DEIM Forum 2019 D8-4

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

丸　千尋†　小林　一郎†

† お茶の水女子大学　〒 112-8610 東京都文京区大塚 2-1-1
E-mail: †{maru.chihiro,koba}@is.ocha.ac.jp

あらまし　Generative Adversarial Network (GAN) は，現実世界の高次元分布をモデル化することができ，異常検知にも適用され始めている．しかし，GAN を用いた異常検知の既存研究は，特定時点の観測値を扱うモデルであるため，観測値自体は正常であるが，その観測値のふるまいが変化する集団型異常を検知することはできない．そこで本研究では，時系列データに存在する集団型異常を GAN を用いて検知することを目的とする．我々は，既存手法の GAN モデルの Encoder に sequence to sequence (seq2seq) の Encoder 側，Generator に seq2seq の Decoder 側，そして Discriminator に Recurrent Neural Network と全結合 Neural Network を採用することで，時系列データに対応可能な GAN モデルを開発した．時系列データの一部の正常な観測値を他の正常な観測値と入れ替えて生成したデータセットを用いて評価を行ったところ，集団型異常を検知するためには，複数の観測値を扱うネットワークを利用する必要があること，我々の GAN モデルは，複数の観測値を扱うネットワークを採用した既存手法と比較して，高い精度で集団型異常を検知できることが明らかになった．

キーワード　GAN，異常検知，集団型異常，時系列データ

1 はじめに

あらゆるものがインターネットと繋がる，Internet of Things の出現により，機械や設備などに取り付けられた各種センサーから膨大な時系列データを容易に収集することが可能になっている．同時に，これらのデータ活用の一つとして，大量のデータをリアルタイムに監視することによって，平常時と異なる状況の発生やその予兆を検知可能な異常検知が盛んに行われている [1]．例えば，クレジットカードの不正利用の検出，病気の診断の援助，サイバーセキュリティの侵入検知，および安全性が重視されるシステムの障害検知など，さまざまなアプリケーションで広く使用されている．

現在の異常検知においては，教師あり異常検知と教師なし異常検知の 2 つの手法が存在する．教師あり異常検知は，入力データにあらかじめ付けられた正常／異常の正解のラベルに基づき，異常判定モデルを学習する手法である．一方，教師なし異常検知は，入力データに正解ラベルを付けず，正常データのみから成る入力データを用いて異常判定モデルを学習する手法である．これら 2 つの手法のうち，教師あり異常検知では，複数の問題が存在する．まず 1 つ目は，機械や設備などに異常が発生することは稀であるため，異常データを大量に収集することが難しいことである．2 つ目は，ラベル付けは人手で行われるため，ラベル付けがされているデータが少ないことである．したがって，近年，教師なし異常検知が盛んに研究されている．この手法を用いれば，正常なデータのみを用いてあらかじめ異常判定モデルを学習しておくことで，モデルから異なる異常な観測値が与えられた時，異常を検知することが可能となる．

近年，現実世界の高次元分布をモデル化することができる，Generative Adversarial Network (GAN) [2] が提案されており，教師なし異常検知に GAN が適用され始めている [3] [4] [5]．例えば，Efficient GAN [4] は，上記の GAN を用いて，正常データのモデルをあらかじめ学習しておき，このモデルに従わない観測値が与えられた時に異常と判定する．しかし，Efficient GAN は，時系列データに潜む異常を検知することができない．Efficient GAN は特定時点の観測値を扱うモデルであるため，各時点の観測値の単変量もしくは多変量を考慮した際に，他の観測値から値が大きく異なる，点異常／文脈依存型異常を検知することはできる．一方で，観測値自体は正常であるが，その観測値のふるまいが変化する集団型異常については，Efficient GAN は複数の観測値を扱うことができないため，検知することができない．そこで，本研究は，GAN モデルを用いて時系列データに潜む集団型異常を検知することを目的とする．

集団型異常を検知するため，Efficient GAN の Encoder，Generator，Discriminator をそれぞれ複数の観測値を扱えるネットワークに拡張した，Multivariate Anomaly detection with Recurrent Units-GAN (MARU-GAN) を提案する．具体的には，Encoder に sequence to sequence (seq2seq) [6] の Encoder 側，Generator に seq2seq の Decoder 側，そして Discriminator に Recurrent Neural Network (RNN) と全結合 Neural Network (NN) を採用することで，時系列データに対応することが可能になる．この MARU-GAN に対して，正常な時系列データから成る，Secure Water Treatmemt (SWaT) データセット [7] の一部の正常な観測値を他の正常な観測値と入れ替えて生成した新たなデータセットを用いて評価を行った．その結果，集団型異常を検知するためには複数の観測値を扱うネットワークを利用する必要があること，我々の MARU-GAN は複数の観測値を扱うネットワークを採用した既存手法と比較して，高い精度で集団型異常を検知することができることが明らかになった．

本論文の構成は以下の通りである．2 章で関連研究について紹介する．3 章で GAN の詳細，4 章で GAN を用いた異常検知について，それぞれ説明する．そして，提案手法である MARU-GAN について 5 章で説明し，6 章で MARU-GAN の評価実験と結果の考察を行う．最後に 7 章で本論文をまとめる．

2 関 連 研 究

異常検知とは，予測されるふるまいに適合しないデータ内のパターンを見つける技術である．異常検知は，クレジットカードの不正利用の検出，病気の診断の援助，サイバーセキュリティの侵入検知，および安全性が重視されるシステムの障害検知など，さまざまなアプリケーションで広く使用されている．異常検知の既存手法の多くは，統計的な技法，クラスタリングに基づく技法，最近傍法に基づく技法，分類に基づく技法に分類することができる．

また，異常検知は，異常を判定するモデルを構築するため，観測値に加えて，正常か異常かを示すラベルが同時に観測されているデータを用いる場合 (教師あり異常検知) と観測値のみが与えられている場合 (教師なし異常検知) の 2 つに分類できる．しかし，前者の教師あり異常検知には複数の問題が存在する．1 つ目は，システム等に異常が発生することは稀であるため，異常データを大量に収集することが難しいことである．2 つ目は，ラベル付けは人手で行われるため，ラベル付けがされているデータが少ないことである．したがって，近年，教師なし異常検知が盛んに研究されている．教師なし異常検知で異常判定モデルを作るためには，訓練データの中に異常な観測値が含まれていない，含まれていたとしてもその影響は無視できると信じられることが必要である．本章では，この教師なし異常検知の既存手法を説明する．

