一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
CPSY2018-114,DC2018-96（2019-03）

複数のオンライン教師無し異常検知コアを用いた精度改善手法

塚田　峰登†　近藤　正章††　松谷　宏紀†

† 慶應義塾大学大学院 理工学研究科　〒 223-8522 神奈川県横浜市港北区日吉 3-14-1
†† 東京大学大学院 情報理工学系研究科　〒 113-8656 東京都文京区本郷 7-3-1
E-mail: †{tsukada,matutani}@arc.ics.keio.ac.jp, ††kondo@hal.ipc.i.u-tokyo.ac.jp

あらまし　教師無し異常検知においては如何に正常なデータの分布を正確に学習・モデリングできるかが重要になる。この時、実世界の異常検知においては正常データの分布が時間とともに変化する場合があるため、逐次的に変化に追従できる教師無し「オンライン」異常検知手法が解決策の一つとして挙げられる。関連する近年の研究の一つとして、逐次学習アルゴリズムの OS-ELM と、ニューラルネットワーク技術の一つであるオートエンコーダを組み合わせた、FPGA ベースの教師無しオンライン異常検知器が提案されている。OS-ELM は小さな計算量で逐次学習可能なニューラルネットワークであり、これを基にオートエンコーダを構築することで、エッジデバイスへの実装可能性を保ちつつ、オンラインで学習可能な教師無し異常検知器を構成している。しかしながら、OS-ELM そのものは三層に限定されたニューラルネットワークであるため、学習する正常データの多様性（パターン数、分散等）が大きい場合、十分な精度が得られないという問題点がある。そこで、本論文では複数のオートエンコーダインスタンスを用い、インスタンス一つあたりの正常データの多様性を小さくすることで、異常検知精度を改善する手法を提案する。さらに、本論文では複数のインスタンスから算出される異常度スコアを活用することで確信度スコアを導入し、これが異常検知精度の改善に寄与することを示す。

キーワード　FPGA, 機械学習, 異常検知, エッジコンピューティング

A Method for Improving Accuracy using Multiple Online Unsupervised Anomaly Detection Cores

Mineto TSUKADA†, Masaaki KONDO††, and Hiroki MATSUTANI†

† Graduate School of Science and Technology, Keio University 3-14-1, Hiyoshi,Yokohama, JAPAN 223-8522
†† Graduate School of Information Science and Technology, The University of Tokyo 7-3-1, Hongo, Tokyo, JAPAN 113-8656
E-mail: †{tsukada,matutani}@arc.ics.keio.ac.jp, ††kondo@hal.ipc.i.u-tokyo.ac.jp

1. はじめに

教師無し異常検知は、異常データを必要としない異常検知手法である。多くの場合、正常データの分布を学習し、そこから離れたデータを異常データとみなす。そのため、正常データの分布を如何に正確に捉えられるかが重要になるが、実世界の異常検知においては正常データの分布が時系列で変化する場合があり、容易な問題ではない。

上記の問題意識から、近年では教師無し「オンライン」異常検知手法が解決策の一つとして挙げられている。オンライン異常検知では入力されるデータを逐次的に学習するため、正常データの時系列変化に追従できる。また、これを FPGA や ASIC などを用いてエッジデバイスとして実装することで、工場や病院等様々な環境や設備の異常検知が可能になる。

上記に関する近年の研究の一例として、逐次学習アルゴリズムの OS-ELM [1] とニューラルネットワーク技術の一つであるオートエンコーダ [2] を組み合わせた FPGA ベースの教師無しオンライン異常検知器（OSUAD）[3] が提案されている。OS-ELM は誤差逆伝播法のニューラルネットワークに比べて小さな計算量で逐次学習できるアルゴリズムであり、これを基にオートエンコーダを構築することで、リソースの限られたエッジデバイスへの実装可能性を保ちつつ、逐次学習可能な教師無し異常検知器を構成している。しかしながら、OS-ELM そのものは三層に限定されたニューラルネットワークであり、しばし

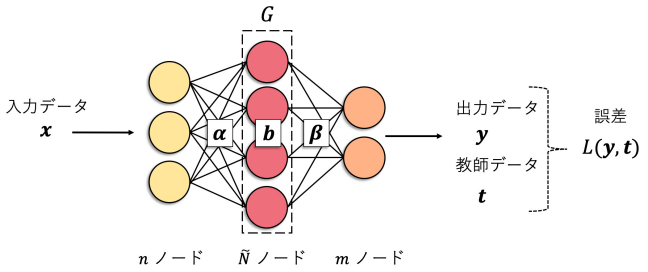

図 1　ELM（Extreme Learning Machine）

ば精度の点で問題が生じる。教師無し異常検知の文脈では、学習する正常データの多様性（パターン数、分散等）が高まるにつれて正常データと異常データの識別が困難になる。異常検知においては多くの場合精度が重要になるため、上記の問題点を解決する必要がある。

