Variational Autoencoder based Anomaly Detection using Reconstruction Probability

Jinwon An jinwon@dm.snu.ac.kr
Sungzoon Cho zoon@snu.ac.kr

December 27, 2015

Abstract

We propose an anomaly detection method using the reconstruction probability from the variational autoencoder. The reconstruction probability is a probabilistic measure that takes into account the variability of the distribution of variables. The reconstruction probability has a theoretical background making it a more principled and objective anomaly score than the reconstruction error, which is used by autoencoder and principal components based anomaly detection methods. Experimental results show that the proposed method outperforms autoencoder based and principal components based methods. Utilizing the generative characteristics of the variational autoencoder enables deriving the reconstruction of the data to analyze the underlying cause of the anomaly.

1 Introduction

An anomaly or outlier is a data point which is significantly different from the remaining data. Hawkins defined an anomaly as an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism [5]. Analyzing and detecting anomalies is important because it reveals useful information about the characteristics of the data generation process. Anomaly detection is applied in network intrusion detection, credit card fraud detection, sensor network fault detection, medical diagnosis and numerous other fields [3].

Among many anomaly detection methods, spectral anomaly detection techniques try to find the lower dimensional embeddings of the original data where anomalies and normal data are expected to be separated from each other. After finding those lower dimensional embeddings, they are brought back to the original data space which is called the reconstruction of the original data. By reconstructing the data with the low dimension representations, we expect to obtain the true nature of the data, without uninteresting features and noise. Reconstruction error

of a data point, which is the error between the original data point and its low dimensional reconstruction, is used as an anomaly score to detect anomalies. Principal components analysis (PCA) based methods belong to this method of detecting anomalies [3].

With the advent of deep learning, autoencoders are also used to perform dimension reduction by stacking up layers to form deep autoencoders. By reducing the number of units in the hidden layer, it is expected that the hidden units will extract features that well represent the data. Moreover, by stacking autoencoders we can apply dimension reduction in a hierarchical manner, obtaining more abstract features in higher hidden layers leading to a better reconstruction of the data.

In this study we propose an anomaly detection method using variational autoencoders (VAE) [8]. A variational autoencoder is a probabilistic graphical model that combines variational inference with deep learning. Because VAE reduces dimensions in a probabilistically sound way, theoretical foundations are firm. The advantage of a VAE over an autoencoder and a PCA is that it provides a probability measure rather than a reconstruction error as an anomaly score, which we will call the reconstruction probability. Probabilities are more principled and objective than reconstruction errors and does not require model specific thresholds for judging anomalies.

2 Background

2.1 Anomaly detection

Anomaly detection methods can be broadly categorized in to statistical, proximity based, and deviation based [1].

Statistical anomaly detection assumes that data is modeled from a specified probability distribution. Parametric models such as mixture of Gaussians or Nonparametric models such as kernel density estimation can be used to define a probability distribution. A data point is defined as an anomaly if the probability of it being generated from the model is below a certain threshold. The advantage of such models is that it gives out probability as the decision rule for judging anomalies, which is objective and theoretically justifiable.

Proximity based anomaly detection assumes that anomalous data are isolated from the majority of the data. There are three ways in modeling anomalies in this way, which are clustering based, density based, and distance based. For clustering based anomaly detection, a clustering algorithm is applied to the data to identify dense regions or clusters that are present in the data. Next, the relationships of the data points to each cluster is evaluated to form an anomaly score. Such criteria include distance to cluster centroids and the size of the closest cluster. If the distance to cluster centroids is above a threshold or the size of the closest cluster is below

a threshold, the data point is defined as an anomaly. Density based anomaly detection define anomalies as data points that lie in sparse regions of the data. For example, if the number of data points within a local region of a data point is below a threshold, it is defined as an anomaly. Distance based anomaly detection uses measurements that are related to the neighboring data points of a given data point. K-nearest neighbor distances can be used in such a way where data points with large k-nearest neighbor distances are defined as anomalies.

Deviation based anomaly detection is mainly based on spectral anomaly detection, which uses reconstruction errors as anomaly scores. The first step is to reconstruct the data using dimension reduction methods such as principal components analysis or autoencoders. Reconstructing the input using k-most significant principal components and measuring the difference between its original data point and the reconstruction leads to the reconstruction error which can be used as an anomaly score. Data points with high reconstruction error are defined as anomalies.

