International Journal of Production Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

# Deep embedding kernel mixture networks for conditional anomaly detection in high-dimensional data

Hyojoong Kim & Heeyoung Kim

**To cite this article:** Hyojoong Kim & Heeyoung Kim (2022): Deep embedding kernel mixture networks for conditional anomaly detection in high-dimensional data, International Journal of Production Research, DOI: 10.1080/00207543.2022.2027040

**To link to this article:** https://doi.org/10.1080/00207543.2022.2027040

Published online: 18 Feb 2022.

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
https://doi.org/10.1080/00207543.2022.2027040

Taylor & Francis
Taylor & Francis Group



# Deep embedding kernel mixture networks for conditional anomaly detection in high-dimensional data

Hyojoong Kim and Heeyoung Kim

Department of Industrial and Systems Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea

**ABSTRACT**

In various industrial problems, sensor data are often used to detect the abnormal state of manufacturing systems. Sensor data are sometimes influenced by contextual variables that are not related to the system health status and may exhibit different behaviours depending on their values, even if the system is in a normal condition. In this case, a conditional anomaly detection method should be used to consider the effects of contextual variables. In this study, we propose a conditional anomaly detection method, particularly for high-dimensional and complex data, using a deep embedding kernel mixture network. The proposed method comprises embedding and kernel mixture networks. The embedding network learns low-dimensional embeddings from high-dimensional data, and the kernel mixture network models the distribution of the learned embeddings conditional on contextual variables. The two networks enable a flexible estimation of conditional density using the high expressive power of deep neural networks. The two networks are trained simultaneously such that the high-dimensional data are embedded into a low-dimensional space, to assist conditional density estimation. The effectiveness of the proposed model is demonstrated using real data examples from the UCI repository and a case study from a tire company.

**ARTICLE HISTORY**
Received 2 July 2021
Accepted 1 January 2022

**KEYWORDS**
Autoencoder; conditional anomaly detection; conditional density estimation; high-dimensional data; kernel mixture network

## 1. Introduction

Owing to the advances in sensing technologies, sensor data are frequently used for anomaly detection for monitoring systems in industrial problems, and various methods have been proposed for the same, e.g. Kim, Huo, and Shi (2014), Jin et al. (2017), Kim et al. (2017), Liu et al. (2018, 2021), Kim and Kim (2018), Tan et al. (2019) and Ko and Kim (2019). Previous methods generally assume a single normal state and detect abnormalities by modelling the normal behaviour and testing the likelihood of observing new observations, or by predicting future observations and evaluating the prediction errors. However, sensor data are often affected by contextual variables (e.g. equipment setting values, product type, or material) irrespective of the system health. In this case, it is more reasonable to assume that the normal behaviour changes according to the state of the contextual variables. For example, in our case study, presented in Section 4, when a certain operation is performed on different products using the same machine, if the meter per minute (MPM) of the motor on the machine changes according to the equipment setting value, the patterns of vibration data change accordingly, although the system health remains normal. This change in normal patterns in accordance with the change in contextual variables renders it difficult to correctly detect anomalies using typical anomaly detection methods by assuming a single normal state, because distinguishing between a normal pattern formed by changes in the values of contextual variables and an abnormal pattern is challenging.

To address this problem, various methods have been proposed to detect the anomalies conditional on contextual variables. For example, a conditional anomaly detection technique was proposed by assuming that contextual data were generated from a Gaussian mixture model (GMM) and response data from another GMM (Song et al. 2007). Subsequently, the conditional distribution of each cluster of response variables given each cluster of contextual variables was estimated, and an anomaly was detected based on the likelihood of observing a new sample. Another study assumed multiple response variables and used a multivariate conditional model given contextual variables, which was used to measure the likelihood of observing a new sample for anomaly detection (Hong and Hauskrecht 2015). A method based on isolation forest (Liu, Ting, and Zhou 2008) was also proposed for anomaly detection conditional on data partitions that are defined on the basis of mixed-type attributes driven by

**CONTACT** Heeyoung Kim heeyoungkim@kaist.ac.kr

expert knowledge (Stripling et al. 2018). More recently, a GMM was used to model response data with consideration of categorical contextual variables, wherein the latent class variable was estimated using the distribution of contextual data (Ohkubo and Nagata 2019). Subsequently, an anomaly was detected based on the likelihood. A method that combines neural networks, low-dimensional representation learning, and noise contrastive estimation was also proposed to consider multiple high cardinality categorical features and estimate the conditional probability distribution (Amin, Garg, and Coskun 2019).

Although previous methods have demonstrated successful performances in many applications of conditional anomaly detection, they do not involve dimension reduction tools and thus could be ineffective in dealing with high-dimensionality and complex relationships in response data, which are typical issues encountered in sensor data in industrial problems. With the rapid development of sensor technology, high-dimensional sensor data can be easily collected by wireless sensor networks, image monitoring systems, and multimedia sensor networks (Deng et al. 2018; Koroniotis et al. 2019; Liu et al. 2021). In the literature, there have been many dimension reduction techniques including principal component analysis (PCA), linear discriminant analysis (LDA), autoencoder, and variational autoencoder (Schölkopf, Smola, and Müller 1998; Kingma and Welling 2014; Goodfellow, Bengio, and Courville 2016; Reddy et al. 2020; Ghahramani et al. 2020). Although PCA and LDA are simple and have shown good performances in many examples (Mika et al. 1999; Hoffmann 2007; Soh, Kim, and Yum 2018), they may not be effective in dealing with complex relationships. Recently, autoencoder and variational autoencoder, which are feature extraction methods based on deep neural networks, have shown powerful dimension reduction performances in many industrial problems (Ko and Kim 2019; Lee and Kim 2020; Hwang and Kim 2020; Kim, Shin, and Kim 2021). They can effectively capture a complex relationship among the variables in high-dimensional data using a deep neural network structure. Recently, for conditional anomaly detection, a variational autoencoder was used to learn low-dimensional embeddings of high-dimensional response data within each of multiple contextual groups assumed from expert knowledge (e.g. product ID) (Jabbar et al. 2019). Subsequently, a new sample is detected as an anomaly if the reconstruction error from the variational autoencoder is large. However, this method cannot be applied when the contextual groups are unknown.

In this study, we propose a new conditional anomaly detection method for complex and high-dimensional data. In particular, we assume that different contextual groups are unknown, and we only know that the normal pattern of response data varies in accordance with the change in contextual variables. We model the distribution of the normal pattern conditional on the contextual variables using a kernel mixture network (KMN) (Ambrogioni et al. 2017), which is a type of deep neural network that models the conditional density as a linear combination of kernel functions centered on training points, where the weights of the kernel functions are determined by the output of the KMN. KMNs allow the flexible estimation of the conditional density using the high expressive power of deep neural networks.

However, directly using the KMN for the conditional density estimation of sensor data may not result in good performances if the data are high dimensional, because kernel density estimation is a challenging task if data is high-dimensional (Nagler and Czado 2016; Wang and Scott 2019). To manage high-dimensional data effectively, we propose to learn low-dimensional embeddings from high-dimensional data and perform kernel density estimation on the low-dimensional embeddings. To learn the low-dimensional embeddings, we use the autoencoder. In our method, the low-dimensional embeddings and kernel density function are learned simultaneously, such that the high-dimensional data are embedded into a low-dimensional space to assist conditional density estimation. We name the proposed method as deep embedding kernel mixture network (DEKMN).

Finally, we detect anomalies using a predetermined threshold on the conditional density function estimated from the training data. That is, we train our model using normal samples in the training phase, and then determine whether a new observation is an anomalous or a normal sample in the detection phase. In this study, we assume a situation where only normal training data are available, because abnormal events rarely occur in practice. Our anomaly detection procedure is detailed in Section 2.2.