そこで、本論文では複数のオートエンコーダインスタンスを用いた精度改善手法を提案する。学習する正常データに対しクラスタリングを行い、各クラスタ毎に一つのインスタンスを割り当てることで、それぞれが学習する正常データの多様性を小さくする。これにより正常データと異常データの識別が容易になり、結果として異常検知精度を改善できる。さらに、各インスタンスから算出される異常度スコアを用いて新たに確信度スコアを導入することで、確信度の低いデータは人間に異常検知の判定を委ねるといった協調手法を可能にする。

本論文の構成は次に示す通りである。はじめに 2. 章で前提となる知識について述べ、3. 章で関連研究について述べる。また、4. 章で提案手法を説明し、5. 章で提案手法の評価を行う。最後に、6. 章で本論文をまとめる。

2. 前提知識

本章では前提知識について述べる。

2.1 ELM と OS-ELM

ELM（Extreme Learning Machine）[4] はニューラルネットワークの派生モデルの一つである。図 1 に示すように、ELM は入力層、隠れ層、出力層の三つの層で構成されるニューラルネットワークであり、バッチサイズ k の n 次元の入力データ $\boldsymbol{x} \in \boldsymbol{R}^{k\times n}$ に対応する m 次元の推論結果 $\boldsymbol{y} \in \boldsymbol{R}^{k\times m}$ は、$\boldsymbol{y} = G(\boldsymbol{x}\cdot\boldsymbol{\alpha}+\boldsymbol{b})\boldsymbol{\beta}$ として得られる。この時、$\boldsymbol{\alpha} \in \boldsymbol{R}^{n\times\tilde{N}}$ は入力層と隠れ層を結合する重みであり、$\boldsymbol{\beta} \in \boldsymbol{R}^{\tilde{N}\times m}$ は隠れ層と出力層を結合する重みを示す。また、$\boldsymbol{b} \in \boldsymbol{R}^{\tilde{N}}$ と G はそれぞれ隠れ層のバイアスと活性化関数を示す。

ELM の学習手法は以下の通りである。まず、重み $\boldsymbol{\alpha}$ を任意の乱数で初期化する。次に、学習データ $\{\boldsymbol{x} \in \boldsymbol{R}^{k\times n}, \boldsymbol{t} \in \boldsymbol{R}^{k\times m}\}$ を用いて、入力データ $\boldsymbol{x}$ の推論結果 $\boldsymbol{y}$ と教師データ $\boldsymbol{t}$ の誤差が 0 であると仮定すると次の等式が成立する。

$$G(\boldsymbol{x}\cdot\boldsymbol{\alpha}+\boldsymbol{b})\boldsymbol{\beta} = \boldsymbol{t} \tag{1}$$

上式を満たす最適解 $\hat{\boldsymbol{\beta}}$ は隠れ層行列 $\boldsymbol{H} \equiv G(\boldsymbol{x}\cdot\boldsymbol{\alpha}+\boldsymbol{b}) \in \boldsymbol{R}^{k\times\tilde{N}}$ を用いて次の式を計算することで求められる。

$$\hat{\boldsymbol{\beta}} = \boldsymbol{H}^{\dagger}\boldsymbol{t} \tag{2}$$

$\boldsymbol{H}^{\dagger}$ は $\boldsymbol{H}$ の擬似逆行列であり、SVD（Singular Value Decomposition）や QRD（QR Decomposition）等の行列分解アルゴリズムを用いて計算できる。最後に、$\boldsymbol{\beta}$ を $\hat{\boldsymbol{\beta}}$ で更新することで学習が完了する。

現在主流の誤差逆伝播法を用いたモデルは $\boldsymbol{\alpha}$ と $\boldsymbol{\beta}$ の両方を最適化する必要があるが、ELM は $\boldsymbol{\beta}$ のみでよい。また、ELM は同じ学習データに対して繰り返し学習する必要がなく一度で完了するため、高速に学習可能なニューラルネットワークとして知られる [4]。しかし、ELM は予め全ての学習データが用意されている場合を想定しており、逐次的に学習データが生成される場合はその都度過去の全学習データを含めて再学習する必要がある。

一方で、OS-ELM（Online Sequential Extreme Learning Machine）[1] は、ELM を逐次的に学習できるように拡張したアルゴリズムである。バッチサイズ k_i の i 番目の学習データ $\{\boldsymbol{x}_i \in \boldsymbol{R}^{k_i\times n}, \boldsymbol{t}_i \in \boldsymbol{R}^{k_i\times m}\}$ が与えられたとすると、以下の誤差を最小化する $\boldsymbol{\beta}_i$ を求める必要がある。

$$\left\| \begin{bmatrix} \boldsymbol{H}_0 \\ \vdots \\ \boldsymbol{H}_i \end{bmatrix} \boldsymbol{\beta}_i - \begin{bmatrix} \boldsymbol{t}_0 \\ \vdots \\ \boldsymbol{t}_i \end{bmatrix} \right\| \tag{3}$$