2.2 Autoencoder and anomaly detection

An autoencoder is a neural network that is trained by unsupervised learning, which is trained to learn reconstructions that are close to its original input. An autoencoder is composed of two parts, an encoder and a decoder. A neural network with a single hidden layer has an encoder and decoder as in equation (1) and equation (2), respectively. W and b is the weight and bias of the neural network and σ is the nonlinear transformation function.

$$h = \sigma(W_{xh}x + b_{xh}) \tag{1}$$

$$z = \sigma(W_{hx}h + b_{hx}) \tag{2}$$

$$\|x - z\| \tag{3}$$

The encoder in equation (1) maps an input vector x to a hidden representation h by a an affine mapping following a nonlinearity. The decoder in equation (2) maps the hidden representation h back to the original input space as a reconstruction by the same transformation as the encoder. The difference between the original input vector x and the reconstruction z is called the reconstruction error as in equation (3). An autoencoder learns to minimize this reconstruction error. The training algorithm for the vanilla autoencoder is shown in algorithm 1 where f_θ and g_ϕ are multilayered neural networks for the autoencoder.

By using the hidden representation of an autoencoder as an input to another autoencoder, we can stack autoencoders to form a deep autoencoder [16]. To avoid trivial lookup table-like representations of hidden units, autoencoders reduces the number of hidden units. Autoencoders with various other regularization has also been developed. Contractive autoencoders use gradients of activations as penalty term and try to model data with sparse activations that only

Algorithm 1 Autoencoder training algorithm

INPUT: Dataset $x^{(1)}, \cdots, x^{(N)}$

OUTPUT: encoder f_ϕ, decoder g_θ

$\phi, \theta \leftarrow$ Initialize parameters

repeat

$E = \sum_{i=1}^{N} \|x^{(i)} - g_\theta(f_\phi(x^{(i)}))\|$ Calculate sum of reconstruction error

$\phi, \theta \leftarrow$ Update parameters using gradients of E(e.g. Stochastic Gradient Descent)

until convergence of parameters ϕ, θ

respond to the true nature of the data [13]. Denoising autoencoders uses a adds noise to the original input vector x and uses this noisy input $\hat{x}$ as the input vector. The difference between the resulting output, the reconstruction of the noisy input, and the original input is used as the reconstruction error. In short, this is training the autoencoder to reproduce the original input x from a noisy input $\hat{x}$. This allows the autoencoder to be robust to data with white noise and capture only meaningful patterns of the data [16].

Autoencoder based anomaly detection is a deviation based anomaly detection method using semi-supervised learning. It uses the reconstruction error as the anomaly score. Data points with high reconstruction are considered to be anomalies. Only data with normal instances are used to train the autoencoder. After training, the autoencoder will reconstruct normal data very well, while failing to do so with anomaly data which the autoencoder has not encountered. Algorithm 2 shows the anomaly detection algorithm using reconstruction errors of autoencoders.

Algorithm 2 Autoencoder based anomaly detection algorithm

INPUT: Normal dataset X, Anomalous dataset $x^{(i)}$ $i = 1, \cdots, N$, threshold α

OUTPUT: reconstruction error $\|x - \hat{x}\|$

$\phi, \theta \leftarrow$ train a autoencoder using the normal dataset X

for i=1 **to** N **do**

$reconstruction\ error(i) = \|x^{(i)} - g_\theta(f_\phi(x^{(i)}))\|$

if $reconstruction\ error(i) > \alpha$ **then**

$x^{(i)}$ is an anomaly

else

$x^{(i)}$ is not an anomaly

end if

end for

2.3 Variational Autoencoder

A variational autoencoder (VAE) is a directed probabilistic graphical model (DPGM) whose posterior is approximated by a neural network, forming an autoencoder-like architecture. Figure 1 shows a typical directed graphical model. In the VAE, the highest layer of the directed graphical model z is treated as the latent variable where the generative process starts. $g(z)$ represents the complex process of data generation that results in the data x, which is modeled in the structure of a neural network. The objective function of a VAE is the variational lowerbound of the marginal likelihood of data, since the marginal likelihood is intractable. The marginal likelihood is the sum over the marginal likelihood of individual data points $\log p_\theta(x^{(1)}, \cdots, x^{(N)}) = \sum_{i=1}^{N} \log p_\theta(x^{(i)})$, where the marginal likelihood of individual data points can be rewritten as follows.

$$\log p_\theta(x^{(i)}) = D_{KL}(q_\phi(z|x)||p_\theta(z)) + \mathcal{L}(\theta, \phi; x^{(i)}) \tag{4}$$

