Semi-Supervised Anomaly Detection by Variational Autoencoder

Rotem Bar

Afeka College of Engineering,

Department Of Intelligent Systems

Tel-Aviv, Israel

rotembaruch@gmail.com

Uria Levkovich

Afeka College of Engineering

Department of Intelligent Systems

Tel-Aviv, Israel

urialevko@gmail.com

Abstract Anomaly detection has many applications in security areas and has become an active research issue of great concern in recent years. The purpose of this article is to present an approach for semi-supervised Anomaly detection based on Variational Autoencoder (VAE). VAE is a class of deep generative models trained by maximizing the lower bound of data distribution. This paper demonstrates the potential of using VAE for anomaly detection based on the reconstruction error of the output. Our work focuses on network attacks detection by examining the kddcup99 dataset. We performed several experiments to address our research questions, such as exmine the encoder and decoder size, the latent space dimension, and adding weight to each component in the loss function. The model presents Accuracy results of 85-98%.

Keywords – Variational Autoencoder; Anomaly Detection; Latent Space; Representing Learning

1. Introduction

With the rapid development of the Internet of Things (IoT) and big data technologies, anomaly detection, also known as outlier detection, has played an increasingly important role, where defective products or failures can be detected as abnormal samples. Anomaly detection [1] can be categorized into three classes depending on the labels: Supervised [2], Semi-supervised [3], and Unsupervised [4]. In our work, we will examine the Semi-supervised approach.

Semi-supervised anomaly detection trains a model with only normal samples and gets a standard "normal" model to learn the characteristics of normal samples. Test samples examined by the "normal" model, samples with significant differences classified as anomalies.

Due to the importance of effective representations [5], representative learning has made significant contributions and demonstrated success in anomaly detection [6]. Representative learning aims to learn a set of basis vectors that can encode all feature data into compact form with a linear or non-linear combination. In applying anomaly detection, the maximum reconstruction error of test samples is calculated to discriminate whether it is abnormal. The ability of this method to reveal the underlying structure of data is the key to solving the problem of anomaly detection. Deep generative models are the most consistent with this representative learning task. Recently, generative adversarial networks (GANs) [7] and variational auto-encoders (VAEs) [9] are the most known for this task. VAE is not only capable of generating a feature output close to the original input, effecting similarity information of similar data, but also providing latent feature vectors.

In this paper, we examine VAE for anomaly detection. We will explore several aspects by defining several experiments and analyzing them. The experiments will be performed on the KDD-CUP'99 dataset.

2. Variational Autoencoder

VAE developed from Auto-Encoder (AE) [8] trained to reconstruct the input data. Auto-Encoder consists of an encoder and a decoder. The encoder maps the input data to a latent variable while reconstructing the input data with the latent variable. Compared with VAE [9], AE uses a simple network layer to denote the latent variable. Thus, Auto-Encoder is more considered a deterministic model that cannot generate new samples. To overcome Auto-Encoder's weakness, VAE adds a variational constrain that the latent variable z is subject to a normal distribution, and the decoder starts with sampling from the distribution. Accordingly, VAE can map the training data to a normal distribution and generate new samples from the distribution.

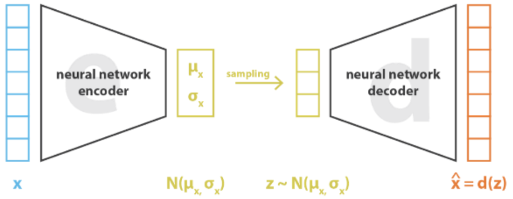


Fig 1: Variational Auto-Encoder Structure

In Fig. 1, $x/\hat{x}$ denote the input/reconstruct data, $\mu/\sigma^{\,2}$ denote Gaussian distribution's mean/variance of latent variable $\boldsymbol{z}$, $\hat{\boldsymbol{z}}$ represents the sample from $N(\mu/\sigma^{\,2})$. The purpose of VAE is to train a network that reconstructs its input data x with $\hat{x}$ as close as possible by minimizing:

$$(1) \qquad L(x/\hat{x}) = \parallel x - \hat{x} \parallel$$

As shown in Fig. 1, VAE is a directed probabilistic graphical model consisting of an encoder and a decoder. Different from Auto-Encoder, VAE is a stochastic generative model in which encoder and decoder are given by probabilistic function $q(z|x)$ and $p(z|x)$. $q(z|x)$ is the approximate posterior called adversarial model or encoder, while $p(z|x)$ is the likelihood of x given z. After fully training, the encoder is learned to approximate the posterior distribution and be able to map the input data x to the latent space. Unlike Auto-Encoder, VAE doesn't represent the latent space with simple value but maps input data to a stochastic variable z. z is subject to Gaussian distribution, which is determined by its mean μ and variance $\sigma^{\,2}$. After encoding, $\hat{z}$ sampled from $N(\mu/\sigma^{\,2)})$, and the decoder is trained to output the reconstructed result with $\hat{z}$. By using the variational inference approach [10] we find that the marginal likelihood of the data is given by:

$$(2) \quad \boldsymbol{log\left(p_\theta(X)\right) = KL\left(q_\phi(z|x) \parallel p_\theta(z|x)\right)} - \mathbb{E}_{q_\phi(z|x)}\boldsymbol{[log\, p_\theta(x|z)] - KL\left(q_\phi(z|x) \parallel p_\theta(z)\right)}$$

Where $\boldsymbol{p_\theta(x|z)}$ represents the true posterior distribution, The first KL divergence expression measures the distance of two distributions. Due to the non-negativity of the KL function, The KL divergence can be obtained straightly if the prior $p(x|z)$ Is subject to a normal distribution $N(\mu/\sigma^{\,2})$., The two other terms in equation 2 are called the Evidence Lower Bound (ELBO).

$$(3) \quad \boldsymbol{\mathcal{L}(\theta,\phi)} = \mathbb{E}_{q_\phi(z|x)}\boldsymbol{[log\, p_\theta(x|z)] - KL\left(q_\phi(z|x) \parallel p_\theta(z)\right)}$$

The left expression equation 3 in the ELBO is the expected log-likelihood of reconstruct data and the KL expression in the ELBO represent the KL measure between approximate posterior and the prior. Because the KL terms are always greater than zero the lower bound of the log evidence in equation 1 is dependent on the ELBO. VAE tries to maximize the log evidence as much as it can by maximizing the ELBO, this leads to the final objective loss function of VAE that expressed as:

$$(4) \quad \boldsymbol{\mathcal{L}(\theta,\phi)} = -\mathbb{E}_{q_\phi(z|x)}\boldsymbol{[log\, p_\theta(x|z)] + KL\left(q_\phi(z|x) \parallel p_\theta(z)\right)}$$

Before deciding, we need to sample $\hat{z}$ from an unknown distribution $\boldsymbol{q(z|x)}$. And it's hard to take the derivative of the sampling process, which makes the network unable to use gradient-based training. To solve it, we will use the reparameterization trick:

$$(5) \qquad Z = \mu + \sigma \times \varepsilon$$

Where $\varepsilon \sim N(0,1)$ represents the sample from the standard normal distribution, μ, and $\sigma \times \varepsilon$ denote the mean and variance of the latent variable distribution obtained by the encoder of VAE. Through the reparameterization trick, as shown in Fig.1, VAE set the sampling process independent of the network to be trained through gradient-based methods directly.

2.1. Beta Variational Autoencoder

β-VAE [11] is a modification of the Variational Autoencoder with a special emphasis to discover disentangled latent factors. Following the same incentive in VAE, we want to maximize the probability of generating real data while keeping the distance between the real and estimated posterior distributions small. The loss function of β-VAE is defined as:

$$(6) \quad \boldsymbol{\mathcal{L}_\beta(\theta,\phi)} = -\mathbb{E}_{q_\phi(z|x)}\boldsymbol{[log\, p_\theta(x|z)]} + \beta\boldsymbol{KL\left(q_\phi(z|x) \parallel p_\theta(z)\right)}$$

When β = 1 the loss function is equivalent to VAE whereas β>1 applies a greater constraint on the latent bottleneck and limits the representation capacity of z. The higher β encourages more efficient latent encoding but can create a trade-off between reconstruction quality

Variational Autoencoder with reconstruction loss weight (Alpha Variational Autoencoder)

To avoid Posterior collapse [12] many prior works used a variational Autoencoder with reconstruction loss weight or α-VAE [13]. In short, α-VAE is a modification of the variational Autoencoder. Following the same incentive in VAE, we want to maximize the probability of generating real data, while keeping the distance between the real and estimated posterior distributions small. The loss function of α -VAE is defined as:

$$(7) \quad \boldsymbol{\mathcal{L}}_{\beta}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \alpha \times (-\mathbb{E}_{q_{\phi}(z|x)}[\boldsymbol{log}\, \boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})]) + \boldsymbol{KL}(\boldsymbol{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \parallel \boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{z}))$$

When α = 1 the loss function is equivalent to VAE and when α >1 it applies a stronger constraint on the reconstruction ability. A higher α encourages more efficient reconstruction quality but can create a trade-off between latent encoding

3. Related Work

This section will review the studies that use representative learning for the anomaly detection task. Sun, J [14] proposes a sparse representation framework that learns dictionaries based on the latent space of variational auto-encoder. The architecture model builds a dictionary learned from the latent space. Experiments were performed on the KDD-CUP, MNIST, and UCSD datasets. The experimental results demonstrate that the proposed algorithm outperforms competing algorithms in all kinds of anomaly detection tasks.

