Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction

Mayu Sakurada
The University of Tokyo
Department of Aeronautics and Astronautics
sakurada@space.rcast.u-tokyo.ac.jp

Takehisa Yairi
The University of Tokyo
Research Center for Advanced Science and Technology
yairi@space.rcast.u-tokyo.ac.jp

ABSTRACT

This paper proposes to use autoencoders with nonlinear dimensionality reduction in the anomaly detection task. The authors apply dimensionality reduction by using an autoencoder onto both artificial data and real data, and compare it with linear PCA and kernel PCA to clarify its property. The artificial data is generated from Lorenz system, and the real data is the spacecrafts' telemetry data. This paper demonstrates that autoencoders are able to detect subtle anomalies which linear PCA fails. Also, autoencoders can increase their accuracy by extending them to denoising autoenconders. Moreover, autoencoders can be useful as nonlinear techniques without complex computation as kernel PCA requires. Finaly, the authors examine the learned features in the hidden layer of autoencoders, and present that autoencoders learn the normal state properly and activate differently with anomalous input.

Categories and Subject Descriptors

I.2.1 [Artificial Intelligence]: Applications and Expert Systems—Industrial automation; I.5.4 [Pattern recognition]: Applications—Signal processing

General Terms

Performance

Keywords

anomaly detection, novelty detection, fault detection, autoencoder, auto-assosiative neural network, denoising autoencoder, dimensionality reduction, nonlinear, spacecrafts

1. INTRODUCTION

Recently, feature learning using the neural network with dimensionality reduction has become popular in Deep Learning context [4]. Actually an autoencoder, which is the neural network with nonlinear dimensionality reduction capability, has been used since the 1990's, often with the name autoassociative neural networks [8], [7]. However, there are still few works in which researchers try to apply those learned features to other data mining tasks. Our idea is to apply them to one of the fundamental data mining tasks: the anomaly detection task. We perform dimensionality reduction using autoencoders to the data which contain anomalies. We investigate the difference in performance to detect anomalies by comparing an autoencoder with other traditional approaches such as linear principal component analysis (hereinafter referred to as PCA), and kernel PCA. Previous works proposed the other extension to the ordinary autoencoder, named denoising autoencoder [13], and we also include this approach in our comparison.

Our work eventually aims to detect anomalies in the spacecrafts' telemetry data by dimensionality reduction technique. Spacecrafts have a complex system and their telemetry data have hundreds of variables. Most of the variables are nonlinearly correlated and temporally dependent. It is difficult for humans to distinguish the abnormal state from the normal state only by the raw data. For this reason, training the machine to learn the normal state and displaying the reconstruction error as the anomaly score is valuable. Thus, in this paper we especially focus on the time series data which consist of 10-100 variables with the nonlinear correlation.

Our contribution is three-fold. First we apply dimensionality reduction using autoencoders to both artificial data and real data, and present that autoencoders are applicable to anomaly detection. Second, we compare the performance among autoencoders, denoising autoencoders, linear PCA and kernel PCA to clarify the property of autoencoders. We found that 1) autoencoders can detect anomalies which linear PCA fails to detect, and also increase the accuracy by extending autoencoders to denoising autoencoders, and 2) autoencoders can avoid complex computation as kernel PCA requires without degrading the quality of detecting performance. Finally, we investigate the learned features in the hidden layer of the autoencoder, and display that they learn the normal state properly and activates differently with anomalous input.

2. RELATED WORK

One of the properties of autoencoders is that they can employ nonlinear dimensionality reduction. There are several papers such as [8], [6], [10] in which the authors investigated its nonlinear property. In [6], they theoretically demonstrated the nonlinearity of autoencoders. In [8], [6], [10], they applied autoencoders to nonlinear anomaly detec-

MLSDA '14, December 2, 2014, Gold Coast, QLD, Australia

http://dx.doi.org/10.1145/2689746.2689747

tion data which is artificially generated. However, the data they used are too simple to simulate the real data. In our work, we generated the data with 25 dimensions from a more complicated nonlinear system using the Lorenz equations.

Some of the previous works applied autoencoders to the real data or the realistic data generated by simulating the real world model [11], [3], [12], [9]. However, these works are insufficient in that either they only use the low dimensional data or they lack the comparison with the other approaches. We applied two kinds of real data: one has 10 dimensions and the other has more than 100 dimensions. Although some works compare an autoencoder with other approaches [7], [15], in this paper we focus on the dimensionality reduction and determine the difference in performances according to the reconstruction error.

3. ANOMALY DETECTION USING AUTOENCODERS

3.1 Anomaly Detection by Dimensionality Reduction

In anomaly detection based on machine learning or data mining, we obtain the model which captures the normal behavior in the training period, and after that we check whether the test data can be fitted with the trained model or not. If the test data is inconsistent with the trained model, we regard it as an anomaly.