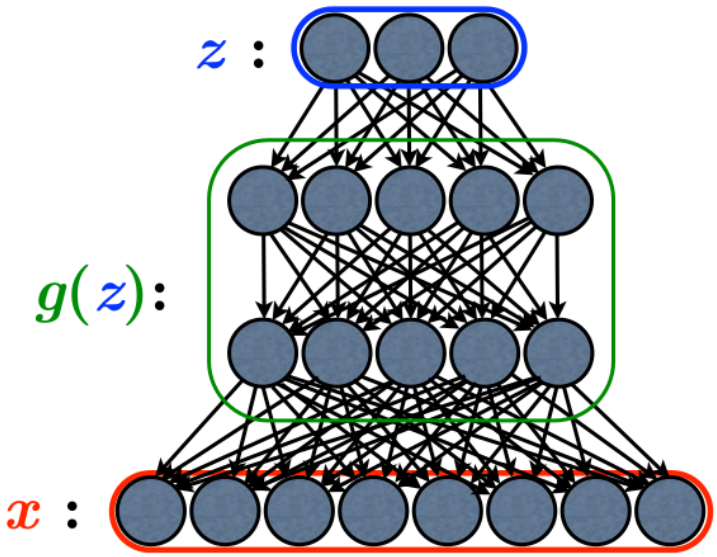

Figure 1: Directed probabilistic graphical model

$q_\phi(z|x)$ is the approximate posterior and $p_\theta(z)$ is the prior distribution of the latent variable z. The first term of the right hand side of equation (4) is the KL divergence of the approximate posterior and the prior. The second term of the right hand side of equation (4) is the variational lowerbound on the marginal likelihood of the data point i. Since the KL divergence term is

always bigger than 0, equation (4) can be rewritten as follows.

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(\theta, \phi; x^{(i)}) \tag{5}$$

$$= E_{q_\phi(z|x^{(i)})}[-\log q_\phi(z|x) + \log p_\theta(x|z)] \tag{6}$$

$$= -D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) + E_{q_\phi(z|x^{(i)})}[\log p_\theta(x|z)] \tag{7}$$

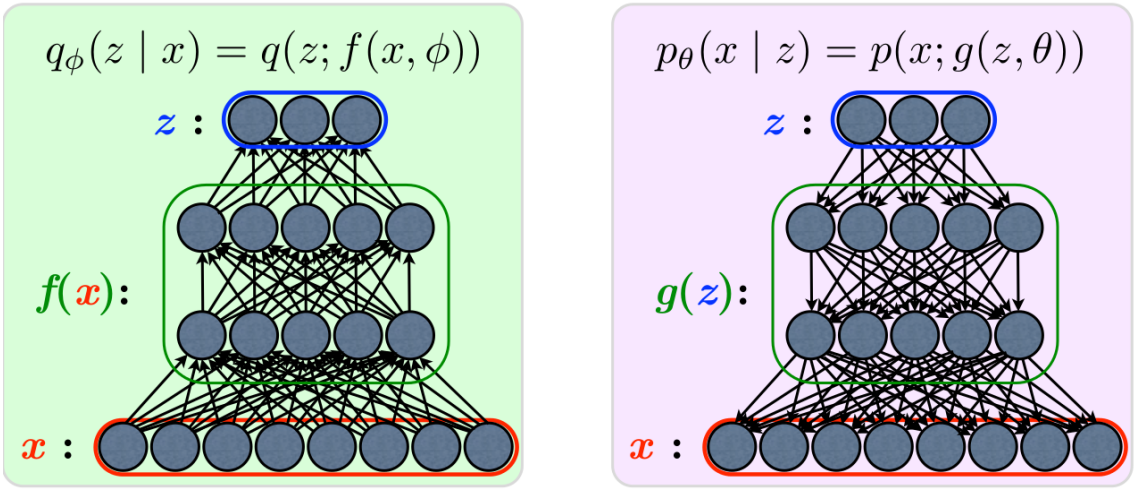

Figure 2: Encoder and decoder of a DPGM

$p_\theta(x|z)$ is the likelihood of the data x given the latent variable z. The first term of equation (7) is the KL divergence between the approximate posterior and the prior of the latent varaible z. This term forces the posterior distrbution to be similar to the prior distribution, working as a regularization term. The second term of equation (7) can be understood in terms of the reconstruction of x through the posterior distribution $q_\phi(z|x)$and the likelihood $p_\theta(x|z)$.

The VAE models the parameters of the approximate posterior $q_\phi(z|x)$ by using a neural network. This is where the VAE can relate to the autoencoder. As shown in figure 2, in the autoencoder analogy, the approximate posterior $q_\phi(z|x)$ is the encoder and the directed probabilistic graphical model $p_\theta(x|z)$ is the decoder. It is worth emphasizing that VAE models the parameters of the distribution rather than the value itself. That is, $f(x, \phi)$ in the encoder ouputs the parameter of the approximate posterior $q_\phi(z|x)$ and to obtain the actual value of the latent variable z, sampling from $q(z; f(x, \phi))$ is required. Thus the encoders and decoders of VAE can be called as ***probabilistic encoders and decoders***. $f(x, \phi)$ being a neural network represents the complex relationship between the data x and the latent variable z. To get the reconstruction $\hat{x}$, given the sample z, the parameter of $p_\theta(x|z)$ is obtained by $g(z, \theta)$ where the reconstruction $\hat{x}$ are sampled from $p_\theta(x; g(z, \theta))$. **To summarize, it is the distribution parameters that are being modeled in the VAE, not the value itself**. The choice for the distributions are open to any kind of parametric distribution. For the distribution of the

latent variable z, which are $p_\theta(z)$ and $q_\phi(z|x)$, the common choice is the isotropic normal, since it is assumed that the relationship among variables in the latent variable space is much more simple than the original input data space. The distributions of the likelihood $p_\theta(x|z)$, figure 3 depends on the nature of the data. If the data is of binary form, Bernoulli distribution is used. If the data is in continuous form, Multivariate Gaussian is used. Figure 3shows the structure of the VAE as a whole.