Nguyen, Q.P [15] present a new technique, Gradient-based Explainable (GEE) Variational Autoencoder for Network Anomaly Detection. GEE comprises VAE and a gradient-based fingerprinting approach for explaining anomalies. Evaluation of GEE on the UGR dataset effectively detects different anomalies and identifies fingerprints that are good representations of these various attacks.

Gonzalez, D. [16] Work compared between the basic Autoencoder and the VAE, in Gonzalez, D, the data set examined is UK-DALE, which contains data of the electricity consumption among elderly. The experiment results show significantly better AUC and F1 scores to the VAE model. The VAE managed to present better data representation than the simple Autoencoder. In addition, the dimensional reduction was performed from input size at 24 to latent space of 2.

4. Methodology

This section will describe the dataset used in our experiments, the evaluation metrics, the anomaly classification role we used, and the experiment process.

4.1. Dataset

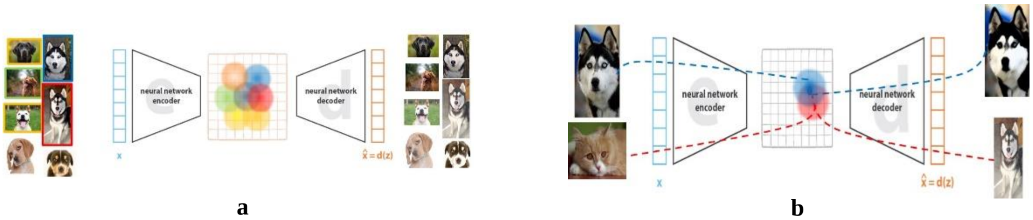
KDD-CUP'99 [17] dataset was introduced in 1999, and it is the primary benchmark of the network intrusion field, which lays a foundation for the research of computer network intrusion. KDD-CUP'99 dataset comprises 5 million network connection records and, each row in the dataset represent a network connection. A network connection is defined as a sequence of TCP packets from start to finish at a specific time under a predefined protocol, such as TCP/UDP, from source IP address to destination IP address. Each network connection is labeled as normal or abnormal (attack). In the KDD-CUP'99 dataset, each connection contains 41 feature attributes. In our experiments, we will use the smaller version of the KDD-CUP99 that contains 10% of the full dataset.

4.1.1. Data Pre-processing

We implemented simple pre-processing [18] steps before using the data for training and testing out models. First, we label the normal network connection as 1 and attack as 0, and the label will be used only in the testing stage. Then we convert each symbolic feature (e.g., connection type - TCP/HTTP, etc.) to numeric value using pandas dummies encoding [19]. By this, the number of features increased to 121. And in the final step, scale the dataset values to the range of 0-1 with a min-max scaler [20].

4.1.2. Train and Test Split

The KDD-CUP99 10% dataset contains much more attack data than normal network data because we train a semi-supervised model to use the most common data in training. Therefore the attack data will be used for training, and normal network connection will consider as anomaly data. We take 80% of the attack data (317394 samples). For validation and testing, we take the same amount of normal network connection and attack connection and split the data such that the validation set consists of 80% of this data.



a

b

Fig 2 – (a) Variational Autoencoder trained on dog imag
reconstruction error while a cat image leads to high reco

4.2. Evaluation Criteria

The ordinary criterion of evaluating the anomaly detection algorithm is an Area Under the ROC Curve (AUC) and Accuracy score. ROC is a curve drawn with a true positive rate (TPR) as the longitudinal coordinate and a false positive rate
(FPR) as the horizontal coordinate. The ROC curve of the perfect model is closer to the axis on the top-left side, and the AUC value is closer to 1. The result of random classification is 0.5. Accuracy score is defined as the proportion of correct prediction among the total number of cases examined.