Our contributions are summarised as follows. First, we propose a new conditional anomaly detection method, DEKMN, for detecting anomalies in high-dimensional and complex response data affected by a set of contextual variables. Benefitting from the high expressive power of deep neural networks, DEKMN can flexibly estimate the conditional density of the complex response data using the KMN while dealing with the high-dimensionality using the autoencoder. Second, DEKMN performs feature extraction and kernel density estimation simultaneously, such that the features are distributed in a latent space in favour of kernel density estimation. Note that if feature extraction is performed independent of the kernel density estimation, important information necessary for kernel density estimation could be lost during

dimension reduction (Hwang and Kim 2020). The combination of KMN and autoencoder into a single model has not been previously studied. We train the DEKMN end-to-end to interactively learn low-dimensional features and kernel density function. The proposed model demonstrated effective anomaly detection performance in our experiments.

The remainder of this paper is organised as follows. Section 2 describes the proposed method. Section 3 evaluates the performance of the proposed method using several UCI datasets. Section 4 presents a case study of a tire company. Finally, the conclusions are provided in Section 5.

## 2. Proposed method

### 2.1. Deep embedding kernel mixture networks

The proposed model comprises two subnetworks: the KMN and the embedding network. Figure 1 shows the overall structure of the proposed model. The lower and upper large boxes represent the embedding network and the KMN, respectively. The embedding network learns the low-dimensional representations from high-dimensional data. The KMN uses contextual variables as inputs and learns the weights of the kernel functions. Given the learned low-dimensional embeddings and kernel weights, the proposed model estimates the conditional density function of the low-dimensional embeddings of high-dimensional sensor data given contextual variables, using weighted kernel density estimation.

More specifically, the embedding network has a structure same as that of a deep autoencoder (Goodfellow, Bengio, and Courville 2016) and provides low-dimensional representations of high-dimensional sensor

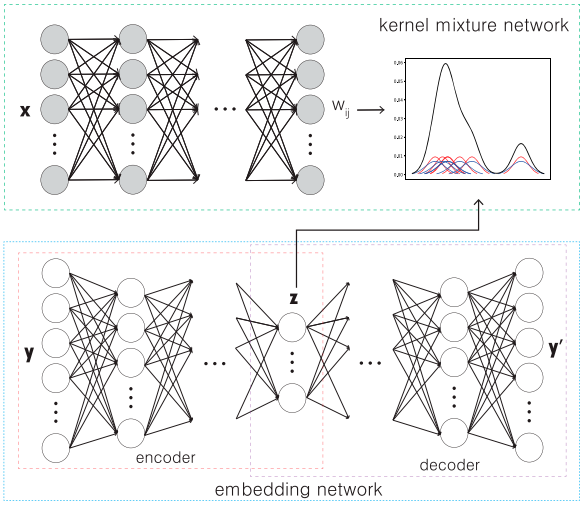



**Figure 1.** Structure of deep embedding kernel mixture networks.

data. The embedding network comprises an encoder and a decoder, which are represented by the dashed red and purple boxes, respectively, in Figure 1. Let $(X, Y)$ be a pair of random variables with realizations $\mathbf{x} \in \mathbb{R}^{D_x}$ representing the $D_x$ contextual variables, and $\mathbf{y} \in \mathbb{R}^{D_y}$ representing sensor data collected from $D_y$ sensors. The encoder $h(\mathbf{y}; \theta_e)$, a deep neural network parameterised by $\theta_e$, uses a high-dimensional observation $\mathbf{y}$ as an input and returns a low-dimensional representation $\mathbf{z} \in \mathbb{R}^{D_z}$ as the output, where $D_z < D_y$. The decoder $g(\mathbf{z}; \theta_d)$, a deep neural network parameterised by $\theta_d$, reconstructs $\mathbf{y}$ from $\mathbf{z}$. Let $\mathbf{y}' \in \mathbb{R}^{D_y}$ denote the reconstructed counterpart of $\mathbf{y}$. The embedding network is trained by minimising the reconstruction error, expressed as

$$L_1(\mathbf{y}, \mathbf{y}') = \|\mathbf{y} - \mathbf{y}'\|^2, \tag{1}$$

where $\|\cdot\|$ is the $L_2$ norm.

The embedding network is then combined with the KMN. The KMN models the conditional density as a combination of kernel functions centered on training points, where the weights of the kernel functions are modelled using a deep neural network that uses the contextual variables $\mathbf{x}$ as input. In particular, combined with the embedding network, the KMN performs conditional density estimation on the latent low-dimensional embeddings $\mathbf{z}$ given the contextual variables. We name this method as a deep embedding kernel mixture network (DEKMN). Using the DEKMN, the density function of $\mathbf{z}$ conditioned on $\mathbf{x}$ is expressed as

$$f(\mathbf{z} \mid \mathbf{x}) = \sum_{i=1}^{N} w_i(\mathbf{x}; \theta_w) K(\mathbf{z}, \mathbf{z}^{(i)}; \sigma), \tag{2}$$

where $\mathbf{z}^{(i)}$ is the low-dimensional representation of the $i$th sample; $i = 1, \ldots, N$, $N$ is the number of training samples; $K(\mathbf{z}, \mathbf{z}^{(i)}; \sigma)$ is the kernel function centered on $\mathbf{z}^{(i)}$ with bandwidth $\sigma$; $w_i(\mathbf{x}; \theta_w)$ is the weight for kernel $K(\mathbf{z}, \mathbf{z}^{(i)}; \sigma)$ and is parameterised by a deep neural network with parameter $\theta_w$. The weights $w_i(\mathbf{x}; \theta_w)$ must satisfy $w_i(\mathbf{x}; \theta_w) \geq 0, i = 1, \ldots, N$, and $\sum_{i=1}^{N} w_i(\mathbf{x}; \theta_w) = 1$ to represent a categorical distribution over the kernels. To ensure this, we use the softmax function in the output layer of the deep neural network for $w_i(\mathbf{x}; \theta_w)$:

$$w_i(\mathbf{x}; \theta_w) = \frac{\exp(o_i(\mathbf{x}))}{\sum_{i=1}^{N} \exp(o_i(\mathbf{x}))}, \tag{3}$$

where $o_i(\mathbf{x}) \in \mathbb{R}$ is the output of the fully connected layer immediately preceding the softmax layer. Unlike the usual kernel density estimation, the KMN adjusts the weights conditional on $\mathbf{x}$, instead of having the same weight for each kernel. This enables similar process conditions, having similar contextual variable values, to have

similar kernel weights, and different process conditions to have different kernel weights. Therefore, we can obtain a conditional kernel density function, which allows us to perform conditional anomaly detection. Moreover, the KMN allows the flexible density estimation of complex, high-dimensional sensor data using the expressive power of deep neural networks.

In this study, we use the Gaussian kernel expressed as

$$K(\mathbf{z}, \mathbf{z}'; \sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^{D_z}} \exp\left\{-\frac{\|\mathbf{z} - \mathbf{z}'\|^2}{2\sigma^2}\right\}, \quad (4)$$

Hence, Eq. (2) can be regarded as a GMM with $N$ components, each having $\mathbf{z}^{(i)}$ as the mean and $\sigma^2$ as the variance. The bandwidth (or standard deviation) $\sigma$ of the Gaussian kernel can be fixed.

