エンコーダ・デコーダ型敵対的生成ネットワークを用いた時系列データの高速異常検知

井手　優介†　　有次　正義††

† 熊本大学大学院自然科学教育部　〒 860-8555 熊本県熊本市中央区黒髪 2 丁目 39 番 1 号
†† 熊本大学大学院先端科学研究部　〒 860-8555 熊本県熊本市中央区黒髪 2 丁目 39 番 1 号
E-mail: †yusuke.i@st.cs.kumamoto-u.ac.jp, ††aritsugi@cs.kumamoto-u.ac.jp

あらまし　近年，時系列データの異常検知手法として敵対的生成ネットワークが注目されている．しかし，敵対的生成ネットワークを用いた既存研究は異常スコアの計算に必要な入力データの復元を行うためにテストサンプル毎に潜在変数の最適化が必要であり，コストが膨大になるという問題がある．そこで本研究では敵対的生成ネットワークを用いて高速に異常を検知することを目的とする．提案手法は Encoder と Decoder により Generator を構築することで，データ空間と潜在空間の双方向のマッピングを訓練時に学習することができ，高速に異常を検知することができる．また，ネットワークに Attention を採用することにより，モデルが重要な特徴量と時刻に注目できるため，高精度に異常を検知することができる．サイバーフィジカルシステムから収集された多変量時系列データを用いた実験では，提案手法が既存手法と比較して高精度かつ高速に異常を検知できることを示す．

キーワード　敵対的生成ネットワーク，時系列データ，異常検知

1 はじめに

異常検知は画像，医療，サイバーフィジカルシステムなど様々な分野で研究されている重要な課題である．例えば，サイバーフィジカルシステムではセンサーから収集されるデータを監視し，異常な振舞いを検知することでサイバー攻撃からシステムを守ることができる [1].

近年では，異常検知手法として敵対的生成ネットワーク [2] が注目されている．敵対的生成ネットワークは Generator と Discriminator と呼ばれる二つのネットワークで構成される生成モデルである．Generator は潜在空間からランダムに生成された潜在変数をデータ空間へマッピングし，本物のデータに類似したデータを生成する．一方で，Discriminator は本物のデータと Generator によって生成されたデータの識別を行う．敵対的生成ネットワークは画像 [3] や音楽 [4] の生成など様々な分野で盛んに研究されているが，高次元データのモデル化において優れているという点から，半教師あり異常検知や教師なし異常検知への適用が注目されている．

例えば，MAD-GAN [5] は多変量時系列データに対する教師なし異常検知手法の一つである．MAD-GAN は LSTM を構成要素とした敵対的生成ネットワークを用い，入力サンプルと復元結果の残差と Discriminator の識別結果を組み合わせたスコアにより異常を検知した．これにより，多変量時系列データに対して，従来の教師なし異常検知手法よりも優れた精度で異常検知を可能にした．

しかし，MAD-GAN はテストサンプルの復元を行うためにサンプル毎に潜在変数を最適化する必要があり，異常検知に膨大なコストを必要とする．そこで本研究では多変量時系列データに対し，敵対的生成ネットワークを用いて高速に異常検知を行う手法を提案する．

提案手法は Encoder と Decoder により Generator を構築することで，データ空間と潜在空間の双方向のマッピングを訓練時に学習することができる．そのため，テストサンプル毎に潜在変数を最適化する必要がなく，コストを削減することができる．また，モデルの Encoder，Decoder，Discriminator の構成要素として Attention を採用する．ネットワークに Attention を採用することで重要な時間と特徴量に注目することが可能となり，高精度に異常を検知することができる．

実験ではサイバーフィジカルシステムから収集される多変量時系列データに対して，提案手法が既存手法と比較して高精度かつ高速に異常を検知できることを示す．

本論文の構成は以下の通りである．2 節で関連研究について述べる．3 節で提案手法に用いる Attention 構造，4 節で提案手法，5 節で評価実験について述べ，6 節で本論文をまとめる．

2 関連研究

異常検知は画像，医療，サイバーフィジカルシステムなど様々な分野で研究されている重要な課題である．例えば，サイバーフィジカルシステムではセンサーから収集されるデータを監視し，異常な振舞いを検知することでサイバー攻撃からシステムを守ることができる [1].