この時、$\boldsymbol{H}_i \equiv G(\boldsymbol{x}_i\cdot\boldsymbol{\alpha}+\boldsymbol{b})$ である。また、i 番目の最適化された重み $\boldsymbol{\beta}_i$ は以下の式で得られる。

$$\begin{aligned} \boldsymbol{P}_i &= \boldsymbol{P}_{i-1} - \boldsymbol{P}_{i-1}\boldsymbol{H}_i^T(\boldsymbol{I}+\boldsymbol{H}_i\boldsymbol{P}_{i-1}\boldsymbol{H}_i^T)^{-1}\boldsymbol{H}_i\boldsymbol{P}_{i-1} \\ \boldsymbol{\beta}_i &= \boldsymbol{\beta}_{i-1} + \boldsymbol{P}_i\boldsymbol{H}_i^T(\boldsymbol{t}_i - \boldsymbol{H}_i\boldsymbol{\beta}_{i-1}) \end{aligned} \tag{4}$$

特に、$\boldsymbol{P}_0$ と $\boldsymbol{\beta}_0$ は以下の式で得られる。

$$\boldsymbol{P}_0 = (\boldsymbol{H}_0\boldsymbol{H}_0^T)^{-1}, \boldsymbol{\beta}_0 = \boldsymbol{P}_0\boldsymbol{H}_0^T\boldsymbol{t}_0 \tag{5}$$

OS-ELM は過去の学習データを記憶する必要が無く、新しく与えられた学習データに対してのみ学習を行えば良い。誤差逆伝播法を用いたニューラルネットワークと比較して、小さな計算量で高速に逐次学習できることが知られている [1]。

2.2 OSUAD

OSUAD は OS-ELM とオートエンコーダを組み合わせた、FPGA ベースの教師無し異常検知器である [3]。オートエンコーダ [2] はニューラルネットワークを用いた次元圧縮アルゴリズムの一つであり、入力データを教師データとして流用し、推論結果として入力データをできるだけ正確に再構成するように学習する。この時、隠れ層のノード数を入力層と出力層のそれよりも小さく設定することで、学習誤差が収束した時に、隠れ層行列を入力データの次元圧縮形式とみなせる。これを異常検知に適用するには、オートエンコーダを与えられた正常データのみで学習させる。すると、正常データの分布から離れたデータ（異常データ）に対しては再構成に失敗し、相対的に推論誤差が大きくなる。よって、そこに閾値を設けることで、入力データが正常か異常かを検出できる。上記の手法は学習時に異常データを必要としないことから、教師無し異常検知アルゴリズムに区分される。

OSUAD の学習アルゴリズムは、式 4 において $\boldsymbol{t}_i = \boldsymbol{x}_i$ とし

たものである。

$$\boldsymbol{P}_i = \boldsymbol{P}_{i-1} - \frac{\boldsymbol{P}_{i-1}\boldsymbol{h}_i^T\boldsymbol{h}_i\boldsymbol{P}_{i-1}}{1+\boldsymbol{h}_i\boldsymbol{P}_{i-1}\boldsymbol{h}_i^T}$$
$$\boldsymbol{\beta}_i = \boldsymbol{\beta}_{i-1} + \boldsymbol{P}_i\boldsymbol{h}_i^T(\boldsymbol{x}_i - \boldsymbol{h}_i\boldsymbol{\beta}_{i-1}) \quad (6)$$

$\boldsymbol{h} \in \boldsymbol{R}^{1\times\tilde{N}}$ は $k=1$ の時の隠れ層行列 $\boldsymbol{H}$ である。OSUAD では FPGA リソース量と学習時の計算量を削減するためにバッチサイズを 1 としている。また、ある入力 $\boldsymbol{x}$ に対する損失値は、損失関数 L を用いて以下のように計算される。

$$loss = L(\boldsymbol{x}, G(\boldsymbol{x}\cdot\boldsymbol{\alpha}+\boldsymbol{b})\boldsymbol{\beta}) \quad (7)$$