In kernel density estimation, selecting an appropriate bandwidth is important, and various methods have been used to select the bandwidth. To avoid the bandwidth selection problem, multiple kernels per center, each with its own bandwidth, can be considered instead of optimising the bandwidth (Ambrogioni et al. 2017). The same approach is used in this study, and the model in Eq. (2) is extended to the following:

$$f(\mathbf{z} \mid \mathbf{x}) = \sum_{i=1}^{N} \sum_{j=1}^{J} w_{ij}(\mathbf{x}; \theta_w) K(\mathbf{z}, \mathbf{z}^{(i)}; \sigma_j), \quad (5)$$

where $J$ is the number of different bandwidths, and $w_{ij}(\mathbf{x}; \theta_w)$ is the weight of the kernel function centered on $\mathbf{z}^{(i)}$ with bandwidth $\sigma_j$. Similar to Eq. (3), the weights $w_{ij}$ are modelled using the softmax function such that $w_{ij} \geq 0$ and $\sum_{i=1}^{N} \sum_{j=1}^{J} w_{ij} = 1$. The loss function of the KMN is obtained using the negative log-likelihood for the model in Eq. (5) as follows:

$$L_2(\mathbf{x}, \mathbf{z}) = -\log \sum_{i=1}^{N} \sum_{j=1}^{J} w_{ij}(\mathbf{x}; \theta_w) K(\mathbf{z}, \mathbf{z}^{(i)}; \sigma_j), \quad (6)$$

Finally, the two subnetworks of the DEKMN are jointly trained to embed the sensor data in a low-dimensional space to assist conditional density estimation. Parameters $\theta_e, \theta_d$, and $\theta_w$ of the DEKMN are estimated by minimising the total loss function, which is a weighted sum of the Eqs.(1) and (6), as follows:

$$L = \sum_{n=1}^{N} L_1(\mathbf{y}^{(n)}, \mathbf{y}'^{(n)}) + \lambda \sum_{n=1}^{N} L_2(\mathbf{x}^{(n)}, \mathbf{z}^{(n)}), \quad (7)$$

where $\lambda$ is a regularisation parameter used to avoid either $L_1$ or $L_2$ dominating the other, and $n$ is the index of the training samples. To minimise the loss, we used the Adam optimiser (Kingma and Ba 2015).

### 2.2. Conditional anomaly detection

In general, anomaly detection methods based on density function estimation detect a new observation as an anomaly if the likelihood for this observation is lower than the predefined threshold. The threshold is typically defined as the critical value at a specified confidence level for a statistical hypothesis test (Huang et al. 2007; Casas, Mazel, and Owezarski 2011). In this study, we use only 'normal' samples to train the model. Once the proposed model is trained, we first measure the likelihood under the proposed model for all training samples. Subsequently, we set a specified percentile in the distribution of the likelihood as the threshold, depending on the desired anomaly rate. In our experiments (as presented in Sections 3 and 4), we set the fifth percentile in the distribution of the likelihood as the threshold, because the false alarm rate of 0.05 is typically used as a default value in practice. Subsequently, we consider a new observation as an anomaly if the likelihood for this observation based on the proposed model is lower than the predetermined threshold. The anomaly detection procedure of this study is summarised in Algorithm 1.

## 3. UCI data experiments

We evaluated the performance of the proposed model using the following datasets from the UCI repository (Dua and Graff 2019): the Abalone, Adult, Australian Credit, German Credit, Annthyroid, Cardiotocography, Heart Disease, and KDD99 datasets. Each dataset contains both categorical and numerical variables. In fact, these datasets are mainly used for classification problems. To use these datasets for anomaly detection problems, we regarded rare classes as anomalies, similar to those in Lu et al. (2016). Subsequently, the rare classes were excluded from the training datasets as we only used normal samples in the training phase. In addition, the categorical variables in each dataset were used as the contextual variables to provide multicategory contextual information by assuming multiple normal states depending on different values of the categorical variables. Table 1 summarises the configuration of the datasets used in the experiments.

We evaluated the anomaly detection performance of the proposed model. We compared the performance of the proposed model with those of the following competing models: conditional anomaly detection (CAD) (Song et al. 2007), deep autoencoding Gaussian mixture model (DAGMM) (Zong et al. 2018), mixture density network (MDN) (Bishop 2006), and KMN (Ambrogioni et al. 2017). A comparison with CAD verifies the performance of the proposed method over one of the representative conditional anomaly detection methods

**Algorithm 1** Anomaly detection procedure

**Training Phase**

**Input**: training normal samples $\mathbf{Y}_{\text{train}} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$, contextual variables $\mathbf{X}_{\text{train}} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, regularisation parameter $\lambda$

**Output**: likelihood of training samples, threshold $T$

1: Train the proposed model with $\mathbf{X}_{\text{train}}$ and $\mathbf{Y}_{\text{train}}$ by minimising the loss function in Eq. (7).

2: Calculate the likelihood under the trained model for all training samples using Eq. (5).
Set the threshold $T$ as the specified (e.g., fifth) percentile in the distribution of the likelihood.

**Detection Phase**

**Input**: new observation $\mathbf{y}_{\text{new}}$, contextual factors $\mathbf{x}_{\text{new}}$, $T$

3: Calculate the likelihood $l_{\text{new}}$ of $\mathbf{y}_{\text{new}}$ under the estimated proposed model.

4: **if** $l_{\text{new}} < T$ **then**
Determine $\mathbf{y}_{\text{new}}$ as an anomalous sample.

5: **else**
Determine $\mathbf{y}_{\text{new}}$ as a normal sample.

6: **end if**

**Table 1.** Description of datasets used in UCI data experiments.

| | Number of variables | | Number of instances | | |
|---|---|---|---|---|---|
| Dataset | Response variables (numerical) | Contextual variables (categorical) | Train | Test (normal) | Test (abnormal) |
| Abalone | 7 | 1 | 2000 | 709 | 126 |
| Adult | 4 | 7 | 23,000 | 11,543 | 3772 |
| Australian Credit | 6 | 8 | 400 | 225 | 65 |
| German Credit | 7 | 13 | 650 | 313 | 37 |
| Annthyroid | 6 | 15 | 3488 | 3178 | 250 |
| Cardiotocography | 20 | 13 | 1000 | 655 | 471 |
| Heart Disease | 5 | 22 | 115 | 50 | 25 |
| KDD99 | 31 | 83 | 9000 | 6000 | 1000 |

discussed in Section 1. The DAGMM is an unsupervised anomaly detection method that employs a GMM over low-dimensional embeddings learned by a deep autoencoder. The DAGMM was considered for comparison, to demonstrate the advantage of using contextual variables in the proposed model for conditional anomaly detection. The MDN is a type of deep neural network that models the conditional density as a GMM, where the weight and component parameters are parameterised by neural networks with contextual variables as inputs. Comparisons with the MDN and KMN demonstrate the advantage of using the latent low-dimensional embeddings instead of the original observations in the proposed model.

The proposed model was implemented using TensorFlow and trained using the Adam optimiser with a learning rate of 0.05. In addition, the regularisation parameter $\lambda$ in Eq. (7) was set to 0.5. Table 2 presents the network structure of the proposed model for each dataset. The network structures were optimised in a heuristic manner by evaluating the performance with various hyperparameter values. More specifically, we gradually increased the values of the hyperparameters (i.e. the number of layers, the number of nodes) until the loss no longer decreased significantly. A set of different bandwidths, [1.0, 0.75, 0.5, 0.25, 0.1], was used for Eq. (5). For the DAGMM, we assumed three mixture components for the GMM part and used an autoencoder structure same as that in the embedding network of our model. For the MDN, we assumed three mixture components for the GMM part and assumed three hidden layers, each with 50 nodes, for the neural network part. For the KMN, we used a structure same as that in the KMN component of our model.