古典的かつ統計的な異常検知は，正常な観測値は確率分布の高い確率の領域で発生し，異常な観測値は確率分布の低い確率の領域で発生するという仮定に基づいている．この手法は，(1) 訓練データが従う確率分布のモデルを仮定し，(2) この確率分布に基づいてテスト用の観測値に対する異常の度合い (異常度) を求める，という 2 つのステップから構成される．代表的な Grubbs 検定は，単変量の訓練データをガウス分布にモデル化し，推定したガウス分布のパラメータ (平均 μ, 分散 σ) を用いて，異常度を算出する [9]．Grubbs 検定を拡張した手法は，M 次元 ($M \geqq 2$) の観測データを多変量ガウス分布にモデル化し，マハラノビス距離を用いて M 次元の観測値を単変量のスカラー値に変換することで，異常度を算出する [10]．他に，複数の確率分布を組み合わせた混合分布モデルを使用する手法がある [11]．これらの手法が有効なのは，観測データが静的な一定値に集まっており，仮定した確率分布に綺麗にモデル化できる場合のみで，実際に利用する際には複数の問題点が存在する．1 つ目は，高次元データになる程，単純な確率分布を仮定したとしても，モデル化することが困難であること，2 つ目は，モデル化した確率分布は固定であるため，時間依存性の高い時系列データには適用することができないことである．

クラスタリングに基づく異常検知は，正常な観測値はクラスタに属し，異常な観測値はどのクラスタにも属さないという仮定に基づく．この手法は，(1) 特定のクラスタリングアルゴリズムを用いて，正常な観測値から成る訓練データをクラスタ化し，(2) テスト用の観測値がどの学習されたクラスタに属するかによって異常度を算出する，という 2 つのステップから構成される．異常度を算出する際には，各クラスタに異常度を割り当てる手法 [12] や，観測値から最も近いクラスタの中心までの距離を異常度とする手法 [13] が使われる．しかし，このクラスタリングに基づく異常検知は，基礎となるクラスタリングアルゴリズムの主な目的がクラスタを見つけることであるため，異常を検知するために最適化されていない．

最近傍法に基づく異常検知は，正常な観測値は密度が高い近傍の集団の中で発生し，異常な観測値は最も近い近傍の集団から離れて発生するという仮定に基づく．この手法は，(1)k 番目に近い近傍への観測値の距離を異常度として用いる，(2) 各観測値の相対密度を異常度として用いる，といった 2 つに分類できる．(1) の k 近傍法は，正常な観測値から成る訓練データが与えられており，このデータセットの中で k 番目に近い近傍への距離を，テスト用の観測値の異常度とする．そして，異常度に対する閾値を用いて，異常かどうかの判定を実施する [14]．他に，異常度を算出するため，距離 d を定数として与え，d 以下の近傍の数 n をカウントする手法も存在する [15]．(2) の相対密度を用いた手法は，テスト用の観測値に対する k 番目に近い観測値を球の半径とし，その球の中に含まれる観測値の数の逆数を異常度とする．これは，密度の低い近傍の集合に属する観測値の方が異常であるという仮定に基づく．最近傍法に基づく異常検知は，分布を仮定する必要がないが，データの変数の次元が高い場合には，個々の変数の寄与が消されてしまうといった欠点が存在する．

分類に基づく異常検知は，(1) 正常な観測値から成るデータを用いて異常判定モデル (分類器) を学習し，(2) 学習した分類器を使って正常／異常の判定を実施する，といった 2 つのステップから構成される．分類器を生成するためのアルゴリズムには，Bayesian Networks (BN)，Support Vector Machines (SVM)，NN などが用いられる．ナイーブ BN を用いた手法 [16] は，テスト用の観測値のクラスごと (正常／異常) の事後確率を推定する．そして，最も大きい事後確率を持つクラスをテスト用の観測値のラベルとする．クラスの事前確率は訓練データを用いて予め推定される．One-class SVM [17] は，SVM を用いて正常な観測値を分類する境界を学習し，学習された境界の中に含まれないテスト用の観測値を異常と判定する．NN を用いた手法は，同数の入力と出力ニューロンを持つ NN を訓練データを用いて学習する．テスト時は，学習された NN を用いて，時刻 t_i のテスト用の観測値 $\boldsymbol{x}^{(i)}$ を復元し，$\boldsymbol{o}^{(i)}$ を得る．そして，復元誤差を時刻 $t_i = 1$ から T まで集計した値 $\sum_{i=1}^{T} \|\boldsymbol{x}^{(i)} - \boldsymbol{o}^{(i)}\|^2$ を異常度とする．NN に Long-Short-Term Memory (LSTM) と Encoder-Decoder モデルを利用した手法が提案され始めている．LSTM を用いた異常検知 [18] は，正常なデータのみから成る訓

練データを用いて，入力された d 点の観測値から次の l 点を予測する LSTM を学習する．そして，学習された LSTM から予測された値と実際の値がどの程度異なるかによって，異常度を算出する．Encoder-Decoder モデルを用いた異常検知 [19] は，正常なデータのみから成る訓練データを用いて，入力された部分時系列をそのまま復元する Encoder-Decoder モデルを学習する．そして，学習された Encoder-Decoder モデルから復元された値と実際の値がどの程度異なるかによって，異常度を算出する．

近年，NN の一つとして，現実世界の高次元分布をモデル化することができる，GAN が提案されており，異常検知にも適用され始めている [3] [4] [5]．特定時点の観測値を扱う Efficient GAN [4] は，他の観測値から大きく異なる観測値を検知するために，GAN を用いることが有用であることを明らかにした．しかし，Efficient GAN は，複数の観測値を扱うことができないため，観測値自体は正常であるが，その観測値のふるまいが変化した集団型異常を検知することができない．MAD-GAN [5] は，時系列データの異常検知のための GAN モデルであるが，時系列データの中の他の観測値から大きく異なる観測値を検知しており，集団型異常を検知することを目的にしていない．そこで，我々は，GAN を用いて時系列データに潜む集団型異常を検知することを目指す．

3 GAN

本章では，Generative Adversarial Network (敵対的生成ネットワーク : GAN) と，GAN の Generator と Discriminator に Encoder という新しいネットワークを導入することで，標準的な GAN を拡張した，Bidirectional GAN (BiGAN) [20] について説明する．

3.1 GAN の詳細

現実世界の複雑な高次元分布をモデル化することができる強力なフレームワークとして，GAN が提案されている [21]．

通常の GAN は，2 つのネットワーク，Generator と Discriminator から構成される．この GAN の全体像を図 1 に示す．Generator (G) は，潜在空間のノイズをデータ空間にマッピングすることにより，実データ $\boldsymbol{x}$ のデータ分布 $p_{data(\boldsymbol{x})}(\boldsymbol{x})$ を学習する．そして，潜在変数 $\boldsymbol{z}$ を与えた時，$p_{data(\boldsymbol{x})}(\boldsymbol{x})$ を用いて実データに近いデータ $G(\boldsymbol{z})$ を生成する．Discriminator (D) は，入力データが与えられた時に，それが実データ（本物）であるのか，Generator によって生成されたデータ (偽物) であるのかを識別し，与えられた入力データが本物である確率 $P(y)$ を出力する．

