithm 1** で示す.

#### Algorithm 1 MTGAN の予測方法

```

1:  $N \leftarrow$  生成する個数
2:  $T \leftarrow$  予測する長さ
3:  $m \leftarrow$  条件データ
4:  $m_{\text{init}} \leftarrow$  条件データ
5:  $List1 \leftarrow []$  {生成した予測値を格納するリスト}
6:  $List2 \leftarrow []$  {生成した系列を格納するリスト}
7: start
8: for  $N$  steps do
9:   for  $T$  steps do
10:    if  $T == 0$  then
11:       $m \leftarrow m_{\text{init}}$ 
12:    else
13:       $m \leftarrow m[1 : ]$ 
14:    end if
15:    生成器に  $m$  を入力して予測値を生成
16:    予測値を  $List1$  に追加
17:     $m$  の末尾に予測値を結合
18:  end for
19:  $List1$  を  $List2$  に追加
20:  $List1 \leftarrow []$ 
21: end for
22: end

```

条件データ  $m$  は, 最初の予測値を生成するために必要な情報である.  $m$  を生成器に入力して予測値を得る. 次に, 予測値を  $m$  の末尾に加えて 1 ステップ先へずらすことで, 次の条件データを得る. 得られた新しい条件データを生成器に入力して, 次の予測値を得る. この再帰的な処理を  $T$  回繰り返すことで, 求めたい予測長の結果を得ることができる. さらに, この処理を  $N$  回繰り返すことで,  $N$  個の異なる予測系列を得ることができる.

#### 3.3 評価方法

本研究では, 生成した複数の予測値に対して, 各時刻における最大値と最小値を取得し, 予測を範囲で表現する. 予測を範囲で表現することによって, 未来の様々な

ケースを想定して意思決定を行うことが可能となる。予測範囲の予測精度を評価する方法として、独自に作成した評価指標である Predicted Area Error (PAE) を利用する。PAE の計算式を式 (24) に示す。

$$\text{Pred}_I = \begin{pmatrix} p_{I,1}^1 & \cdots & p_{I,1}^N \\ \vdots & \ddots & \vdots \\ p_{I,T}^1 & \cdots & p_{I,T}^N \end{pmatrix}, \quad (21)$$

$$\text{Pred}_D = \begin{pmatrix} p_{D,1}^1 & \cdots & p_{D,1}^N \\ \vdots & \ddots & \vdots \\ p_{D,T}^1 & \cdots & p_{D,T}^N \end{pmatrix},$$

$$a_t = |\max(p_{d,t}^1 \cdots p_{d,t}^N) - y_t|, \quad (22)$$

$$b_t = |\min(p_{d,t}^1 \cdots p_{d,t}^N) - y_t|, \quad (23)$$

$$\text{PAE} = \sum_{t=0}^T |a_t - b_t| \quad (24)$$

各時刻における最大値と実測値の差の絶対値を  $a$ 、最小値と実測値の差の絶対値を  $b$  とする。  $T$  は予測長、  $D$  は次元数、  $N$  は生成した個数、  $y_t$  は時刻  $t$  における実測値を表す。 PAE は、予測範囲が実測値とずれている程、値が大きくなり、値がゼロに近いほど予測精度が高いことを示す。

PAE は GAN 同士の予測精度比較にしか利用できない。決定的予測手法と予測精度の比較を行うために、Mean Squared Error (MSE) を用いた評価も行う。MSE の計算式を式 (25) に示す。

$$\text{MSE} = \frac{1}{n} \sum_{t=0}^T (\hat{y}_t - y_t)^2 \quad (25)$$

GAN については、各時刻における予測範囲の平均値を取得することで、実測値との比較を行う。

## 4 検証実験

本研究では、MTGAN, ForGAN, LSTNet, GRU の 4 種類の手法で PAE と MSE を用いて予測精度の比較を行う。

### 4.1 実験データセット

実験に用いるデータセットは、道路占有率 (Traffic)、消費電力量 (Electricity)、太陽光発電量 (Solar-Energy) の 3 種類 [11] である。各データセットの詳細を表 1 に示す。

データセット名	T	D	時間間隔
Traffic	17,544	862	1hour
Electricity	26,304	321	15minutes
Solar-Energy	52,560	137	10minutes

表 1 データセットの詳細

Traffic は、2015 年から 2016 年までのサンフランシスコ湾岸区域の高速道路における道路占有率を記録したデータである。Electricity は、2012 年から 2014 年までの 321 世帯の消費電力量 (kWh) を記録したデータである。Solar-Energy は、2006 年のアラバマ州の異なる地点の太陽光発電所で発電された発電量を記録したデータである。各データセットの一部を図 4~6 に示す。