異常検知は訓練に使用可能なラベルの違いにより，教師あり異常検知，半教師あり異常検知，教師なし異常検知の三つに分類することができる [6]．正常データと異常データの両方を含むデータセットを訓練に用いる場合を教師あり異常検知と呼ぶ．教師あり異常検知では入力されたデータが正常と異常のどちら

に属するか予測するモデルを訓練し，モデルの予測結果によってテストデータから異常を検知する．正常データのみを訓練に用いる場合を半教師あり異常検知と呼ぶ．半教師あり異常検知では訓練データから正常データの振舞いをモデルに学習させ，振舞いの違いによってテストデータから異常を検知する．訓練データを必要としない場合を教師なし異常検知と呼ぶ．教師なし異常検知ではテストデータのほとんどが正常データであると仮定し，データの振舞いの違いから異常を検知する．また，ラベルなしデータを正常データと仮定し訓練データとして用いることで，半教師あり異常検知を教師なし異常検知に適用可能である．

一般的に，入手できる異常データの数は正常データの数に比べて非常に少ない．また，データへのラベル付けはその分野の専門家が手作業で行うため，全てのデータへラベルを付けることは容易ではない．そのため，正常データのみで訓練可能な半教師あり異常検知やラベルなしデータのみで実行可能な教師なし異常検知を行うことは重要である．

近年，半教師あり異常検知，教師なし異常検知手法として高次元データのモデル化に優れた敵対的生成ネットワークが注目されている [5], [7], [8]．敵対的生成ネットワークを用いた異常検知ではテストデータとその復元結果の非類似度を異常スコアとする．ただし，復元を行うために必要な潜在変数の生成には三つの方法がある．一つ目は Encoder を用いずに潜在変数を生成する方法である．この方法では標準の敵対的生成ネットワークと同じく Generator と Discriminator のみでモデルを構築し，異常検知対象となるテストデータに最も似たデータを生成できるようにランダムに生成した潜在変数の最適化を行う．この方法を利用した研究として MAD-GAN [5] がある．MAD-GAN は多変量時系列データに対して，LSTM を用いてモデルを構築することで時系列の特徴を捉え，既存の教師なし異常検知手法と比較して優れた精度で異常検知を可能とした．二つ目は BiGAN [9] を用いる方法である．BiGAN は標準の敵対的生成ネットワークにデータ空間から潜在空間へのマッピングを行う Encoder を追加したモデルである．これにより，訓練時にデータ空間と潜在空間の双方向のマッピングが可能となるため，テストデータを復元するための潜在変数を生成することができる．BiGAN を利用した研究として MARU-GAN [7] がある．MARU-GAN は seq2seq [10] を用いて EfficientGAN [11] を拡張したモデルであり，観測値自体は正常だが振舞いが異なる集団型異常の検知を可能とした．三つ目は Autoencoder を用いる方法である．この方法では Generator を Autoencoder により構築することで，BiGAN と同様に訓練時にデータ空間と潜在空間の双方向のマッピングを学習することができる．Autoencoder を利用した研究として BeatGAN [8] がある．BeatGAN は 1 次元 CNN または全結合を用いることで，単変量と多変量の両方の時系列データに対して既存研究を上回る精度を達成した．

本研究では Autoencoder を利用した敵対的生成ネットワークを構築することで，多変量時系列データに対して異常検知を行う．提案手法は Encoder，Decoder，Discriminator のそれぞれに LSTM を用いることで多変量時系列の特徴を捉える．また，各ネットワークにはデータの重要度を捉える Attention を採用する．これにより，入力の重要な部分に注目し高精度に異常を検知することを可能にする．次節で提案手法に用いる Attention について説明する．

3 Attention

Attention は機械翻訳 [12] をはじめとして，画像 [13] や医療 [14] など様々な分野のニューラルネットワークで用いられる重要な構造の一つである．Attention を導入することにより，目的を遂行するために重要な部分に対してモデルが注意を払うことが可能となり，精度や解釈性を改善することができるという利点がある．

Attention を用いたモデルの例として RAIM [14] がある．RAIM は電子健康記録データに対する臨床予測モデルであり，Attention を用いることで臨床判断の根拠の可視化や精度の向上を達成した．RAIM で提案された *Multi-Channel Attention* について説明する．

Multi-Channel Attention : *Multi-Channel Attention* の構造を図 1 に示す．*Multi-Channel Attention* は入力された時系列の時間と特徴量に関して，RNN が過去の時系列から抽出した特徴を参照しながら重要度を計算する．