上式で計算される損失値が閾値を上回れば、$\boldsymbol{x}$ を異常データとみなせる。

3. 関 連 研 究

本章では本研究の関連研究について述べる。

3.1 オートエンコーダを用いた異常検知の精度改善手法

オートエンコーダを用いた異常検知手法にはいくつかの精度改善手法が提案されている。Jinwon らは通常のオートエンコーダを変分オートエンコーダ [5] に置き換えることで説明性と精度が向上することを示した [6]。変分オートエンコーダは隠れ層行列の値に任意の分布（主に正規分布）を仮定するモデルであり、異常検知の際に損失値だけでなく、入力データがどの程度正常データの分布から離れているかを推測できる。しかしながら変分オートエンコーダは誤差逆伝播法を用いたニューラルネットワークを対象としており、OS-ELM にはそのまま適用できない。確認できる限り、OS-ELM ベースのオートエンコーダを用いた異常検知の精度改善手法は前例がなく、また誤差逆伝搬法を用いたニューラルネットワークベースのオートエンコーダにおいても、複数インスタンス化やそれを利用した確信度の導入は行われていない。

4. 提　　案

本章では本研究の提案手法について述べる。

4.1 複数インスタンス化による異常検知精度の改善

2.2 節で述べたように、OSUAD は OS-ELM ベースのオートエンコーダを用いて正常データを学習する。これにより、正常データの分布から離れたデータ（異常データ）が入力された際に高い損失値が出力される。しかしながら、OS-ELM そのものは三層に限定されたニューラルネットワークであり、正常データの多様性（パターン数、分散等）が大きくなるほど、正常データと異常データの識別が困難になる。そこで、本節では複数のオートエンコーダインスタンスを用い、単一のインスタンスが学習する正常データの多様性を小さくすることで異常検知精度を改善する手法を提案する。提案手法は**初期化フェーズ**と**オンライン推論フェーズ**の二つのフェーズで構成され、それぞれについて順に述べる。

始めに、提案手法の初期化フェーズ（図 2）について述べる。このフェーズでは、主にモデルの初期化を行う。予め用意した正常データを $\boldsymbol{X} \in \boldsymbol{R}^{N\times n}$ とし、N 個の n 次元データが含まれるとする。これを学習用に用いる $\boldsymbol{X}_{train} \in \boldsymbol{R}^{N_{train}\times n}$ と後述

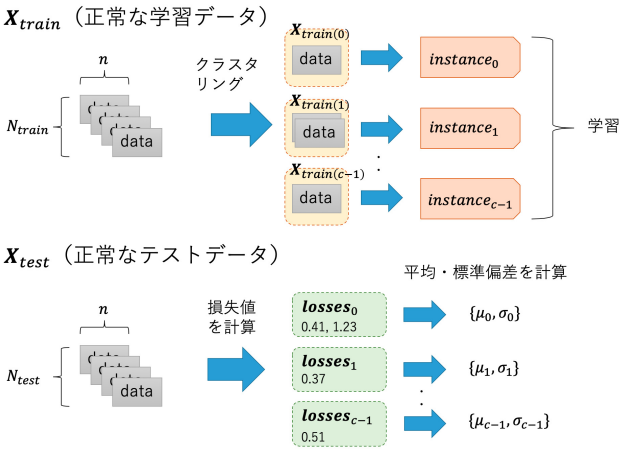

図 2　初期化フェーズ

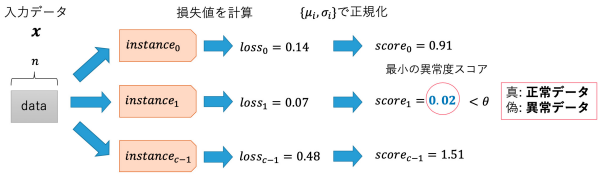

図 3　オンライン推論フェーズ

のパラメタの計算時に用いるテストデータ $\boldsymbol{X}_{test} \in \boldsymbol{R}^{N_{test}\times n}$（$N = N_{train} + N_{test}$）に分割する。

図 2 の上部に $\boldsymbol{X}_{train}$ の処理手順を示す。まず、$\boldsymbol{X}_{train}$ に含まれる N_{train} 個のデータを c 個のクラスタに分類する。この際には K-Means 等のクラスタリングアルゴリズムを用いる。次に、i（$0 \leqq i \leqq c-1$）番目のクラスタ $\boldsymbol{X}_{train(i)}$ を用い、i 番目のオートエンコーダインスタンス（単一の OSUAD に相当）を学習する。このように、正常データをクラスタリングすることで同じ正常パターンを持つデータを同じクラスタに分類できる。そして各クラスタに対し一つのインスタンスを割り当てることで、学習する正常データの多様性が小さくなり、異常データの識別が容易になる。

図 2 の下部に $\boldsymbol{X}_{test}$ の処理手順を示す。$\boldsymbol{X}_{test}$ は、各インスタンスの損失値を正規化するためのパラメタを得るために用いる。まず、各テストデータを全てのインスタンスに入力し、対応する損失値を計算する。この時、あるテストデータに対し i 番目のインスタンスが最小の損失値を出力したとすると、対応する損失値を集合 $\boldsymbol{losses}_i$ の要素として加える。全てのテストデータに対し上記の処理を行ったら、それぞれの $\boldsymbol{losses}_i$ の平均値 μ_i と標準偏差 σ_i を求め、i 番目のインスタンスの正規化パラメタとする。以上で初期化フェーズは完了する。