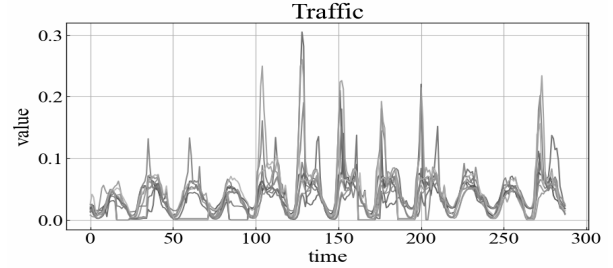


図 4 Traffic

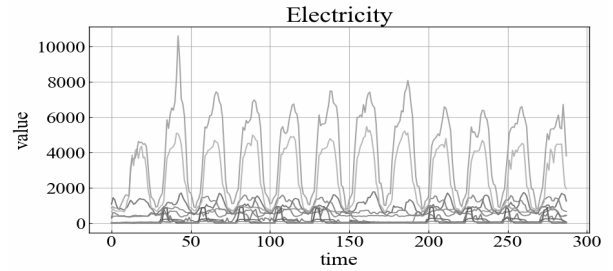


図 5 Electricity

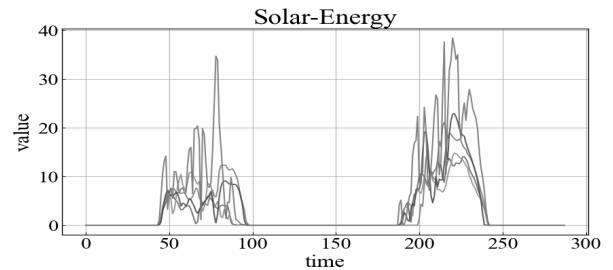


図 6 Solar-Energy

データは、周期的なパターンを持っており、24 時間周期でデータが変動している。さらに Traffic と Electricity は、週単位でも周期性が見られる。これらの周期性を捉えることで、より高度な予測が可能になると考えられる。また、各データは、多変量時系列データであり、多変量入力に対して、局所的な特徴を抽出することで、ForGAN では困難であった特徴を捉えることが可能になると考えられる。

### 4.2 実験内容

実験は、ホールドアウト検証で行う。Traffic では、2 日後、Electricity と Solar-Energy では 1 日後の予測を行う。Traffic では最後から 2 日前以降をテストデータとし、Electricity と Solar-Energy は、最後から 1 日前以降をテストデータとする。それ以前のデータは学習データとする。

MTGAN では、RNNSkip 層のハイパーパラメータ  $p$  を手動で設定する必要がある。Traffic は朝と夕方の方のピーク、Electricity は生活リズム、Solar-Energy は日照時間に合わせて 24 時間周期で変動している。そのため、Traffic は  $p = (60 \div 60_{\min}) \times 24 = 24$ 、Electricity は  $p = (60 \div 15_{\min}) \times 24 = 96$ 、Solar-Energy は  $p = (60 \div 10_{\min}) \times 24 = 144$  に設定する。

各データセットでは、スケールを統一するため、標準化を行う。標準化の式を式 (26) に示す。

$$x_{std} = \frac{x - \bar{x}}{s} \quad (26)$$

$\bar{x}$  は平均,  $s$  は標準偏差を表す。この変換により、データは平均 0, 分散 1 に正規化される。

予測範囲のグラフでは、カーネル密度推定 [12] を用いて予測値に対応する確率を色の濃さで表現する。

### 4.3 実験結果

実験結果を表 2 に示す。PAE の結果から、ForGAN と比較して MTGAN の方が全てのデータセットにおいて高い精度を示した。予測結果の図 7, 9, 11 より、MTGAN は、上下の変動を追従できていることが確認できる。また、再帰的に予測値を生成しているため、予測の後半には誤差が増加するが、図 7, 9, 11 より MTGAN は長期的な予測も対応できていると言える。一方 ForGAN は、PAE が MTGAN と比較して非常に大きく、予測結果の図 8, 10, 12 から予測が外れていることが確認できる。これより、ForGAN は多変量時系列データには適用が困難であると言える。

MSE の結果から、Traffic で MTGAN は LSTNet と同程度の予測精度が確認された。GAN は学習が不安定であるため、MTGAN は LSTNet と比較して予測精度が低下すると予想していたが、Electricity と Solar-Energy では、LSTNet を上回る結果となった。LSGAN の仕組みを損失関数に用いて学習を安定化させたことが影響したと考えられる。GRU は、Traffic と Solar-Energy において、最も悪い予測精度であり、Electricity でも ForGAN の次に悪く、多変量時系列データには向かないことが明らかになった。