$X = \{\boldsymbol{x}_t, t = 1, \ldots, T\}$ を長さ T の時系列とし，$\boldsymbol{x}_t \in \mathbb{R}^d$ を時刻 t における d 次元のベクトルとする．X をウィンドウサイズ s_w，ステップサイズ s_s で分割した場合の i 番目の部分時系列を $X_i \in \mathbb{R}^{s_w \times d}$ とすると，時間方向の重要度 $\boldsymbol{\alpha}_i$ は以下の式により計算される．

$$\boldsymbol{s}_i^{time} = \tanh(W_h^{\alpha} \cdot \boldsymbol{h}_{i-1} + X_i \cdot \boldsymbol{w}_X^{\alpha} + \boldsymbol{b}^{\alpha}) \tag{1}$$

$$\alpha_{ij} = \frac{\exp(s_{ij}^{time})}{\Sigma_{j'=1}^{s_w} \exp(s_{ij'}^{time})}, \text{ for } j = 1, \ldots, s_w \tag{2}$$

ただし，$W_h^{\alpha} \in \mathbb{R}^{s_w \times |\boldsymbol{h}|}$，$\boldsymbol{w}_X^{\alpha} \in \mathbb{R}^{d \times 1}$，$\boldsymbol{b}^{\alpha} \in \mathbb{R}^{s_w \times 1}$ は Attention が学習するパラメータであり，$\boldsymbol{h}_{i-1}$ は RNN が X_{i-1} から抽出した隠れ状態ベクトルである．

同様に，特徴量方向の重要度 $\boldsymbol{\beta}_i$ は以下の式により計算される．

$$\boldsymbol{s}_i^{feature} = \tanh(W_h^{\beta} \cdot \boldsymbol{h}_{i-1} + X_i^{\mathrm{T}} \cdot \boldsymbol{w}_X^{\beta} + \boldsymbol{b}^{\beta}) \tag{3}$$

$$\beta_{ik} = \frac{\exp(s_{ik}^{feature})}{\Sigma_{k'=1}^{d} \exp(s_{ik'}^{feature})}, \text{ for } k = 1, \ldots, d \tag{4}$$

ただし，$W_h^{\beta} \in \mathbb{R}^{d \times |\boldsymbol{h}|}$，$\boldsymbol{w}_X^{\beta} \in \mathbb{R}^{s_w \times 1}$，$\boldsymbol{b}^{\beta} \in \mathbb{R}^{d \times 1}$ は Attention が学習するパラメータである．

提案手法ではこの Attention をモデルに採用することで，時系列の特徴をより正確に捉え，高精度に異常を検知する．

4 提 案 手 法

本研究の目的は多変量時系列データが与えられた時，敵対的生成ネットワークを用いて高速に異常を検知することである．

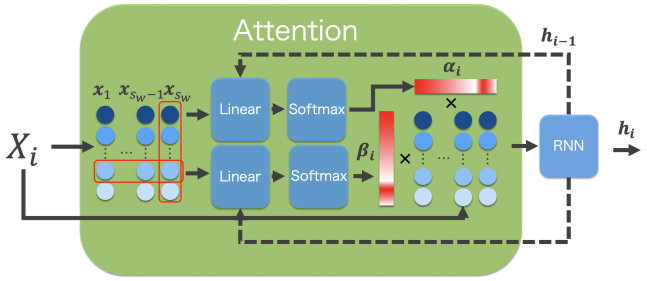

図 1　Multi-Channel Attention

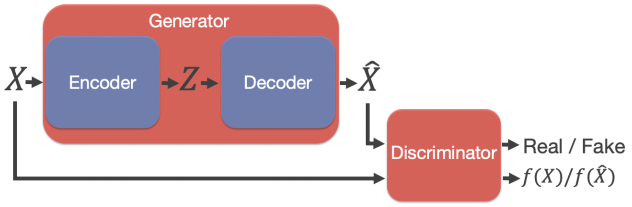

図 2　提案モデルの全体像

以下では本論文に必要な定義を行う．

$X = \{\boldsymbol{x}_t, t = 1, \ldots, T\}$ を長さ T の時系列とし，$\boldsymbol{x}_t \in \mathbb{R}^d$ を時刻 t における d 次元のベクトルとする．本研究では 3 節で述べた Attention を提案手法に採用するため，この多変量時系列 X をウィンドウサイズ s_w，ステップサイズ s_s で分割する．分割後に生成された i 番目の部分時系列を $X_i \in \mathbb{R}^{s_w \times d}$ とする．

