

# 識別に悪影響を及ぼす変数の推定・影響排除が可能な autoencoder ベース識別器の提案

石井 良尚<sup>1,a)</sup> 高梨 昌樹<sup>1,b)</sup> 三田 勝史<sup>1,c)</sup>

受付日 2018年5月23日, 採録日 2018年9月7日

**概要:** Neural Networks ベースの識別器 (NN 識別器) は, 学習用データが網羅的に収集されていない場合, 本来識別に無関係な変数を利用して識別するように学習してしまい, テストデータに対する識別精度が悪化することがある. この精度悪化を回避するための手段として, 学習用データに対する変数選択の実施が考えられる. しかし, 既存の変数選択手法を NN 識別器に対して適用する場合, 以下に示す 2 つの課題が存在する. 1 つは, NN 識別器においては入力変数と識別結果との依存関係を解析するのが困難なため, 効率的に識別精度を向上させる変数選択ができないという課題であり, もう 1 つは, クラスごとに異なる変数選択ができないという課題である. そこで本稿では, 上記課題を克服可能な autoencoder (AE) ベースの識別器である「マルチ AE 識別器」を提案する. マルチ AE 識別器は, 再構成誤差を利用することで識別精度を悪化させた原因の変数をクラスごとに独立して高精度に自動推定できる. 加えて, 推定した変数の影響を排除して識別できる. また本稿では, 実センサデータセットを用いた評価実験を行い, マルチ AE 識別器が実際に上記の特長を示すことを検証する.

**キーワード:** ニューラルネットワーク, 識別器, オートエンコーダ, 再構成誤差, 変数選択

## AE-based Classifier Estimating and Eliminating Variables that Deteriorate the Classification

YOSHINAO ISHII<sup>1,a)</sup> MASAKI TAKANASHI<sup>1,b)</sup> KATSUSHI SANDA<sup>1,c)</sup>

Received: May 23, 2018, Accepted: September 7, 2018

**Abstract:** Neural Networks-based classifiers (NN classifiers) may degrade the classification accuracy for the test data if the learning data is not exhaustively collected. That is because it inappropriately learns classes with some biased variables involved in the learning data. In order to avoiding the deterioration of the classification, it is simply conceivable to select each set of variables necessary for learning each class. However, there are two significant problems when we apply the variable selection to a NN classifier. One is that we cannot analyze how each variable affects the classification result in the NN classifier. The other is that the conventional NN classifier allows us to determine only a single set of variables for learning all of the classes. Therefore, we proposed a “multi-AE classifier” which is a new autoencoder (AE) based classifier. This classifier enables us to derive the reconstruction error for each variable, and the error enables us to estimate variables deteriorating the classification accuracy. This also allows us to construct each independent set of variables to classify each class. As a result, we achieve high accuracy classification. We verify the effectiveness of the multi-AE classifier with artificial and experimental data set.

**Keywords:** neural networks, classifier, autoencoders, reconstruction error, variable selection

<sup>1</sup> 株式会社豊田中央研究所  
Toyota Central R&D Laboratories, Inc., Nagakute, Aichi  
480–1118, Japan

a) e1597@mosk.tytlabs.co.jp

b) e1392@mosk.tytlabs.co.jp

c) e1175@mosk.tytlabs.co.jp

### 1. はじめに

近年, 識別性能の高さから Neural Networks (NN) ベースの識別器が注目され, センサデータを用いた異常検知, 状態推定等のタスクで利用され始めている. しかし, 一般

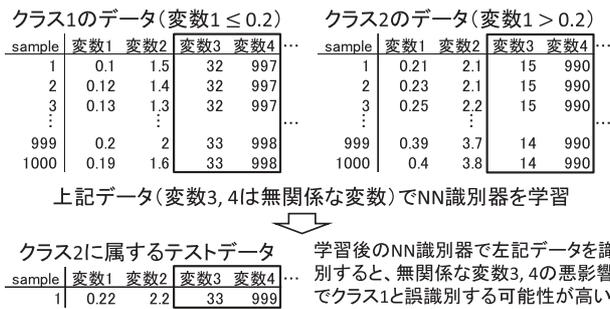


図 1 学習用データの偏りによる識別への悪影響

Fig. 1 A harmful effect on classification due to bias of train data.

に異常時のデータや稀な状態のセンサデータを網羅的に収集することは困難であり、NN 識別器に対する学習用データが不足することが多い。学習用データにおいて各クラスのデータが網羅的に収集されていない場合、各クラスの学習用データは、データ空間上で本来とりうる値の範囲に対して、一部の範囲に偏った値となる可能性がある。

学習用データに偏りが生じていると、識別精度に悪影響を及ぼす可能性がある。その例について、図 1 を用いて説明する。図 1 のように変数 1~4 の 4 変数からなるデータがあり、変数  $1 \leq 0.2$  となるデータ群をクラス 1、変数  $1 > 0.2$  となるデータ群をクラス 2 と定義し、それぞれ学習したとする。ここで、変数 3, 4 はクラス識別に無関係な変数であり、データを網羅的に収集できれば、クラス 1, 2 で同一の値の分布となることが期待される。しかし、学習データの不足により、クラス 1, 2 間で変数 3, 4 の値に偏りが生じた場合、本来クラス 2 に属すると識別されるべきテストデータが、クラス 1 に属すると誤識別される可能性が発生する。

このように、学習用データに偏りが生じている場合、NN 識別器は本来識別に無関係な変数や誤った特徴を用いて識別するように学習してしまう可能性が高く、そのように学習された NN 識別器は未知のテストデータに対して高い識別精度を出しにくい。以降、本来識別に無関係ながら識別結果に影響を与える変数や誤った特徴を導く変数を、識別に悪影響を及ぼす変数と称する。

NN 識別器において、このような識別に悪影響を及ぼす変数による識別精度悪化を回避するための手段として、学習用データに対する変数選択（特徴選択、入力選択とも呼ばれる）[1] の適用が考えられる。変数選択の手法を適用することで識別に悪影響を及ぼす変数を学習用データから自動的に推定でき、それらの変数を取り除いて学習することでテストデータの識別精度を向上させられる可能性がある。既存の変数選択手法は、なんらかの評価基準を定義し、その評価値が最大になるように学習に用いる変数を選択する。具体的には、元の全変数集合を  $X$ 、評価値を計算する関数を  $C$  としたとき、以下の式 (1) を満たす変数の部分集

合  $T$  を導出する。

$$T = \arg \max_{T' \subseteq X} C(T') \quad (1)$$

これらの手法は、 $T$  の求め方によって大きく 3 種類に大別できる [2]。各々について NN 識別器への適用の観点から概説する。

1 つめはフィルタと呼ばれる方式 [3] の手法群で、NN 識別器の学習を介さず、学習用データのみを用い、たとえば変数間の相関等を用いた評価基準の元で変数選択を行う。これらの手法は、NN 識別器の識別精度への影響を考慮せずに変数選択を行うため、識別に悪影響を与える変数を積極的に除去する機能がない。

