

# Shapley Values of Reconstruction Errors of PCA for Explaining Anomaly Detection

Naoya Takeishi

Center for Advanced Intelligence Project

RIKEN

Tokyo, Japan

naoya.takeishi@riken.jp

**Abstract**—We present a method to compute the Shapley values of reconstruction errors of principal component analysis (PCA), which is particularly useful in explaining the results of anomaly detection based on PCA. Because features are usually correlated when PCA-based anomaly detection is applied, care must be taken in computing a value function for the Shapley values. We utilize the probabilistic view of PCA, particularly its conditional distribution, to exactly compute a value function for the Shapley values. We also present numerical examples, which imply that the Shapley values are advantageous for explaining detected anomalies than raw reconstruction errors of each feature.

**Index Terms**—anomaly detection, fault detection and isolation, interpretable machine learning, Shapley values, principal component analysis

## I. INTRODUCTION

Anomaly detection based on machine learning has been actively studied and now plays an important role in various industrial applications such as fraud detection in finance [1], intrusion detection [2], and fault detection of mechanical systems [3]. Up to date, there have been proposed many types of anomaly detection algorithms based on different assumptions and technical principles (see, e.g., [4]–[6]).

In this work, we focus on the so-called *unsupervised anomaly detection* [4] problems, wherein the reference data used for learning a model contain only nominal, non-anomalous states of a target system. Moreover, we concentrate on the use of *reconstruction errors* as the measure of deviation from the nominal status. In fact, unsupervised anomaly detection using reconstruction errors is a popular technical setting. For example, reconstruction errors of principal component analysis (PCA) and autoencoders (and their variants) have been utilized in various applications (see, e.g., [7]–[10] and the references therein).

Reconstruction errors are used for not only detection but also explaining anomalies, to some extent, in practice. Now, let us suppose that we are monitoring the reconstruction errors of data comprising multiple features. Let  $\mathbf{x} = [x_1 \cdots x_d]^T \in \mathbb{R}^d$  be a data instance (with  $d$  features) and  $\hat{\mathbf{x}}$  be a reconstruction of  $\mathbf{x}$ . Then, the reconstruction error  $e(\mathbf{x})$  is defined by

$$e(\mathbf{x}) = \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 = \sum_{i=1}^d (\hat{x}_i - x_i)^2. \quad (1)$$

As  $e(\mathbf{x})$  is the sum of the reconstruction errors of the features,  $x_1, \dots, x_d$ , if all the features are scaled appropriately, we

may anticipate that a feature having larger reconstruction errors is more suspicious. However, is it truly explaining the features' contribution to the total reconstruction error,  $e(\mathbf{x})$ ? Not necessarily so it would be. A large error on a feature,  $x_i$ , may stem from anomalous behavior of another feature,  $x_j$ . Thus, by simply looking at errors of each feature, we may fail to localize detected anomalies.

This preliminary work focused an alternative mean to explain reconstruction errors for localizing detected anomalies. In particular, we introduce a way to compute the *Shapley values* [11] of reconstruction errors. The Shapley value is a notion originally used to distribute a gain of a coalitional game to the players. It has been utilized also for explaining outputs of machine learning models [12]–[14]. The Shapley value would be useful for localizing detected anomalies, too. In an analogy to a game, the raw reconstruction error of each feature is (a part of) the final gain of a game possessed separately by each player (i.e., feature) and not necessarily represents the true contribution of each player; e.g., the gain one possesses may be generated by another. In contrast, the Shapley value is known as a quantity that has desirable properties to measure the contribution generated by each player.

In this paper, we show how a value function for the Shapley values of reconstruction errors, particularly for PCA, is exactly computed using the probabilistic view of PCA. Moreover, we provide numerical examples, which exhibit the potential utility of the Shapley values of reconstruction errors to explain anomaly detection.

## II. BACKGROUND

### A. Anomaly Detection Using Reconstruction Errors

One of the most popular kinds of unsupervised anomaly detection methods utilizes the *reconstruction errors*, which are the deviation between test data and the corresponding reconstructions computed by a model trained on a reference dataset. Below we summarize the abstract procedures of such methods and examples of popular methodologies.

Suppose that a dataset is a set of column vectors  $\mathbf{x} \in \mathbb{R}^d$ . As the first step of anomaly detection, we learn a model  $(\mathbf{f}, \mathbf{g})$  on a dataset that contains no (or few) anomalies. Here,  $\mathbf{f}: \mathbb{R}^d \rightarrow \mathbb{R}^p$  (usually  $d > p$ ) is a map (so-called encoder) from the original data space to some latent compressed representation space, and  $\mathbf{g}: \mathbb{R}^p \rightarrow \mathbb{R}^d$  is a map (decoder) that approximately

restores the original data from the compressed representations. After learning  $\mathbf{f}$  and  $\mathbf{g}$ , we compute the reconstruction error<sup>1</sup> on a new data-point  $\mathbf{x}$  using Eq. (1) with  $\hat{\mathbf{x}} = \mathbf{g}(\mathbf{f}(\mathbf{x}))$ . If  $e(\mathbf{x})$  is small, it is likely that the new data-point  $\mathbf{x}$  does not deviate from the training data, and thus it is regarded as normal. In contrast, if  $e(\mathbf{x})$  is large, we will suspect that some features of the new data-point  $\mathbf{x}$  are anomalous.