正常な M 個の部分時系列から構成される訓練データ $\mathcal{X}_{train} = \{X_i, i = 1, \ldots, M\}$ と正常と異常の両方を含む N 個の部分時系列から構成されるテストデータ $\mathcal{X}_{test} = \{X_i, i = 1, \ldots, N\}$ が与えられたとする．このとき，まず訓練データ $\mathcal{X}_{train}$ を用いて提案モデルの訓練を行う．提案モデルは Generator G と Discriminator D から構成される敵対的生成ネットワークである．また，Generator は Encoder G_E と Decoder G_D から構成される．提案モデルを訓練後，訓練済みのモデルを用いてテストデータ $\mathcal{X}_{test}$ から異常検知を行う．具体的には，テストデータの各部分時系列 X_i の各時刻 t に正常な場合を 0，異常な場合は 1 のラベルを割り当てる．異常を検知するために，本研究では入力サンプルとその復元結果の残差に基づく異常スコアを定義する．もし推論時に正常データが入力された場合，提案モデルは正常データを復元できるように学習しているため正しく復元することができる．その結果，残差は小さくなり異常スコアは低くなる．しかし，異常データが入力された場合，正常データしか学習していないモデルではその特徴を捉えきれず正しく復元することができない．その結果，残差は大きくなり異常スコアは高くなる．よって，入力とその復元結果の残差により，異常検知を行うことができる．以降では表記の簡略化のため，部分時系列 X_i を X と表記する．

4.1　提案モデル

提案モデルを図 2 に示す．提案モデルは Generator と Discriminator で構成されており，Generator は Encoder と Decoder で構成されている．各ネットワークには 3 節で述べた Attention を採用しているため，入力されたデータの重要な時刻と特徴量に注目しながら処理することができる．各ネットワークの詳細について説明する．

4.2　Generator

Generator の構造を図 3 に示す．Generator G は Encoder G_E と Decoder G_D により構成されており，それぞれのネットワークは Attention 層，LSTM 層，全結合層により構成されている．Attention 層は入力された多変量時系列データの時間と特徴量の両方に関する重要度を計算し，入力を重み付けしたものを出力する．LSTM 層は Attention 層により重み付けされたデータの特徴量間の相関と時間依存性を捉える．全結合層は LSTM 層の出力に対して，各時刻の特徴ベクトル毎に独立してマッピングを行う．多変量時系列データ X が与えられた時，Encoder は X を潜在空間にマッピングした潜在変数 $Z \in \mathbb{R}^{s_w \times L}$ を出力する．Z は X の特徴を圧縮したものであり，L は潜在変数の次元数である．

潜在変数 Z が与えられた時，Decoder は Z をデータ空間にマッピングした $\hat{X} \in \mathbb{R}^{s_w \times d}$ を出力する．$\hat{X}$ は Encoder への入力である X を復元したものである．Generator を訓練するために，以下の三つの損失関数を定義する．

Adversarial Loss：Generator は正常データの分布を捉え，正常なサンプルを生成する必要がある．そのため，Discriminator が本物のサンプルと識別できない現実的なサンプルを生成できるように学習を行う．Adversarial Loss は以下のように定義される．

$$L_{adv} = \mathbb{E}_{X \sim p_X}[\log D(X)] + \mathbb{E}_{X \sim p_X}[\log(1 - D(G(X))] \tag{5}$$

ただし，$D(X)$ は Discriminator が出力する確率である．

Feature Loss：敵対的生成ネットワークでは，敵対する Generator と Discriminator が互いに自分の損失を最小にしようとするために学習が不安定になることが知られている．Salimans らはその対策として Discriminator の中間層の特徴量を一致させる *feature matching* [15] を提案した．提案手法ではこの損失関数を採用することで，提案モデルの訓練の安定化を図る．Feature Loss は以下のように定義される．

$$L_{feature} = \mathbb{E}_{X \sim p_X} |f(X) - f(G(X))|_2 \tag{6}$$

提案手法では $f(X)$ として Discriminator の LSTM 層の出力を使用する．

Reconstruct Loss：提案手法は入力とその復元結果の残差により異常を検知するため，入力サンプルを正確に復元する必要がある．よって，入力サンプルと復元結果の L1 距離を損失関数として採用する．Reconstruct Loss は以下のように定義される．