We considered three performance measures as follows:

- Accuracy = number of correct assessments/number of all assessments,
- Specificity = number of true negatives/(number of true negatives + number of false positives),
- Sensitivity = number of true positives/(number of true positives + number of false negatives),

where true positives mean the anomalous samples that were correctly identified as anomalous, false positives

**Table 2.** Network structure of proposed model for UCI datasets.

| Dataset | Network | Layer | Input dimension | Activation | Output dimension |
|---|---|---|---|---|---|
| Abalone | Embedding network | Input | 7 | ReLU | 10 |
| | | 1st hidden | 10 | ReLU | 5 |
| | | 2nd hidden | 5 | ReLU | 10 |
| | | Output | 10 | ReLU | 7 |
| | Kernel mixture network | Input | 3 | ReLU | 50 |
| | | 1st hidden | 50 | ReLU | 50 |
| | | Output | 50 | Softmax | 100 |
| Adult | Embedding network | Input | 4 | ReLU | 4 |
| | | 1st hidden | 4 | ReLU | 3 |
| | | 2nd hidden | 3 | ReLU | 4 |
| | | Output | 4 | ReLU | 4 |
| | Kernel mixture network | Input | 83 | ReLU | 100 |
| | | 1st hidden | 100 | ReLU | 200 |
| | | 2nd hidden | 200 | ReLU | 500 |
| | | Output | 500 | Softmax | 500 |
| Australian Credit | Embedding network | Input | 6 | ReLU | 6 |
| | | 1st hidden | 6 | ReLU | 4 |
| | | 2nd hidden | 4 | ReLU | 4 |
| | | 3rd hidden | 4 | ReLU | 4 |
| | | 4th hidden | 4 | ReLU | 6 |
| | | output | 6 | ReLU | 6 |
| | Kernel mixture network | Input | 34 | ReLU | 50 |
| | | 1st hidden | 50 | ReLU | 100 |
| | | Output | 100 | Softmax | 100 |
| German Credit | Embedding network | Input | 7 | ReLU | 10 |
| | | 1st hidden | 10 | ReLU | 10 |
| | | 2nd hidden | 10 | ReLU | 4 |
| | | 3rd hidden | 4 | ReLU | 10 |
| | | 4th hidden | 10 | ReLU | 10 |
| | | Output | 10 | ReLU | 7 |
| | Kernel mixture network | Input | 52 | ReLU | 100 |
| | | 1st hidden | 100 | ReLU | 200 |
| | | 2nd hidden | 200 | ReLU | 250 |
| | | Output | 250 | Softmax | 250 |
| Anntyroid | Embedding network | Input | 6 | ReLU | 10 |
| | | 1st hidden | 10 | ReLU | 3 |
| | | 2nd hidden | 3 | ReLU | 10 |
| | | Output | 10 | ReLU | 6 |
| | Kernel mixture network | Input | 15 | ReLU | 100 |
| | | 1st hidden | 100 | ReLU | 250 |
| | | 2nd hidden | 250 | ReLU | 250 |
| | | Output | 250 | Softmax | 500 |
| Cardiotocography | Embedding network | Input | 20 | ReLU | 20 |
| | | 1st hidden | 20 | ReLU | 10 |
| | | 2nd hidden | 10 | ReLU | 4 |
| | | 3rd hidden | 4 | ReLU | 10 |
| | | 4th hidden | 10 | ReLU | 20 |
| | | Output | 20 | ReLU | 20 |
| | Kernel mixture network | Input | 13 | ReLU | 50 |
| | | 1st hidden | 50 | ReLU | 150 |
| | | 2nd hidden | 150 | ReLU | 200 |
| | | Output | 200 | Softmax | 200 |
| Heart disease | Embedding network | Input | 5 | ReLU | 5 |
| | | 1st hidden | 5 | ReLU | 3 |
| | | 2nd hidden | 3 | ReLU | 5 |
| | | Output | 5 | ReLU | 5 |
| | Kernel mixture network | Input | 22 | ReLU | 100 |
| | | 1st hidden | 100 | ReLU | 300 |
| | | 2nd hidden | 300 | ReLU | 500 |
| | | Output | 500 | Softmax | 575 |

(*continued*).

**Table 2.** Continued.

| Dataset | Network | Layer | Input dimension | Activation | Output dimension |
|---|---|---|---|---|---|
| KDD99 | Embedding network | Input | 31 | ReLU | 50 |
| | | 1st hidden | 50 | ReLU | 20 |
| | | 2nd hidden | 20 | ReLU | 4 |
| | | 3rd hidden | 4 | ReLU | 20 |
| | | 4th hidden | 20 | ReLU | 50 |
| | | Output | 50 | ReLU | 31 |
| | Kernel mixture network | Input | 83 | ReLU | 200 |
| | | 1st hidden | 200 | ReLU | 500 |
| | | 2nd hidden | 500 | ReLU | 500 |
| | | Output | 500 | Softmax | 1000 |

**Table 3.** Mean and standard error (in parentheses) of accuracy, specificity, and sensitivity of competing models for UCI datasets.

| | | Model | | | | |
|---|---|---|---|---|---|---|
| Dataset | Criteria | CAD | DAGMM | MDN | KMN | Proposed model |
| Abalone | Accuracy | 0.790 (0.013) | 0.915 (0.014) | 0.917 (0.018) | 0.923 (0.018) | **0.943** (0.016) |
| | Specificity | 0.831 (0.015) | 0.924 (0.017) | **0.946** (0.017) | 0.928 (0.021) | 0.944 (0.019) |
| | Sensitivity | 0.560 (0.046) | 0.870 (0.032) | 0.756 (0.036) | 0.899 (0.026) | **0.931** (0.020) |
| Adult | Accuracy | 0.812 (0.039) | 0.890 (0.028) | 0.879 (0.025) | 0.910 (0.022) | **0.928** (0.014) |
| | Specificity | **0.930** (0.048) | 0.903 (0.036) | 0.923 (0.033) | 0.912 (0.025) | 0.929 (0.018) |
| | Sensitivity | 0.528 (0.020) | 0.852 (0.039) | 0.725 (0.020) | 0.902 (0.023) | **0.917** (0.012) |
| Australian Credit | Accuracy | 0.704 (0.012) | 0.784 (0.017) | 0.762 (0.013) | 0.828 (0.016) | **0.896** (0.017) |
| | Specificity | 0.862 (0.020) | 0.920 (0.014) | 0.920 (0.017) | 0.939 (0.010) | **0.964** (0.013) |
| | Sensitivity | 0.159 (0.028) | 0.314 (0.034) | 0.214 (0.041) | 0.444 (0.049) | **0.664** (0.050) |
| German Credit | Accuracy | 0.822 (0.020) | 0.878 (0.022) | 0.896 (0.009) | 0.911 (0.011) | **0.948** (0.010) |
| | Specificity | 0.903 (0.021) | 0.925 (0.020) | 0.933 (0.007) | 0.941 (0.009) | **0.965** (0.009) |
| | Sensitivity | 0.136 (0.034) | 0.482 (0.048) | 0.582 (0.035) | 0.660 (0.039) | **0.803** (0.029) |
| Annthyroid | Accuracy | 0.791 (0.013) | 0.912 (0.014) | 0.915 (0.018) | 0.920 (0.018) | **0.935** (0.016) |
| | Specificity | 0.821 (0.046) | 0.912 (0.017) | 0.931 (0.017) | 0.919 (0.021) | **0.935** (0.019) |
| | Sensitivity | 0.551 (0.046) | 0.851 (0.032) | 0.742 (0.036) | 0.887 (0.026) | **0.921** (0.020) |
| Cardiotocography | Accuracy | 0.481 (0.021) | 0.651 (0.017) | 0.692 (0.018) | 0.623 (0.017) | **0.798** (0.011) |
| | Specificity | 0.528 (0.048) | 0.752 (0.024) | 0.781 (0.021) | 0.624 (0.021) | **0.821** (0.013) |
| | Sensitivity | 0.421 (0.024) | 0.512 (0.021) | 0.503 (0.031) | 0.621 (0.019) | **0.721** (0.016) |
| Heart Disease | Accuracy | 0.566 (0.019) | 0.636 (0.021) | 0.716 (0.022) | 0.779 (0.023) | **0.858** (0.019) |
| | Specificity | 0.735 (0.049) | 0.804 (0.048) | 0.892 (0.051) | 0.902 (0.061) | **0.924** (0.031) |
| | Sensitivity | 0.235 (0.061) | 0.308 (0.089) | 0.364 (0.095) | 0.538 (0.084) | **0.731** (0.051) |
| KDD99 | Accuracy | 0.855 (0.033) | 0.897 (0.028) | 0.908 (0.022) | 0.879 (0.030) | **0.917** (0.019) |
| | Specificity | **0.997** (0.010) | 0.985 (0.005) | 0.995 (0.001) | 0.988 (0.009) | 0.995 (0.003) |
| | Sensitivity | 0.128 (0.231) | 0.370 (0.193) | 0.391 (0.154) | 0.220 (0.221) | **0.451** (0.098) |

mean the normal samples that were incorrectly identified as anomalous, true negatives mean the normal samples that were correctly identified as normal, and false negatives mean the anomalous samples that were incorrectly identified as normal. For all competing models, we set the fifth percentile in the distribution of the likelihood in the training phase as the threshold, to determine anomalous samples in the detection phase.