2 つめは組み込みと呼ばれる方式の手法群で、Lasso [4] やランダムフォレスト [5] のように、学習アルゴリズム内に変数選択の評価基準が組み込まれている手法である。NN 識別器の場合、たとえばスパース正則化 [6] やアテンション [7] 等の利用がこれにあたる。これらの手法は、学習用データをうまく識別するよう変数選択しながら学習する機能を持つが、悪影響を及ぼす変数を除去するように学習する機能は備わっていない。

3 つめはラッパーと呼ばれる方式 [8] の手法群で、NN 識別器の学習時のクロスバリデーション (CV) 評価基準値を最大にするように変数選択を行う。この際の変数選択の戦略は手法によって異なる。ラッパー方式のアプローチをとる場合、他のアプローチと異なり、NN 識別器の CV 評価基準値を最大にする過程で、偏りが生じた変数の悪影響を検知でき、その変数を除外できる可能性がある。そのため、NN 識別器の識別に悪影響を及ぼす変数を学習時に変数選択で除去するためには、ラッパー方式が有望である。

しかし、ラッパー方式の既存手法を NN 識別器に対して適用するには、以下の課題が存在する [9]。

- NN 識別器において入力変数と識別結果との依存関係を解析するのは困難であり、識別精度を向上させるための変数選択の戦略は、ある程度試行錯誤的なアプローチをとらざるを得ない。そのため、識別に悪影響を及ぼす変数の特定に計算コストがかかる。
- 識別に悪影響を及ぼす変数がクラスごとに異なる状況が発生しうが、NN 識別器ではクラスごとに異なる変数選択を行えず、すべてのクラスに対して共通の変数を取捨選択することしかできない。

そこで本稿では、以下の特徴を持つ新しい NN ベース識別器である「マルチ AE 識別器」を提案する。

- あるクラスのデータを誤識別した際に、どの変数が誤識別に寄与したかを直接的に導出できる。
- クラスごとに独立した変数選択ができる。

マルチ AE 識別器は、NN モデルの中でも autoencoder (AE) [10] と呼ばれるモデルを応用して実現する。次章において、提案するマルチ AE 識別器に関連する、既存の NN

識別器に対するラッパー方式の変数選択手法について述べ、マルチ AE 識別器と既存手法の違いについて言及する。

## 2. 関連研究

センサデータをはじめとする実データを用いた識別タスクでは、学習用データの次元数やサンプル数が多くなることもあり、その場合には NN 識別器の学習に多くの時間がかかる。このような状況では、変数選択の組合せを網羅的に探索し、評価基準値を最大にする選択を行う戦略を取ることは計算コストの観点から現実的でない [9]。

そこで、変数選択がされていない状態から探索を開始し、各入力変数が CV 評価基準値（以降、基準値と略記する）の向上に貢献した度合いを推定し、貢献度の低い変数から除外していく探索戦略（sequential backward elimination [11]；SBE）をとることで、計算コストを抑えながら基準値を効率的に向上させる探索を実現する手法が提案されてきた [12], [13], [14], [15], [16], [17]。

これらの手法は、各変数の貢献度の推定方法によって大きく 2 種類に分類することができる。1 種類めは、貢献度の推定に基準値を直接利用する手法である。文献 [12] では、基準値として CV 識別精度を採用したうえで、ある 1 つの変数を除外した際の CV 識別精度変化を計測し、精度が向上するかほとんど下がらない場合はその変数を除外し、その元で NN 識別器を再学習する。その後、再びある変数を除外して精度変化を計測する、というプロセスを繰り返すことで、試行錯誤的に基準値を向上させるための探索を行う。探索時の変数除去による精度変化を顕著にするため、目的関数にネットワークの重みを用いたペナルティ項を追加している点に特徴がある。文献 [13] も同様のアプローチをとるがペナルティ項が異なり、ネットワークの重みではなく活性化関数の導関数を元にする値を用いる。文献 [14], [15] では、基準値として誤差関数（loss function）の値を採用する。文献 [14] では、ある変数をデータから削除して学習した際の誤差を変数ごとに導出し、最も誤差が低くなる変数を除外していくという戦略をとる。文献 [15] では、ある変数をランダム値に置き換えた場合の誤差を変数ごとに導出し、置き換える前と比べて誤差が低くなった変数を除外するという戦略をとる。これらの基準値を直接向上させることを狙う手法は、学習用データに対するオーバーフィッティングが起りやすいという欠点が存在する。

2 種類めは、学習用データに対するオーバーフィッティングを避けた変数選択を行うため、基準値とは異なる、識別に本質的に貢献した入力変数を推定する尺度を定義し、その尺度に基づき変数選択を行う手法である。文献 [16] では、基準値には CV 識別精度を採用する一方で、変数選択は各入力変数が出力値に対して与えた影響度の推定値を用いて行う。具体的には、ある入力ニューロンからある出力ニューロンへ到達可能な各パスにおける重みパラメータの

積を計算し、それらの絶対値の和を求め、その値の大きさを、その入力変数の推定貢献度として定義している。この推定貢献度が低い変数から除外していき、基準値が最大となったときの変数選択を結果として出力する。文献 [17] では、重みパラメータだけでなく、中間層の活性も考慮した推定貢献度を定義している。この貢献度を元に、相関が高い入力変数間で、貢献度が低い方を冗長な変数として除外することで CV 識別精度の向上を狙うアプローチをとる。変数選択による精度改善は、一般的に 2 種類めに属する手法の方が高くなる傾向にあり [16], [18], マルチ AE 識別器は 2 種類めに属する手法である。

しかし、2 種類目に属する既存手法が定義する推定貢献度は、出力に対して入力がどの程度影響を与えたかを近似的に数値化するものであり、識別に悪影響を及ぼす変数を特定できるとは限らない。一方で、提案するマルチ AE 識別器では、再構成誤差を尺度として利用することで、識別に悪影響を及ぼした変数を直接的に導出し変数選択することができる。

また、一般的な NN 識別器では次元数の異なるデータを直接的に扱うことができないため、どちらに属する既存手法もクラスごとに異なる変数選択ができない。データによってはクラスごとに識別に悪影響を及ぼす変数が異なることがあり、その場合はクラスごとに異なる変数選択を行う方が識別器の識別精度向上につながる事が検証されている [19], [20], [21]。提案するマルチ AE 識別器は、NN ベースの識別器ながらクラスごとに異なる変数選択が可能であり、既存手法よりも効果的に変数選択できることが期待される。