$$L_{rec} = \mathbb{E}_{X \sim p_X} |X - G(X)|_1 \tag{7}$$

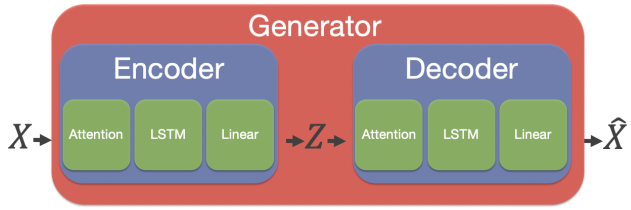

図 3　Generator の構造

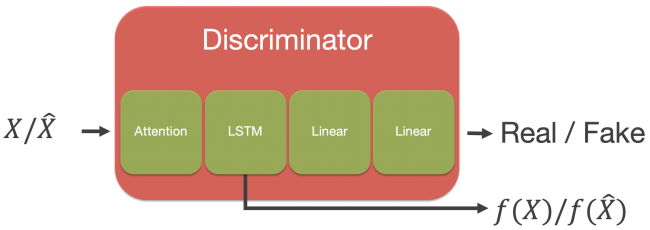

図 4　Discriminator の構造

Generator は上記の関数を組み合わせた以下の損失関数が最小になるように学習する．

$$L_G = \lambda_a L_{adv} + \lambda_f L_{feature} + \lambda_r L_{rec} \tag{8}$$

ただし，λ_a，λ_f，λ_r は各損失関数の重みを決めるパラメータである．

4.3　Discriminator

Discriminator の構造を図 4 に示す．Discriminator は Attention 層，LSTM 層，二つの全結合層から構成されている．このネットワークの最初の三層は Encoder と同じ役割をもつ．ただし，三層目の全結合層は特徴量方向を 1 にマッピングした $\mathbb{R}^{s_w \times 1}$ のベクトルを出力する．最後の全結合層は入力された特徴ベクトルを時間方向に関して 1 にマッピングすることで，Discriminator に入力された時系列全体を考慮した確率を出力する．X または $\hat{X}$ が与えられた時，Discriminator は入力が本物のサンプルか Generator によって復元されたサンプルかを識別する．具体的には，入力が本物のサンプルである確率を出力する．Discriminator は以下の目的関数が最大になるように学習する．

$$L_D = \mathbb{E}_{X \sim p_X}[\log D(X)] + \mathbb{E}_{X \sim p_X}[\log(1 - D(G(X))] \tag{9}$$

4.4　異常スコア

提案手法では入力とその復元結果の残差に基づき異常を検知する．具体的には，以下のように異常スコアを定義する．

$$A(X) = (1-\lambda)|f(X) - f(\hat{X})|_2 + \lambda |X - \hat{X}|_1 \tag{10}$$

ただし，λ は比率を決めるパラメータである．全てのテストサンプルに対して異常スコアを算出後，スコアの最大値が 1，最小値が 0 になるように以下の式で正規化する．

$$A(\hat{X_{test}}) = \frac{A(X_{test}) - min(A(X_{test}))}{max(A(X_{test})) - min(A(X_{test}))} \tag{11}$$

正規化後，閾値 τ を超えるスコアを持つ時刻を異常と判断し，そうでない時刻を正常と判断する．

5　実　　験

本節では提案手法の異常検知性能を評価する．具体的には，Precision，Recall，F1 スコアを用いた精度評価と実行速度評価を行う．また，Attention 層から得られた重要度を考察する．

5.1　データセット

本研究ではオープンデータである SWaT（Secure Water Treatment）データセット [1] を用いて実験を行う．SWaT は安全なサイバーフィジカルシステム設計の調査を目的として作られた水処理施設のテストベッドから収集された時系列データセットである．データは 24 個のセンサーと 27 個のアクチュエータから 11 日間，1Hz で収集され，合計で 51 属性，946,719 点が収集されている．前半の 7 日間は通常運転が行われており，全ての点が正常である．一方で，後半の 4 日間には実際のサイバー攻撃を想定して標的となるセンサーや期間の異なる 36 の異常を発生させており，正常と異常の両方の点を含む．

本研究では攻撃が発生しない前半 7 日間に収集された点を訓練データ，攻撃が発生する後半 4 日間に収集された点をテストデータとして用いる．ただし，データ収集を開始してからシステムが安定するまで 5 時間かかるため，訓練データの最初の 21,600 点は訓練データに含めない．結果として，訓練データは 475,200 点，テストデータは 449,919 点となった．前処理として各データは平均 0，分散 1 に標準化し，ウィンドウサイズ 30，ステップサイズ 10 で分割する．これにより生成された各部分時系列を用いて訓練，評価を行う．