次に、提案手法のオンライン推論フェーズ（図 3）について述べる。このフェーズでは、入力データに対しオンラインで異常検知を行い、正常データか異常データかの判定を行う。まず n 次元の入力データ $\boldsymbol{x} \in \boldsymbol{R}^n$ が与えられたとすると、全 c 個のインスタンスの損失値を計算する。i 番目のインスタンスの損失値を $loss_i$ とすると、初期化フェーズ時に得られたパラメタ

Algorithm 1 Compute Confidence Score

for $i = 0$ to $c-1$ **do**
  $score_i \leftarrow \frac{loss_i - \mu_i}{\sigma_i}$
end for
$is\_abnormal \leftarrow false$
if $min\{score_i, i = 0, 1, \ldots, c-1\} > \theta$ **then**
  $is\_abnormal \leftarrow true$
end if
$max\_score \leftarrow max\{score_i, i = 0, 1, \ldots, c-1\}$
for $i = 0$ to $c-1$ **do**
  $score\_inv_i \leftarrow max\_score - score_i$
end for
for $i = 0$ to $c-1$ **do**
  $prob_i \leftarrow \frac{\exp(score\_inv_i)}{\sum_{j=0}^{c-1} \exp(score\_inv_j)}$
end for
$entropy = -\frac{\sum_{i=0}^{c-1} prob_i \log_2(prob_i)}{\log_2(c)}$
if $is\_abnormal == false$ **then**
  $conf \leftarrow 1 - entropy$
else
  $conf \leftarrow entropy$
end if

$\{\mu_i, \sigma_i\}$ を用いて、以下の正規化処理を行う。

$$score_i = \frac{loss_i - \mu_i}{\sigma_i} \tag{8}$$

各インスタンスの損失値 $loss_i$ を正規化した値 $score_i$ を**異常度スコア（abnormality score）**として定義する。

さて、各インスタンスは正常データのクラスタを用いて学習されている。つまり、全てのインスタンスの異常度スコアが、ある閾値 θ よりも高ければ、対応する入力データは、学習に使用したいずれの正常データのクラスタにも類似しない異常なデータであるといえる。よって、以下の不等式が真である場合に、入力データは異常データであると推測できる。

$$min\{score_i, i = 0, 1, \ldots, c-1\} > \theta \tag{9}$$

逆に式 9 が偽である場合、入力データは正常データであると推測できる。この入力データを学習データとして使用したい場合、どのインスタンスで学習を行うかを決定する必要があるが、このデータに類似するデータで学習されたインスタンス、つまり、最小の異常度スコアを算出したインスタンスで学習するのが妥当であろう。以上がオンライン推論フェーズの処理手順である。

4.2 確信度スコアの導入

OSUAD は入力データが正常か異常かを自動で識別できる。しかし、中には識別が困難なデータがあり、その際には誤判定が発生する可能性が高まる。そこで、本節では、複数のオートエンコーダインスタンスから出力された異常度スコアを活用し、モデルの推論結果にどの程度信頼性があるかを示す**確信度スコア（confidence score）**を提案する。このスコアを導入することにより、例えばあるデータに対する推論結果の確信度が小さい場合、そのデータの判定を人に任せることで、誤判定を防ぐことができる。

アルゴリズム 1 に確信度スコアを計算するための具体的な処理手順を示す。まず、ある入力 $\boldsymbol{x}$ が与えられたとすると、各インスタンスの損失値 $loss_i$ を計算し、式 8 を用いて異常度スコア $score_i$ を計算する。次に、式 9 を用いて $\boldsymbol{x}$ が正常か異常かの判定を行い、結果を $is\_abnormal$ に格納する。ここまでは、4.1 節で述べたオンライン推論フェーズの処理と同一である。続いて、異常度スコアの最大値（max_score）を用いて $score\_inv_i \leftarrow max\_score - score_i$ とし、それぞれの異常度スコアの大小関係を逆転させる。これにより、例えば $\boldsymbol{x}$ が i 番目のインスタンスの正常パターンに類似する時、$score_i$ は最小値をとると期待できるが、$score\_inv_i$ は逆に最大値をとると期待できる。また、これらの値にソフトマックス関数を適用することで、$prob_i$ を $\boldsymbol{x}$ が i 番目のインスタンスの正常パターンに属する確率とみなせる。