Among many choices of the model,  $(\mathbf{f}, \mathbf{g})$ , a linear model or its mixtures are popular because of their simplicity, computational efficiency, and stability. Particularly, PCA [see, e.g., 15] is a well-known classical method for computing reconstruction errors. In this work, we focus on the use of PCA for anomaly detection, so we review PCA in the next subsection. Of course, a linear model may be insufficient to capture characteristics of data in practice. In such cases, a popular alternative is to use autoencoders, which are a neural network model that can compress and approximately restore data. For the use of autoencoders or their variants for anomaly detection, see, e.g., Zhou *et al.* [9] and Zong *et al.* [10]. Finally, we note that there is a recent study by Antwarg *et al.* [16] in which they try to explain the reconstruction of autoencoders using a method related to the one used in our work. We discuss the difference between their work and ours in Section V-A.

### B. Principal Component Analysis

In this work, we focus on the anomaly detection using the reconstruction errors computed via PCA [see, e.g., 15], in which both an encoder and a decoder are linear models. For the convenience in the main part of this work, we adopt the probabilistic view of PCA [17]. In this view, the data  $\mathbf{x} \in \mathbb{R}^d$  and the compressed representations (i.e., principal component scores)  $\mathbf{z} \in \mathbb{R}^p$  are regarded as Gaussian random variables, and probabilistic distributions on them are defined as follows (note that (4) is just a direct consequence of (2) and (3)):

$$p(\mathbf{x} | \mathbf{z}) = \mathcal{N}_{\mathbf{x}}(\mathbf{W}\mathbf{z} + \mathbf{b}, \sigma^2 \mathbf{I}), \quad (2)$$

$$p(\mathbf{z}) = \mathcal{N}_{\mathbf{z}}(\mathbf{0}, \mathbf{I}), \quad (3)$$

$$p(\mathbf{x}) = \mathcal{N}_{\mathbf{x}}(\mathbf{b}, \mathbf{C}), \quad (4)$$

where  $\{\mathbf{W}, \mathbf{b}, \sigma^2\}$  is the set of the parameters, and

$$\mathbf{C} = \sigma^2 \mathbf{I} + \mathbf{W}\mathbf{W}^\top. \quad (5)$$

Here, the column of  $\mathbf{W} \in \mathbb{R}^{d \times p}$  are principal components,  $\mathbf{b} \in \mathbb{R}^d$  is the interception parameter, and  $\sigma^2 \in \mathbb{R}_{>0}$  is the observation noise variance. It is known [17] that the maximum likelihood estimation of  $\mathbf{W}$  coincides with the result of the classical (non-probabilistic) PCA in the limit of  $\sigma^2 \rightarrow 0$ . The maximum likelihood estimation of  $\mathbf{W}$  can be obtained via eigendecomposition of a data covariance matrix or iterations of the EM algorithm.

The optimal least-squares reconstruction by the probabilistic PCA [17] is given as

$$\hat{\mathbf{x}} = \mathbf{B}\mathbf{x}, \quad \text{where} \quad \mathbf{B} = \mathbf{W}(\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top. \quad (6)$$

<sup>1</sup>We may define the reconstruction error  $e(\mathbf{x})$  as the norm of a part of  $\hat{\mathbf{x}} - \mathbf{x}$  when the features of interest are limited. The methods discussed in this paper apply even to this case only with a slight modification.

Therefore, the corresponding reconstruction error is given in a quadratic form:

$$e(\mathbf{x}) = \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 = \text{Tr}((\mathbf{I} - \mathbf{B})\mathbf{x}\mathbf{x}^\top). \quad (7)$$

### C. Shapley Values

The *Shapley value* [11] is a concept in coalitional games. It is a method to distribute the total gain of a game to the players and has desirable properties such as efficiency, symmetry, and linearity. It has also been utilized in explaining machine learning models in several contexts of machine learning [12]–[14], [18]–[20]. In these contexts, a gain is the output of a machine learning model, and a player is each of the features fed into the model.