Anomaly detection using dimensionality reduction is based on the assumption that data has variables correlated with each other and can be embedded into a lower dimensional subspace in which normal samples and anomalous samples appear significantly different [2]. In the training phase, we have normal data as training set $\{\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(m)\}$. Assuming each data sample $\boldsymbol{x}(i) \in \mathbb{R}^D$ is represented by a vector of D different variables. We compress the data into lower dimensional latent subspace and reproduce the output $\{\hat{\boldsymbol{x}}(1), \hat{\boldsymbol{x}}(2), \ldots, \hat{\boldsymbol{x}}(m)\}$ so that the reconstruction error in Eq. 1 becomes small.

$$Err(i) = \sqrt{\sum_{j=1}^{D} (x_j(i) - \hat{x_j}(i))^2} \tag{1}$$

After we determine the subspace, in the test phase, we project test data into the subspace and reconstruct the original data. We use the reconstruction error shown in Eq. 1 as the anomaly score. The reconstruction error has low values if test samples are normal instances that satisfy the normal correlation learned during the test phase, while the error becomes large with anomalous samples.

There are several dimensionality reduction techniques. In this work, we choose representative linear and nonlinear techniques, linear PCA and kernel PCA, as the baseline to be compared with autoencoders.

Kernel PCA [5] performs nonlinear mapping to the high-dimensional feature space by a kernel function, and then employ linear PCA in the feature space. In this work, we use the Gaussian kernel $k(\boldsymbol{x}(i), \boldsymbol{x}(j)) = \exp(-\frac{||\boldsymbol{x}(i) - \boldsymbol{x}(j)||^2}{\boldsymbol{\sigma}^2})$. In kernel PCA, the reconstruction error is computed in the feature space, not in the original observation space. To obtain the reconstructed data in the original space, we must solve the pre-image problem which has high computational complexity. Furthermore, kernel PCA basically requires to hold all the training samples, which is also computationally expensive. Therefore, compared to kernel PCA, autoencoders have the advantages in computation cost.

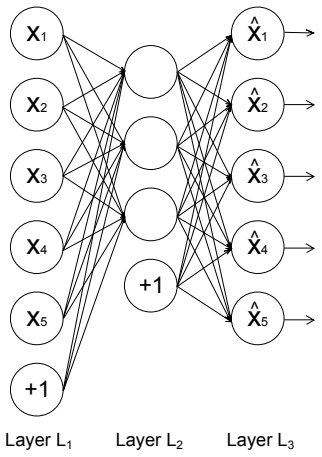

Figure 1: Autoencoder [1]

3.2 Dimensionality Reduction by Autoencoders

First of all, an autoencoder is an unsupervised neural network, whose objective is to learn to reproduce input vectors $\{\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(m)\}$ as outputs $\{\hat{\boldsymbol{x}}(1), \hat{\boldsymbol{x}}(2), \ldots, \hat{\boldsymbol{x}}(m)\}$. Fig. 1 shows an autoencoder. In this figure, Layer L_2 is the hidden layer, whereby the inputs are compressed into a small number of neurons. Activation of unit i in layer l is given by Eq. 2:

$$a_i^{(l)} = f\left(\sum_{j=1}^{n} W_{ij}^{(l-1)} a_j^{(l-1)} + b_i^{(1)}\right) \tag{2}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ denote weight and bias parameters respectively. In the first layer, i.e., the input layer, $\boldsymbol{a}^{(1)} = \boldsymbol{x}$, and in the last layer, i.e., the output layer, $\boldsymbol{a}^{(3)} = \hat{\boldsymbol{x}}$. For the activate function f, we used sigmoid function in hidden layers, but in the output layer, we used linear function since we don't pre-scale every input example to a specific interval like $[-1, 1]$.

During the training period, we minimize the objective function shown in Eq. 3 with respect to $\boldsymbol{W}$ and $\boldsymbol{b}$. The objective function includes the regularization term, and the parameter λ determines the strength of regularization.

$$J(\boldsymbol{W}, \boldsymbol{b}) = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{2} \left\|\boldsymbol{x}(i) - \hat{\boldsymbol{x}}(i)\right\|^2\right) + \frac{\lambda}{2} \sum_{l=1}^{n_l - 1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_l + 1} \left(W_{ji}^{(l)}\right)^2 \tag{3}$$

where n_l denotes number of layers in the network and s_l denotes number of units in layer L_l.