なお、AE を用いた変数選択手法は複数提案されている [22], [23] が、基本的に組み込み方式を採用しており、その変数選択の目的は性質の良い再構成の実現であり、直接的に識別精度を向上させることを目的としていない。マルチ AE 識別器は AE をベースにマルチクラス識別とラッパー方式の変数選択を同時に行うことができ、この点で既存の AE を用いた変数選択手法と異なっている。

以降の章にて、マルチ AE 識別器のベースとなる AE について概説し、提案するマルチ AE 識別器の詳細について述べた後、実センサデータを用いた評価実験を行う。

## 3. Autoencoder

### 3.1 概要

AE は NN モデルの一種で、入力と出力のニューロン数が同一なネットワーク構造を持ち、 $\{\mathbf{x}_i\}$  ( $i = 1, \dots, N$ ) を学習用データ、 $\{\bar{\mathbf{x}}_i\}$  ( $i = 1, \dots, N$ ) を各学習用データに対応する AE からの出力とすると、一般的に以下の式 (2), (3) で定義されるデータごとの再構成誤差  $e_i$  の相加平均  $em$  を最小化するようにモデルパラメータを学習することを特徴とする。ただし、 $N$  はサンプル数である。

$$em = \frac{1}{N} \sum_{i=1}^N e_i, \quad (2)$$

$$e_i = \sum_{d=1}^D (x_{i,d} - \bar{x}_{i,d})^2. \quad (3)$$

ただし、 $D$  は学習用データの次元数であり、 $x_{i,d}$ ,  $\bar{x}_{i,d}$  ( $d \in \{1, \dots, D\}$ ) は  $\mathbf{x}_i$ ,  $\bar{\mathbf{x}}_i$  の  $d$  番目の要素を表す。以降、 $\mathbf{x}_i$ ,  $\bar{\mathbf{x}}_i$  の各要素ごとの二乗誤差  $(x_{i,d} - \bar{x}_{i,d})^2$  を、変数ごとの再構成誤差と呼ぶ。

### 3.2 1 クラス識別器としての利用

一般的に AE は、中間層のニューロン数を入力層よりも小さくして学習することで、入力の特徴をうまく表す低次元空間への写像として利用されるが、1 クラス識別問題に対する識別器としても利用することが可能である [24], [25]. AE を 1 クラス識別器として利用する場合は、以下の手順で識別を行う。はじめに、ある 1 つのクラスを持つ学習用データのみで AE を学習させる。次に、学習済みの AE に対してあるテストデータを入力し、その再構成誤差を計算する。このとき、計算された再構成誤差があらかじめ設定された閾値より小さい場合、そのテストデータは学習に用いたデータ群と空間的に似通った座標に存在する、すなわち、学習に用いたデータ群と同じクラスに属すると識別できる。反対に、再構成誤差が閾値より大きい場合は、学習に用いたデータ群とは異なるクラスに属すると識別できる。すなわち、AE を 1 クラス識別器として利用する場合は、再構成誤差の大きさに基づいて識別を行う。

ここで、あるテストデータに対して、AE が誤識別した場合を考える。このとき、誤識別の原因は再構成誤差が大きくなったためであり、変数ごとの再構成誤差の大きさは、そのまま誤識別への寄与度と見なすことができる [26]. つまり、誤識別が発生した際には、変数ごとの再構成誤差の大きさを参照し、その値の大きな変数が、誤識別の要因となった変数であると推定できる。

## 4. マルチ AE 識別器

### 4.1 AE を用いた変数選択方法とマルチクラス識別に対する課題

前節で述べたように、1 クラス識別器としての AE は、変数ごとの再構成誤差を利用することで誤識別の要因となった変数を推定できる。この特性を利用することで、AE を用いた識別を行う際に、効果的なラッパー方式の変数選択が実現できると考えられる。具体的には、学習用データに対して CV を実施し、誤識別したデータ群における変数ごとの再構成誤差の値が大きな変数を学習や識別に利用しないようにする戦略をとることで、効率的な SBE 方式の変数選択が実現できると期待される。一般的な NN 識別器では誤識別の原因となった変数を直接的に導出できないため、

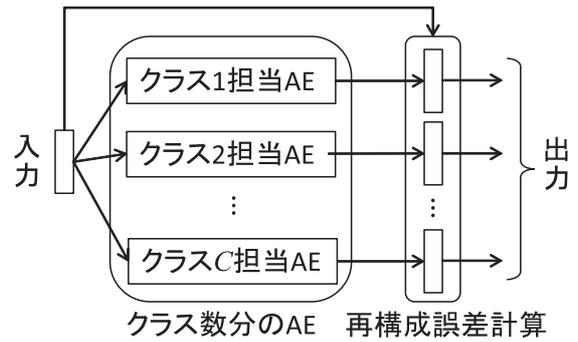


図 2 マルチ AE 識別器のネットワーク構造

Fig. 2 The network structure of the multi-AE classifier.

識別に悪影響を与えた変数の探索は試行錯誤的にならざるを得ないが、AE を用いた識別ではその変数を直接的に推定できるため、変数選択のコストが低く抑えられると考えられる。

ただし、上記のとおり、基本的に AE は 1 クラスの識別問題にしか適用できない。マルチクラスの識別問題を AE で解く方法として、クラス数と同数の AE を用意し、それぞれ異なるクラスのデータを再構成するよう独立に学習させて、テストデータに対して各 AE から出力される再構成誤差の大小でデータが属するクラスを識別することが考えられる。しかし、各 AE は異なるデータで学習されるため、計算される再構成誤差の尺度が一致せず、その大小で識別することは適切ではないと推察される。

そこで次節において、この AE の欠点を克服しながら、1 章にて述べた 2 つの特徴を持つマルチ AE 識別器を提案する。

## 4.2 提案手法

### 4.2.1 ネットワーク構造

提案するマルチ AE 識別器のネットワーク構造を図 2 に示す。マルチ AE 識別器は、学習用データに含まれるクラス数  $C$  と同数の AE を持ち、各 AE は担当するクラスに属するデータのみを再構成することを目的としている。以降、クラス  $c \in \{1, \dots, C\}$  のデータの再構成を担当する AE を、 $AE(c)$  と表記する。