Let  $D = \{1, \dots, d\}$  be the set of feature indices,  $S \subseteq D$  be an index subset, and  $v(S)$  be a value function that expresses the output of a machine learning model given only the features indexed in  $S$ . Then, the Shapley value of  $v$  with regard to the  $i$ -th feature,  $\varphi_i(v)$ , is defined as

$$\begin{aligned} \varphi_i(v) &= \sum_{S \subseteq D \setminus \{i\}} \gamma(S) (v(S \cup \{i\}) - v(S)), \\ \gamma(S) &= \frac{|S|! (d - 1 - |S|)!}{d!}. \end{aligned} \quad (8)$$

In fact, the following expression is equivalent to (8), i.e.,

$$\varphi_i(v) = \frac{1}{d!} \sum_{O \in \pi(d)} \left( v(\text{Pre}_i(O) \cup \{i\}) - v(\text{Pre}_i(O)) \right), \quad (9)$$

where  $\pi(d)$  is the set of permutations of  $(1, \dots, d)$ , and  $\text{Pre}_i(O)$  is the set of feature indices that precede  $i$  in an order  $O$ ; e.g.,  $\text{Pre}_2((1, 4, 2, 3)) = \{1, 4\}$ .

There have been proposed several ways to define the value function  $v(\cdot)$  to use Shapley values for explaining machine learning models. In Shapley value regression [12], [18],  $v(S)$  is defined as the coefficient of determination of models using features in  $S$ , with which they measure contributions of features to the explained variance. In another line of studies [13], [19], [20],  $v(S)$  is defined as the output of a model trained using the features in  $S$ . As a naïve implementation of this value needs retraining of the model for different  $S$  [19], the consequent studies [13], [20] approximate it by integrating a full model's outputs with regard to the features absent in  $S$ . A similar idea is adopted in the method called Shapley additive explanation (SHAP) [14], which defines  $v(S)$  as the output of a local surrogate model.

The exact computation of  $\varphi(v)$  using (8) or (9) is intractable for a moderate number of features,  $d$ , because we have to consider all subsets of  $D \setminus \{i\}$  for the summation, whose number increases exponentially with regard to  $d$ . Hence, we need some approximation, and a couple of methods to this end are known. A straightforward option is to approximate the summation in (9) simply by a Monte Carlo method, i.e., by randomly choosing *some* subsets of  $D \setminus \{i\}$  for the summation. In this paper, we adopt this Monte Carlo strategy. Another popular approximation method is the one used in SHAP [14], namely kernel SHAP. Kernel SHAP utilizes a

sample weighting function (a kernel) with which a linear local surrogate model coincides with the approximation of the Shapley values. Considering the latter type of approximation is an open problem in our context.

### III. SHAPLEY VALUES OF RECONSTRUCTION ERRORS OF PCA

In this work, we aim to explain the reconstruction errors of PCA using the idea of the Shapley values. In other words, we would like to compute the Shapley values when the value function,  $v(S)$ , is defined via the reconstruction errors of PCA on the features indexed in  $S$ . To this end, the main problem of ours is how to define  $v(S)$  in a way that facilitates an efficient computation. We consider two possible formulations to define  $v(S)$  for our purpose, namely  $v_1$  in (12) and  $v_2$  in (13), and elaborate the latter as we find it more convenient.

#### A. Notation

Let us introduce notations used hereafter. Without loss of generality, we consider that a data-point  $\mathbf{x} \in \mathbb{R}^d$  is a vertical concatenation of  $\mathbf{x}_{S^c} \in \mathbb{R}^{d-|S|}$  and  $\mathbf{x}_S \in \mathbb{R}^{|S|}$ , i.e.,

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{S^c} \\ \mathbf{x}_S \end{bmatrix}, \quad (10)$$

where  $\mathbf{x}_S$  denotes the subvector of  $\mathbf{x}$  corresponding to the indices in  $S$  (e.g.,  $\mathbf{x}_{\{1,3\}} = [x_1 \ x_3]^\top$ ), and  $S^c = D \setminus S$  denotes the complement of  $S$ . Analogously, we can rearrange and partition the matrices  $\mathbf{C}$  and  $\mathbf{B}$  in (5) and (6), respectively, as

$$\begin{aligned} \mathbf{C} &= \begin{bmatrix} \mathbf{C}_{S^c} & \mathbf{C}_{S^c,S} \\ \mathbf{C}_{S^c,S}^\top & \mathbf{C}_S \end{bmatrix} \quad \text{and} \\ \mathbf{B} &= \begin{bmatrix} \mathbf{B}_{S^c} & \mathbf{B}_{S^c,S} \\ \mathbf{B}_{S^c,S}^\top & \mathbf{B}_S \end{bmatrix}, \end{aligned} \quad (11)$$

where  $\mathbf{C}_{S^c} \in \mathbb{R}^{(d-|S|) \times (d-|S|)}$ ,  $\mathbf{C}_{S^c,S} \in \mathbb{R}^{(d-|S|) \times |S|}$ ,  $\mathbf{C}_S \in \mathbb{R}^{|S| \times |S|}$ , and analogously for  $\mathbf{B}$ . Here,  $\mathbf{C}_S$  is the submatrix of  $\mathbf{C}$  corresponding to the features in  $S$ , and so on.