Generator の最終的な目標は，Discriminator に本物と識別されるような，実データに近いデータを生成することである．一方，Discriminator は，Generator に騙されないよう，与えられた入力データが本物であるのか，偽物であるのかを正確に識別することを目標とする．GAN の学習においては，以下の目的関数 (1) を Generator と Discriminator で共有し，Discriminator に関しては目的関数を最大化，Generator に関しては最小化するように交互に学習することで，上記の目標を達成する．

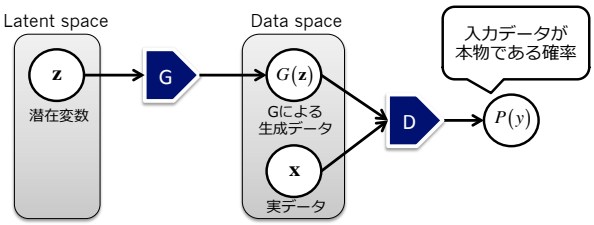

図 1　GAN の全体像

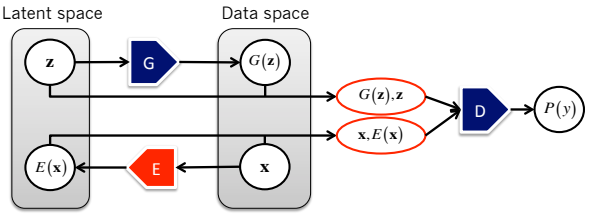

図 2　BiGAN の全体像

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

ここで，$p_{data}(\boldsymbol{x})$ は実データ分布，$p_z(z)$ は潜在変数 $\boldsymbol{z}$ の分布で，訓練データを用いて学習される．$G(\boldsymbol{z})$ は潜在変数 $\boldsymbol{z}$ を Generator に与えた時に生成されるデータを表す．さらに，$D(\boldsymbol{x})$ と $D(G(\boldsymbol{z}))$ は，入力データ $\boldsymbol{x}$ もしくは $G(\boldsymbol{z})$ が Discriminator に与えられた時に Discriminator によって出力される，入力データが本物である確率を表す．

3.2 BiGAN : Encoder を導入した GAN

標準の GAN には，潜在変数 $\boldsymbol{z}$ をデータ空間にマッピングする Generator は存在するが，データ $\boldsymbol{x}$ を潜在空間にマッピングする逆の機能は含まれていない．そこで，標準の GAN に Encoder と呼ばれる，データ $\boldsymbol{x}$ を潜在空間にマッピングし $E(\boldsymbol{x})$ を得るネットワークを導入した，BiGAN が提案されている．Encoder を導入することにより，Discriminator に，データ ($G(\boldsymbol{z})$ または $\boldsymbol{x}$) だけでなく，データと同時に潜在変数も入力する ($(G(\boldsymbol{z}), \boldsymbol{z})$ または $(\boldsymbol{x}, E(\boldsymbol{x}))$ ことで，Discriminator の識別の精度を向上させることが可能となる．BiGAN の全体像を図 2 に示す．

BiGAN における Encoder, Generator, Discriminator の学習は，式 (1) を拡張した下記の目的関数を共有し，Discriminator に関しては目的関数を最大化，Encoder と Generator に関しては最小化するように交互に行われる．

$$\begin{aligned} V(D,E,G) \;=\; & \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log(D(\boldsymbol{x}, E(\boldsymbol{x})))] \\ & + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(\boldsymbol{z}), \boldsymbol{z}))] \end{aligned} \tag{2}$$

式 (2) より，Discriminator の入力が，データだけでなく，データと潜在変数の組になっていることがわかる．Encoder は，Discriminator に実データ $\boldsymbol{x}$ を本物だと信じてもらえるよう，データを潜在空間にマッピングするように学習される．

4 GAN を用いた特定時点の観測値の異常検知

GAN を異常検知に用いた研究は少ない．その中でも，Zenati

ら [4] は，GAN を用いた異常検知の手法である Efficient GAN を提案し，画像およびネットワーク侵入データセットにおいて，最先端の性能を達成した．本章では，Efficient GAN の詳細，異常検知における異常の種類，そして Efficient GAN における問題点について説明する．

4.1 Efficient GAN

Efficient GAN は，3.2 節の BiGAN を異常検知に利用した手法である．BiGAN の Encoder，Generator，Discriminator を正常なデータのみから成る訓練データを用いて式 (2) で学習する．これにより，学習された GAN モデルは正常データを反映したものになっており，このモデルに従わないデータが入力として与えられた時，異常だと判定することが可能になる．Efficient GAN においては，Generator は正常データの分布に従って，潜在変数 $\boldsymbol{z}$ から本物に似たデータ $G(\boldsymbol{z})$ を生成する．Encoder は正常データの分布に従って，実データ $\boldsymbol{x}$ を潜在変数 $E(\boldsymbol{x})$ にマッピングする．Discriminator は正常な実データ $\boldsymbol{x}$ もしくは，Generator によって生成されたデータ $G(\boldsymbol{z})$ が，本物であるか，偽物であるかを正確に分類するように学習される．

この正常データのみを用いて学習された GAN モデルを用いて，未知の入力データ $\boldsymbol{x}$ に対して，異常度 $A(\boldsymbol{x})$ (式 (3)) を算出する．$A(\boldsymbol{x})$ は，再構築損失 $L_G(\boldsymbol{x})$ (式 (4)) と識別損失 $L_D(\boldsymbol{x})$ (式 (5)) の 2 つの項から構成される．α は係数である．$A(\boldsymbol{x})$ の値が大きくなるほど，$\boldsymbol{x}$ が異常であるということを意味する．

$$A(\boldsymbol{x}) = \alpha L_G(\boldsymbol{x}) + (1-\alpha) L_D(\boldsymbol{x}) \tag{3}$$

$$L_G(\boldsymbol{x}) = \|\boldsymbol{x} - G(E(\boldsymbol{x}))\|_1 \tag{4}$$

$$L_D(\boldsymbol{x}) = \sigma(D(\boldsymbol{x}, E(\boldsymbol{x})), 1) \tag{5}$$

再構築損失 $L_G(\boldsymbol{x})$ は，未知の入力データ $\boldsymbol{x}$ と再構築されたデータ $G(E(\boldsymbol{x}))$ の $L1$ ノルムである．$G(E(\boldsymbol{x}))$ は，Encoder を使って $\boldsymbol{x}$ を潜在変数 $E(\boldsymbol{x})$ にマッピングした後，$E(\boldsymbol{x})$ を Generator に与えることで再構築されたデータである．Encoder と Generator は正常データを用いて学習されているため，$\boldsymbol{x}$ が正常である場合，再構築された $G(E(\boldsymbol{x}))$ は $\boldsymbol{x}$ に似たデータとなるはずである．一方，$\boldsymbol{x}$ が異常な場合は，Encoder と Generator が対応していないため，$G(E(\boldsymbol{x}))$ は $\boldsymbol{x}$ と大きく異なるデータとなる．よって，$L_G(\boldsymbol{x})$ の値が大きくなる．