For each dataset, we performed 10 experiments by randomly splitting the training and test datasets. Table 3 presents the average results with standard errors in parentheses. The best scores are indicated in bold. As shown, the proposed model performs best in terms of all performance measures for all datasets, except three cases: for the Abalone dataset, the MDN yielded the best specificity value, while for Adult and KDD99 datasets, the CAD yielded the best specificity values; however, they yielded significantly lower sensitivity values for the corresponding datasets. Table 4 summarises the average training time (in seconds) of the competing methods for each dataset with standard errors in parentheses. Although the proposed model required relatively longer training time, it is still computationally efficient to be used in practice.

Finally, to verify whether the contextual variables appropriately determined the weights of kernel functions in our model, we compared the distribution of the estimated kernel weights for each category of the contextual variables. Figure 2 illustrates the case for the Abalone dataset. Recall that the Abalone dataset has one categorical variable, and we used it as a contextual variable. It has three categories, and Figure 2 compares the distribution of the kernel weights for the three categories. In each subfigure, the $x$-axis represents the index of the kernel functions and $y$-axis represents the weight for each

**Table 4.** Training time of the competing methods for UCI data experiments (in seconds).

| | Model | | | | |
|---|---|---|---|---|---|
| Dataset | CAD | DAGMM | MDN | KMN | Proposed model |
| Abalone | 42.89 (3.35) | 68.49 (2.59) | 27.10 (4.05) | 48.31 (4.82) | 57.12 (3.49) |
| Adult | 69.49 (2.74) | 110.84 (2.87) | 55.05 (3.51) | 98.42 (4.02) | 92.12 (3.82) |
| Australian Credit | 38.53 (4.06) | 57.79 (4.40) | 27.66 (2.39) | 51.96 (3.00) | 49.45 (3.26) |
| German Credit | 39.44 (3.50) | 60.33 (2.66) | 28.87 (2.52) | 53.44 (3.45) | 50.13 (2.42) |
| Annthyroid | 36.81 (2.84) | 56.47 (3.96) | 26.52 (4.62) | 47.43 (3.87) | 46.12 (3.12) |
| Cardiotocography | 40.11 (3.72) | 60.26 (4.75) | 30.31 (4.49) | 54.62 (3.11) | 55.15 (3.84) |
| Heart Disease | 18.50 (3.17) | 27.90 (3.25) | 11.83 (3.79) | 25.15 (3.39) | 26.12 (4.65) |
| KDD99 | 51.30 (2.75) | 78.41 (4.80) | 39.72 (2.93) | 72.24 (3.64) | 71.23 (3.12) |

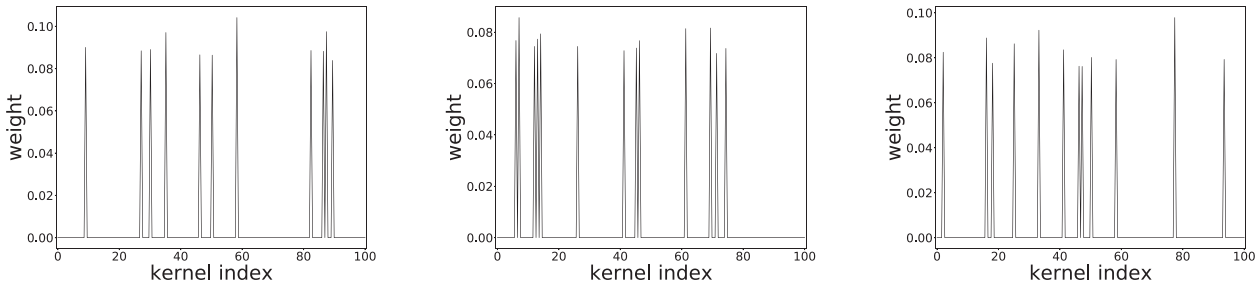



**Figure 2.** Distribution of kernel weights for each contextual variable category in Abalone dataset. (a) cluster 1, (b) cluster 2, (c) cluster 3.

kernel function. The distribution of the kernel weights (i.e. the kernel indices of nonzero weights) was exactly the same within the same category and differed significantly across different categories. This indicates that the kernel weights were appropriately determined depending on the contextual variables, which resulted in effective conditional anomaly detection.

## 4. Case study from tire company

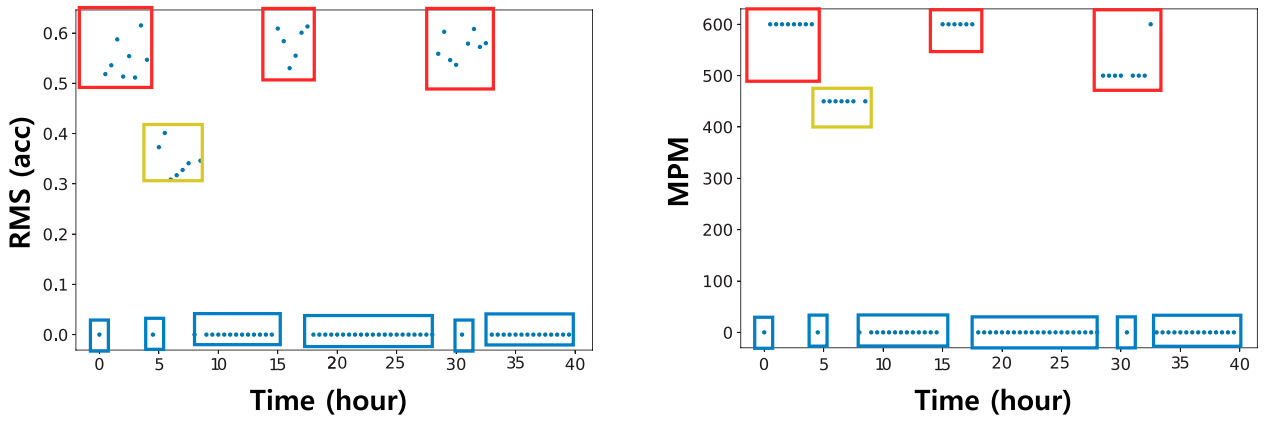
### 4.1. Motivation

Our study was motivated by a real-world example from a tire company, where high-dimensional sensor data were collected from many sensors and used to monitor the health condition of the manufacturing machine. For monitoring, the sensor data points collected over a certain time period are often summarised as the root mean square (RMS). Figure 3(a) illustrates the RMS of the acceleration values obtained from an accelerator attached to the calendering machine when the machine is in good condition. Although the machine is in good condition, we observed that the RMS values varied over time. Interestingly, as illustrated in Figure 3(b), the meter per minute (MPM) values of a motor of the same calendering machine varied over time with the same trend as the RMS, and can be categorised into three groups, as shown in the red, yellow, and blue boxes. In fact, the MPM values were not observed values from the motor; instead, they were set manually based on expert knowledge, according to various factors, such as cord type (e.g. steel cord or nylon cord) and amount of loaded rubber. For example, when the machine was idle, the MPM values were set to zero (blue box). We observed that these different MPM setting values resulted in different RMS values for even the normal state.