#### B. Value Function by Retraining

Let  $\mathbf{W}^{(S)} \in \mathbb{R}^{|S| \times p}$  be the parameter of probabilistic PCA in (2) that was learned on data comprising  $\mathbf{x}_S$ . Then, we may define a value function as

$$\begin{aligned} v_1(S) &= \frac{1}{|S|} e^{(S)}(\mathbf{x}_S), \\ e^{(S)}(\mathbf{x}_S) &= \text{Tr}((\mathbf{I} - \mathbf{B}^{(S)})\mathbf{x}_S\mathbf{x}_S^\top), \end{aligned} \quad (12)$$

where  $\mathbf{B}^{(S)} = \mathbf{W}^{(S)}((\mathbf{W}^{(S)})^\top \mathbf{W}^{(S)})^{-1}(\mathbf{W}^{(S)})^\top$ . In words, (12) defines a value function  $v_1(S)$  as the reconstruction error of  $\mathbf{x}_S$  with regard to the PCA trained only on the subset of features,  $S$ . Obviously, we need to retrain PCA for each  $S$  to compute the Shapley value of (12), which causes difficulties as follows:

- As training PCA needs eigendecomposition of a  $|S| \times |S|$  matrix or iterations of the EM algorithm, retraining may become inefficient for a moderate  $d$ .
- We have to maintain many models or training data itself, which is inefficient in terms of memory.

---

#### Algorithm 1 Monte Carlo approximation of Shapley values of reconstruction errors of PCA

---

**Input:** estimated parameters of PCA ( $\mathbf{W} \in \mathbb{R}^{d \times p}$ ,  $\sigma^2 \in \mathbb{R}_{>0}$ ; we assume  $\mathbf{b} = \mathbf{0}$ ), a test data-point  $\mathbf{x}$ , a target feature index  $i$ , and the number of Monte Carlo iterations  $Q$

**Output:** a Shapley value  $\varphi_i(v_2)$

- 1:  $\mathbf{C} \leftarrow \sigma^2 \mathbf{I} + \mathbf{W}\mathbf{W}^\top$
  - 2:  $\mathbf{B} \leftarrow \mathbf{W}(\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top$
  - 3:  $\varphi_i \leftarrow 0$
  - 4: **for**  $q = 1$  to  $Q$  **do**
  - 5:   draw an order  $O_q$  randomly from  $\pi(D)$
  - 6:    $\varphi_{i,q} \leftarrow v(\text{Pre}_i(O_q) \cup \{i\}) - v(\text{Pre}_i(O_q))$ , using (16)
  - 7:    $\varphi_i \leftarrow \varphi_i + \varphi_{i,q}/Q$
  - 8: **end for**
- 

- It is challenging to consider a definite strategy to determine the dimensionality of the latent variable of PCA for different  $S$ , especially for small  $|S|$ .

#### C. Value Function by Marginalization

To alleviate the difficulties of the retraining strategy, we consider to avoid retraining by performing marginalization with regard to unused features, as done in literature [13], [14], [20]. That is, we define a value function as

$$v_2(S) = \frac{1}{d} \mathbb{E}_{p(\mathbf{x}_{S^c} | \mathbf{x}_S)}[e(\mathbf{x})]. \quad (13)$$

We note that  $e(\mathbf{x})$  is simply the reconstruction error on the full test data-point,  $\mathbf{x}$ , computed with the PCA trained on all the features, as defined in (1).

In literature [13], [14], [20], the marginalization relies on an assumption of statistical independence,  $p(\mathbf{x}_{S^c} | \mathbf{x}_S) = p(\mathbf{x}_{S^c})$ . However, when an anomaly detection based on reconstruction error is valid, there usually exists dependence between features, which may make the approaches based on the independence assumption [13], [14], [20] less meaningful.

Meanwhile, we can exactly compute (13) *without the independence assumption* using the probabilistic view of PCA in (2)–(4) as follows. First, the conditional distribution of  $\mathbf{x}_{S^c}$  given  $\mathbf{x}_S$ , namely  $p(\mathbf{x}_{S^c} | \mathbf{x}_S)$ , is immediately derived from (4) as

$$p(\mathbf{x}_{S^c} | \mathbf{x}_S) = \mathcal{N}_{\mathbf{x}_{S^c}}(\mathbf{C}_{S^c,S} \mathbf{C}_S^{-1} \mathbf{x}_S, \mathbf{C}_{S^c,S} \mathbf{C}_S^{-1} \mathbf{C}_{S^c,S}^\top), \quad (14)$$