識別損失 $L_D(\boldsymbol{x})$ は，未知の入力データ $\boldsymbol{x}$ と，それを Encoder を使ってマッピングした潜在変数 $E(\boldsymbol{x})$ の組を Discriminator が本物であると識別する確率と，クラス 1 の交差エントロピー損失 σ である．ここで，クラス 1 は入力データ $\boldsymbol{x}$ が本物であることを意味する．交差エントロピーでは，$D(\boldsymbol{x}, E(\boldsymbol{x}))$ の値が 0 に近くなる，すなわち Discriminator によって入力データ $\boldsymbol{x}$ が偽物であると識別される程，$L_D(\boldsymbol{x})$ の値が大きくなる．Discriminator は正常データの識別を正確に行えるように学習されているため，異常データ $\boldsymbol{x}$ が与えられると，Discriminator は $\boldsymbol{x}$ を偽物だと識別し，$D(\boldsymbol{x}, E(\boldsymbol{x}))$ が 0 に近い値となる．その結果，$L_D(\boldsymbol{x})$ の値が大きくなる．

上記の $A(\boldsymbol{x})$ を用いて，全ての未知の入力 $\boldsymbol{x}$ に対して異常度を算出し，上位 N%のデータを異常と判定する．

4.2 異常検知における異常の種類

異常検知における異常は，3 種類に分類することができる [22]．1 つ目は，他の観測値から大きく異なる観測値を検知する，点異常である．例えば，気温の年間の推移において，気温が 100 度である観測値は点異常であると判定される．2 つ目は，多変量のデータにおいて，特定の状況において異常である観測値を検知する，文脈依存型異常である．例えば，6 月の気温において，気温が 0 度である観測値は文脈依存型異常であると判定される．年間を通して気温が 0 度になる可能性はあるが，6 月に 0 度になることは，通常起こり得ないことである．これは，時期と気温の 2 つの変数を考えた時に異常だと判定される．3 つ目は，他のデータと比べ，ふるまいが異なる観測値の集まりを検知する，集団型異常である．これは，観測値自体は正常であるが，その観測値が複数集まった時のふるまいが変化したことを意味しており，時系列データに存在する異常である．例えば，人間の心電図のふるまいが変化した時，その観測値の集まりは集団型異常であると判定される．

点異常，文脈依存型異常はそれぞれ，1 点の観測値の 1 次元，K 次元 (K>1) を見れば解くことができる．一方で，集団型異常は N 点の観測値の K 次元 (K≧1) を見れば解くことができる．

4.3 Efficient GAN の問題点

4.1 節の Efficient GAN における Encoder，Generator，Discriminator は，観測値を個々に扱うネットワークであるため，特定時点の観測値の異常である点異常および文脈依存型異常を検知することはできる．しかし，観測値自体は正常であるが，その観測値のふるまいに異常が存在するような複数の観測値を扱うことによって異常を検知することが可能となる集団型異常を検知することはできない．そこで本研究では，GAN モデルを用いて，時系列データに潜む集団型異常を検知することを目的とする．

5 GAN を用いた時系列データの異常検知

本稿では，4 章の Efficient GAN を拡張し，時系列データを処理可能な GAN を用いた集団型異常検知の手法を説明する．Seq2seq について説明した後，提案する GAN モデルの Encoder，Generator，Discriminator についてそれぞれ説明する．

5.1 Seq2seq

Seq2seq とは，機械翻訳やメディア変換に用いられる NN の一種であり，Encoder，Decoder と呼ばれる 2 つの別々の RNN で構成される．まず Encoder で入力された部分時系列を低次元のベクトルに圧縮する．この圧縮されたベクトルには，入力部分時系列の特徴が含まれている．その後，Decoder で低次元のベクトルから部分時系列を復元する．

本稿では，MARU-GAN の Encoder に seq2seq の Encoder 側，Generator に seq2seq の Decoder 側を採用する．

5.2 提 案 手 法

本稿では，長さ T の部分時系列 $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T)}\}$ を考え

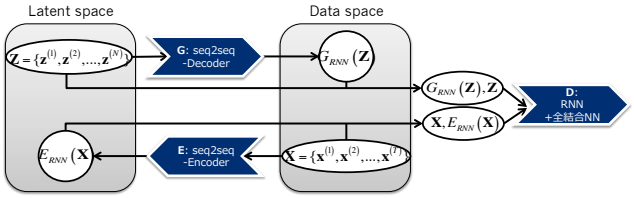

図 3　MARU-GAN の全体像

る．時刻 t_i の観測値 $\boldsymbol{x}^{(i)} \in \mathbb{R}^M$ は，M 個の変数を持つ M 次元ベクトルである．本稿では，このような長さ T の複数の部分時系列 $\boldsymbol{X}$ が時系列データを構成していると仮定する．部分時系列 $\boldsymbol{X}$ を扱うため，Efficient GAN の Encoder，Generator，Discriminator をそれぞれ複数の観測値を扱えるネットワークに拡張した，Multivariate Anomaly detection with Recurrent Units-GAN (MARU-GAN) を提案する．MARU-GAN の全体像を図 3 に示す．Efficient GAN の Encoder に seq2seq の Encoder 側，Generator に seq2seq の Decoder 側，Discriminator に RNN と全結合 NN を採用する．MARU-GAN の Encoder，Generator，Discriminator について，詳細に説明する．

5.2.1　Encoder

Encoder には，seq2seq の Encoder 側を採用する．Encoder のネットワークを図 4 に示す．Encoder は長さ T の固定長の部分時系列 $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T)}\}$ を入力部分時系列 $\boldsymbol{X}$ として受け取り，Encoder の最後の隠れ状態 $(\boldsymbol{h}_1^{(T)}, \boldsymbol{h}_2^{(T)}, \boldsymbol{h}_3^{(T)})$ を $E_{RNN}(\boldsymbol{X})$ として出力する．この $E_{RNN}(\boldsymbol{X})$ には，入力部分時系列の特徴が圧縮されている．本稿では，Encoder の隠れ層を 3 ユニットとして実装した．Encoder は式 (6) の目的関数を最小化するように学習される．

$$V(E) = \mathbb{E}_{\boldsymbol{X} \sim p_{data}(\boldsymbol{X})}[\log(D_{RNN}(\boldsymbol{X}, E_{RNN}(\boldsymbol{X})))] \tag{6}$$