確信度スコアは、上記で得られた確率 $prob_i$ の平均情報量を基に計算する。平均情報量はある事象系の情報量示す指標であり、この値が大きいほど情報量が大きいとみなせる。具体的には、平均情報量 $-\sum_{i=0}^{c-1} prob_i \log_2(prob_i)$ を最大値 $\log_2(c)$ で割り、[0,1] に規格化したものを $entropy$ とする。さらに、モデルが正常（$is\_abnormal$ が $false$）、または異常（$is\_abnormal$ が $true$）と判断した場合に、確信度スコア（$conf$）をそれぞれ、$1 - entropy$、$entropy$ とする。モデルが正常と判定した場合、仮に $\boldsymbol{x}$ が i 番目のインスタンスの正常パターンに非常に近いとすると、$prob_i$ は際立って高い値になると期待される。この時、平均情報量は小さくなり、$entropy$ が小さいほど判定の確信度が高いとみなせる。よって、この場合は $1 - entropy$ を確信度スコアとして利用できる。一方、モデルが異常と判定した場合、仮に本当に $\boldsymbol{x}$ が異常データであるとすると、どの正常パターンにも属さないことになる。この場合、特定の確率 $prob_i$ に大きな値が出力されるよりも、全体的に確率値が分散されることが期待される。この時、平均情報量は大きくなり、$entropy$ が大きいほど判定の確信度が高いとみなせる。よって、この場合は $entropy$ を確信度スコアとして利用できる。

5. 評　　価

本章では、4.1 節で提案した複数インスタンス化による異常検知精度の改善手法と、4.2 節で提案した確信度スコアの有効性について評価を行う。全評価に共通し、表 1 のマシンを評価環境とする。

5.1 比 較 対 象

本評価では次の三種類の実装: ①**OSUAD-Sim**、②**OSUAD-Multi**、③**OSUAD-Conf** を用いる。OSUAD-Sim は OSUAD をソフトウェアで実装したものであり、OSUAD-Multi と OSUAD-Conf は、それぞれ OSUAD-Sim に 4.1 節の複数インスタンス化と 4.2 節の確信度スコアを適用したも

表 1　評 価 環 境

OS	Ubuntu 16.04 LTS
CPU	Intel Core i5-3470S 2.9GHz
GPU	NVIDIA GTX 1080Ti 12GB
DRAM	DDR4 RAM 16GB

図 4　MNIST のサンプル画像

図 5　Fashion-MNIST のサンプル画像

のである。確信度スコアの計算は複数インスタンス化が前提であり、また、複数インスタンス化は OSUAD-Sim が前提であるため、OSUAD-Conf は OSUAD-Multi を、OSUAD-Multi は OSUAD-Sim を内包する。これらは Python の科学計算ライブラリの一つである Numpy（ver. 1.16.0）を用いて実装しており、計算精度は単精度浮動小数点とする。OSUAD-Sim は OSUAD と全く同一な学習・推論アルゴリズムを実装しているが、OSUAD は単精度浮動小数点ではなく、整数部 22bit、小数部 10bit の固定小数点を用いているため、異常検知精度における両者の差は計算精度のみである。各実装の設定としては、OSUAD-Conf と OSUAD-Multi 内のオートエンコーダインスタンスは全て OSUAD-Sim と同じものを用いる。OSUAD-Sim の入力層、出力層のノード数は 784（後述のデータセットの次元数が $28 \times 28 = 784$ であるため）とし、隠れ層のノード数は [3] を参考に 32 とする。また、活性化関数は恒等関数 $G(\boldsymbol{x}) = \boldsymbol{x}$ を用い、損失関数は絶対平均誤差 $L(\boldsymbol{t}, \boldsymbol{y}) = \frac{1}{m}\sum_{i=0}^{m-1}|t_i - y_i|$ を用いる。

5.2　評 価 手 法

評価用のデータセットとしては、図 4 の MNIST [7] と、図 5 の Fashion-MNIST [8] を用い、前者を正常データ、後者を異常データとして利用する。両者は共に 28×28 のグレースケール画像のデータセットであり、60,000 枚の学習データと 10,000 枚のテストデータから構成される。正常な学習データ（4.1 節の $\boldsymbol{X}_{train}$ に相当）は MNIST の学習データから 10,000 をランダム抽出したものを用い、正常なテストデータ（4.1 節の $\boldsymbol{X}_{test}$ に相当）は MNIST のテストデータから 5,000 枚をランダム抽出したものを用いる。各実装はこれらを用いて 4.1 節の初期化フェーズの処理を行い、学習とパラメタの計算を行う。特に、OSUAD-Sim は OSUAD-Multi のインスタンス数が 1 の場合の処理を行う。MNIST のテストデータの残り 5,000 枚と、Fashion-MNIST のテストデータからランダムに抽出した 500 枚合わせた合計 5,500 枚の入力データ（正常データ数 : 異常データ数 = 10 : 1）に対し、各実装がいかに正確に異常データを検出できているかを評価する。この時の評価指標としては、**F 値**（f-measure）を用いる。F 値は**精度**（precision）と**再現率**（recall）の調和平均であり、それぞれを f、p、r とすると、以下の式で定義される。

$$p = \frac{TP}{TP+FP}, r = \frac{TP}{TP+FN}, f = \frac{2pr}{p+r} \qquad (10)$$

式 10 中の TP、FP、FN はそれぞれ True Positive（異常データに対し異常と推論した数）、False Positive（正常データに対し異常と推論した数）、False Negative（異常データに対し正常と推論した数）を意味する。F 値は誤判定である FP と FN を