5.2　比 較 手 法

提案手法の性能を評価するために，NoAttention，BeatGAN [8]，MAD-GAN [5] と比較を行った．

提案手法は潜在変数を 32，各中間層のユニット数を 128，LSTM の層数を 3 に設定した．また，Generator の損失関数のハイパーパラメータはそれぞれ $\lambda_a = 1, \lambda_f = 0.1, \lambda_r = 10$ に設定した．Generator は学習率を $g_{lr} = 0.0002$，ハイパーパラメータを $\beta_1 = 0.5, \beta_2 = 0.999$ に設定した Adam を用いて最適化した．一方で，Discriminator は学習率を $d_{lr} = 0.00002$，$\beta_1 = 0.5, \beta_2 = 0.999$ に設定した Adam を用いて最適化した．訓練はバッチサイズ 100，エポック 500 で行い，最も F 値が高いエポックの結果を精度評価に採用した．異常スコアの計算に使用するハイパーパラメータは $\lambda = 0.6$ に設定した．閾値 τ は 0.1 刻みで変化させ，F 値が最も高くなったものを採用した．結果的に $\tau = 0.3$ となった．

NoAttention は提案手法の各ネットワークに採用した Attention 層を取り除いたモデルである．この手法と比較を行うことで Attention 層が精度の向上に貢献しているか確認する．各ネットワークのユニット数や層数，実験に使用するハイパーパ

表 1　精度評価の実験結果

	Precision	Recall	F 値
BeatGAN	0.9931	0.6228	0.7655
MAD-GAN	0.9897	0.6374	0.7754
NoAttention	0.9895	0.6084	0.7535
提案手法	0.9083	0.6824	0.7793

ラメータは提案手法と同様の設定にした．ただし，異常スコアの計算に使用するハイパーパラメータは F 値が最も高くなるものを採用するため，$\lambda = 0.7$，$\tau = 0.5$ に設定した．

BeatGAN は論文 [8] で用いられた全結合層のモデルを元にユニット数とハイパーパラメータを設定した．具体的には，潜在変数を 10，Encoder のユニット数を 256-128-32-10，Decoder のユニット数を 10-32-128-256-1530，Discriminator のユニット数を 256-128-32-1 に設定した．それぞれのネットワークの最適化は Adam を用いて行い，学習率 $lr = 0.0001$，ハイパーパラメータはそれぞれ $\beta_1 = 0.5$,$\beta_2 = 0.999$ に設定した．バッチサイズは 64，エポックは 300 に設定し訓練を行った．異常スコアは入力サンプルと復元結果の L2 距離とし，閾値は提案手法と同じく 0.1 刻みで変化させ，F 値が最も高くなったものを採用した．結果的に閾値は $\tau = 0.1$ となった．BeatGAN は全結合層を用いてマッピングを行うため，多変量時系列データは 1530 次元のベクトルに平坦化し入力した．

MAD-GAN は論文 [5] で SWaT に対して異常検知を行っていたため，そのパラメータを元にモデルを構築した．具体的には，潜在変数を 15 に設定し，Generator をユニット数 100，層数 3 に設定した LSTM とユニット数 5 に設定した全結合層で，Discriminator をユニット数 100，層数 1 に設定した LSTM とユニット数 1 に設定した全結合層で構築した．Generator は学習率 $lr = 0.1$ の勾配降下法を用いて最適化を行い，Discriminator は学習率 $lr = 0.001$，ハイパーパラメータをそれぞれ $\beta_1 = 0.9$,$\beta_2 = 0.999$ に設定した Adam を用いて最適化した．バッチサイズは 500，エポックは 100 に設定し訓練を行った．ただし，前述の通り MAD-GAN は論文で SWaT を用いた異常検知の精度評価を行っていたため，精度は論文に記述されたものを採用し，実行速度は本実験で得られた結果を採用する．

本実験は 128GB のメモリ，Intel Xeon W-2123 3.6GHz の CPU と NVIDIA TITAN V を搭載した Linux のマシン上で実施した．また，プログラムは Python3.6 で記述し，深層学習ライブラリとして Pytorch を使用した．