where we assume that the data are centered so that  $\mathbf{b} = \mathbf{0}$  without loss of generality. Note that  $\mathbf{C}_\cdot$ 's are the submatrices of  $\mathbf{C}$  defined in (11). Second, the total reconstruction error,  $e(\mathbf{x})$  in (1), can be decomposed as

$$\begin{aligned} e(\mathbf{x}) &= \text{Tr}((\mathbf{I} - \mathbf{B})\mathbf{x}\mathbf{x}^\top) \\ &= \text{Tr}((\mathbf{I} - \mathbf{B}_{S^c})\mathbf{x}_{S^c}\mathbf{x}_{S^c}^\top) \\ &\quad - 2 \text{Tr}(\mathbf{B}_{S^c,S}\mathbf{x}_{S^c}\mathbf{x}_S^\top) \\ &\quad + \text{Tr}((\mathbf{I} - \mathbf{B}_S)\mathbf{x}_S\mathbf{x}_S^\top), \end{aligned} \quad (15)$$

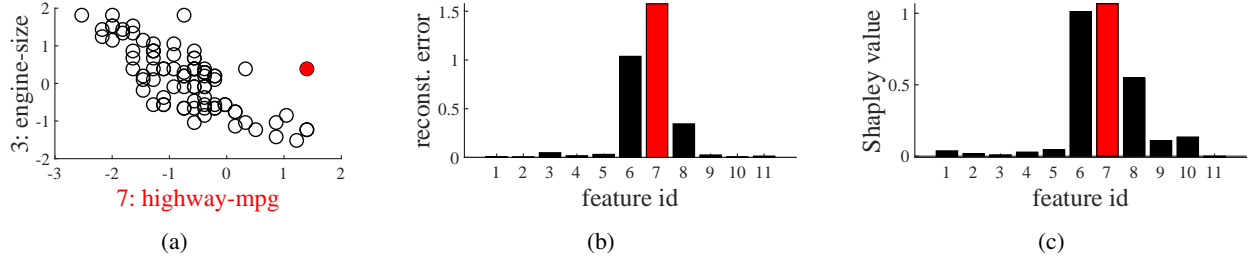


Fig. 1: (a) Scatter plot of normalized test data in a test trial, wherein a filled circle indicates the artificial anomalous data-point. In this test trial, the feature #7 of the indicated data-point was made anomalous. (b) Reconstruction errors of each feature of the anomalous data-point. (c) Shapley values of reconstruction error of each feature of the anomalous data-point.

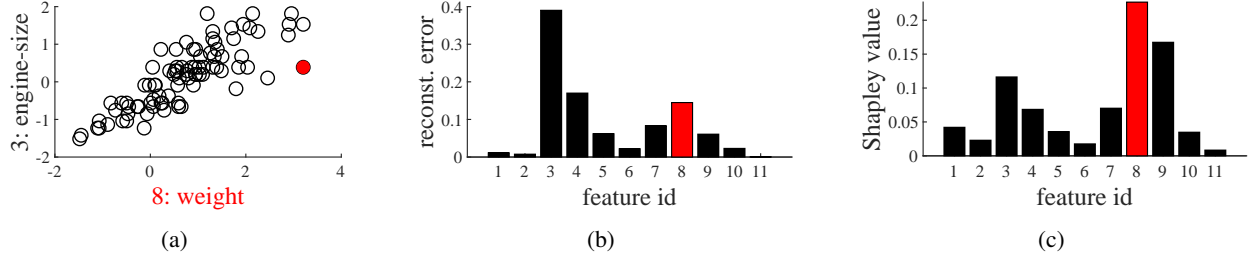


Fig. 2: As in Fig. 1, but for another test trial, in which the feature #8 of the indicated data-point was made anomalous.

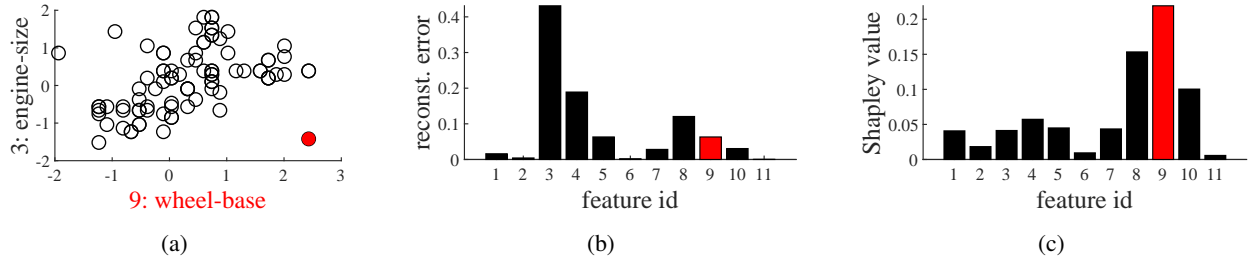


Fig. 3: As in Fig. 1, but for yet another test trial, in which the feature #9 of the indicated data-point was made anomalous.