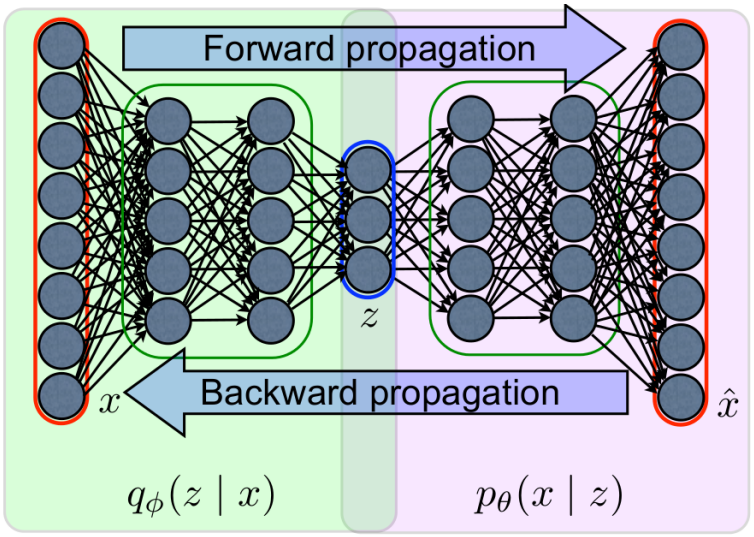

Figure 3: Variational autoencoder as an encoder and decoder

The main difference between a VAE and an autoencoder is that the VAE is a stochastic generative model that can give calibrated probabilities, while an autoencoder is a deterministic discriminative model that does not have a probabilistic foundation. This is obvious in that VAE models the parameters of a distribution as explained above.

Backpropagation is used to train the VAE. Since the second term on equation (7) should be calculated through Monte Carlo methods, Monte Carlo gradient methods have to be used. However it is known that traditional Monte Carlo gradient methods used to optimize variational lower bound suffers from very high variance and is thus not suitable for use [10]. VAE overcomes this by using a reparameterization trick that uses a random variable from a standard normal distribution instead of a random variable from the original distribution. The random variable $z \sim q_\phi(z|x)$ is reparmeterized by a deterministic transformation $h_\phi(\epsilon, x)$ where ϵ is from a standard normal distribution.

$$\tilde{z} = h_\phi(\epsilon, x) \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, 1) \tag{8}$$

The reparmeterization should ensure that $\tilde{z}$ follows the distribution of $q_\phi(z|x)$. This is much

more stable than directly using the Monte Carlo gradient method. The algorithm for training the VAE is shown in algorithm 3.

Algorithm 3 Variational autoencoder training algorithm

INPUT: Dataset $x^{(1)}, \cdots, x^{(N)}$

OUTPUT: probabilistic encoder f_ϕ, probabilistic decoder g_θ

$\phi, \theta \leftarrow$ Initialize parameters

repeat

for i=1 **to** N **do**

Draw L samples from $\epsilon \sim \mathcal{N}(0, 1)$

$z^{(i,l)} = h_\phi(\epsilon^{(i)}, x^{(i)}) \quad i = 1, \cdots, N$

end for

$E = \sum_{i=1}^{N} -D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) + \frac{1}{L}\sum_{l=1}^{L}(\log p_\theta(x^{(i)}|z^{(i,l)}))$

$\phi, \theta \leftarrow$ Update parameters using gradients of E (e.g. Stochastic Gradient Descent)

until convergence of parameters ϕ, θ

3 Proposed method

We propose an anomaly detection method that uses a VAE to calculate the anomaly score in terms of a probability which we call the reconstruction probability.

3.1 Algorithm

The algorithm of the proposed method is in algorithm 4. The anomaly detection task is conducted a semi-supervised framework, using only data of normal instances for training the VAE. The probabilistic encoder f_ϕ and decoder g_θ both paramterizes an isotropic normal distribution in the latent variable space and the original input variable space, respectively. For testing, a number of samples are drawn from the probabilistic encoder of the trained VAE. For each sample from the encoder, the probabilistic decoder outputs the mean and variance parameter. Using these parameters, the probability of the original data generating from the distribution is calculated. The average probability is used as an anomaly score and is called the reconstruction probability. **The reconstruction probability that is calculated here is the Monte Carlo estimate of $E_{q_\phi(z|x)}[\log p_\theta(x|z)]$, the second term of the right hand side of equation (7).** Data points with high reconstruction probability is classified as anomalies.