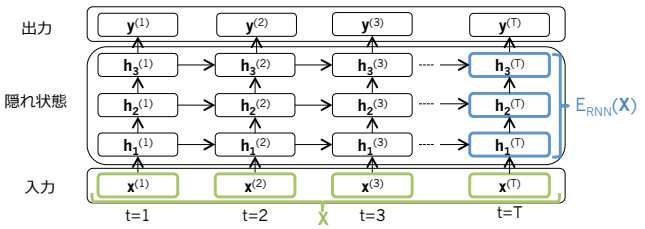

図 4　Encoder の全体像

5.2.2　Generator

Generator には，seq2seq の Decoder 側を採用する．Generator のネットワークを図 5 に示す．Generator は潜在変数 $\boldsymbol{Z}$ を受け取り，それを Generator の最初の隠れ状態 $(\boldsymbol{s}_1^{(1)}, \boldsymbol{s}_2^{(1)}, \boldsymbol{s}_3^{(1)})$ に設定する．そして，開始を表す入力を与えると，各時刻 t_i で $\boldsymbol{x}^{(i)}$ が出力される．全ての $\boldsymbol{x}^{(i)}$ を結合した部分時系列 $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T)}\}$ を $G_{RNN}(\boldsymbol{Z})$ とする．本稿では，Generator の隠れ層を 3 ユニットとして実装した．Generator は式 (7) の目的関数を最小化するように学習される．

$$V(G) = \mathbb{E}_{\boldsymbol{Z} \sim p_{\boldsymbol{Z}}(\boldsymbol{Z})}[\log(1 - D_{RNN}(G_{RNN}(\boldsymbol{Z}), \boldsymbol{Z}))] \tag{7}$$

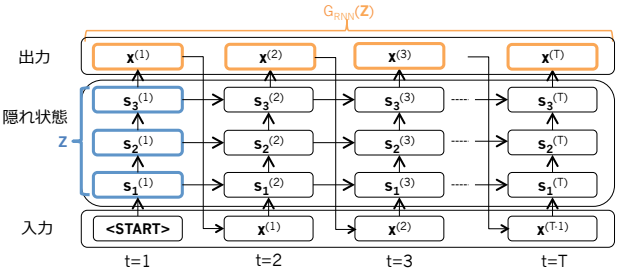

図 5　Generator の全体像

5.2.3　Discriminator

Discriminator には，RNN と全結合 NN の 2 種類の NN を用いる．Encoder の全体像を図 6 に示す．Discriminator は，データ (部分時系列) と潜在変数を結合したベクトルを入力として受け取る．RNN では，部分時系列 $G_{RNN}(\boldsymbol{Z})$ もしくは，$\boldsymbol{X}$ を入力として，最後の隠れ状態 $\boldsymbol{h}_{dis}^{(T)}$ を得る．この $\boldsymbol{h}_{dis}^{(T)}$ には，入力された部分時系列の特徴が圧縮されている．そして，得られた $\boldsymbol{h}_{dis}^{(T)}$ と，潜在変数 $\boldsymbol{Z}$ もしくは，$E_{RNN}(\boldsymbol{X})$ を結合し，全結合 NN に入力する．全結合 NN は，入力された部分時系列が本物である確率 $P(y)$ を出力し，これが Discriminator の出力となる．Discriminator は式 (8) の目的関数を最大化するように学習される．

$$\begin{aligned} V(D) = {} & \mathbb{E}_{\boldsymbol{X} \sim p_{data}(\boldsymbol{X})}[\log(D_{RNN}(\boldsymbol{X}, E_{RNN}(\boldsymbol{X})))] \\ & + \mathbb{E}_{\boldsymbol{Z} \sim p_{\boldsymbol{Z}}(\boldsymbol{Z})}[\log(1 - D_{RNN}(G_{RNN}(\boldsymbol{Z}), \boldsymbol{Z}))] \end{aligned} \tag{8}$$

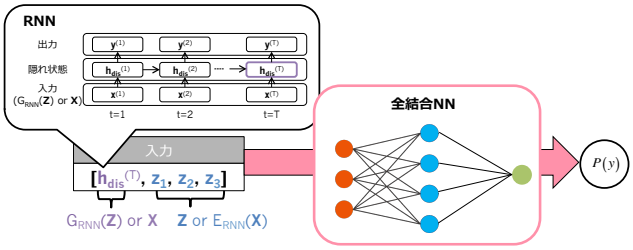

図 6　Discriminator の全体像

5.2.4　異常度の算出

5.2 節の Encoder，Generator，Discriminator を正常な部分時系列のみから構成される訓練データを用いて学習した後，各部分時系列 $\boldsymbol{X}$ の異常度 $A_{RNN}(\boldsymbol{X})$ を，式 (3) を拡張した式 (9) を用いて算出する．

$$A_{RNN}(\boldsymbol{X}) = \alpha L_{G_{RNN}}(\boldsymbol{X}) + (1 - \alpha) L_{D_{RNN}}(\boldsymbol{X}) \tag{9}$$

$$L_{G_{RNN}}(\boldsymbol{X}) = \|\boldsymbol{X} - G_{RNN}(E_{RNN}(\boldsymbol{X}))\|_1 \tag{10}$$

$$L_{D_{RNN}}(\boldsymbol{X}) = \sigma(D_{RNN}(\boldsymbol{X}, E_{RNN}(\boldsymbol{X})), 1) \tag{11}$$

$L_{G_{RNN}}$ (式 (10)) においては，入力部分時系列 $\boldsymbol{X}$ に含まれる観測値ごとに算出した値を集計して $\boldsymbol{X}$ の $L_{G_{RNN}}$ とする．$L_{D_{RNN}}(\boldsymbol{X})$ (式 (11)) の $D_{RNN}(\boldsymbol{X}, E_{RNN}(\boldsymbol{X}))$ では，Discriminator が部分時系列 $\boldsymbol{X}$ ごとに本物だと識別する確率を出力する．上記の $A_{RNN}(\boldsymbol{X})$ を用いて，全ての未知の入力部分時系列 $\boldsymbol{X}$ に対して異常度を算出し，上位 N%の部分時系列を異常と判定する．