また、ネットワーク中の再構成誤差計算部にて、入力  $\mathbf{x}_i$  と  $AE(c)$  の出力  $\bar{\mathbf{x}}_i^c$  の再構成誤差  $e_i^c$  を以下の式 (4) にて計算する。

$$e_i^c = \frac{1}{D} \sum_{d=1}^D (x_{i,d} - \bar{x}_{i,d}^c)^2. \quad (4)$$

なお、 $D$  で平均する理由については後述する。入力  $\mathbf{x}_i$  に対するマルチ AE 識別器の最終的な出力  $\mathbf{z}_i$  は、 $\mathbf{z}_i = \{e_i^1, \dots, e_i^C\}$  となる。データの識別時には、入力  $\mathbf{x}_i$  の属するクラスは、以下の式 (5) で得られる  $c_{\min}$  であると予測する。

$$c_{\min} = \arg \min_{c \in \{1, \dots, C\}} e_i^c. \quad (5)$$

つまり、入力データの属するクラスは、最も小さな再構成誤差となった AE が再構成を担当したクラスであると予測する。このように、マルチ AE 識別器は各 AE の再構成誤差を基準として識別を行う。

#### 4.2.2 学習方法

$\mathbf{x}_i$  の属するクラスを  $y_i \in \{1, \dots, C\}$  とすると、マルチ AE 識別器は以下の誤差関数  $Loss$  を最小化するように学習する。

$$Loss = \frac{1}{N} \sum_{i=1}^N \left( e_i^{y_i} + \alpha \sum_{c=1, c \neq y_i}^C \max(0, \beta e_i^c - e_i^c) \right). \quad (6)$$

ただし、 $\alpha (> 0)$ 、 $\beta (\geq 1)$  は、以下で述べるペナルティ項の重みを決定するパラメータである。外側の  $\Sigma$  内の第 1 項は、入力データが属するクラスの再構成を担当する AE の再構成誤差を最小化するための項である。第 2 項はヒンジ関数をベースとするペナルティ項であり、入力データが属するクラスの再構成を担当する AE の再構成誤差を、そのほかすべての AE の再構成誤差よりも相対的に小さくするための項である。両項を最小化するように学習することで、学習後のマルチ AE 識別器は、各学習用データが属するクラスの再構成を担当する AE が、各学習用データを最も良く再構成するようになり、式 (5) を用いて学習用データの識別が可能になる。これは、各 AE を独立に学習するのではなく、すべての AE を同時に学習するため、学習用データに対する各 AE 間の再構成誤差の尺度が一致することによる。学習用データのクラスごとのデータ数に偏りがある場合においても、各データが属するクラスを担当する AE の再構成誤差を最小にするように学習される性質は変化しないため、学習後の各 AE 間の誤差尺度は一致し正常な識別が可能である。なお、もし  $\alpha = 0$  とした場合は、各 AE を独立して学習する場合と一致し、各 AE 間の再構成誤差の尺度が一致するように学習されない。

また、すでに変数選択が済んでおり、クラス  $c$  の識別に利用しない変数の集合  $X_c \subseteq \{1, \dots, D\}$  が与えられている場合、以下の 2 つの制約をモデルに課しながら学習および識別することで  $X_c$  が AE( $c$ ) に与える影響を排除することができる。

(1) AE( $c$ ) の再構成誤差を計算する際、 $X_c$  に含まれる変数の再構成誤差を 0 として計算する。すなわち、式 (4) を以下の式に変更する。

$$e_i^c = \frac{1}{D - |X_c|} \sum_{d=1, d \notin X_c}^D (x_{i,d} - \bar{x}_{i,d}^c)^2. \quad (7)$$

ただし、 $|\cdot|$  は要素数を表す。なお、式 (4) および式 (7) にて、各クラスの学習に用いる変数の数で再構成誤差を平均する理由は、各 AE 間における変数ごとの再構成誤差の尺度を、変数選択の数によらず同一に保つた

めである。

(2) AE( $c$ ) の入力層において、 $X_c$  に関係する重みパラメータをつねに 0 に固定する。具体的には、 $X_c$  に関係する重みパラメータの初期値を 0 にしたうえで、学習途中にパラメータの更新が起こらないようにする。

なお、 $X_c$  はクラスごとに異なるものを設定可能である。すなわち、クラスごとに異なる変数選択が可能である。

#### 4.2.3 変数選択法—識別に悪影響を与える変数の推定方法

学習後のマルチ AE 識別器に対してクラス  $c$  に属するデータを入力した際に、誤識別した場合を考える。誤識別は、AE( $c$ ) の再構成誤差が最小とならず、相対的に大きな値となったために起きている。すなわち、3 章で述べた一般的な AE の場合と同様に、AE( $c$ ) の変数ごとの再構成誤差の値が大きい変数が識別に悪影響を与えた変数であると推定できる。そのため、誤差の大きな変数の影響を排除する ( $X_c$  に追加する) ことで、精度改善に効果的な変数選択が実現できる。以下において、マルチ AE 識別器を用いた変数選択 ( $X_c$  導出) の具体的な手順を述べる。

手順 1  $X_c$  ( $c = 1, \dots, C$ )、および  $\bar{X}_c$  ( $c = 1, \dots, C$ ) を空にする。  $cnt = 0$  に設定する。

手順 2 各 AE( $c$ ) の変数ごとの再構成誤差を保存する空のリスト群  $List(c)$  を用意する。

手順 3 現在の  $X_c$  のもと、学習用データに対して  $N_p$ -fold cross-validation を実施する ( $N_p$  はパラメータ)。各 fold では、識別に失敗したバリデーションデータに対し、各データが属するクラス  $c$  に対応する AE( $c$ ) の変数ごとの再構成誤差を求め、それぞれ  $List(c)$  に追加する。ただし、 $X_c$  に含まれる変数の再構成誤差は 0 とする。また、全 fold を通じたクラスごとの平均識別精度  $a_c$ 、および全 fold を通じた全クラスの平均識別精度  $a_{all}$  を計算する。

手順 4 すべての  $X_c$  が空のとき、 $a_{max} = a_{all}$ 、 $a_{first} = a_{all}$  と設定し、手順 6 に移行する。そうでないとき、手順 5 に移行する。

手順 5  $a_{first} > a_{all}$  ならば  $cnt := cnt + 1$ 、そうでなければ  $cnt := 0$  とする。その後、 $cnt < cnt_{max}$  のとき、 $a_{max} \leq a_{all}$  ならば  $\bar{X}_c := X_c$ 、 $a_{max} := a_{all}$  として手順 6 に移行し、 $a_{max} > a_{all}$  ならばそのまま手順 6 に移行する。 $cnt = cnt_{max}$  ならば変数選択を終了し  $\bar{X}_c$  を結果として出力する。ただし、 $cnt_{max}$  はパラメータである。

手順 6  $a_c < a_{all}$  であるクラス  $c$  ごとに、 $List(c)$  のサンプル方向の中央値が最大の変数を  $X_c$  に追加し、手順 2 に戻る。

上記のように、マルチ AE 識別器では、クラスごとに異なる変数選択を行いながら CV 識別精度を向上させ、その精度が最大となったときの変数選択を結果として出力する。変数選択の探索戦略は、多くの既存手法と同様 [27] に、低