Algorithm 4 Variational autoencoder based anomaly detection algorithm

INPUT: Normal dataset X, Anomalous dataset $x^{(i)}$ $i = 1, \cdots, N$, threshold α

OUTPUT: reconstruction probability $p_\theta(x|\hat{x})$

$\phi, \theta \leftarrow$ train a variational autoencoder using the normal dataset X

for i=1 **to** N **do**

$\boldsymbol{\mu}_{\boldsymbol{z}^{(i)}}, \boldsymbol{\sigma}_{\boldsymbol{z}^{(i)}} = f_\theta(z|x^{(i)})$

draw L samples from $z \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}^{(i)}}, \boldsymbol{\sigma}_{\boldsymbol{z}^{(i)}})$

for l=1 **to** L **do**

$\boldsymbol{\mu}_{\hat{\boldsymbol{x}}^{(i,l)}}, \boldsymbol{\sigma}_{\hat{\boldsymbol{x}}^{(i,l)}} = g_\phi(x|z^{(i,l)})$

end for

$\textit{reconstruction probability}(i) = \frac{1}{L}\sum_{l=1}^{L} p_\theta(x^{(i)}|\boldsymbol{\mu}_{\hat{\boldsymbol{x}}^{(i,l)}}, \boldsymbol{\sigma}_{\hat{\boldsymbol{x}}^{(i,l)}})$

if $\textit{reconstruction probability}(i) < \alpha$ **then**

$x^{(i)}$ is an anomaly

else

$x^{(i)}$ is not an anomaly

end if

end for

3.2 Reconstruction Probability

The reconstruction probability is calculated by the stochastic latent variables that derive the parameters of the original input variable distribution. What is being reconstructed is the parameters of the input variable distribution, not the input variable itself. This is essentially the probability of the data being generated from a given latent variable drawn from the approximate posterior distribution. Because a number of samples are drawn from the latent variable distribution, this allows the reconstruction probability to take into account the variability of the latent variable space, which is one of the main distinctions between the proposed method and the autoencoder based anomaly detection. It is possible to use other distributions of the input variable space that fits for the data. For continuous data, the normal distribution can be used as in algorithm 4. For binary data, Bernoulli distributions can be used. In case of the distribution of the latent variable space, a simple continuous distribution such as a isotropic normal distribution is preferred. This can be justified by the assumption of spectral anomaly detection that the latent variable space is much more simple compared to the input variable space.

3.3 Difference from an autoencder based anomaly detection

Reconstruction probability is different from reconstruction error from an autoencoder in two ways. First, latent variables are stochastic variables. In autoencoders, latent variables are defined by deterministic mappings. However since VAE uses the probabilistic encoder for modeling the distribution of the latent variables rather than the latent variable itself, the variability of the latent space can be taken into account from the sampling procedure. This extends the expressive power of the VAE compared to the autoencoder in that even though normal data and anomaly data might share the same mean value, the variability can differ. Presumably anomalous data will have greater variance and show lower reconstruction probability. Since deterministic mappings of autoencoders can be thought as a mapping to the mean value of a dirac delta distribution, the autoencoder lacks the ability to address variability.

Second, reconstructions are stochastic variables. Reconstruction probability considers not only the difference between the reconstruction and the original input, but also the variability of the reconstruction by considering the variance parameter of the distribution function. This property enables selective sensitivity to reconstruction according to variable variance. Variables with large variance would tolerate large differences in the reconstruction and the original data as normal behavior while those with small variance will lower the reconstruction probability significantly. This is also a feature that the autoencoder lacks in due to its deterministic nature.

Third, reconstructions are probability measures. Autoencoder based anomaly detection uses reconstruction errors as anomaly scores, which are difficult to calculate if the input variables are heterogeneous. To sum up the differences of heterogeneous data, a weighted sum is required. The problem is, a universal objective method to decide the appropriate weight is not available since the weights will be different depending on the data you have. Moreover, even after the weights are decided, deciding the threshold for reconstruction error is cumbersome. There will be no clear cut threshold that is objective. In contrast, the calculation of the reconstruction probability doesn't require weighting of the reconstruction error of heterogeneous data since the probability distribution of each variable allows them to be separately calculated by its own variability. Also a 1% probability is always a 1% for any data. Thus deciding the threshold of the reconstruction error is much more objective, reasonable, and easy to understand than that of the reconstruction error.

4 Experimental Results

VAE based anomaly detection with reconstruction probability is compared with other reconstruction based methods such as autoencoder based and PCA based methods.

4.1 Datasets and setup

Datasets used for anomaly detection are MNIST dataset [9] and KDD cup 1999 network intrusion dataset (KDD) [6]. Datasets are divided into normal data and anomaly data according to their class labels. To apply Semi-supervised learning, the training data consists of 80% of the normal data and the test data consists of remaining 20% of the normal data and all of the anomaly data. Thus models are trained only with normal data and tested with both normal and anomalous data.