5.3　実 験 結 果

精度評価：精度評価の結果を表 1 に示す．表 1 に示す通り，提案手法は最も高い Recall と F 値を達成した．これは提案手法が LSTM 層により多変量時系列の特徴を正確に捉えることができるだけでなく，Attention 層により重要な部分の選択が可能となったためである．この結果は Attention 層の有無だけが異なる NoAttention の結果と比較して，F 値を 0.0258 改善させたことからもわかる．Precision は他の手法と比較して劣っているが，異常検知においては誤検知を防ぐことよりも多くの異常を検知できることの方が重要である．さらに，提案手法は最も高い F 値を示した．

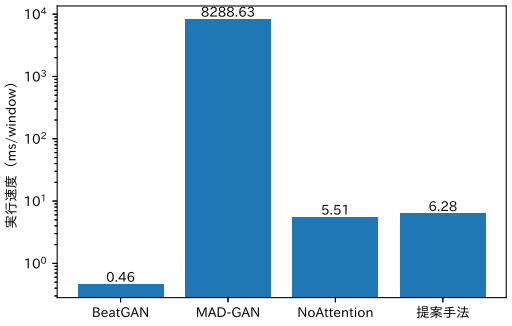

図 5　実験結果（実行速度）

実行速度評価：実行速度評価の結果を図 5 に示す．BeatGAN が最も高速であり，提案手法は 3 番目の速度となった．これは，ネットワーク構造の違いによるものだと考えられる．LSTM 層は時間関係を考慮するために時刻順に入力を処理する必要があり，並列処理することができず，全結合層や CNN 層と比較して実行により多くの時間を必要とする．そのため，全結合層のみで構成された BeatGAN は最も高速に処理することができる．また，NoAttention は LSTM 層を用いているため BeatGAN と比較して実行速度は遅いが，Attention 層を採用していないため提案手法と比較すると処理工程が少なく，より高速に処理することができる．その結果，LSTM 層と Attention 層の両方を用いた提案手法は BeatGAN と NoAttention と比較すると遅くなったと考えられる．しかし，提案手法は訓練時に潜在空間とデータ空間の双方向のマッピングを学習できるため，テストサンプル毎に最適化が必要な MAD-GAN と比較すると 1320 倍高速に実行することができる．さらに，精度評価の通り，提案手法は最も精度良く異常を検知できるという利点がある．

5.4　Attention の重要度の考察

本節では提案手法が異常を検知する際に Attention 層で得られる重要度に関して考察を行う．この重要度は入力したデータに対してネットワークが注目した部分を表している．提案モデルでは Encoder，Decoder，Discriminator のそれぞれに Attention 層を導入しており，それぞれで特徴量と時刻の両方の重要度を計算する．本研究では，正常か異常かの判断に特に関係していると考えられる Discriminator の Attention 層で出力される重要度の考察を行う．

図 6 に Attention 層から出力された重要度のヒートマップを示す．ヒートマップは異常が発生した各時刻における各特徴量の重要度を表している．ただし，正常な場合と比較しやすくするため異常発生前後も含めて表示している．青線の内側が異常が発生した区間であり，外側が正常な区間である．重要度は値が高いほど白に近く低いほど黒に近い．赤枠で囲った部分は異常が発生した特徴量の重要度である．