い計算コストながらも精度改善が期待できる SBE 方式を採用している。変数選択の終了条件は、変数選択を行っていない状態でのバリデーション識別精度と比べて  $cnt_{\max}$  ステップ連続して精度が低下した場合である。  $cnt_{\max}$  を大きくするほど計算コストが増加する代わりに探索する変数の組合せが増え、より良い  $a_{\max}$  時の変数選択結果が得られやすくなる。

## 5. 評価実験

### 5.1 実験方法

本章では、提案したマルチ AE 識別器の評価実験を行う。実験は 3 種類行い、各々は以下の観点からマルチ AE 識別器を評価するために行う。

**実験 1** 変数選択がされていない条件下にて、センサデータを元にした識別問題に対して、一般的な NN 識別器との識別性能と遜色ないか。

**実験 2** 識別に悪影響を及ぼす変数を意図的に組み込んだセンサデータに対して、マルチ AE 識別器にその変数を特定させ、特定した変数の影響を排除するよう再学習した際に、識別精度が向上するか。

**実験 3** 実センサデータに対して学習用データのみを用いて変数選択を行い、テストデータに対する識別精度が向上するか。

実験を通じて用いるデータセットは、スマートフォンのセンサデータから所持者の状態（歩き、座り等）を予測する HAR データ [28]、および同種のデータである AAL データ [29] である。各変数値は min-max normalization によって最小値が  $-1.0$ 、最大値が  $1.0$  となるように正規化する。表 1 に各データセットの概要を示す。

以下において、各実験の実験方法について述べる。実験 1 では、各データに対してマルチ AE 識別器と NN 識別器のネットワークパラメータの初期値を変えて 100 回学習・テストを繰り返し、その平均識別精度を比較する。また、マルチ AE 識別器のパラメータ  $\alpha$  を 0 に設定した場合、すなわち、クラスごとに各 AE を独立に学習した場合における平均識別精度も同様に算出し比較する。以降において、平均識別精度と記載されている場合、それは実験 1 と同様にネットワークパラメータの初期値を変えて 100 回学習・テストを繰り返した際に得られる平均識別精度を意味することとする。

実験 2 は、以下に述べる方法で行う。はじめに、HAR データの全変数 561 種から、10 種（変数リストの後半の 10 変数）の変数のみを抽出したデータを作成する。この変数の限定は、以下で説明する方法で付与する変数値の偏りによる悪影響を顕著に生じさせるための処置である。次に、変数を限定した 10 次元 HAR データの学習用データに対して、値の偏りを生じさせた「学習用バイアスデータ」を作成する。学習用バイアスデータは、クラスごとに異なる

表 1 データセットの概要

Table 1 The overview of datasets.

データ	クラス数	次元数	学習用データ数	テストデータ数
HAR	6	561	7,352	2,947
AAL	6	561	4,252	1,492

変数を 1 つ乱択し、選択された変数における学習用データの各値を、 $[-1.0, 1.0]$  の範囲内からランダムに決定したクラスごとに異なる固定値の近傍に分布する値（決定された固定値に平均 0、分散 0.04 の正規分布を元に発生させたノイズを加えた値）に置き換えることで作成する。以降、値に偏りが生じている変数をバイアス変数と表記する。学習用バイアスデータを用いて識別器を学習させると、バイアス変数による偏りから得られる偽の特徴を利用して識別するよう学習されるため、テストデータに対する識別精度は元学習用データを用いる場合と比べて悪化する。

その後、変数選択がされていない条件下にて、学習用バイアスデータを用いてマルチ AE 識別器を学習する。学習後、テストデータの識別を行い、その識別精度  $ACC_{\text{before}}$ 、および誤識別したテストデータ集合を導出する。次に、集合内のテストデータをクラスごとに分割し、分割ごとに学習済マルチ AE 識別器に入力する。そして、各分割のクラス  $c$  を担当する  $AE(c)$  における、変数ごとの再構成誤差をサンプルごとに計算し、誤差の中央値が最大の変数が、クラス  $c$  におけるバイアス変数であると推定する（4.2.3 項の手順 6）。最後に、式 (6), (7) を用いて、クラスごとに推定された各変数 ( $X_c$ ) の影響を排除するようにマルチ AE 識別器を学習用バイアスデータで再学習し、再学習後モデルのテストデータに対する識別精度  $ACC_{\text{after}}$  を求める。

この実験を、バイアス変数の組合せと偏らせる値を 100 通りに変化させて試行し、各試行における  $ACC_{\text{after}}$  が  $ACC_{\text{before}}$  に比べて向上するか検証する。また、各  $ACC_{\text{after}}$ ,  $ACC_{\text{before}}$  の平均値を、正解のバイアス変数を設定した場合の平均値および 10 次元の元学習用データを用いた場合の平均識別精度と比較する。さらに、NN 識別器を用い、10 次元の元学習用データと学習用バイアスデータを用いた際の平均識別精度、およびバイアス変数の正解が分かっている前提の元で全バイアス変数をデータから除去した 4 次元の学習用データを用いて学習・テストした際の平均識別精度とも比較する。

実験 3 は、HAR, AAL データに対する変数選択を行う前のマルチ AE 識別器の平均識別精度と、3.3 節にて述べた方法で変数選択を行った後の平均識別精度を比較し、その精度変化を評価する。また、比較手法として、NN 識別器に対するラッパー方式の変数選択手法として広く知られている ANNIGMA [16]、および MFS [17] 手法を採用し、NN 識別器に対する手法適用前後の平均識別精度変化を求め、マルチ AE 識別器と比較する。

表 2 各実験で用いるニューロン数と活性化関数

Table 2 The number of neurons and the activation functions used in each experiment.

実験	データ	手法	ニューロン数	活性化関数
1	HAR	MAE	100	{relu, none}
		NN	100	{sigmoid, none}
	AAT	MAE	1,000	{relu, none}
		NN	1,000	{relu, none}
2	HAR	MAE	100	{relu, none}
	(10次元)	NN	100	{sigmoid, none}
3	HAR	MAE	100	{relu, none}
		NN	100	{sigmoid, none}
	AAT	MAE	1,000	{relu, none}
		NN	100	{sigmoid, none}

## 5.2 実験条件

はじめに、全実験で共通して利用するパラメータ等について述べる。マルチ AE 識別器の各 AE と NN 識別器のネットワーク構造は、中間層が 1 層の全結合ネットワークを採用し、実装には Chainer [30] (ver. 1.21.0) を用いる。最適化手法は RMSpropGraves [31] を用い、各パラメータは  $\{\text{lr, alpha, momentum, eps}\} = \{0.001, 0.95, 0.95, 0.0001\}$  と設定する。なお、学習率 lr に関しては、マルチ AE 識別器では 40 エポックごとに、NN 識別器では 20 エポックごとに 0.5 倍する。バッチサイズは 200 とし、エポック数が 200 を過ぎた時点で学習を終了する。マルチ AE 識別器のパラメータ  $\alpha, \beta$  は、 $\alpha = 10.0, \beta = 2.0$  を用いる。