For the MNIST dataset, a model was trained with each digit class labeled as an anomaly and the other digits labeled as normal. This results in datasets with 10 different anomalies. We will refer the digit class i that is labeled as an anomaly as the anomaly digit i. Total number of data are 60,000, with the same number of instances for each digit. Only min max scaling was applied as preprocessing to make each pixel value to be within 0 and 1.

The KDD cup dataset consists of classes of five main classes, which are DoS, R2L, U2R, Probe and *Normal*. The first 4 classes are anomalies and the *Normal* class is normal. Each of the first 4 classes is considered as anomalies. For each anomaly class, normal data were defined in two different ways. The first is defining normal data as the data with class of *Normal* only. Since the model is trained only with normal data, this yields an identical model for each anomaly class. Another definition of normal data is all the data except for the specified anomaly class. This yields different training data for each anomaly class and also has much more training data than the former definition of normal data. We will refer the first method of defining normal data as *only normal method* and the second will be called *except anomaly method*. The number of instances for each class is shown in table 1. For categorical variables, one hot coding was used to transform it into numeric values. For numeric variables, standardization of 0 mean and unit variance was applied as preprocessing.

Table 1: KDD dataset

class	number of instances
Normal	972,770
DOS	3,914,580
probe	41,070
R2L	11,260
U2R	520

4.2 Model setup

For the VAE, the encoder and decoder are both a single hidden layer with 400 dimensions. The latent dimension is 200 dimensions. For the autoencoder(AE), we used a two hidden layer denoising autoencoder with 400, 200 dimensions for the first and second hidden layer, respectively. The second layer was trained by stacking the previous layer output. For the PCA, we used linear PCA (PCA) and a kernel PCA (kPCA) with a Gaussian kernel. Parameters of the Gaussian kernel was estimated using cross-validation. VAE uses the reconstruction probability as the anomaly score, while the other models uses the reconstruction error as the anomaly score.

4.3 Performance evaluation

Area under the curve of the receiver operating characteristic (AUC ROC) and average precision or Area under the curve of the precision recall curve (AUC PRC) and f1 score are used for evaluation. The f1 score was obtained by deciding the threshold for the binary decision from the validation dataset f1 score.

4.3.1 MNIST dataset

Table 2 shows the performance of each model for the anomaly datasets. VAE outperforms other models most of the time. For all models, performance is low When the digit 1, 7 and 9 is the anomaly digit. The other performance measures of the VAE in detail are shown in table 3. It can be seen that cases when digits 1 and 7, 9 is the anomaly digit show low performance in AUC PRC and f1 score as also. This seems related with how the data structure itself is hard to reconstruct. Analyzing the reconstructions reveals the possible reason for this result.

4.3.2 Reconstruction of MNIST dataset

Figure 4 shows the data samples and its reconstructions that have a low reconstruction probability for each anomaly digit. That is, it shows the detected anomalies for each anomaly digit data. It can be easily noticed that except for 1, 7 and 9, the VAE does not reconstruct the data of the anomaly digit well, correctly leading to defining it as an anomaly. For anomaly digit 1, it is seen that 1 is not detected as anomalies. 7 and 9 also follow similarly with other digits judged as anomalies. The reason for the VAE not working for these digits can be understood if we see the data samples and its reconstructions that have a high reconstruction probability for each anomaly digit shown in figure 5. That is, these data samples are judged to be normal data.

All anomaly digits except for anomaly digit 1 show only samples of 1 to have the highest reconstruction probability. Even in the case where 1 was in the anomaly data and not given to the training data as in anomaly digit 1 shows 1 with 7 and 9 with high reconstruction probability.

Table 2: MNIST AUC ROC performance

Anomaly digit	VAE	AE	PCA	kPCA
0	0.917	0.825	0.785	0.694
1	0.136	0.135	0.205	0.231
2	0.921	0.874	0.798	0.801
3	0.781	0.761	0.632	0.638
4	0.808	0.727	0.682	0.702
5	0.862	0.792	0.627	0.598
6	0.848	0.812	0.733	0.720
7	0.596	0.508	0.512	0.560
8	0.895	0.869	0.493	0.502
9	0.545	0.548	0.41	0.420

Table 3: MNIST VAE evaluation

Anomaly digit	AUC PRC	f1 score
0	0.517	0.537
1	0.063	0.205
2	0.644	0.598
3	0.251	0.332
4	0.337	0.381
5	0.325	0.427
6	0.432	0.433
7	0.148	0.212
8	0.499	0.49
9	0.104	0.21