MARU-GAN による異常検知のアルゴリズムを Algorithm 1 に示す．

Algorithm 1 MARU-GAN based anomaly detection

Require: K,$\boldsymbol{Z}$,$\boldsymbol{X}$,α,N

for K epochs **do**
  Training:
  Generate time-series subsequences from the latent variables
  $\boldsymbol{Z}$, <START> $\Rightarrow G_{RNN}(\boldsymbol{Z})$
  Map from time-series subsequences to the latent space
  $\boldsymbol{X} \Rightarrow E_{RNN}(\boldsymbol{X})$
  Discriminate time-series subsequences
  $D_{RNN}(G_{RNN}(\boldsymbol{Z}), \boldsymbol{Z})$
  $D_{RNN}(\boldsymbol{X}, E_{RNN}(\boldsymbol{X}))$
  Update the parameters by maximizing $V(D)$
  $V(D) = \mathbb{E}_{X\sim p_{data}(X)}[\log(D_{RNN}(\boldsymbol{X}, E_{RNN}(\boldsymbol{X})))] + \mathbb{E}_{\boldsymbol{Z}\sim p_{\boldsymbol{Z}}(\boldsymbol{Z})}[\log(1 - D_{RNN}(G_{RNN}(\boldsymbol{Z}), \boldsymbol{Z}))]$ (Equation (8))
  Update the parameters by minimizing $V(G)$
  $V(G) = \mathbb{E}_{\boldsymbol{Z}\sim p_{\boldsymbol{Z}}(\boldsymbol{Z})}[\log(1 - D_{RNN}(G_{RNN}(\boldsymbol{Z}), \boldsymbol{Z}))]$ (Equation (7))
  Update the parameters by minimizing $V(E)$
  $V(E) = \mathbb{E}_{\boldsymbol{X}\sim p_{data}(\boldsymbol{X})}[\log(D_{RNN}(\boldsymbol{X}, E_{RNN}(\boldsymbol{X})))]$ (Equation (6))
  Save the parameters of Encoder, Generator, and Discriminator in the current epoch
  Validating:
  Compute anomaly scores $A_{RNN}(\boldsymbol{X})$ using validation dataset
  $L_{G_{RNN}}(\boldsymbol{X}) = \|\boldsymbol{X} - G_{RNN}(E_{RNN}(\boldsymbol{X}))\|_1$ (Equation (10))
  $L_{D_{RNN}}(\boldsymbol{X}) = \sigma(D_{RNN}(\boldsymbol{X}, E_{RNN}(\boldsymbol{X})), 1)$ (Equation (11))
  $A_{RNN}(\boldsymbol{X}) = \alpha L_{G_{RNN}}(\boldsymbol{X}) + (1 - \alpha)L_{D_{RNN}}(\boldsymbol{X})$ (Equation (9))
  Define N% of the time-series subsequences with the highest $A_{RNN}(\boldsymbol{X})$ as anomalous
  Compute F-value using the anomaly detection results
end for
Testing:
Restore the model with the highest F-value in K epochs
Compute anomaly scores $A_{RNN}(\boldsymbol{X})$ using test dataset based on the above model
$L_{G_{RNN}}(\boldsymbol{X}) = \|\boldsymbol{X} - G_{RNN}(E_{RNN}(\boldsymbol{X}))\|_1$ (Equation (10))
$L_{D_{RNN}}(\boldsymbol{X}) = \sigma(D_{RNN}(\boldsymbol{X}, E_{RNN}(\boldsymbol{X})), 1)$ (Equation (11))
$A_{RNN}(\boldsymbol{X}) = \alpha L_{G_{RNN}}(\boldsymbol{X}) + (1 - \alpha)L_{D_{RNN}}(\boldsymbol{X})$ (Equation (9))
Define N% of the time-series subsequences with the highest $A_{RNN}(\boldsymbol{X})$ as anomalous
Compute F-value, accuracy, and false positive rate using the anomaly detection results

6 実　　験

5.2 節で提案した MARU-GAN が時系列データに潜む集団型異常を検知可能か，SWaT データセット [7] を用いて本章で評価する．我々は，正常な観測値のみから構成される SWaT データセットに，集団型異常を人工的に作成し混ぜることで，新たに生成されたデータセットを用いて実験を行った．比較対象として，GAN を用いた特定時点の観測値の異常を検知する手法である，Efficient GAN，Encoder-Decoder モデルを異常検知に用いた手法である，EncDec-AD，そして LSTM を異常検知に用いた手法である，LSTM-AD を採用した．このうち，EncDec-AD と LSTM-AD は，集団型異常検知に向けた手法である．

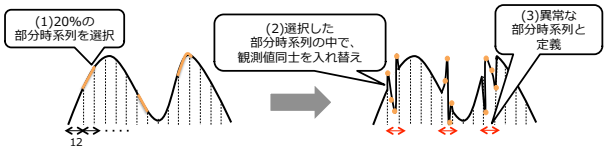

図 7　疑似集団型異常時系列データの生成方法

6.1　実験データ: SWaT データセット

SWaT (Secure Water Trearment) データセットは，最新の水処理プラントを縮小したレプリカから収集された 51 次元のデータから構成される．SWaT における水の浄化プロセスは，6 段階のプロセス (プロセス 1: 原水の供給と貯蔵，プロセス 2: 薬品注入，プロセス 3: 限外ろ過，プロセス 4: 脱塩素処理，プロセス 5: 逆浸透，プロセス 6: 供給する水の貯蔵) から構成されている．それぞれのプロセスにおいて，センサー・アクチュエータが 1 秒ごとに複数の指標 (水位，流量，伝導率，pH，酸化還元電位，差圧，水圧) を測定し，プロセス全体で 51 種類の指標を測定する．

本研究では，稼働している水処理プラントのレプリカから連続した 7 日間に収集された，475,200 点の正常な観測値から成るデータセットを用いて各手法の評価を行った．

6.2　集団型異常の生成

6.1 節の SWaT データセットは正常な観測値のみから構成されたデータセットである．そこで，集団型異常を人工的に生成し，SWaT データセットに混ぜることで，新しい実験データセットを生成する．

事前処理として，475,200 点の正常な SWaT データを訓練／検証／テストデータとして 8:1:1 に分割し，部分時系列の長さが $T = 12$ のデータセットを生成する．その結果，訓練／検証／テストデータは，それぞれ 31,680 部分時系列，3,960 部分時系列，3,960 部分時系列となった．訓練データは，モデルを訓練するために使われる．検証データは，各 epoch の学習後の評価に使われ，最終的に F 値が最も高い epoch のモデルを選択するために使われる．テストデータは，検証データを用いて選択された epoch のモデルに対して，異常検知の精度を評価するために使われる．これらのデータのうち，各異常検知のモデルは，正常なデータのみを用いて学習されるため，訓練データはこのまま利用できる．一方，検証・テストデータは評価に使われるため，集団型異常を含む必要がある．

集団型異常とは，観測値自体は正常だが，その観測値のふるまいが変化したタイプの異常である．したがって，正常な観測値と他の正常な観測値を入れ替えることで対応した．具体的な集団型異常の生成方法を下記に示す．生成方法の概要を図 7 に示す．

（1）　各データ (検証／テスト) の中で，20%の部分時系列をランダムに選択

（2）　選択した部分時系列の中で，観測値同士をランダムに入れ替え

（3）　選択した 20%の部分時系列を異常と定義

(2) で選択された 20%の各部分時系列に含まれる T 点の観測値を，他の選択された部分時系列の観測値とランダムに入れ替えることで，観測値自体は正常な値であるが，その観測値のふるまいに異常が存在する集団型異常を生成することが可能となる．上記で選択された 20%を異常と定義し，MARU-GAN の異常検知の結果と比較する．