Recently a denoising autoencoder [13], which is one of the extensions of an autoencoder, has been developed. The idea is to learn an over-complete set of basis vectors to represent input vectors, so that our basis vectors can capture structures and patterns inherent in the input data better. At the same time, in order to avoid highly compressed encoding which is usually highly entangled, we can encode the input

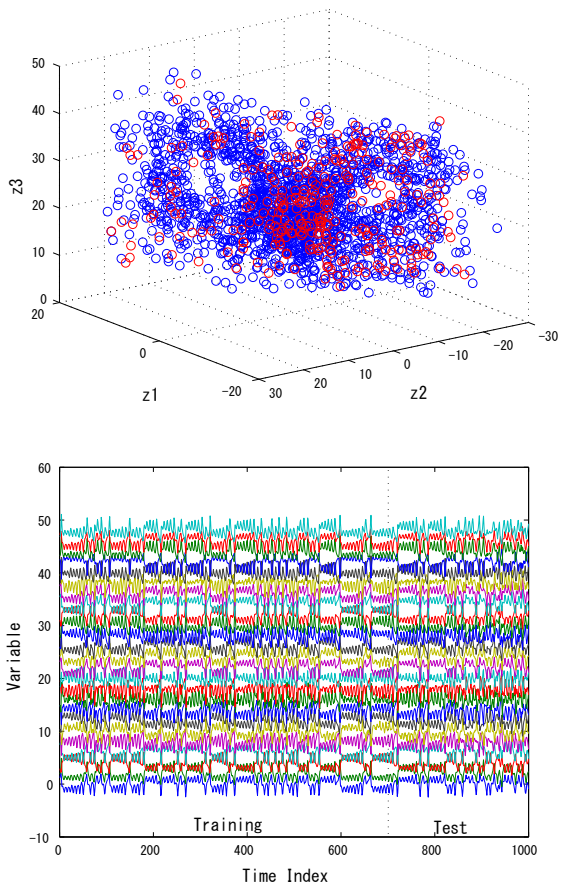

Figure 2: Top: Normal $\{\boldsymbol{z}(1), \boldsymbol{z}(2), ..., \boldsymbol{z}(849)\}$ (blue) and anomalous $\{\boldsymbol{z}(850), \boldsymbol{z}(851), ..., \boldsymbol{z}(1000)\}$ (red) data from Lorenz system. Bottom: Normalized 25 dimensional Lorenz system data $\boldsymbol{x}$.

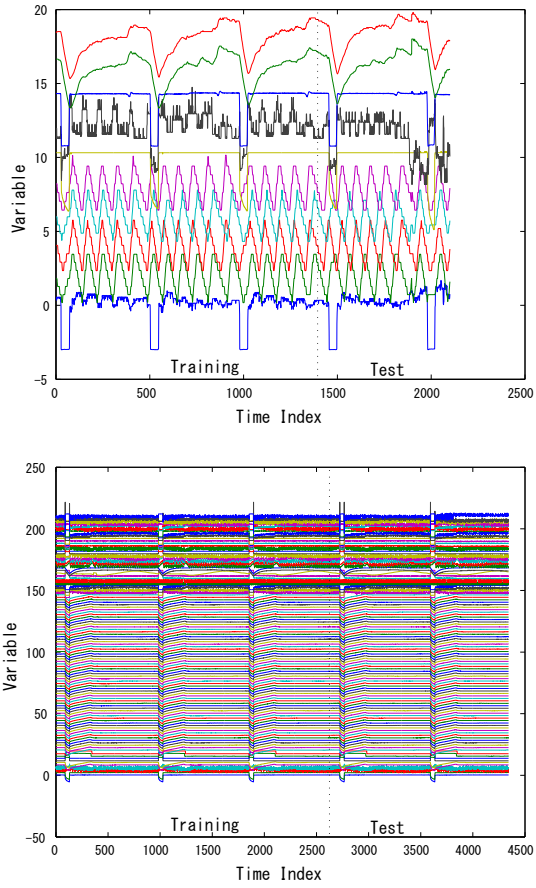

Figure 3: Top: Normalized data of Satellite-A. Bottom: Normalized data of Satellite-B.

with small subset of neurons. We can achieve this by increasing the number of hidden units and adding some noise to the input. There are some ways in adding the noise to each input, but in this work, we destruct the input by randomly choosing a fixed number of components of the input to be 0, which is sometimes called as the salt-and-pepper noise [14].

4. EXPERIMENTAL SETUP

We performed dimensionality reduction on each data by 4 methods: linear PCA, an autoencoder, a denoising autoencoder and kernel PCA. In each method, the number of latent space dimension was adjusted manually. For autoencoders and denoising autoencoders, we adjusted several parameters in the objective function (Eq. 3) as $\lambda = 0.00001$, $\beta = 3$, $\rho = 0.01$. The destruction level, i.e., the probability of that each element is forced to 0, is fixed to 0.1. We compared the performances based on the reconstruction error in Eq. 1.