次に、各実験ごとに利用するネットワークパラメータ等について述べる。ネットワーク構造および活性化関数は表 2 に示すものを利用する。ただし、表 2 内の手法における MAE と NN は、それぞれマルチ AE 識別器と NN 識別器を意味し、活性化関数の  $\{-, \}$  は左側が入力層–中間層間の、右側が中間層–出力層間の活性化関数を意味し、none は活性化関数を用いないことを意味する。なお、実験 1 と 3 において、NN 識別器の AAT で利用するニューロン数と活性化関数が異なるのは、実験 3 で用いる両比較手法が活性化関数として sigmoid 関数を前提とするためであり [16], [17], AAT データに対する平均識別精度は実験 1 での設定を用いる方が高くなる (実験結果を参照)。また、マルチ AE 識別器では、中間層のニューロン数を入力層のニューロン数より大きく設定する場合、すなわち、実験 1, 3 の AAT データおよび実験 2 において、各 AE の学習結果が恒等写像となることを防ぐため、中間層からの出力値に L1-正則化 [32] を設定する。具体的には、式 (6) の目的関数を以下の  $Loss_{\text{mod}}$  に変更して学習する。

$$Loss_{\text{mod}} = Loss + \frac{\lambda}{N} \sum_{i=1}^N \sum_{j=1}^J |h_j^{y_i}|. \quad (8)$$

ただし、 $J$  は各 AE の中間層のニューロン数であり、 $|h_j^{y_i}|$  は  $AE(y_i)$  の中間層における  $j$  番目のニューロンの出力値

表 3 各データセットに対する各手法の識別精度

Table 3 The identification accuracy of each method for the datasets.

データ	手法	識別精度 [%]
HAR	MAE	96.22
	NN	95.58
	AE-dist	87.38
AAL	MAE	88.17
	NN	87.54
	AE-dist	82.12

の絶対値を表す。パラメータ  $\lambda$  は共通して 0.1 を用いる。なお、上記パラメータ群は、試行錯誤的に探索した中で、最良の結果となったものを採用している。

また、実験 3 におけるマルチ AE 識別器の  $cnt_{\text{max}}$  は 4 とし、 $N_p$  および両比較手法の変数選択尺度導出時における CV の fold 数は 10 とする。また、ANNIGMA の変数選択戦略は BE (SBE 方式と同等、文献 [16] を参照) とし、探索は全通り (561 通り) の変数選択を試行した時点で終了する。MFS に関しては、変数選択戦略として Opt-II (文献 [17] を参照) を採用する。

## 5.3 実験 1 に対する実験結果

各データセットに対する各手法の平均識別精度を表 3 に示す。なお、AE-dist は独立に AE を学習して識別する手法 (マルチ AE 識別器の  $\alpha = 0$  の場合) を意味する。結果から、今回用いたデータセットに対するマルチ AE 識別器の基本的な識別性能は、NN 識別器と遜色ないことが確認できた。また、各 AE を独立に学習して識別する手法の識別精度は、マルチ AE 識別器、および NN 識別器よりも低くなることが確認できた。このことから、クラス数分の AE を用いて再構成誤差を元に識別する際には、各 AE の再構成誤差の尺度を一致させるよう学習する必要性があることを検証できた。

## 5.4 実験 2 に対する実験結果

マルチ AE 識別器の各試行における識別精度の改善、すなわち、 $(ACC_{\text{after}} - ACC_{\text{before}}) [\%]$  を図 3 に示す。加えて、各手法の各条件の元での平均識別精度を表 4 に示す。なお、表 4 中のデータバイアスにおける「なし」は、バイアス変数を設定していない 10 次元の学習用元データを利用したことを、「あり」は学習用バイアスデータを利用したことを意味する。また、変数選択における「なし」は変数選択を行わずに学習して識別したことを、「手法が推定」はマルチ AE 識別器が自動特定したバイアス変数の影響を除外するよう学習して識別したことを意味する。「正解付与」は、マルチ AE 識別器においては正解の変数選択が行われた状況で学習して識別したことを意味し、NN 識別器においてはバイアス変数をすべてデータから除去した 4 次

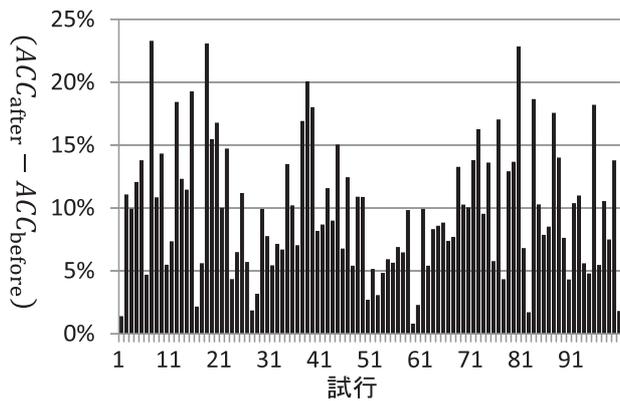


図 3 HAR データを元にした 10 次元トイデータに対する各試行における変数選択による精度改善

Fig. 3 The accuracy improvement by variable selection in each trial on the 10-dimensional toy data based on HAR dataset.

表 4 HAR データを元にした 10 次元トイデータに対する変数選択による精度改善

Table 4 The improvement of accuracy by variable selection for the 10-dimensional toy data based on HAR dataset.

手法	データバイアス	変数選択	識別精度 [%]
MAE	なし	なし	64.18
	あり	なし	47.91
	あり	手法が推定	57.69
	あり	正解付与	59.60
NN	なし	なし	66.64
	あり	なし	48.96
	あり	正解付与	54.56

元データで学習して識別したことを意味する。

これらの結果から、以下のことがいえる。

- 図 3 より、バイアス変数が異なる 100 回の試行の全試行において、マルチ AE 識別器が識別に悪影響を及ぼしたと推定した変数の影響を取り除くように再学習することで、識別精度が向上していることが確認できる。このことから、マルチ AE 識別器は、変数値の偏りから生じる悪影響によって低下した識別精度を改善可能な変数選択を実現できることが検証できた。また、表 4 の MAE における「手法が推定」時の精度改善 (9.78 ポイント) と「正解付与」時の精度改善 (11.69 ポイント) の差が小さいことが確認でき、マルチ AE 識別器が高精度で悪影響を及ぼした変数を特定できることが検証できた。
- 表 4 より、バイアス変数の正解値を与えられ、それらをすべてデータ中から削除した NN 識別器の精度 (データバイアス: あり, 変数選択: 正解付与) よりも、マルチ AE 識別器が自動で変数選択し再学習した場合の精度 (データバイアス: あり, 変数選択: 手法が推定) と、正解の変数選択が与えられた場合の精度 (データバイアス: あり, 変数選択: 正解付与) の方が

表 5 実センサデータに対する変数選択による精度改善

Table 5 The improvement of accuracy by variable selection for the real sensor data.