where  $B$ .'s are the submatrices of  $B$  in (11). Finally, using (14) and (15),  $v_2(S)$  in (13) is computed by

$$\begin{aligned} v_2(S) &= \frac{1}{d} \mathbb{E}_{p(\mathbf{x}_{S^c} | \mathbf{x}_S)} [e(\mathbf{x})] \\ &= \frac{1}{d} \left( \text{Tr}((\mathbf{I} - \mathbf{B}_{S^c}) \mathbf{V}_{S^c}) + \text{Tr}((\mathbf{I} - \mathbf{B}_{S^c}) \mathbf{m}_{S^c} \mathbf{m}_{S^c}^\top) \right. \\ &\quad \left. - 2 \text{Tr}(\mathbf{B}_{S^c, S} \mathbf{x}_S \mathbf{m}_{S^c}^\top) + \text{Tr}((\mathbf{I} - \mathbf{B}_S) \mathbf{x}_S \mathbf{x}_S^\top) \right), \end{aligned} \quad (16)$$

where we use notations

$$\mathbf{m}_{S^c} = \mathbf{C}_{S^c, S} \mathbf{C}_S^{-1} \mathbf{x}_S, \quad (17)$$

$$\mathbf{V}_{S^c} = \mathbf{C}_{S^c, S} \mathbf{C}_S^{-1} \mathbf{C}_{S^c, S}^\top. \quad (18)$$

Unlike the retraining strategy, the marginalization strategy does not require to keep many models or training data because (16) can be computed only from the parameters of the full model. A computational bottleneck of (16) lies in the computation of  $\mathbf{C}_{S^c, S} \mathbf{C}_S^{-1}$ , but this can be computed relatively efficiently in general.

#### D. Approximation of Shapley values

Given the value function in (16), we approximate its Shapley values using a Monte Carlo approximation of (9). We summarize the overall procedures in Algorithm 1. For the detailed deviations of the Monte Carlo approximation, consult e.g., Štrumbelj *et al.* [20].

### IV. NUMERICAL EXAMPLES

#### A. Artificial Anomalies

We concretely investigated how the Shapley values were different from the reconstruction errors using a multivariate dataset to which we inserted artificial anomalies. We used the 2004 New Car and Truck Data, which is available online<sup>2</sup>, as the base dataset. It contains specifications of vehicles released in 2004, such as price, fuel efficiency, engine size, and car size; in total, it comprises  $d = 11$  features. The original dataset consists of 428 instances. We eliminated 41 instances including missing values and used the first 300 instances for the training and the last 87 for the test. We repeated the test

<sup>2</sup>[jse.amstat.org/jse\\_data\\_archive.htm](http://jse.amstat.org/jse_data_archive.htm)

phase for different settings of how an artificial anomaly is created; in each test, we replaced the value of a feature of a data-point to the maximum value of the same feature in the test set. Thus, we conducted  $11 \times 87 = 957$  test trials in total. The dimensionality of PCA was set to  $p = 8$ .

In Figs. 1, 2, and 3, we exemplify three test trials. Each figure comprises: (a) a scatter plot of an “anomalous” feature against the feature #3 (engine size), (b) the reconstruction errors of each feature on the anomalous instance, and (c) the corresponding Shapley values computed via the proposed method. In Fig. 1, an explanation that would be derived from the reconstruction errors and the one from the Shapley values roughly coincide, i.e., the feature #7 (highway miles per gallon), which was made anomalous in fact, indicates the largest value in both criteria. In contrast, in Figs. 2 and 3, the reconstruction errors are suspicious whereas the Shapley values are somewhat trustworthy; e.g., in Fig. 2, though the anomaly was inserted to the feature #8 (weight), the reconstruction error shows its largest value at feature #3.

We also examined how often anomalous features are identified correctly by looking at the reconstruction errors or the Shapley values. In Table I, we report the Hits@ $n$ , which denotes the rate that a truly anomalous feature exists in the set of features having the  $n$  largest values of a criterion (the errors or the Shapley). Table I is for two experimental cases, MAX and MIN: MAX denotes the experimental setting explained above, i.e., an anomaly is generated by replacing a feature value by its maximum. MIN is its slightly differed version, in which an anomaly is generated by the replacement with minimum values. From Table I, we can observe that the Shapley values are advantageous to identify anomalous features.