6.3 比較手法

MARU-GAN の比較手法として，Efficient GAN と，集団型異常検知に向けた手法である EncDec-AD，LSTM-AD を採用した．各異常検知の手法について説明する．これらの比較手法は，観測値 $\boldsymbol{x}^{(i)}$ ごとに異常判定を行うため，6.2 節で選択された 20%の部分時系列に含まれる T 点の観測値全てを異常と定義し，観測値ごとに評価を実施した．本稿では，$T = 12$ としたため，検証・テストデータにおける異常と定義された観測値は，3,960 部分時系列 ×20%×12 点=9,504 点である．

6.3.1 Efficient GAN

4.1 節で説明した Efficient GAN を用いる．Efficient GAN は，観測値 $\boldsymbol{x}^{(i)}$ ごとに異常度を算出するため，異常度が高い上位 20%の観測値を異常と判定する．

6.3.2 EncDec-AD

EncDec-AD は Encoder-Decoder モデルを用いた異常検知の手法である．まず，正常なデータのみから成る訓練データの一部を用いて，Encoder で入力部分時系列 $\boldsymbol{X}$ を低次元のベクトルに圧縮し，Decoder で圧縮されたベクトルから時系列 $\boldsymbol{X'}$ を復元する Encoder-Decoder モデルを学習する．EncDec-AD は，入力部分時系列 $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T)}\}$ と Encoder-Decoder モデルによって復元される部分時系列 $\boldsymbol{X'} = \{\boldsymbol{x}^{(1)'}, \boldsymbol{x}^{(2)'}, \cdots, \boldsymbol{x}^{(T)'}\}$ に対して，目的関数 $\sum_{i=1}^{T} \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i)'}\|^2$ を最小化するように学習される．そして学習で使用されなかった残りの訓練データを用いて，学習された Encoder-Decoder モデルで復元された時系列 $\boldsymbol{X}^{(i)'}$ と，入力部分時系列 $\boldsymbol{X}^{(i)}$ における各観測値 $\boldsymbol{x}^{(i)}$ のエラーベクトル $\boldsymbol{e}^{(i)} = |\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i)'}|$ を算出し，$\boldsymbol{e}^{(i)}$ が従う正規分布 $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ の平均 $\boldsymbol{\mu}$ と標準偏差 $\boldsymbol{\Sigma}$ を求める．このパラメータを用い，テストデータの各観測値 $\boldsymbol{x}^{(i)}$ に対して，異常度 $A^{(i)} = (\boldsymbol{e}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{e}^{(i)} - \boldsymbol{\mu})$ を算出し，この値が高い上位 20%の観測値を異常と判定する．

6.3.3 LSTM-AD

LSTM-AD は，LSTM を用いた異常検知の手法である．まず，正常なデータのみから成る訓練データの一部を用いて，d 点から l 点を予測する LSTM を学習する．LSTM-AD は，LSTM によって予測される l 点 $\{\boldsymbol{x}^{(1)'}, \boldsymbol{x}^{(2)'}, \ldots, \boldsymbol{x}^{(l)'}\}$ と実データ $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(l)}\}$ に対して，目的関数 $\sum_{i=1}^{l} \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i)'}\|^2$ を最小化するように学習される．そして，学習で使用されなかった残りの訓練データを用いて，学習された LSTM から予測された各観測値 $\boldsymbol{x}^{(i)'}$ と，実際の $\boldsymbol{x}^{(i)}$ のエラーベクトル $\boldsymbol{e}^{(i)} = |\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i)'}|$ を算出し，$\boldsymbol{e}^{(i)}$ が従う正規分布 $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ の平均 $\boldsymbol{\mu}$ と標準偏差 $\boldsymbol{\Sigma}$ を求める．このパラメータを用い，テストデータの各観測値 $\boldsymbol{x}^{(i)}$ に対して，異常度 $A^{(i)} = (\boldsymbol{e}^{(i)} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{e}^{(i)} - \boldsymbol{\mu})$ を算出し，この値が高い上位 20%の観測値を異常と判定する．

表 1　MARU-GAN のハイパーパラメータ

	ユニット数	層数	ドロップアウト率
Encoder			
$E(\boldsymbol{X})$: RNN	100	3	0.0
Generator			
$G(\boldsymbol{Z})$: RNN	100	3	0.0
Discriminator			
$D(\boldsymbol{X})$: RNN	100	1	0.2
$D(\boldsymbol{X}, \boldsymbol{Z})$: 全結合 NN	1	1	0.0

表 2　実験設定

データセット	SWaT データセット
勾配法	Adam
ハイパーパラメータ	$\alpha = 1e-5, \beta_1 = 0.5, \beta_2 = 0.999, \varepsilon = 1e-8$
時系列の長さ T	12
バッチサイズ	50
Epoch 数	1000
潜在変数の次元	100

表 3　実験結果

手法	F 値	Accuracy	False positive 率
Efficient GAN [4]	0.20	0.68	0.20
EncDec-AD [19]	0.19	0.68	0.20
LSTM-AD [18]	0.50	0.80	0.12
MARU-GAN	**0.72**	**0.89**	**0.07**

6.4 実験設定

本稿の実験で用いたハイパーパラメータについて説明する．表 1 に MARU-GAN の Encoder，Generator，Discriminator の詳細，表 2 に実験設定を示す．比較のため，各表のハイパーパラメータは，先行研究に基づいて調節した．1000epoch の各 epoch の学習後に検証データを用いて F 値を算出し，F 値が最も高い epoch のモデルを使ってテストデータで評価を実施した．6.2 節で異常と定義された部分時系列と，MARU-GAN によって算出される異常度が高い 20%の部分時系列を比較し，F 値，Accuracy，False positive 率を求める．

6.5 実験結果と考察

実験結果を表 3 に示す．MARU-GAN は全ての評価値において，他の既存手法よりも高い精度を達成することができた．

Efficient GAN のような特定時点の観測値を扱うモデルでは，観測値自体は正常であるが，その観測値のふるまいが変化した集団型異常を検知することが不可能であった．Efficient GAN では，F 値が実験データに占める異常の割合 20%とほぼ変わらず，つまり，異常をランダムに選択してしまっているのと区別がつかない．従って，集団型異常を検知するためには，複数の観測値を扱うためのネットワークを利用する必要があることが明らかになった．