4.1 Artificial Data

We prepared the nonlinear simulated data using the Lorenz system. The Lorenz system consists of the following equations:

$$\begin{aligned} \dot{z}_1(t) &= \sigma(z_2(t) - z_1(t)) \\ \dot{z}_2(t) &= z_1(t)(\rho - z_3(t))z_2(t) \\ \dot{z}_3(t) &= z_1(t)z_2(t) - \beta z_3(t) \end{aligned} \tag{4}$$

We set three parameters σ, ρ and β to 28, 10 and 8/3 respectively. According to Eq. 4, first we generated the vector $\boldsymbol{z}(t) = (z_1(t) \quad z_2(t) \quad z_3(t))^T$. We sampled 1000 vectors by running this simulation for 100[s] with the sampling rate 0.1[s], with the small observation noise and system transition noise. To generate the anomalous data, after sampling we flipped the values from $z_3(850)$ to $z_3(1000)$ horizontally so that z_3 aligns in reverse chronological order after 850th. To generate the high dimensional vector $\boldsymbol{x}(t)$, first we made the matrix $\boldsymbol{W} \in \mathbb{R}^{25\times 3}$ whose components we randomly chose from the interval $(-5, 5)$. Then we multiplied $\boldsymbol{W}$ by each vector $\boldsymbol{z}(t)$, i.e., $\boldsymbol{x}(t) = \boldsymbol{W}\boldsymbol{z}(t)$. We divided 1000 samples into two, which are 700 training samples $\{\boldsymbol{x}(1), \boldsymbol{x}(2), ..., \boldsymbol{x}(700)\}$ and 300 test samples $\{\boldsymbol{x}(701), \boldsymbol{x}(702), ..., \boldsymbol{x}(1000)\}$, with the latter half of the test samples including anomalies. Fig. 2 shows the distribution of the 1000 vectors of $\boldsymbol{z}$ and the data of $\boldsymbol{x}$ after normalized to a mean of zero and a variance of 1.

4.2 Real Data

We used two kinds of the spacecraft telemetry data in our real data experiment: Satellite-A and Satellite-B. Satellite-A and Satellite-B have 17 and 106 continuous sensor measurements respectively. Spacecraft telemetry data have many different sensor measurements and in general these inputs are correlated with each other [16], [17]. This means that we can remove redundant inputs and represent each data sample as a lower dimensional vector.

Fig. 3 shows the data after we normalized each data so that the mean and variance become 0 and 1 respectively.

5. RESULT AND DISCUSSION

5.1 Artificial Data

If we look at Fig. 4, it is clear that the reconstruction error becomes far bigger after the 150th in the test set with an autoencoder, a denoising autoencoder and kernel PCA. Since this data includes verified anomalies after 150th, it means that the anomaly detection with nonlinear techniques was successful. Linear PCA, however, has failed to show a significant difference between the anomalous and the normal data. We can also see that in Tab. 1, in which linear PCA performs poorly in the Lorenz row. This is a good example that nonlinear dimensionality reduction technique, like autoencoders, can learn the nonlinear correlation between a lot of variables and succeed to detect anomalies, while linear PCA, which employs linear dimensionality reduction, fails and misses anomalies. In the Lorenz row in Tab. 1, we can also notice that the denoising autoencoder performs better than the ordinary autoencoder. In this case, we succeeded to increase the accuracy by extending autoencoders to denoising autoencoders.

5.2 Real Data

We can see from Fig. 5 and Fig. 6 that basically all dimensionality reduction methods succeeded to detect anomalies on both spacecrafts' data. We can see in Fig. 5 and in the row Sat-A in Tab. 1 that, although the performances of linear PCA and an autoencoder are almost the same, a denoising autoencoder performs better than linear PCA and an autoencoder. Unlike the experiment on the Lorenz system data and Satellite-A data, we can see in the Sat-B row in Tab. 1 that a denoising autoencoder fails to increase the accuracy. In this case, since the detecting performance is already good enough with the ordinary autoencoder, adding noise to the input rather gives a bad effect. We tried this experiment with several different numbers of dimensions. Linear PCA turned out to be very sensitive to the number of latent dimensions, and it was harder to tune the number of latent dimension. Autoencoders can detect anomalies even with relatively high latent dimensions while linear PCA can't.

When compared to kernel PCA, the autoencoder and denoising autoencoder performed either better or the same as kernel PCA. Kernel PCA, however, requires heavy computation. By using autoencoders, we don't need to hold all the training samples and we can avoid memory intensive kernel computation. Also, in autoencoders we can compare the original and reconstructed data in the original observation space, and we don't need to solve complex pre-image problem which kernel PCA requires. In fact, the overall running time including training and test phase was more than an hour in kernel PCA, while the autoencoder and denoising autoencoder only took several minutes.