データ	手法	識別精度 [%]		精度変化 [pp *1]
		変数選択前	変数選択後	
HAR	MAE	96.22	96.49	+0.27
	ANNIGMA	95.58	95.73	+0.15
	MFS	95.58	95.65	+0.07
AAT	MAE	88.17	88.38	+0.21
	ANNIGMA	85.33	83.70	-1.63
	MFS	85.33	85.11	-0.22

表 6 HAR データセットにて選択された変数

Table 6 The selected variables in HAR dataset.

クラス 1	2	3	4	5	6
—	—	—	78,79,80,158,	80,78,79,160,	—
			557,160,159	558,557,40	—

表 7 AAL データセットにて選択された変数

Table 7 The selected variables in AAL dataset.

クラス 1	2	3	4	5	6
—	80,79,558,	—	558,557,	558,79,80,	—
	78,556,		40,160,	39,557,	
	157,557		159,78,38	160,159	

識別精度が高いことを確認できる。これは、クラスごとに異なる変数選択ができず、あるクラスの識別には不要であるが他のクラスの識別には必要である変数を取り除かざるを得ない NN 識別器に対し、クラスごとに異なる変数選択が可能で、あるクラスに対して選択された変数はそのクラス以外の識別には利用可能なマルチ AE 識別器の方が、より多くの情報を利用して学習および識別できるからであるといえる。

### 5.5 実験 3 に対する実験結果

各データセットに対する各手法の変数選択前後の平均識別精度、およびその精度変化を表 5 に示す。また、マルチ AE 識別器において、各データセットで変数選択により影響排除された変数番号 (one-based) を表 6、表 7 に示す。なお、比較手法の変数選択により除外された変数の数は、HAR データでは ANNIGMA で 443、MFS で 254 変数であり、AAT データでは ANNIGMA で 208、MFS で 259 変数であった。加えて、各データセットに対するマルチ AE 識別器の変数選択の過程 (3.3 節における手順 2 から 6 の繰り返しステップ) における  $a_{all}$  の推移を図 4、図 5 に示す。なお、 $cnt_{max} = 4$  のため各々 15、14 ステップにて変数選択が終了するが、図 4、図 5 では確認のため、終了せずに変数選択を続行した場合における 25 ステップまでの推移を記載してある。

\*1 Percent point を意味する。

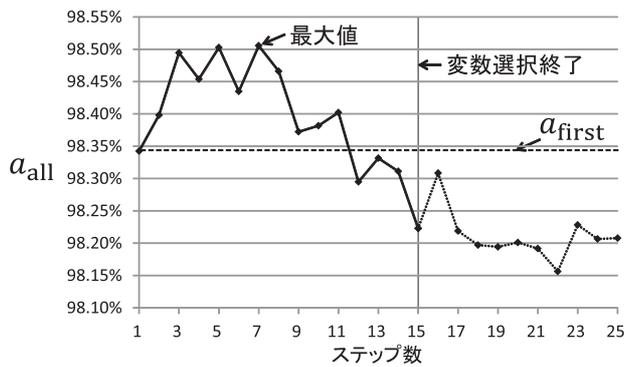


図 4 HAR データセットにおける各変数選択ステップの CV 識別精度

Fig. 4 The cross-validation accuracy of each variable selection step in HAR dataset.

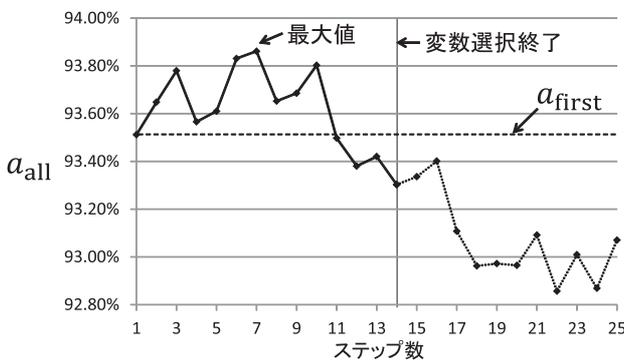


図 5 AAL データセットにおける各変数選択ステップの CV 識別精度

Fig. 5 The cross-validation accuracy of each variable selection step in AAL dataset.

表 5 より、両データセットともマルチ AE 識別器による変数選択により、テストデータに対する識別精度が向上したことが確認できる。また、表 6、表 7 より、クラスごとに異なる変数選択が実現できたことが分かる。さらに、表 5 より、マルチ AE 識別器の精度向上度合いは、クラスごとに変数選択できない両比較手法よりも大きくなることが確認できた。これより、マルチ AE 識別器によるクラスごとの変数選択は、実センサデータの識別精度向上に貢献できることが示された。

また、図 4、図 5 より、CV 識別精度が最大値となったステップはともに 7、変数選択が終了したステップがともに 10 数ステップと、全 561 変数に対して少ないステップで探索が終了することが分かった。さらに、終了後 10 ステップ以内に再度  $\alpha_{all}$  が  $\alpha_{first}$  を超えないことも確認できた。マルチ AE 識別器の探索終了までにかかったステップ数は、マルチ AE 識別器と同様の探索戦略をとる ANNIGMA において CV 識別精度が最大となったステップ数 443、および 208 と比較すると小さい値となっており、マルチ AE 識別器の再構成誤差を利用した SBE 方式の探索戦略が効率的であることを検証できた。

## 6. おわりに

本稿では、識別に悪影響を及ぼす変数を自動的に推定(変数選択)し、それら変数の影響を排除して学習および識別が可能な AE ベースの識別器であるマルチ AE 識別器を提案した。マルチ AE 識別器による変数選択では、変数ごとの再構成誤差を用いることで、識別精度を悪化させた原因の変数を高精度に導出可能であり、クラスごとに異なる変数選択が可能である。また、実データを用いた評価実験を行い、上記特性が有効に働くことを確認した。

今後の課題としては、画像データ等の今回の実験で用いたセンサデータとは異なる特性を持つデータセットへの適用、マルチ AE 識別器の各 AE のネットワーク構造を変化させた際の影響確認、クラス数が多いデータに対する計算コスト削減方法の提案があげられる。