### B. Anomaly Detection Datasets

We also applied the proposed method to real-world anomaly detection datasets. We downloaded the datasets summarized as Outlier Detection DataSets [21], which are available online<sup>3</sup>. From the category of multi-dimensional point datasets, we selected 10 datasets with which a PCA-based anomaly detection worked well. Each dataset comprises a set of  $m_{\text{good}}$  normal instances and a set of  $m_{\text{bad}}$  anomalous instances. We used the first  $m_{\text{good}} - m_{\text{bad}}$  elements of the normal set for training, and the remaining elements of the normal set and the whole anomalous set for a test. The dimensionality of PCA was chosen so that it could explain about 95% of the variance of the training data.

As conducting case studies for each dataset is unrealistic due to the lack of precise understanding of causes of the anomalies, we just examined the discrepancy between reconstruction errors and their Shapley values in test sets. Concretely, we investigated the correlation coefficient between them. In Table II, we show the median values<sup>4</sup> of the correlation coefficients

TABLE I: Hits@ $n$  for two experimental cases, MAX and MIN.

	MAX		MIN	
	Hits@1	Hits@3	Hits@1	Hits@3
reconstruction error	.316	.605	.271	.471
Shapley values	.484	.801	.484	.710

TABLE II: Datasets property ( $d$  and  $m = m_{\text{good}} + m_{\text{bad}}$ ) and the correlation coefficients between reconstruction errors and their Shapley values. The medians (via the Fisher transformation) of the coefficients computed for each feature are reported.  $r_{\text{all}}$ ,  $r_{\text{good}}$ , and  $r_{\text{bad}}$  denote the correlation coefficients for the whole, the normal part, and the anomalous part of the test sets, respectively.

dataset	$d$	$m$	$r_{\text{all}}$	$r_{\text{good}}$	$r_{\text{bad}}$
CARDIO	21	1831	.866	.893	.797
FORESTCOVER	10	286048	.756	.536	.808
IONOSPHERE	33	351	.984	.986	.985
MAMMOGRAPHY	6	11183	.854	.268	.854
MUSK	166	3062	.945	.987	.949
SATIMAGE-2	36	5803	.975	.993	.981
SHUTTLE	9	49097	.869	.958	.893
VOWELS	12	1456	.883	.833	.877
WBC	30	278	.956	.955	.943
WINE	13	129	.817	.785	.657

computed for each feature individually. There, the correlation on a full test set ( $r_{\text{all}}$ ), the one on the normal part of the test set ( $r_{\text{good}}$ ), and the one on the anomalous part of the test set ( $r_{\text{bad}}$ ) are reported. We can observe that the reconstruction errors and their Shapley values are strongly correlated (say, with  $r > 0.9$ ) in many cases, whereas the correlation is moderate (say,  $r < 0.7$ ) in some cases. These results do not completely preclude the use of reconstruction errors for explaining anomalies, but one should be aware that there are cases where the Shapley values do not completely agree with the reconstruction errors of each feature.

## V. DISCUSSION

### A. Related Work

Several researchers have been working on explaining anomaly detection. Dang *et al.* [22], Micenková *et al.* [23], and Dang *et al.* [24] proposed ways to explain anomaly detection based on nearest neighbors by looking for a set of features (i.e., a subspace) that maximally discriminates an outlier from neighboring inliers. Similarly, Liu *et al.* [25] developed a method to interpret anomalies by learning classifiers that distinguish normal and anomalous data-points. Moreover, Macha *et al.* [26] proposed a framework to generate descriptions of outliers via clustering in subspaces and its summarization. In another context, Tang *et al.* [27] proposed an interpretable definition of outliers for categorical data, which is understandable in terms of common and uncommon components of anomalous and normal values. Furthermore, Gupta *et al.* [28] developed a method to optimize pictorial explanations of anomalies. The types of anomaly detection

<sup>3</sup>odds.cs.stonybrook.edu/

<sup>4</sup>We computed the medians of the correlation coefficients via the Fisher transformation  $z = \log((1+r)/(1-r))/2$  and the inverse transformation  $r_{\text{median}} = \tanh(z_{\text{median}})$ .

methods premised in these studies are basically different from ours (i.e., reconstruction error-based), but seeking connections is an interesting future direction of research.

One of the existing studies that are mostly related to ours is the one by Antwarg *et al.* [16], in which they use kernel SHAP [14] to explain the reconstruction errors of an autoencoder. A possible difficulty of their methods lies in the assumption of feature independence premised in kernel SHAP. As features are often dependent in anomaly detection applications, considering feature correlations may be essential in some practices. We will be working on comparing our current method (and its extensions) with kernel SHAP.

### B. Direct and Indirect Extensions

One of the promising extensions of our method is application to variants of PCA. For example, the mixtures of PCAs [29] are useful to capture multimodality and in fact, utilized in anomaly detection applications (see, e.g., Yairi *et al.* [8]). An extension to sparse PCA [30] is also interesting to ensure interpretability of both a model and reconstruction errors.