Table 1: The average AUC of the 4 different methods on the 3 different data. LPCA, AE, dAE and KPCA denotes linear PCA, an autoencoder, a denoising autoencoder and kernel PCA respectively. The first row Lorenz has the results on the artificial data using the Lorenz system, and last two rows Sat-A and Sat-B has the results on the real data of two kinds of spacecrafts' telemetry data.

	LPCA	AE	dAE	KPCA
Lorenz	0.5104	0.6473	0.7011	0.7045
Sat-A	0.8852	0.8847	0.9354	0.8862
Sat-B	0.9764	0.9763	0.8355	0.7689

Furthermore, we visualized the activation of a part of the neurons in the hidden layer in Fig. 7. We can see that the anomalous data is significantly different from the normal data in the latent space. This means that denoising autoencoders is able to learn the meaningful features to reproduce normal state and these learned features can't be used to reproduce anomalous input.

6. CONCLUSION AND FUTURE WORK

In this study, we demonstrated examples of applying feature learning by autoencoders to anomaly detection, which is one of the fundamental data mining tasks. Another contribution was comparison of autoencoders with linear PCA and kernel PCA on the artificial data and real data. We clarify the property and the effectiveness of autoencoders based on that comparison. In addition, we examined the learned features in the hidden layer to show the different activations with normal input and anomalous input, which has not been done before.

At the moment we manually tune the parameters of the regularization term of autoencoders, the destruction level of denoising autoencoders, the number of latent dimensions, and so on. Further detailed investigation for those parameters will be necessary in future work. Also, additional comparison with other techniques like vector quantization PCA, mixture probabilistic PCA, which are known as hybrid of clustering and dimensionality reduction [16], [17], would be interesting for clarifying the property of autoencoders. We regard each data sample at each time index as independent, i.e., we disregard time sequence. Although the performance is already good enough without temporal information, we can add the information by giving autoencoders a data vector including current as well as past samples, and see how it improves the performance.

7. REFERENCES

[1] UFLDL Tutorial. http://ufldl.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity. [Online; accessed 10-August-2014].

[2] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. ACM Computing Surveys, 41(3):1–58, July 2009.

[3] S. Hawkins, H. He, G. Williams, and R. Baxter. Outlier detection using replicator neural networks. In Proceedings of the Fifth International Conference and Data Warehousing and Knowledge Discovery, pages 170–180, 2002.