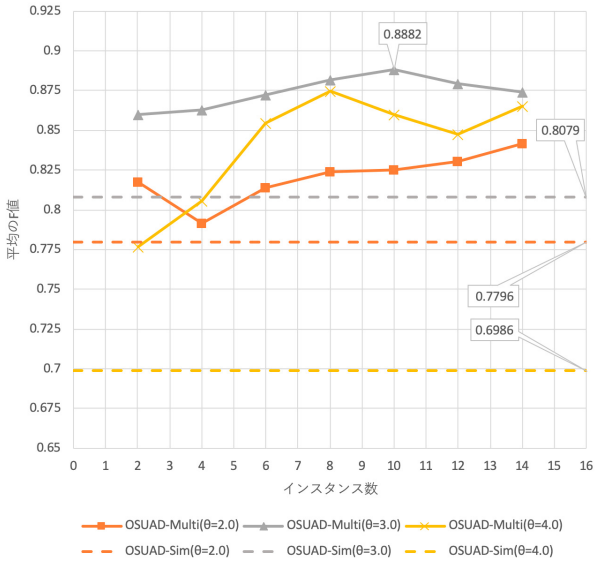

図 6　複数インスタンス化による F 値への影響

同時に考慮する指標であり、総合的な異常検知能力を示す指標として用いられる。通常、この値が最も高くなるように閾値等のパラメタを調節する [9]。

5.3　評 価 結 果

図 6 に、閾値 $\theta = \{2.0, 3.0, 4.0\}$ の時の、複数インスタンス化による F 値への影響を示す。横軸はインスタンス数 $= \{2, 4, 6, 8, 10, 12, 14\}$ の時の F 値をプロットしている。これらの F 値は、全て 50 回同じ手順で実験を繰り返した際に得られた値の平均値である。また、実線は OSUAD-Multi で、破線はベースラインの OSUAD-Sim（インスタンス数 1）を示す。尚、図 6 においては OSUAD-Conf は含まれないため、確信度スコア以前の複数インスタンス化による F 値の影響を示す。

評価結果としては、OSUAD-Sim の F 値の最大値（$\theta = 3.0$）と OSUAD-Multi の F 値の最大値（$\theta = 3.0$、インスタンス数 $= 10$）を比較すると、提案手法は 8.03%の改善が見られた。4.1 節で述べたように、提案手法ではまずクラスタ数を指定し、正常な学習データをその数分にクラスタリングする。そして、得られた各クラスタに対し一つのオートエンコーダインスタンスを割り当てる。よって、正常データのパターン数（この場合、MNIST のクラス数）がインスタンス数に近い程高い F 値が得られ、正常データのパターン数よりもインスタンス数が小さくなるほどあまり大きな改善が得られないと期待されるが、実際に、OSUAD-Multi($\theta = 2.0$)、OSUAD-Multi($\theta = 3.0$)、OSUAD-Multi($\theta = 4.0$) はそれぞれインスタンス数が 8、10、14 の時に最も高い F 値が得られており、多くの場合でインスタンス数が小さくなるほど F 値が小さくなっている。よって、正常データのパターン数が既知である場合や、その数が推測できる時は、その数分のインスタンスを用意することで相対的に大きな F 値の改善が得られると考えられる。

しかし、正常データのパターン数が予測不能な場合もある。この場合、インスタンス数を真の正常データのパターン数に一致させるのは難しいが、評価結果として OSUAD-Multi は全ての閾値 θ、インスタンス数において OSUAD-Sim よりも高い

表 2　確信度スコアの F 値への影響

	OSUAD-Sim	OSUAD-Multi	OSUAD-Conf ($\gamma = 0.1$)	OSUAD-Conf ($\gamma = 0.2$)	OSUAD-Conf ($\gamma = 0.3$)
θ	3.0	3.0	3.0	3.0	3.0
インスタンス数	1	10	10	10	10
精度	0.944	0.961	0.921	0.956	0.929
再現率	0.706	0.826	0.887	0.913	0.963
F 値	0.808	0.888	0.904	0.934	0.946
難データ率	NA	NA	0.0245	0.0763	0.146

値が得られている。よって、必ずしも真の正常パターンの数とインスタンス数が一致しなくとも、複数のインスタンスに正常データを分散させることで単一のインスタンスに割り当てられる正常データの多様性が小さくなり、結果として異常データの検出が容易になると考えられる。