Extensions to nonlinear and/or non-Gaussian models, with which conditional distributions cannot be obtained easily, are also an interesting and demanding direction of research. To this end, one possible option is to approximate the conditional distributions in some tractable ways. For example, approximation for using robust distributions (e.g., *t*-distribution) would be useful to improve anomaly detection performance.

Reconstruction errors are always nonnegative, but the raw Shapley values may take negative values, which is basically challenging to interpret. Also, explanations by the Shapley values are usually not parsimonious. To overcome these issues, approximation methods such as kernel SHAP and their variants would be useful.

### ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP19K21550.

### REFERENCES

- [1] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Generation Computer Systems*, vol. 55, pp. 278–288, 2016.
- [2] P. García-Teodoro, J. E. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, vol. 28, no. 1-2, pp. 18–28, 2009.
- [3] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mechanical Systems and Signal Processing*, vol. 107, pp. 241–265, 2018.
- [4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.
- [5] G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, and E. Schubert, "On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 891–927, 2016.
- [6] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS ONE*, vol. 11, no. 4, e0152173, 2016.
- [7] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 219–230, 2004.
- [8] T. Yairi, N. Takeishi, T. Oda, Y. Nakajima, N. Nishimura, and N. Takata, "A data-driven health monitoring method for satellite housekeeping data based on probabilistic clustering and dimensionality reduction," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 3, pp. 1384–1401, 2017.
- [9] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 665–674.
- [10] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *Proceedings of the 6th International Conference on Learning Representations*, 2018.
- [11] L. S. Shapley, "A value for *n*-person games," in *Contributions to the Theory of Games*, ser. Annals of Mathematics Studies 28, vol. II, Princeton: Princeton University Press, 1953, pp. 307–317.
- [12] S. Lipovetsky, "Entropy criterion in logistic regression and Shapley value of predictors," *Journal of Modern Applied Statistical Methods*, vol. 5, no. 1, pp. 95–106, 2006.
- [13] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and Information Systems*, vol. 41, no. 3, pp. 647–665, 2014.
- [14] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 4765–4774.
- [15] I. T. Jolliffe, *Principal component analysis*, 2nd. Springer, 2002.
- [16] L. Antwarg, B. Shapira, and L. Rokach, "Explaining anomalies detected by autoencoders using SHAP," Tech. Rep., 2019, arXiv:1903.02407.
- [17] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B*, vol. 61, no. 3, pp. 611–622, 1999.
- [18] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Applied Stochastic Models in Business and Industry*, vol. 17, no. 4, pp. 319–330, 2001.
- [19] E. Štrumbelj, I. Kononenko, and M. Robnik Šikonja, "Explaining instance classifications with interactions of subsets of feature values," *Data & Knowledge Engineering*, vol. 68, no. 10, pp. 886–904, 2009.
- [20] E. Štrumbelj and I. Kononenko, "An efficient explanation of individual classifications using game theory," *Journal of Machine Learning Research*, vol. 11, pp. 1–18, 2010.
- [21] S. Rayana, *ODDS library*, 2016. [Online]. Available: <http://odds.cs.stonybrook.edu>.
- [22] X.-H. Dang, B. Micenková, I. Assent, and R. T. Ng, "Local outlier detection with interpretation," in *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Artificial Intelligence, vol. 7908, 2013, pp. 304–320.
- [23] B. Micenková, R. T. Ng, X.-H. Dang, and I. Assent, "Explaining outliers by subspace separability," in *Proceedings of the 13th IEEE International Conference on Data Mining*, 2013, pp. 518–527.
- [24] X.-H. Dang, I. Assent, R. T. Ng, A. Zimek, and E. Schubert, "Discriminative features for identifying and interpreting outliers," in *Proceedings of the 30th IEEE International Conference on Data Engineering*, 2014, pp. 88–99.
- [25] N. Liu, D. Shin, and X. Hu, "Contextual outlier interpretation," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, Jul. 2018, pp. 2461–2467.
- [26] M. Macha and L. Akoglu, "Explaining anomalies in groups with characterizing subspace rules," *Data Mining and Knowledge Discovery*, vol. 32, no. 5, pp. 1444–1480, 2018.
- [27] G. Tang, J. Pei, J. Bailey, and G. Dong, "Mining multidimensional contextual outliers from categorical relational data," *Intelligent Data Analysis*, vol. 19, no. 5, pp. 1171–1192, 2015.
- [28] N. Gupta, D. Eswaran, N. Shah, L. Akoglu, and C. Faloutsos, "Beyond outlier detection: LookOut for pictorial explanation," in *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science, vol. 11051, 2019, pp. 122–138.
- [29] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.
- [30] Y. Guan and J. G. Dy, "Sparse probabilistic principal component analysis," in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, 2009, pp. 185–192.