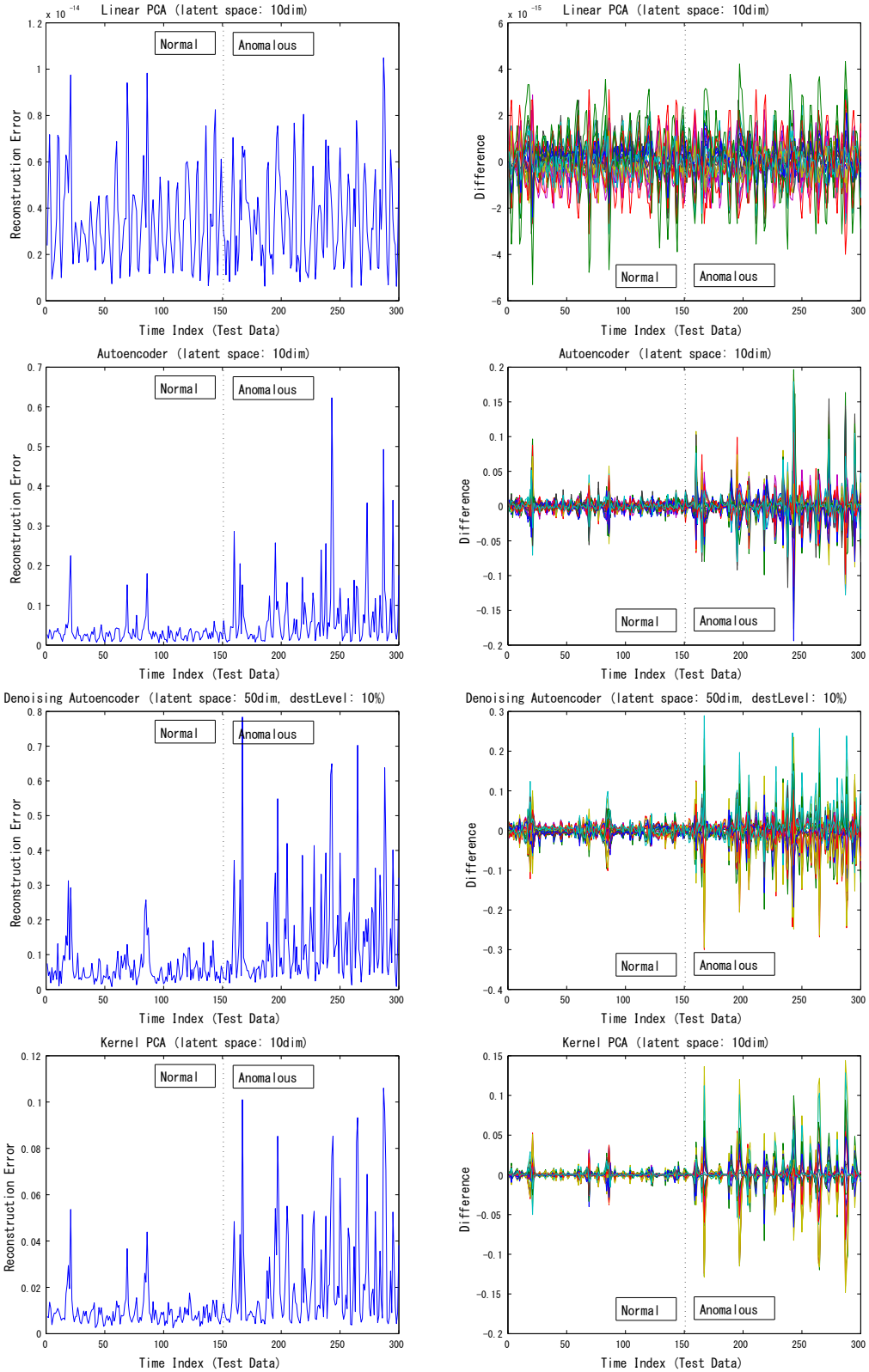

Figure 4: Result on Lorenz system data. The reconstruction error (left column) and the difference between the original and reconstructed data (right column) of linear PCA, an autoencoder, a denoising autoencoder and kernel PCA.