In Figure 3, if an anomaly occurs in the state indicated in yellow and the RMS values increase, the resulting RMS pattern may be similar to the normal pattern in the state indicated in red, which has relatively larger RMS values. Therefore, it is difficult to determine whether an anomaly has occurred. However, if we can use the state information (i.e. states indicated by red or yellow), we can easily determine whether an anomaly has occurred with respect to the state in yellow, or whether the condition is normal with respect to the state in red. This motivated us to study conditional anomaly detection.

### 4.2. Data description

We applied the proposed model to a dataset provided by a global tire company. The dataset contained acceleration data obtained from 42 accelerometers attached to a calendering machine, acquired at a frequency of 8192 Hz for 1 $\sim$ 4 s every 30 min. The 8192 vibration data points collected every 30 min were summarised as a single data point, i.e. the RMS. A total of 700 normal samples (i.e.700 RMS data points) were used for training. The contextual variables were 30 manipulable setting parameters for the calendering machine, including the MPM of the




(a) RMS over time (b) MPM over time

**Figure 3.** Left panel shows RMS of raw acceleration observed from the sensor of calendering machine. Right panel shows MPM values of a motor of the same calendering machine during the same interval as in the left panel.

calendering machine motor, input rubber temperature, and angle and gap between the rollers of the calendering machine.

Our dataset contained only normal samples. To evaluate the performance of the proposed model in detecting anomalies, we generated abnormal samples for testing. The abnormal samples were generated using the anomalous observations of the same calendering machine during different time periods for which the contextual variable data were not collected. We discovered that it was difficult to distinguish between normal and abnormal observations under a time series, but the difference became more evident when we applied wavelet transform to the data and focussed on the wavelet coefficients. We discovered that the detail coefficients at level 2 for normal and abnormal observations differed significantly when 7-level Haar wavelet transform was used.

Based on this observation, we generated abnormal samples by performing 7-level Haar wavelet transform of the original normal time series data. After performing the wavelet transform of the normal data, we randomly selected $4 \sim 6$ indices for the detail coefficients at level 2 and added $0.1\times$ the maximum of all approximation coefficients to the selected detail coefficients such that the resulting detail coefficients were similar to those of real abnormal samples. Subsequently, we reconstructed the abnormal time series using the changed coefficients. Figure 4(a) illustrates a normal time series and Figure 4(b) illustrates an abnormal time series based on the normal time series in Figure 4(a) using the generation approach explained above. Figures 4(c) and 4(d) depict the detail coefficients at level 2 of the 7-level Haar wavelet transform of the time series in Figures 4(a) and 4(b), respectively. We excluded the normal observations that were used to generate abnormal samples in the training dataset to avoid the duplicate use of those observations.

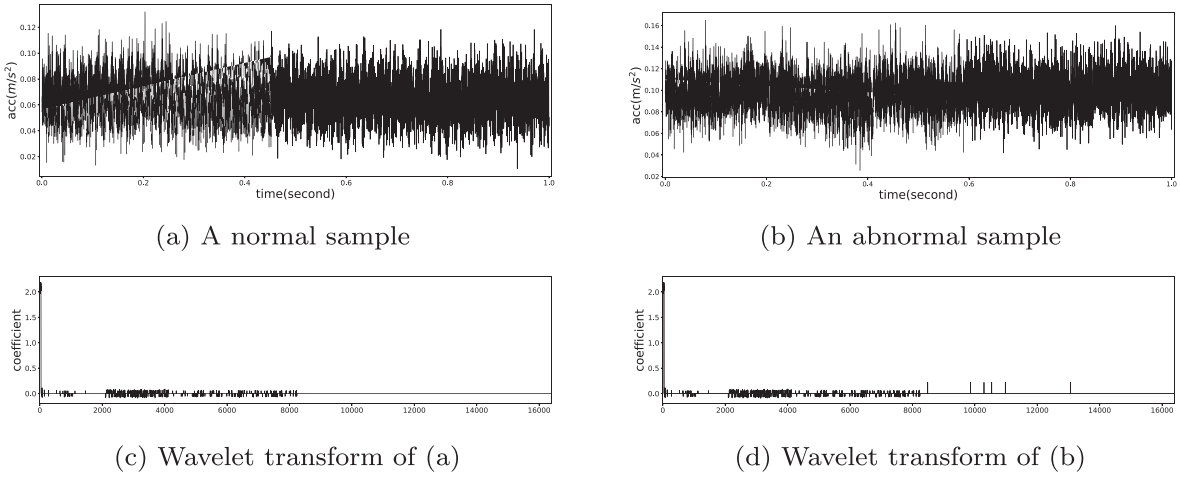
**Table 5.** Network structure of proposed model for calendering machine dataset.

| Network | Layer | Input dim. | Activation | Output dim. |
|---|---|---|---|---|
| Embedding network | Input | 336 | ReLU | 100 |
| | 1st hidden | 100 | ReLU | 50 |
| | 2nd hidden | 50 | ReLU | 20 |
| | 3rd hidden | 20 | ReLU | 5 |
| | 4th hidden | 5 | ReLU | 20 |
| | 5th hidden | 20 | ReLU | 50 |
| | 6th hidden | 50 | ReLU | 100 |
| | Output | 100 | ReLU | 336 |
| Kernel mixture network | Input | 30 | ReLU | 100 |
| | 1st hidden | 100 | ReLU | 500 |
| | 2nd hidden | 500 | ReLU | 500 |
| | Output | 500 | softmax | 500 |

### 4.3. Results

The network structure of the proposed model is presented in Table 5. The bandwidths of the kernel function, learning rate, and regularisation parameter were the same as in the settings of the simulated examples. In the training phase, 700 normal samples were used. In the testing phase, 300 normal samples and 50 abnormal samples were used to evaluate the performance of the model.

First, we evaluated the density estimation performance of the proposed model using the test data. We computed the likelihood of the normal and abnormal samples using Eq. (5). On average, the likelihood of the normal samples was 0.83 (standard error 0.031), whereas that of the abnormal samples was 0.287 (standard error 0.389). It was clear that the likelihood for the normal samples was significantly higher than that for the abnormal samples, indicating that the density function for the




(a) A normal sample (b) An abnormal sample

(c) Wavelet transform of (a) (d) Wavelet transform of (b)

**Figure 4.** Illustration of normal and abnormal samples in upper row and their detail coefficients at level 2 of 7-level Haar wavelet transform in lower row.

**Table 6.** Mean and standard error (in parentheses) of accuracy, specificity, and sensitivity of competing models.

| Model | Accuracy | Specificity | Sensitivity |
|---|---|---|---|
| CAD | 0.225 (0.003) | 0.096 (0.003) | **1** (0.000) |
| DAGMM | 0.941 (0.007) | 0.972 (0.005) | 0.752 (0.050) |
| MDN | 0.845 (0.019) | 0.828 (0.022) | 0.944 (0.017) |
| KMN | 0.795 (0.037) | 0.803 (0.043) | 0.744 (0.051) |
| Proposed model | **0.972** (0.007) | **0.973** (0.007) | 0.962 (0.014) |

normal samples was estimated well using the proposed model.

Next, we evaluated the anomaly detection performance of the proposed model and compared it with those of the competing models considered in Section 3. For all competing models, we set the fifth percentile in the distribution of the likelihood in the training phase as the threshold for determining anomalous samples in the detection phase. We performed 10-fold cross-validation. Table 6 shows a comparison of the results of the CAD, DAGMM, MDN, KMN, and the proposed model. The best result for each performance measure is indicated in bold face. It was evident that the proposed model outperformed the other methods in terms of accuracy and specificity. In terms of sensitivity, CAD yielded the perfect score; however, it demonstrated a significantly lower accuracy and specificity. In fact, CAD, a non neural-network based model, demonstrated a significantly lower performance than other models based on neural networks. This indicates that neural networks are advantageous in managing high-dimensional data. The DAGMM demonstrated good performances in terms of accuracy and specificity; however, its performance in terms of sensitivity was significantly lower than those of others. Furthermore, by comparing the KMN and the proposed model, we discovered that using the low-dimensional representations instead of the original data resulted in significant improvements in the performance measures. The average computation time (in seconds) for each method is presented in Table 7. Although the proposed model required relatively longer computation time, it is still computationally efficient to be used in practice.