From this it can be inferred that because the structure of 1 being a single vertical stroke is so simple, the VAE learned that structure from the other parts of the data. For example, a vertical stroke is included in almost any digit, in the center in 4, in the right side in 7 and 9 if written in a rigid manner without much curvature. This might have given data for the VAE to learn the components of structures. Even though the VAE used in this experiment is a rather shallow one with three hidden layers, it still seems that it has worked as a hierarchical model capturing features that make up the structure of the data. This is evident when looking at the samples shown in figure 5 for anomaly digit 1, where 7 and 9 appear to have high reconstruction probabilities when 1 is absent. The low performance of anomaly digit 7 and 9 can be understood in a similar sense. The vertical stroke make up a large part of 7 and 9 since it has less protruding parts from the vertical stroke compared to other digits as shown in figure 5. Additionally for 7, it seems that a particular way of writing 7 (with a second horizontal stroke in the middle of the digit 7) seems to be detected as anomalies.

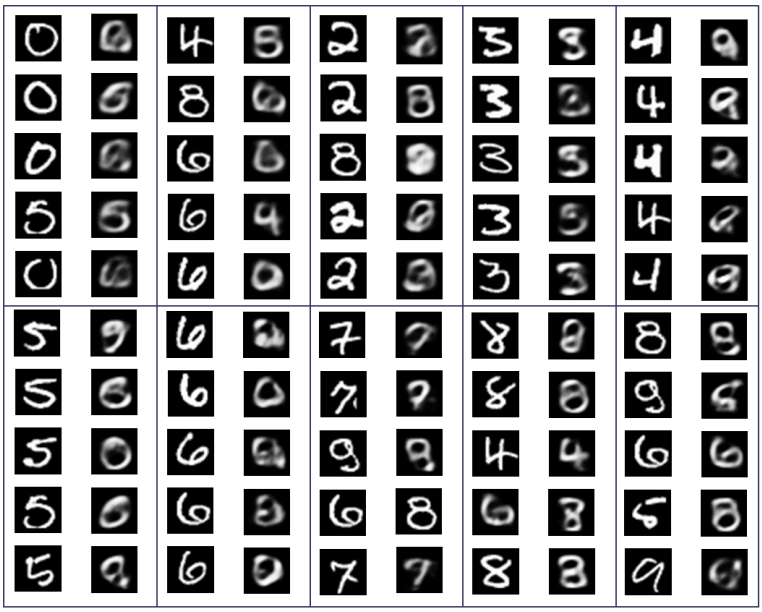

Figure 4: Reconstruction of digits with low reconstruction probability

The upper row section are reconstructions of anomaly digit 0 to 4. The lower row section, 5 to 9. The left column of each section is the original data and the right column is the reconstruction

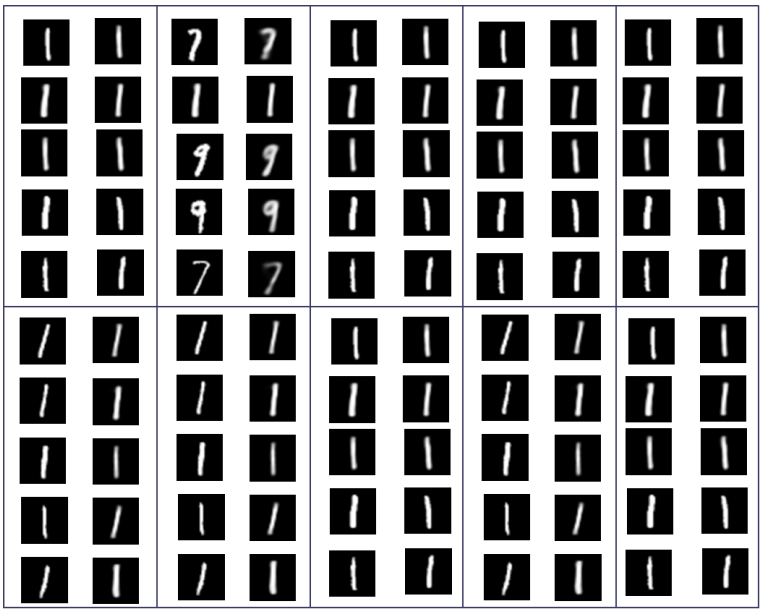

Figure 5: Reconstruction of digits with high reconstruction probability

same configuration as Figure 4.

4.4 KDD dataset

Table 4 and table 5 shows the performance of VAE of each model on the anomaly dataset. Table 4 are models trained with only the *Normal* class data. All models are trained with the same dataset regardless of the anomaly class. Table 5 are models that are trained with all the data except for the anomaly class data, which means the training data of models differ by anomaly classes. The VAE performs better than the other models except for the case in training with only *Normal* classes with Probe as the anomaly class where they are at par. PCA seems to lack in performance implying that linear relationships of data are not sufficient to capture the underlying structure of the data. Kernel PCA has not been much successful either.