表 2 は確信度スコアを導入した時の F 値への影響を示す。表中の OSUAD-Sim と OSUAD-Multi は図 6 において最も高い F 値を記録した設定（前者は $\theta = 3.0$、後者は $\theta = 3.0$、インスタンス数 10）を用いており、OSUAD-Conf は OSUAD-Multi の設定を用いる。また、γ は確信度スコアの閾値であり、これを下回るデータは**難データ**とし、全 5,500 枚の入力データに対する割合を**難データ率**とする。難データに区分されたデータは精度、再現率、F 値の計算に含めないものとし、表中の評価結果は全て同じ実験を 50 回繰り返した際に得られた平均値とする。

評価結果としては、γ を高い値に設定するほど、より高い F 値が得られている。確信度スコアはモデルの推論結果にどの程度信頼性があるかを示す指標であり、その閾値である γ を高くすれば、確信度の高い推論結果のみが F 値の計算に考慮されるため、この結果は妥当である。しかし、γ を大きくすればするほど、難データの割合が増加する。例えば、難データは人が検査することで異常検知を行う、という用途を考えた場合、難データの数が大きくなるほど人への負担や必要な人員数が増加すると考えられる。よって、この場合では許容される難データ率を超えない範囲で、最大の F 値を記録した設定を用いるのが適切だと考えられる。本研究の評価においては、$\gamma = 0.3$ の時以外は一割に満たない難データ率で OSUAD-Multi よりも F 値が改善しており、特に $\gamma = 0.2$ の時は OSUAD-Multi と比較して F 値が 4.6%改善している。この数値は、あくまで難データを F 値の計算から除外した結果であり、難データの検査を人に任せ、その精度が 100%であると仮定すると、より高い値が得られる。

6. 結　　論

本論文では、FPGA ベースの逐次学習型教師無し異常検知器である OSUAD [3] の異常検知精度を向上させる手法として、複数インスタンス化（4.1 節）と確信度スコア（4.2 節）を提案した。前者を適用することにより、単一のインスタンスに割り当てられる正常データの多様性（パターン数、分散等）が小さくなり、異常データの識別が容易になる。また、後者を導入することで、確信度の低い推論結果が得られた場合、そのデータの検査を人に任せるといった協調手法が可能になる。二つのオープンデータセットを用いた評価の結果、インスタンス数によらずベースラインを上回る F 値が得られ、正常データのパターン数とインスタンス数が近い場合には F 値が最大 8.03%向上した。また、上記に加えて確信度スコアを導入することにより、一割未満の難データ率（確信度が閾値を下回ったデータの割合）で F 値が 4.6%改善された。

今後の課題としては、本論文の提案手法の実機実装が考えられる。現時点ではシミュレーションのみであるため、例えばインスタンス数を増加させた時の面積や消費電力への影響を評価する必要がある。また、現状の確信度スコアの設計は確率論的な背景に欠けるため、その点を補完する必要がある。その他に、本論文では二つのオープンデータセットしか用いていないため、本文中の議論に汎用性が無い可能性がある。そのため、今後はより多数のデータセットを用いて評価を行いたい。

謝辞　本研究の一部は、JSPS 科研費 JP16H02816 および JST CREST JPMJCR1785 の助成による。

文　　献

[1] N.Y. Liang, G.B. Huang, P. Saratchandran, and N. Sundararajan, “A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks,” IEEE Transactions on Neural Networks, vol.17, no.6, pp.1411–1423, Nov. 2006.

[2] G. Hinton and R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” Science, vol.313, no.5786, pp.504–507, 2006.

[3] M. Tsukada, M. Kondo, and H. Matsutani, “OS-ELM-FPGA: An FPGA-Based Online Sequential Unsupervised Anomaly Detector,” Proceedings of the International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Platforms, Aug. 2018. http://www.arc.ics.keio.ac.jp/ matutani/papers/tsukada_heteropar2018.pdf.

[4] G.B. Huang, Q.Y. Zhu, and C.K. Siew, “Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks,” Proceedings of the International Joint Conference on Neural Networks, pp.985–990, July 2004.

[5] D.P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” arXiv preprint arXiv:1312.6114, 2013.

[6] A. Jinwon and C. Sungzoon, “Variational autoencoder based anomaly detection using reconstruction probability,” Special Lecture on IE, vol.2, pp.1–18, 2013.

[7] Y. Lecun and C. Cortes, “MNIST handwritten digit database,” http://yann.lecun.com/exdb/mnist/, 2010.

[8] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms,” https://github.com/zalandoresearch/fashion-mnist, 2017.

[9] 井出剛, 杉山将, “異常検知と変化検知,” pp.8–11, 講談社, 2015.