Finally, to investigate whether the kernel weights of each observation were estimated adaptively, depending on the contextual variables, we compared the estimated kernel weights of observations with different MPM values, indicated in three different colors in Figure 3. Figure 5 illustrates the estimated kernel weights of six observations, two per each color of red, yellow, and blue. As shown, the weights were similar for observations with similar MPM values, whereas they differed significantly for observations with significantly different MPM values, as expected. Thus, this enabled successful conditional anomaly detection.

## 5. Conclusion

We proposed a method for detecting anomalies in high-dimensional and complex response data affected by contextual variables. The proposed method uses an autoencoder to learn low-dimensional embeddings from high-dimensional response data. The learned embeddings are

**Table 7.** Training time of the competing methods for the case study (in seconds).

| | CAD | DAGMM | MDN | KMN | Proposed model |
|---|---|---|---|---|---|
| Training time | 19.05 (3.04) | 27.16 (2.59) | 11.54 (5.03) | 20.24 (4.05) | 23.12 (2.16) |

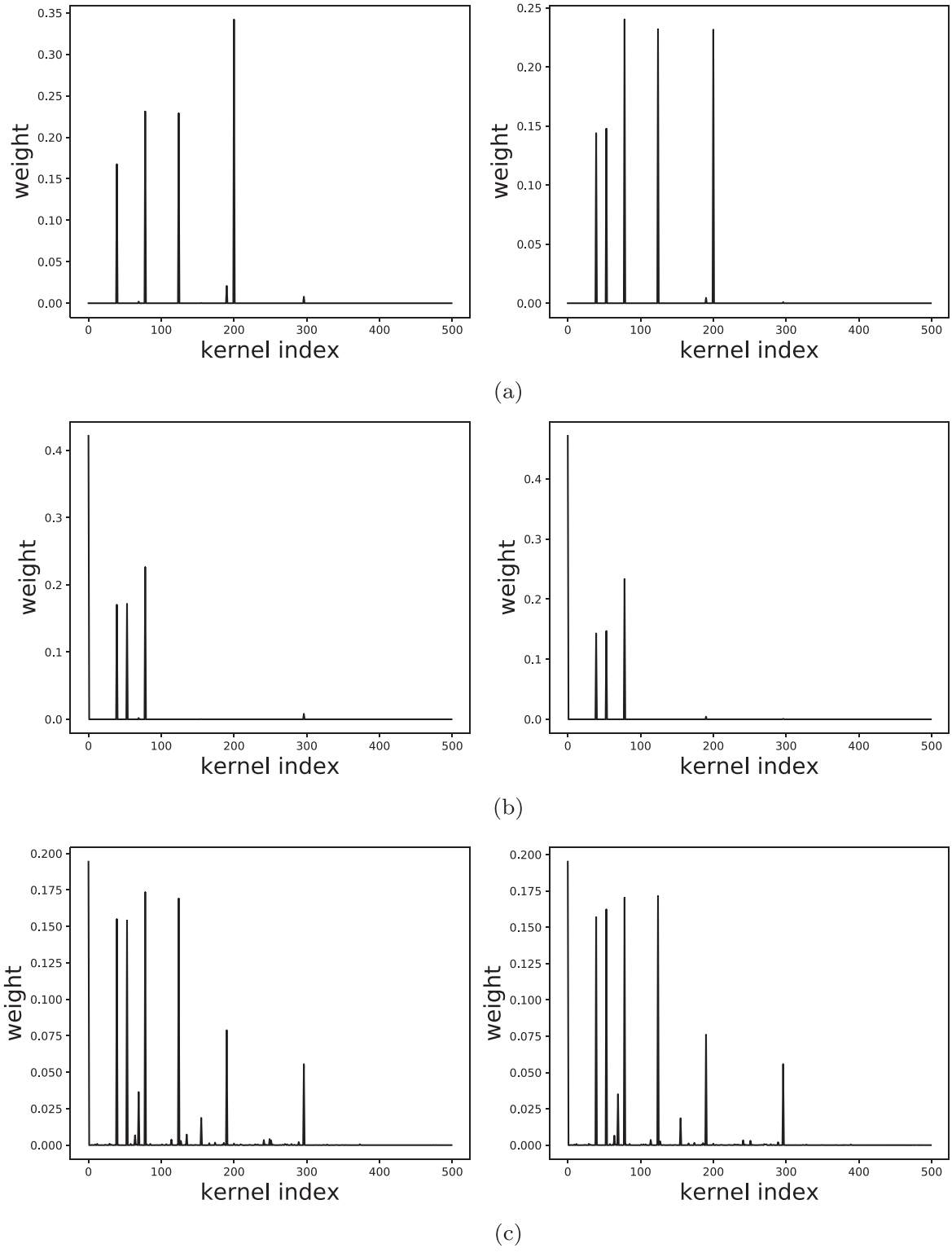



**Figure 5.** Illustration of estimated kernel weights of observations with different MPM values, indicated in different colors of red (first row), yellow (second row), and blue (third row) in Figure 3. (a) cluster 1, (b) cluster 2, (c) cluster 3.

then fed into the KMN. The KMN models the density of the learned embeddings conditional on the contextual variables, where the weights for the kernel functions are determined by the output of the KMN. Benefitting from the high expressive power of deep neural networks, the autoencoder and KMN allow flexibility in dimension reduction and density estimation, respectively. After training the model using normal observations, we detected anomalies using a predetermined threshold on the estimated conditional density function. We trained the proposed model in an end-to-end manner such that low-dimensional embeddings and conditional density functions were learned interactively. Our results indicated that the proposed model can learn embeddings that are suitable for conditional density estimation. The experimental results demonstrated that the proposed model performed significantly better than the other competing methods in terms of conditional anomaly detection. In this study, we did not consider the autocorrelation of time series data. In future studies, the proposed model may be extended to consider autocorrelation.

## Acknowledgements

The authors would like to thank the referees, the associate editor, and the editor for reviewing this article and providing valuable comments.

## Data availability statement

The datasets in Section 3 are available at http://archive.ics.uci.edu/ml, and the dataset in Section 4 is not publicly available due to confidentiality.

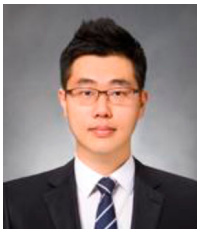
## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) [grant numbers 2018R1C1B6004511, 2020R1A4A10187747].

## Notes on contributors



*Hyojoong Kim* received a B.S. degree in industrial engineering from Hanyang University in Korea and an M.S. degree in industrial and systems engineering from KAIST. He is currently a PhD candidate in industrial and system engineering at KAIST. His research interests include machine learning and applied statistics.

*Heeyoung Kim* received a B.S. degree in industrial engineering from KAIST, an M.S. degrees in statistics and industrial engineering from the Georgia Institute of Technology and KAIST, and a PhD degree in industrial engineering from the Georgia Institute of Technology. She is an associate professor with the Department of Industrial and Systems Engineering, KAIST. She was a Senior Member of Technical Staff with AT&T Laboratories. Her research interests include applied statistics and machine learning

## References

Ambrogioni, Luca, Umut Güçlü, Marcel A. J. van Gerven, and Eric Maris. 2017. "The Kernel Mixture Network: A Nonparametric Method for Conditional Density Estimation of Continuous Random Variables." Preprint arXiv:1705.07111

Amin, Mohammad Ruhul, Pranav Garg, and Baris Coskun. 2019. "CADENCE: Conditional Anomaly Detection for Events using Noise-contrastive Estimation." In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, London, United Kingdom, 71–82.