## 参考文献

- [1] Guyon, I. and Elissee, A.: An introduction to variable and feature selection, *Journal of Machine Learning Research*, pp.1157–1182 (2003).
- [2] Kumar, V. and Minz, S.: Feature Selection: A literature Review, *Smart Computing Review* (2014).
- [3] Sheinvald, J., Dom, B. and Niblack, W.: A modelling approach to feature selection, *Proc. 10th International Conference on Pattern Recognition*, pp.535–539 (1990).
- [4] Tibshirani, R.: Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society, Series B (Methodological)*, pp.267–288 (1996).
- [5] Liaw, A. and Matthew, W.: Classification and regression by randomForest, *R news 2.3*, pp.18–22 (2002).
- [6] Ng, A.Y.: Feature selection, L1 vs. L2 regularization, and rotational invariance, *Proc. 21st International Conference on Machine Learning*, p.78, ACM (2004).
- [7] Bahdanau, D., Cho, K. and Bengio, Y.: Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).
- [8] Doak, J.: An evaluation of feature selection methods and their application to computer security, Technical Report, Davis CA: University of California, Department of Computer Science (1992).
- [9] May, R., Dandy, G. and Maier, H.: Review of input variable selection methods for artificial neural networks, *Artificial Neural Networks-methodological Advances and Biomedical Applications* (2011).
- [10] Hinton, G.E. and Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks, *Science*, Vol.313, No.5786, pp.504–507 (2006).
- [11] Liy, H. and Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining, *Boston: Kluwer Academic Publishers* (1998).
- [12] Setiono, R. and Liu, H.: Neural-network feature selector, *IEEE Trans. Neural Networks*, Vol.8, No.3, pp.654–662 (1997).
- [13] Verikas, A. and Bacauskiene, M.: Feature selection with neural networks, *Pattern Recognition Letters*, Vol.23, No.11, pp.1323–1335 (2002).
- [14] Mao, J., Mohiuddin, K. and Jain, A.K.: Parsimonious network design and feature selection through node pruning, *Proc. 12th IAPR International Conference on Pat-*

- tern Recognition*, pp.622–624 (1994).
- [15] Bowden, G.J., Nixon, J.B., Dandy, G.C., Maier, H.R. and Holmes, M.: Forecasting chlorine residuals in a water distribution system using a general regression neural network, *Mathematical and Computer Modelling*, pp.469–484 (2006).
- [16] Hsu, C.N., Huang, H.J. and Dietrich, S.: The ANNIGMA-wrapper approach to fast feature selection for neural nets, *IEEE Trans. Systems, Man, and Cybernetics*, Vol.32, No.2, pp.207–212 (2002).
- [17] Gasca, E., Sanchez, J.S. and Dietrich, S.: Eliminating redundancy and irrelevance using a new MLP-based feature selection method, *Pattern Recognition*, Vol.39, No.2, pp.313–315 (2006).
- [18] Kabir, M.M., Islam, M.M. and Murase, K.: A new wrapper feature selection approach using neural network, *Neurocomputing*, Vol.73, No.16–18, pp.3273–3283 (2010).
- [19] Lannoy, G., François, D. and Verleysen, M.: Class-specific feature selection for one-against-all multiclass svms, *European Symposium on Artificial Neural Networks*, pp.269–274 (2011).
- [20] Singh, P., Verma, A. and Chaudhari, N.S.: Devanagiri handwritten numeral recognition using feature selection approach, *International Journal of Intelligent Systems and Applications*, Vol.6, No.12 (2014).
- [21] Azmoodeh, A., Dehghantanha, A. and Choo, K.K.R.: Robust Malware Detection for Internet Of (Battlefield) Things Devices Using Deep Eigenspace Learning, *IEEE Trans. Sustainable Computing* (2018).
- [22] Wang, S., Ding, Z. and Fu, Y.: Feature Selection Guided Auto-Encoder, *Proc. AAAI Conference on Artificial Intelligence*, pp.2725–2731 (2017).
- [23] Han, K., Wang, Y., Zhang, C., Li, C. and Xu, C.: AutoEncoder Inspired Unsupervised Feature Selection, *Proc. International Conference on Acoustics, Speech and Signal Processing* (2018).
- [24] Manevitz, L. and Yousef, M.: One-class document classification via neural networks, *Neurocomputing*, Vol.70, No.7–9, pp.1466–1481 (2007).
- [25] Perera, P. and Patel, V.M.: Learning Deep Features for One-Class Classification, arXiv preprint arXiv:1801.05365 (2018).
- [26] Tagawa, T., Tadokoro, Y. and Yairi, T.: Structured denoising autoencoder for fault detection and analysis, *Asian Conference on Machine Learning*, pp.99–111 (2015).
- [27] Romero, E. and Sopena, J.M.: Performing feature selection with multilayer perceptrons, *IEEE Trans. Neural Networks*, Vol.19, No.3, pp.431–441 (2008).
- [28] Reyes-Ortiz, J.L., Oneto, L., Samà, A., Parra, X. and Anguita, D.: Transition-Aware Human Activity Recognition Using Smartphones, *Neurocomputing*, Vol.171, pp.754–767 (2016).
- [29] Anguita, D., Ghio, A., Oneto, L., Parra, X. and Reyes-Ortiz, J.L.: A Public Domain Dataset for Human Activity Recognition Using Smartphones, *21th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (2013).
- [30] Chainer: A flexible framework for neural networks (online), available from (<https://chainer.org/>) (accessed 2018-07-26).
- [31] Graves, A.: Generating sequences with recurrent neural networks, arXiv preprint arXiv:1308.0850 (2013).
- [32] Arpit, D., Zhou, Y., Ngo, H. and Govindaraju, V.: Why

regularized auto-encoders learn sparse representation?, *Proc. International Conference on Machine Learning*, pp.134–144 (2015).



石井 良尚 (正会員)

平成 21 年慶應大学理工学部卒業。平成 23 年同大学大学院修士課程修了。同年 (株) 豊田中央研究所入社。最適化手法, ソフトウェアの自動テスト, 機械学習を用いた異常検知の研究に従事。



高梨 昌樹

平成 13 年北海道大学工学部卒業。平成 15 年同大学大学院修士課程修了。平成 19 年同大学院博士課程修了。同年 (株) 豊田中央研究所入社。高分解能推定法, 超広帯域無線, 無線通信信号処理, 電力線通信, 機械学習の研究に従事。博士 (工学)。電子情報通信学会, IEEE 各会員。



三田 勝史

平成 10 年名古屋大学工学部情報工学科卒業。平成 12 年同大学大学院博士前期課程修了。同年 (株) 豊田中央研究所入社。平成 22 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。移动通信システムに関するデジタル信号処理の研究に従事。平成 18 年 IEEE Chester Sall Award 受賞。