EncDec-AD と LSTM-AD は，複数の観測値を扱うモデルであるにも関わらず，集団型異常を検知することができなかった．

EncDec-AD は，異常データの異常度が正常データの異常度と変わらないため，異常を検知することができなかった．

EncDec-AD は，正常データのみから成る訓練データを用いて，入力部分時系列 $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T)}\}$ と Encoder-Decoder モデルによって復元される部分時系列 $\boldsymbol{X'} = \{\boldsymbol{x}^{(1)'}, \boldsymbol{x}^{(2)'}, \cdots, \boldsymbol{x}^{(T)'}\}$ に対して，目的関数 $\sum_{i=1}^{T} \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i)'}\|^2$ を最小化するように学習される．目的関数を最適化すると，正常データのみを使って Encoder-Decoder モデルを学習しているにも関わらず，訓練データ (正常データ) に存在しない異常データに対しても，学習した Encoder-Decoder モデルを使ってほぼ完全に復元することが可能であった．Encoder-Decoder モデルは，Encoder 側で入力時系列の特徴を圧縮したベクトルを使って Decoder 側で復元を行っている．よって，未来が明らかな状態で復元が行われているため，異常なデータに対してもほぼ完全に復元できてしまうと考えられる．そのため，異常度を算出する際に用いるエラーベクトル $\boldsymbol{e}^{(i)} = |\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i)'}|$ の値が，正常データであっても，異常データであっても，変わらないため，異常を検知することができなかった．一方，MARU-GAN の異常度 $A_{RNN}(\boldsymbol{X}) = \alpha L_{G_{RNN}}(\boldsymbol{X}) + (1-\alpha) L_{D_{RNN}}(\boldsymbol{X})$ においても，EncDec-AD と同様に，再構築誤差 $L_{G_{RNN}}(\boldsymbol{X}) = \|\boldsymbol{X} - G_{RNN}(E_{RNN}(\boldsymbol{X}))\|_1$ を用いて異常の検出を行っている．しかし，MARU-GAN の目的関数 (式 (2)) は EncDec-AD とは異なっており，正常な部分時系列を生成するように学習される．つまり，完全な復元を目的にしていない．学習後，異常な部分時系列も正常な部分時系列と判定されるように再構築されるが，MARU-GAN の場合，それを避けることができる．その結果が再構築誤差 $L_{G_{RNN}}$ で得られるため，MARU-GAN は集団型異常を検知することが可能となる．

LSTM-AD は，異常な部分時系列の前半の観測値を異常と識別することができなかった．これは，LSTM に入力される観測値が少ない前半である程，予測のために必要な手がかりが少ないため，上手く予測ができなかったことが原因と考えられる．また，複数の観測値 d 点間に異常データのようなランダム性が存在する場合，その後の観測値 l 点を上手く予測することができなかった．その結果，頻繁に l 点が異常だと判定されてしまった．

以上より，集団型異常を検知するためには，複数の観測値を扱うネットワークを利用する必要があること，提案モデルである MARU-GAN は，複数の観測値を扱うネットワークを採用した既存手法と比較して，高い精度で集団型異常を検知できることが明らかになった．

7 おわりに

本稿では，特定時点の観測値を扱う GAN モデルのネットワークを複数の観測値を扱えるネットワークに拡張することによって，時系列データに潜む集団型異常を検知可能な GAN モデルを提案した．提案モデルに対して，時系列データの一部の正常な観測値を他の正常な観測値と入れ替えて生成したデータセットを用いて評価を行った．その結果，集団型異常を検知するためには，複数の観測値を扱うネットワークを利用する必要があること，我々の GAN モデルは，複数の観測値を扱うネットワークを採用した既存手法と比較して，高い精度で集団型異常を検知できることが明らかになった．

文　献

[1] 山西健司, データマイニングによる異常検知, 共立出版, 2009.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. ”Generative adversarial nets,” In Advances in Neural Information Processing Systems, pp. 2672–2680, 2014.

[3] T. Schlegl, P. Seebock, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, ”Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” In International Conference on Information Processing in Medical Imaging, pp. 146–157, 2017.

[4] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, ”Efficient GAN-based anomaly detection,” arXiv:1802.06222, 2018.

[5] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S. K. Ng, ”MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks,” arXiv:1901.04997, 2019.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, ”Sequence to sequence learning with neural networks,” In Advances in Neural Information Processing Systems, pp. 3104–3112, 2014.

[7] A.P. Mathur, and N. O. Tippenhauer, ”SWaT: A water treatment testbed for research and training on ICS security,” In Cyber-physical Systems for Smart Water Networks, pp. 31–36, 2016.

[8] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, ”WADI: A water distribution testbed for research in the design of secure cyber physical systems,” In Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, pp. 25–28, 2017.

[9] F. E. Grubbs, ”Procedures for detecting outlying observations in samples,” Technometrics, Vol. 11, No. 1, pp. 1–21, 1969.

[10] J. Laurikkala, M. Juhola, E. Kentala, N. Lavrac, S. Miksch, and B. Kavsek, ”Informal identification of outliers in medical data,” In Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology, Vol. 1, pp. 20–24, 2000.

[11] E. Eskin, ”Anomaly detection over noisy data using learned probability distributions,” In Proceedings of the International Conference on Machine Learning, 2000.

[12] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, ”A density-based algorithm for discovering clusters in large spatial databases with noise,” In Proceedings of Second International Conference on Knowledge Discovery and Data Mining, Vol. 96, No. 34, pp. 226–231, 1996.

[13] R. Smith, A. Bivens, M. Embrechts, C. Palagiri, and B. Szymanski, ”Clustering approaches for anomaly based intrusion detection,” In Proceedings of Intelligent Engineering Systems through Artificial Neural Networks, pp. 579–584, 2002.

[14] S. Ramaswamy, R. Rastogi, and K. Shim, ”Efficient algorithms for mining outliers from large data sets,” In ACM Sigmod Record, Vol. 29, No. 2, pp. 427–438, 2000.

[15] E. M. Knorr, R. T. Ng, and V. Tucakov, ”Distance-based outliers: algorithms and applications,” the VLDB Journal—the International Journal on Very Large Data Bases, Vol. 8, No. 3–4, pp. 237–253, 2000.

[16] O. Taylor, and D. Addison, ”Novelty detection using neural network technology,” In Proceedings of International Congress on Condition Monitoring and Diagnostic Engineering Management, pp. 731–743, 2000.

[17] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, ”Estimating the support of a high-dimensional distribution,” Neural Computation, Vol. 13, No. 7, pp. 1443–1471, 2001.

[18] P. Malhotra, L. Vig, G, Shroff, and P. Agarwal, ”Long short term memory networks for anomaly detection in time series,” European Symposium on Artificial Neural Networks, pp. 89–94, 2015.

[19] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, ”LSTM-based encoder-decoder for multi-sensor anomaly detection,” arXiv:1607.00148, 2016.

[20] J. Donahue, P. Krahenbuhl, and T. Darrell, ”Adversarial feature learning,” arXiv:1605.09782, 2016.

[21] GAN Zoo, https://github.com/hindupuravinash/the-gan-zoo

[22] V. Chandola, A. Banerjee, and V. Kumar, ”Anomaly detection: A survey,” ACM Computing Surveys, Vol. 41, No. 3, 2007.