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Casas, Pedro, Johan Mazel, and Philippe Owezarski. 2011. "Unada: Unsupervised Network Anomaly Detection using Sub-space Outliers Ranking." In *International Conference on Research in Networking*, Valencia, Spain, 40–51. Springer.

Deng, Xiaowu, Peng Jiang, Xiaoning Peng, and Chunqiao Mi. 2018. "Support High-order Tensor Data Description for Outlier Detection in High-dimensional Big Sensor Data." *Future Generation Computer Systems* 81: 177–187.

Dua, Dheeru, and Casey Graff. 2019. *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml

Ghahramani, Mohammadhossein, Yan Qiao, MengChu Zhou, Adrian O. Hagan, and James Sweeney. 2020. "AI-based Modeling and Data-driven Evaluation for Smart Manufacturing Processes." *IEEE/CAA Journal of Automatica Sinica* 7 (4): 1026–1037.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Cambridge, MA: MIT Press. http://www.deeplearningbook.org.

Hoffmann, Heiko. 2007. "Kernel PCA for Novelty Detection." *Pattern Recognition* 40 (3): 863–874.

Hong, Charmgil, and Milos Hauskrecht. 2015. "MCODE: Multivariate Conditional Outlier Detection." Preprint arXiv: 1505.04097

Huang, Ling, XuanLong Nguyen, Minos Garofalakis, Michael I. Jordan, Anthony Joseph, and Nina Taft. 2007. "In-network PCA and Anomaly Detection." In *Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, 617–624.

Hwang, Jonghyun, and Heeyoung Kim. 2020. "Variational Deep Clustering of Wafer Map Patterns." *IEEE Transactions on Semiconductor Manufacturing* 33 (3): 466–475.

Jabbar, Eva, Philippe Besse, Jean-Michel Loubes, and Christophe Merle. 2019. "Conditional Anomaly Detection for Quality and Productivity Improvement of Electronics Manufacturing Systems." In *International Conference on*

*Machine Learning, Optimization, and Data Science*, Tuscany, Italy, 711–724. Springer.

Jin, Yan, Shuai Huang, Guan Wang, and Houtao Deng. 2017. "Diagnostic Monitoring of High-dimensional Networked Systems Via a LASSO-BN Formulation." *IISE Transactions* 49 (9): 874–884.

Kim, Heeyoung, Xiaoming Huo, and Jianjun Shi. 2014. "A Single Interval Based Classifier." *Annals of Operations Research* 216 (1): 307–325.

Kim, Hyojoong, and Heeyoung Kim. 2018. "Functional Logistic Regression with Fused Lasso Penalty." *Journal of Statistical Computation and Simulation* 88 (15): 2982–2999.

Kim, Keunseo, JunCheol Shin, and Heeyoung Kim. 2021. "Locally Most Powerful Bayesian Test for Out-of-Distribution Detection using Deep Generative Models." In *Advances in Neural Information Processing Systems*, 34.

Kim, Keunseo, Hengameh Zabihi, Heeyoung Kim, and Uichin Lee. 2017. "TrailSense: A Crowdsensing System for Detecting Risky Mountain Trail Segments with Walking Pattern Analysis." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1 (3): 1–31.

Kingma, Diederik P., and Jimmy Ba. 2015. "Adam: A Method for Stochastic Optimization." In *International Conference on Learning Representations*, San Diego, CA, USA.

Kingma, Diederik P., and Max Welling. 2014. "Auto-encoding Variational Bayes." In *International Conference on Learning Representations*, Banff, AB, Canada.

Ko, Taeyoung, and Heeyoung Kim. 2019. "Fault Classification in High-dimensional Complex Processes Using Semi-supervised Deep Convolutional Generative Models." *IEEE Transactions on Industrial Informatics* 16 (4): 2868–2877.

Koroniotis, Nickolaos, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. 2019. "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-iot Dataset." *Future Generation Computer Systems* 100: 779–796.

Lee, Hyuck, and Heeyoung Kim. 2020. "Semi-supervised Multi-label Learning for Classification of Wafer Bin Maps with Mixed-type Defect Patterns." *IEEE Transactions on Semiconductor Manufacturing* 33 (4): 653–662.

Liu, Jie, Jianlin Guo, Philip Orlik, Masahiko Shibata, Daiki Nakahara, Satoshi Mii, and Martin Takáč. 2018. "Anomaly Detection in Manufacturing Systems using Structured Neural Networks." In *2018 13th World Congress on Intelligent Control and Automation (WCICA)*, Changsha, China, 175–180. IEEE.

Liu, Chenang, Zhenyu Kong, Suresh Babu, Chase Joslin, and James Ferguson. 2021. "An Integrated Manifold Learning Approach for High-dimensional Data Feature Extractions and Its Applications to Online Process Monitoring of Additive Manufacturing." *IISE Transactions* 53 (11): 1215–1230.

Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. 2008. "Isolation Forest." In *2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, 413–422. IEEE.

Lu, Yen-Cheng, Feng Chen, Yating Wang, and Chang-Tien Lu. 2016. "Discovering Anomalies on Mixed-type Data Using a Generalized Student-*t* Based Approach." *IEEE Transactions on Knowledge and Data Engineering* 28 (10): 2582–2595.

Mika, Sebastian, Bernhard Schölkopf, Alex J. Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. 1999. "Kernel PCA and De-noising in Feature Spaces." In *Advances in Neural Information Processing Systems*, Denver, Colorado, USA, 536–542.

Nagler, Thomas, and Claudia Czado. 2016. "Evading the Curse of Dimensionality in Nonparametric Density Estimation with Simplified Vine Copulas." *Journal of Multivariate Analysis* 151: 69–89.

Ohkubo, Masato, and Yasushi Nagata. 2019. "Conditional Anomaly Detection Based on a Latent Class Model." *Total Quality Management & Business Excellence* 30 (sup1): S227–S239.

Reddy, G. Thippa, M. Praveen Kumar Reddy, Kuruva Lakshmanna, Rajesh Kaluri, Dharmendra Singh Rajput, Gautam Srivastava, and Thar Baker. 2020. "Analysis of Dimensionality Reduction Techniques on Big Data." *IEEE Access* 8: 54776–54788.

Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller. 1998. "Nonlinear Component Analysis As a Kernel Eigenvalue Problem." *Neural Computation* 10 (5): 1299–1319.

Soh, Woojin, Heeyoung Kim, and Bong-Jin Yum. 2018. "Application of Kernel Principal Component Analysis to Multi-characteristic Parameter Design Problems." *Annals of Operations Research* 263 (1–2): 69–91.

Song, Xiuyao, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. 2007. "Conditional Anomaly Detection." *IEEE Transactions on Knowledge and Data Engineering* 19 (5): 631–645.

Stripling, Eugen, Bart Baesens, Barak Chizi, and Seppe van den Broucke. 2018. "Isolation-based Conditional Anomaly Detection on Mixed-attribute Data to Uncover Workers' Compensation Fraud." *Decision Support Systems* 111: 13–26.

Tan, Yingshui, Baihong Jin, Alexander Nettekoven, Yuxin Chen, Yisong Yue, Ufuk Topcu, and Alberto Sangiovanni-Vincentelli. 2019. "An Encoder-decoder based Approach for Anomaly Detection with Application in Additive Manufacturing." In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Boca Raton, FL, USA, 1008–1015. IEEE.

Wang, Zhipeng, and David W. Scott. 2019. "Nonparametric Density Estimation for High-dimensional Data–Algorithms and Applications." *Wiley Interdisciplinary Reviews: Computational Statistics* e1461: 1–41.

Zong, Bo, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection." In *International Conference on Learning Representations*, Vancouver, BC, Canada.