手法	評価指標	データセット		
		Traffic	Electricity	Solar-Energy
MTGAN	PAE	<b>13.0204</b>	<b>10.6244</b>	<b>17.5248</b>
	MSE	0.4551	<b>0.6859</b>	<b>0.6648</b>
ForGAN	PAE	29.2095	67.2545	222.5383
	MSE	1.5248	2.0820	1.1030
LSTNet	MSE	<b>0.4075</b>	0.8015	1.2919
GRU	MSE	2.0704	1.6389	1.2957

表 2 実験結果

## 5 おわりに

本研究では、多変量時系列予測において複数の予測可能性を示す多変量時系列 GAN モデルを構築した。

実験では、提案手法である MTGAN と従来手法の ForGAN, LSTNet, GRU で予測精度の比較を行った。独自に作成した指標で GAN 同士の比較、MSE を用いて 4 手法での比較を行った。GAN 同士の比較では、全てのデータセットにおいて MTGAN が ForGAN と比較して高い予測精度を示した。MSE での比較では、MTGAN が LSTNet と同程度の予測精度を示し、一部のデータセットでは LSTNet を上回った。

今後の課題として、周期性のないデータでの検証実験やパラメータの調整が挙げられる。周期性のないデータに対しては、LSTNet ではモデル化が難しいことが想定

されるため、他のアーキテクチャを ForGAN に組み込むことも一つの手として考えられる。また、MTGAN による予測範囲は幅が狭いため、幅を制御する方法の考案も今後の課題である。

### 参考文献

- [1] Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, Vol. 1, No. 1, pp. 125–151, 2014.
- [2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc., 2014.
- [3] Alireza Koochali, Peter Schichtel, Andreas Dengel, and Sheraz Ahmed. Probabilistic forecasting of sensory data with generative adversarial networks - forgan. *IEEE Access*, Vol. 7, pp. 63868–63880, 2019.
- [4] Jeffrey L Elman. Finding structure in time. *Cognitive science*, Vol. 14, No. 2, pp. 179–211, 1990.
- [5] Kyunghyun Cho, B van Merriënboer, Caglar Gulcehre, F Bougares, H Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014.
- [6] 挺屋友幹, 片山晋, 浦野健太, 青木俊介, 米澤拓郎, 河口信夫ほか. Transformer モデルを用いた人流の時系列予測. マルチメディア, 分散協調とモバイルシンポジウム 2021 論文集, Vol. 2021, No. 1, pp. 900–908, 2021.
- [7] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.
- [8] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [9] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2794–2802, 2017.
- [10] Yann Lecun and Y. Bengio. Convolutional networks for images, speech, and time-series. 01 1995.
- [11] GUOKUN LAI. Multivariate time series data sets. <https://github.com/laiguokun/multivariate-time-series-data> \#multivariate-time-series-data-sets (参照 2023-05-29) .
- [12] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, Vol. 33, No. 3, pp. 1065–1076, 1962.

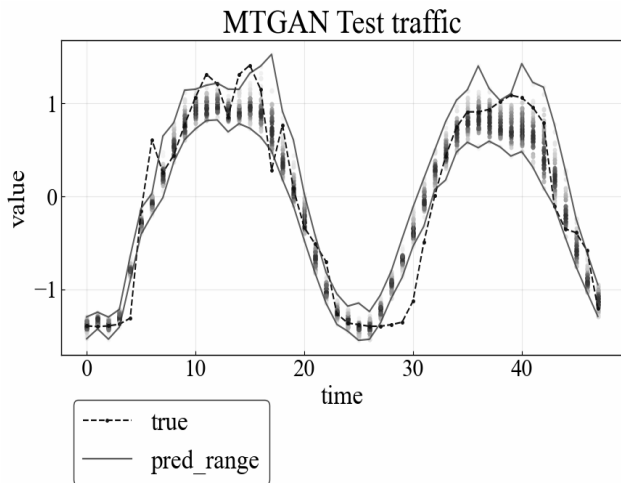


図 7 Traffic のテスト結果例 (MTGAN)

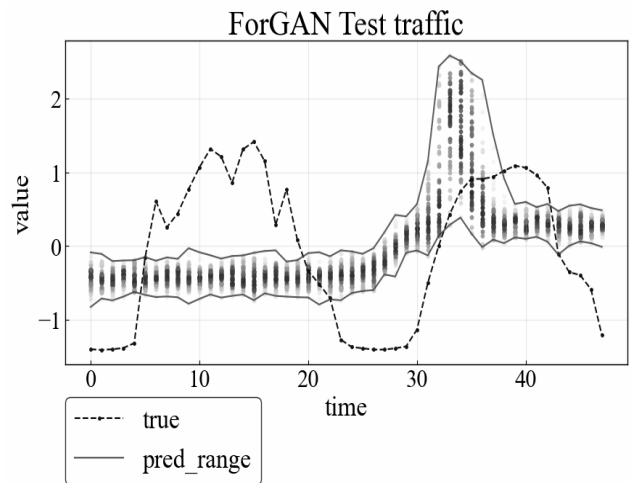


図 8 Traffic のテスト結果例 (ForGAN)