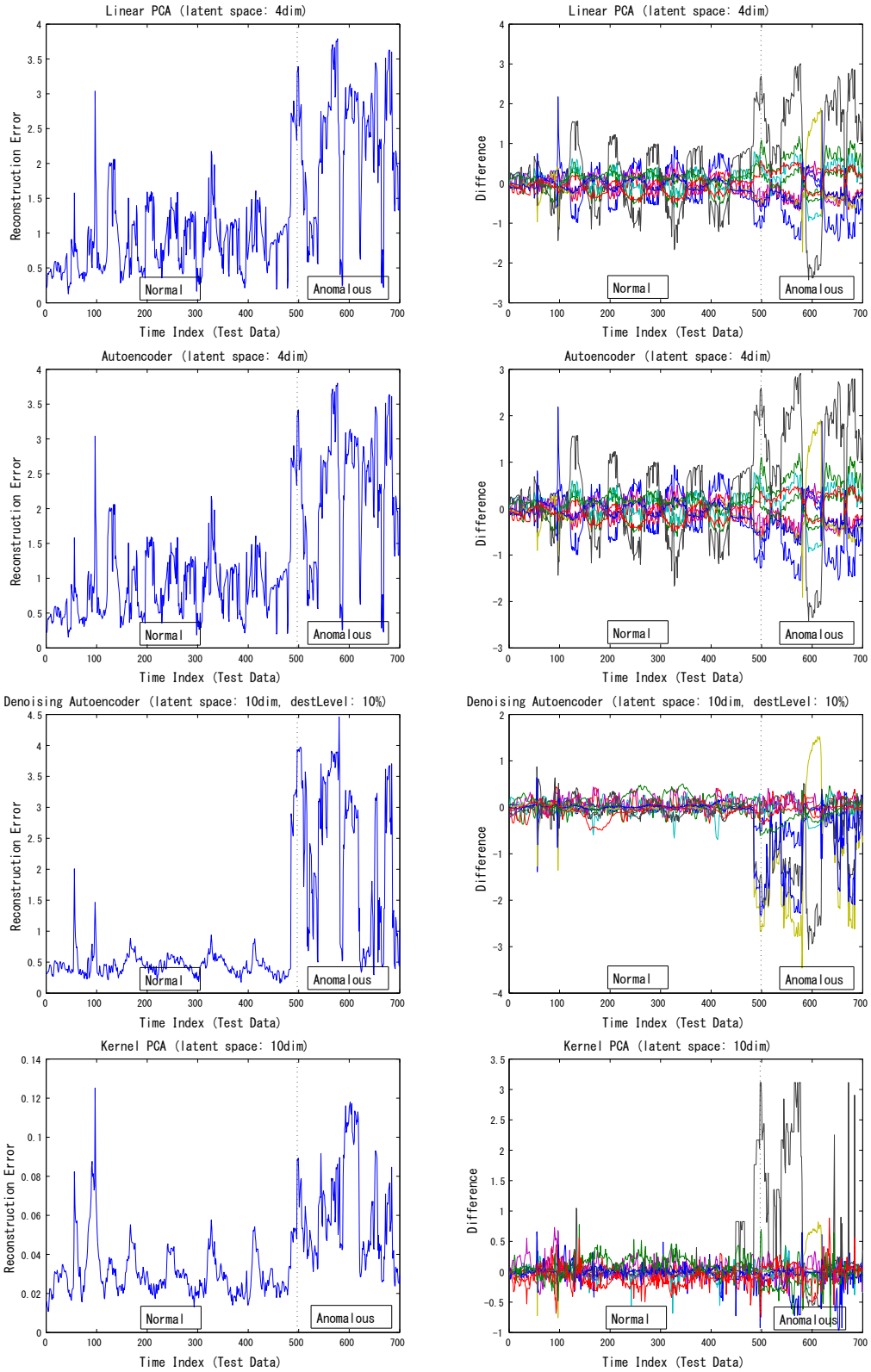

Figure 5: Result on Satellite-A data. The reconstruction error (left column) and the difference (right column) of linear PCA, an autoencoder, an denoising autoencoder and kernel PCA are shown from top to bottom.

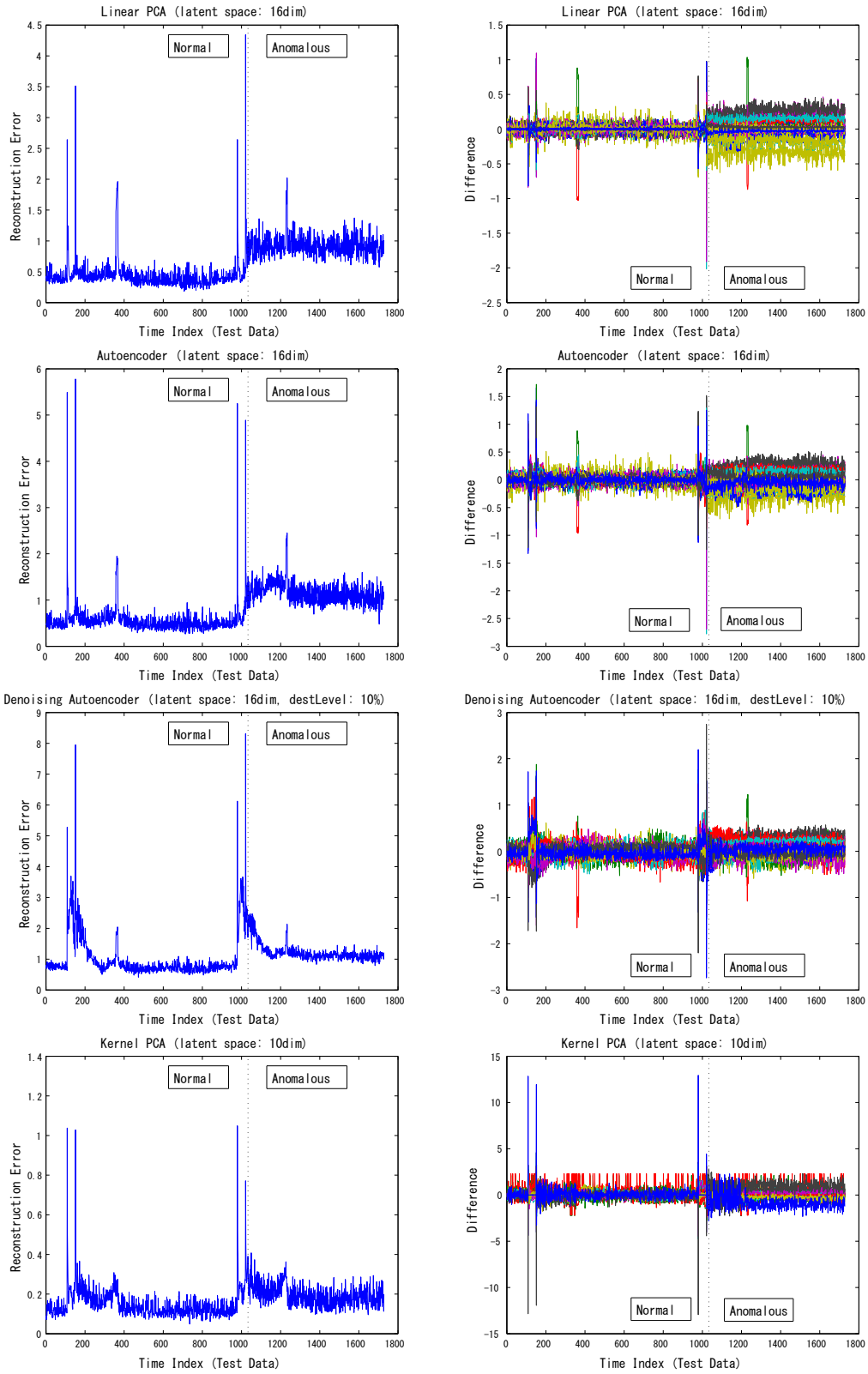

Figure 6: Result on Satellite-B data. The reconstruction error (left column) and the difference (right column) of linear PCA, an autoencoder, a denoising autoencoder and kernel PCA are shown from top to bottom.