図 6 から Attention 層で出力された重要度が異常発生部分の

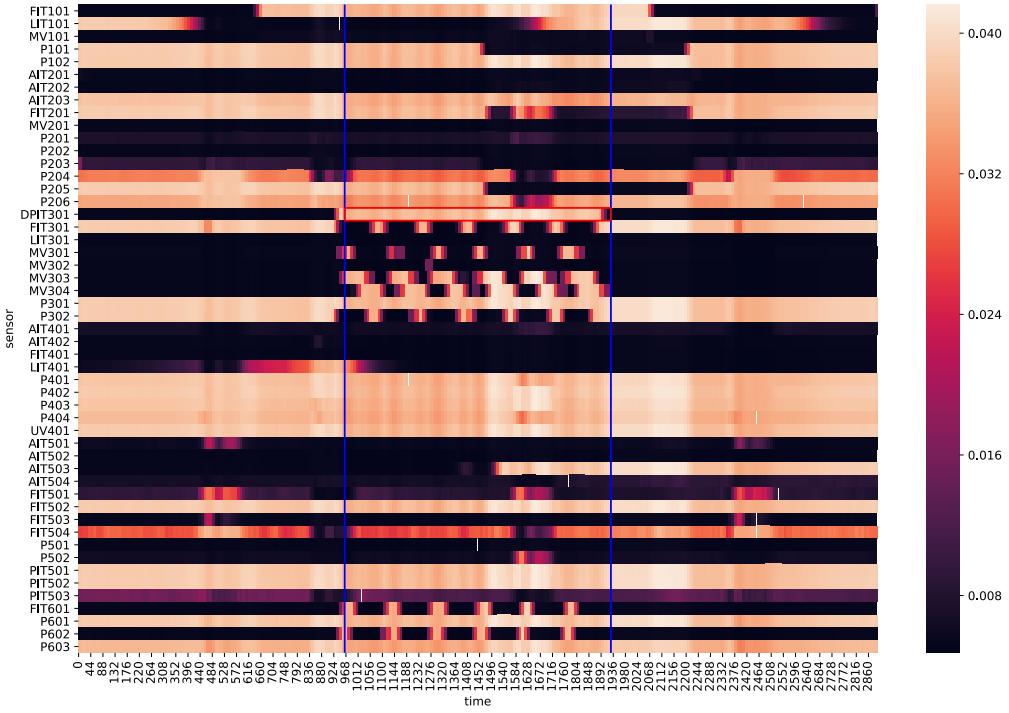

図 6　重要度のヒートマップ

みで高く，それ以外の時刻で低くなっていることがわかる．この傾向は他の検知された異常にもみられた．また，異常が発生したセンサー以外にも異常区間にのみ重要度が高くなる特徴量（FIT301， MV301，MV303 など）があること確認できる．これらの特徴量の波形を確認すると，異常の要因ではないが正常時にない動きをしていることを発見した．このことから，Attention 層は正常時にない振舞いをする特徴量に注目して異常を判断したと考えられ，重要度が異常の要因発見に関連する可能性が示唆された．

6　おわりに

本論文では，多変量時系列データに対して，敵対的生成ネットワークを用いて高精度かつ高速に異常を検知できる手法を提案した．提案手法は Encoder と Decoder を用いて Generator を構築することによりデータ空間と潜在空間の双方向のマッピングが可能である．また，各ネットワークに採用した Attention によりデータの重要な部分に注目することができる．SWaT データセットを用いた評価実験では，提案手法が既存手法と比較して高精度かつ高速に異常を検知できることを示した．さらに，Attention が正常時とは異なる振舞いをする特徴量に対して高い重要度を割り当てることが確認された．

今後の展望として，センサーとアクチュエータで入力方法を変更することが挙げられる．提案手法ではセンサーとアクチュエータを同時に入力したが，アクチュエータは二値しか取らない信号のためセンサーと比較して振舞いを捉えることが難しい．そのため，センサーのみ復元する対象に設定し，アクチュエータをセンサーの特徴を捉える補助的な情報として与えることでモデルがデータの分布を捉えやすくなると考えられる．

文　献

[1] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In Grigore Havarneanu, Roberto Setola, Hypatia Nassopoulos, and Stephen Wolthusen, editors, *Critical Information Infrastructures Security*, pp. 88–99, Cham, 2017.

[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. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.

[3] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[4] Olof Mogren. C-RNN-GAN: A continuous recurrent neural network with adversarial training. In *Constructive Machine Learning Workshop (CML) at NIPS 2016*, p. 1, 2016.

[5] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial

networks. In *International Conference on Artificial Neural Networks*, pp. 703–716. Springer, 2019.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, Vol. 41, No. 3, pp. 1–58, 2009.

[7] 丸千尋, 小林一郎. 敵対的生成ネットワークを用いた時系列データの異常検知への取り組み. 第 11 回データ工学と情報マネジメントに関するフォーラム（DEIM フォーラム 2019）, 3 2019.

[8] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. BeatGAN: anomalous rhythm detection using adversarially generated time series. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4433–4439. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[9] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[10] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pp. 3104–3112. Curran Associates, Inc., 2014.

[11] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.

[12] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[13] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97 of *Proceedings of Machine Learning Research*, pp. 7354–7363, Long Beach, California, USA, 09–15 Jun 2019.

[14] Yanbo Xu, Siddharth Biswal, Shriprasad R. Deshpande, Kevin O. Maher, and Jimeng Sun. RAIM: recurrent attentive and intensive model of multimodal patient monitoring data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pp. 2565–2573, 2018.

[15] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training GANs. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pp. 2234–2242. Curran Associates, Inc., 2016.