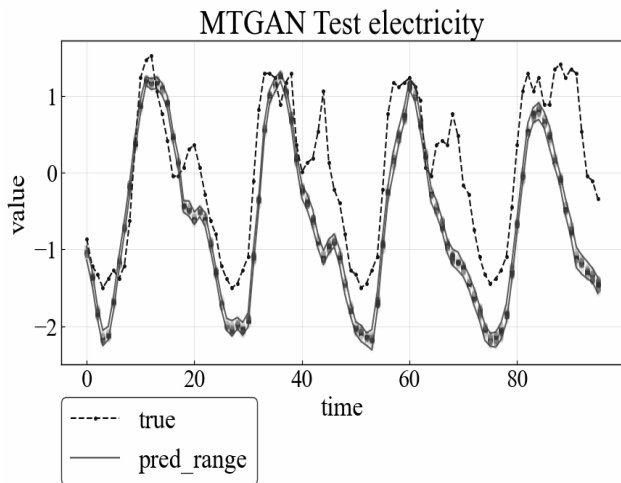


図 9 Electricity のテスト結果例 (MTGAN)

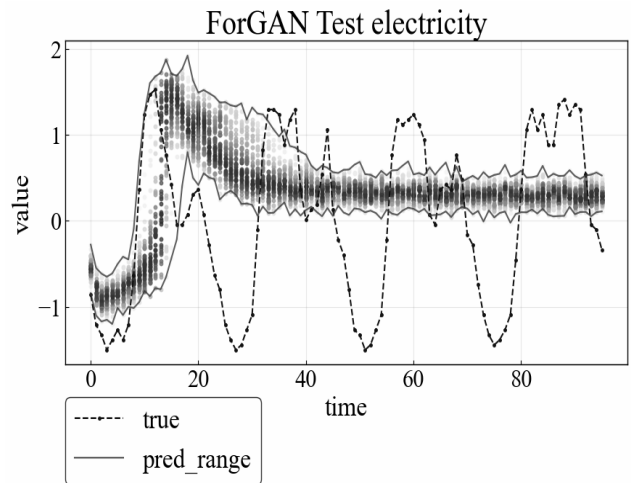


図 10 Electricity のテスト結果例 (ForGAN)

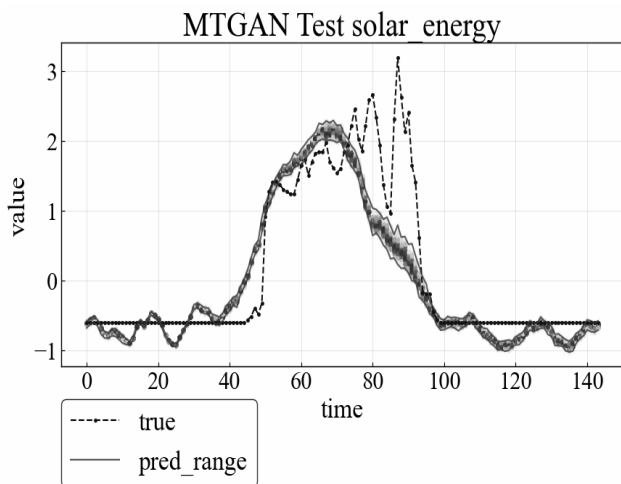


図 11 Solar-Energy のテスト結果例 (MTGAN)

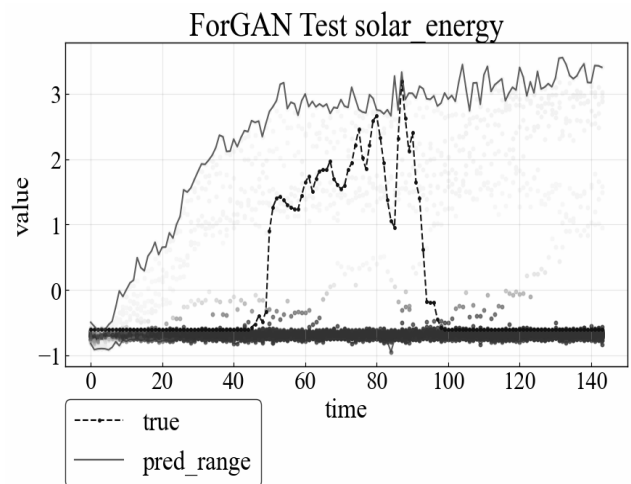


図 12 Solar-Energy のテスト結果例 (ForGAN)