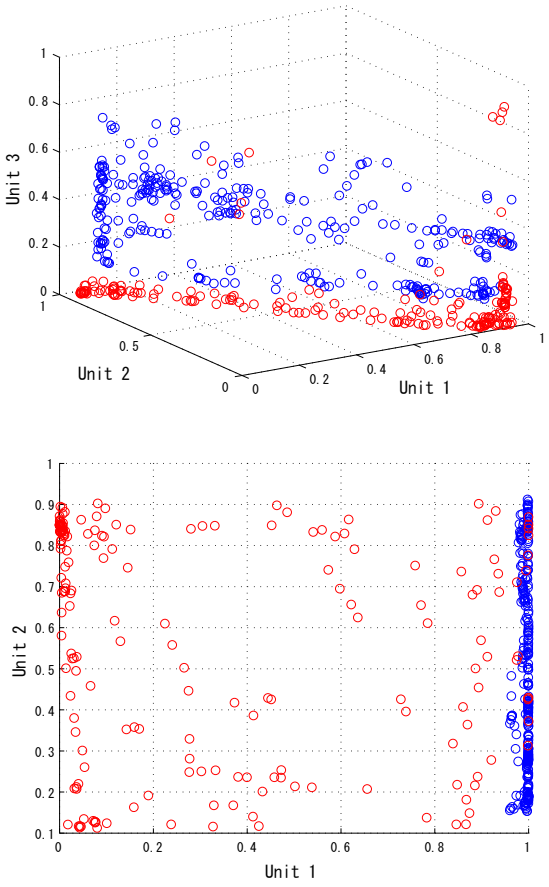

Figure 7: Top: An example of the activation in three of neurons in the hidden layer of the denoising autoencoder with normal input (blue) and anomalous input (red). Bottom: Another example of the activation in two of the neurons in the hidden layer of the denoising autoencoder with normal input (blue) and anomalous input (red). In these two figures, the hidden units of the denoising autoencoder activate in a different way with anomalous input.

[4] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504–507, 2006.

[5] H. Hoffmann. Kernel pca for novelty detection. Pattern Recognition, 40(3):863–874, 2007.

[6] B. Hwang and S. Cho. Characteristics of auto-associative mlp as a novelty detector. In Proceedings of the International Joint Conference on Neural Networks, volume 5, pages 3086–3091, 1999.

[7] N. Japkowicz, C. Myers, and M. Gluck. A novelty detection approach to classification. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, volume 1, pages 518–523, 1995.

[8] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. AIChE J., 37(2):233–243, 1991.

[9] M. Martinelli, E. Tronci, G. Dipoppa, and C. Balducelli. Electric power system anomaly detection using neural networks. In Knowledge-Based Intelligent Information and Engineering Systems, volume 3213 of Lecture Notes in Computer Science, pages 1242–1248. 2004.

[10] S. O. Song, D. Shin, and E. S. Yoon. Analysis of novelty detection properties of auto-associators. In Proceedings of the International Congress on Condition Monitoring and Diagnostic Engineering Management, pages 577–584, 2001.

[11] C. Surace, K. Worden, and G. Tomlinson. A novelty detection approach to diagnose damage in a cracked beam. In Proceedings of SPIE, pages 947–953, 1997.

[12] B. Thompson, R. Marks, J. Choi, M. El-Sharkawi, M.-Y. Huang, and C. Bunje. Implicit learning in autoencoder novelty assessment. In Proceedings of the 2002 International Joint Conference on Neural Networks, volume 3, pages 2878–2883, 2002.

[13] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, pages 1096–1103, 2008.

[14] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11:3371–3408, Dec. 2010.

[15] G. Williams, R. Baxter, H. He, S. Hawkins, and L. Gu. A comparative study of rnn for outlier detection in data mining. In Proceedings of the International Conference on Data Mining, page 709, 2002.

[16] T. Yairi, M. Inui, A. Yoshiki, Y. Kawahara, and N. Takata. Spacecraft telemetry data monitoring by dimensionality reduction techniques. In Proceedings of SICE Annual Conference, pages 1230–1234, Aug 2010.

[17] T. Yairi, T. Tagawa, and N. Takata. Telemetry monitoring by dimensionality reduction and learning hidden markov model. In Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space, 2012.