4.4.1 Comparison between *normal only method* and *anomaly except method*

anomaly except method shows better performance except for the case when the anomaly class is DoS. This may be due to the fact that DoS class is the class with the most instances. table 1 shows that DoS class data makes up nearly 80% of the total data. Not using DoS class data for training the VAE would severely under-fit the VAE, resulting in a model that can not distinguish DoS class data from other classes. For other anomaly classes, incorporating other data, most notably the bulky DoS class data helps improve performance from *normal only method* where the model with only *Normal* data was the training data. Also it can be seen that the autoencoder in *anomaly except method* shows better performance than the VAE in *normal only method* in anomaly classes R2L, U2R and nearly matches Probe. This exemplifies the axiom, more data is often better than a better algorithm.

4.4.2 Comparison between anomaly classes

Table 6 and table 7 shows other performance evaluation metrics of the VAE. DoS shows both high AUC ROC and high AUC PRC. But for R2L and U2R the AUC ROC score is good but the AUC PRC score is bad. This is due to size of the class of R2L and U2R. AUC ROC does not take into account the actual number of data but its percentage of the data. This makes anomaly data that are very small in size to peform better in terms of AUC ROC. However, AUC PRC takes into account the actual number of the data of the anomaly class. Because there are so few data, it is hard for the model to differentiate them from the normal data. U2R, has only 520 samples which is abount 0.001% of the total data. This makes the AUC PRC very low. Even though the f1 score,the peak of AUC PRC, is much bigger than AUC PRC it is still low compared to other anomaly classes.

Table 4: KDD AUC ROC (only normal)

Anomaly	VAE	AE	PCA	kPCA
DoS	0.795	0.727	0.585	0.590
R2L	0.777	0.773	0.705	0.712
U2R	0.782	0.781	0.698	0712
Probe	0.944	0.946	0.832	0.821

Table 5: KDD AUC ROC (except anomaly)

Anomaly	VAE	AE	PCA	kPCA
DoS	0.744	0.685	0.785	0.780
R2L	0.786	0.782	0.502	0.514
U2R	0.921	0.806	0.717	0.760
Probe	0.970	0.968	0.647	0.645

Table 6: KDD VAE (only normal)

Anomaly	AUC ROC	AUC PRC	f1 score
DoS	0.795	0.944	0.981
R2L	0.777	0.17	0.406
U2R	0.782	0.084	0.324
Probe	0.944	0.751	0.791

Table 7: KDD VAE (except anomaly)

Anomaly	AUC ROC	AUC PRC	f1 score
DoS	0.744	0.935	0.979
R2L	0.786	0.135	0.389
U2R	0.921	0.0096	0.347
Probe	0.970	0.706	0.720

5 Conclusion

We have introduced a anomaly detection method using the reconstruction probability from the variational autoencoder. The reconstruction probability incorporates the probabilistic characteristics of the variational autoencoder by taking into account the concept of variability. The reconstruction probability being a probability measure makes it a much more objective and principled anomaly score than the reconstruction error of autoencoder and PCA based methods. Experimental results show that the proposed method outperforms autoencoder based and PCA based methods. Due to its generative characteristics, it is also possible to derive the reconstruction of the data to analyze the underlying cause of the anomaly.

6 Acknowledgment

This work was supported by the BK21 Plus Program(Center for Sustainable and Innovative Industrial Systems, Dept. of Industrial Engineering, Seoul National University) funded by the Ministry of Education, Korea (No. 21A20130012638), the National Research Foundation(NRF) grant funded by the Korea government(MSIP) (No. 2011-0030814), and the Institute for Industrial Systems Innovation of SNU.

References

[1] Charu C Aggarwal. *Outlier analysis*. Springer Science & Business Media, 2013.

[2] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[4] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, pages 2962–2970, 2015.

[5] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[6] Seth Hettich and SD Bay. The uci kdd archive [http://kdd. ics. uci. edu]. irvine, ca: University of california. *Department of Information and Computer Science*, page 152, 1999.

[7] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[9] Yann LeCun and Corinna Cortes. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.

[10] John Paisley, David Blei, and Michael Jordan. Variational bayesian inference with stochastic search. *arXiv preprint arXiv:1206.6430*, 2012.

[11] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[12] Salah Rifai, Grégoire Mesnil, Pascal Vincent, Xavier Muller, Yoshua Bengio, Yann Dauphin, and Xavier Glorot. Higher order contractive auto-encoder. In *Machine Learning and Knowledge Discovery in Databases*, pages 645–660. Springer, 2011.

[13] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 833–840, 2011.

[14] Tara N Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. Auto-encoder bottleneck features using deep belief networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 4153–4156. IEEE, 2012.

[15] Takaaki Tagawa, Yukihiro Tadokoro, and Takehisa Yairi. Structured denoising autoencoder for fault detection and analysis. In *Proceedings of the Sixth Asian Conference on Machine Learning*, pages 96–111, 2014.

[16] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.