4.3. Anomaly classification rule

In our experiment, we used the simplest way to decide if a data point is an anomaly (normal connection in our case) or not. We believed that anomaly data that was not present in the training data or not in the same domain as the training data will lead to large reconstruction error. Using the criteria from section 4.2 we can find the optimal reconstruction error threshold value for determinant if a data point is anomaly data or not.

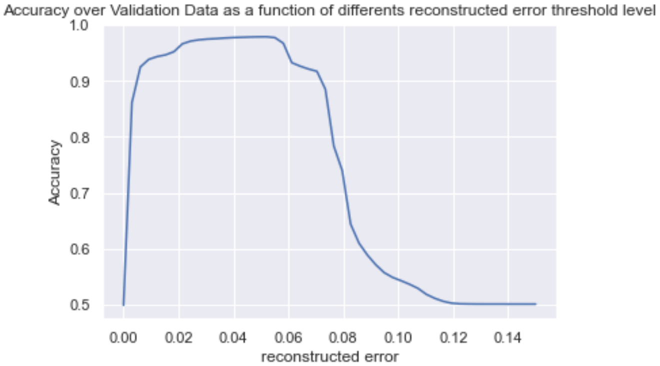


Fig 3: The accuracy as a function of reconstruction error of α-VAE with two encoder and decoder layers and 4D latent space over the validation data. Best accuracy achieved in reconstruction error value of 0.052

4.4. Experiments

On each one of the VAEs introduced in section 2, we perform the following experiments:

Experiment I **-** Change the number of encoder and decoder layers and see if anomaly detection accuracy changes [18].

Experiment II – Change the latent space dimension and see if anomaly detection accuracy changes [18].

In each of those experiments, we train the model using ten epochs with a learning rate of 0.0001 and batch size of 64 [18]. After each training, we find the best reconstruction error threshold value that leads to the highest Accuracy and AUC values on the validation set and measure the model performance with the threshold on the testing set.

5. Results

In each VAE architecture represented in section 2, we change the number of encoder and decoder layers to the following values - 2,3 and 4. And we also change the latent space dimension to the following values – 2, 5, and 10. When training an β-VAE, we set the β value to 5, and when training an α-VAE model, we set the α value to 100 [18]. Overall we end up with nine different models for each architecture.

In fig 4a, we can see the validation set's different VAEs latent space representation. As can be observed, VAE and β-VAE do not clearly distinguish between normal and abnormal data, but both give a good representation of the distribution of z, while α-VAE offers a less accurate representation of the distribution of z, but has good separation between the two classes.

In fig 4b we can see the different VAEs reconstruction errors of the validation set. As can be observed, β-VAE has the poor performance distinguish between normal and abnormal data, while α-VAE offers the best separation between the 2 classes in the value of reconstruction error

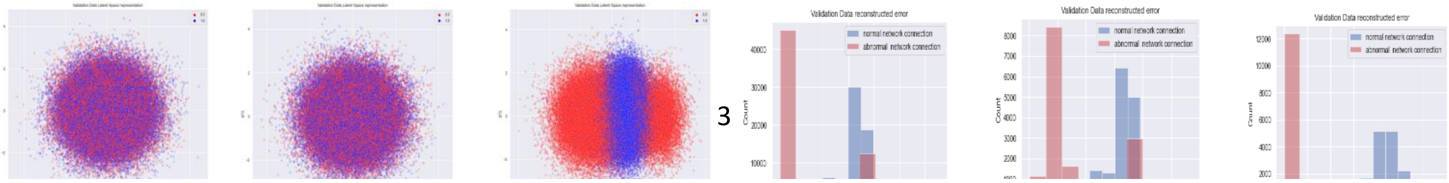


Fig 4 – (a) Latent space representation of Validation data in different VAE architecture (from left to right: VAE, β-VAE, and α-VAE).
(b) reconstruction error of Validation data in different VAE architecture (from left to right: VAE, β-VAE, and α-VAE). In all graphs, red color represents an abnormal network connection, blue represents a normal network connection, each VAE has 2 layers in the encoder and the decoder and 2D latent space

		Accuracy score		
Latent space dim	encoder and decoder layers	VAE	β-VAE	α-VAE
2	**2**	0.854	0.854	0.974
	3	0.854	0.854	0.973
	4	0.854	0.854	0.97
3	**2**	0.853	0.853	0.953
	3	0.854	0.854	0.977
	4	0.854	0.854	0.971
4	**2**	0.854	0.854	**0.98**
	3	0.854	0.854	0.977
	4	0.854	0.854	0.971

Table 1 – Accuracy score of each one of the Variational Auto-Encoders trained in our experiments

As can be observed in table 1, the number of layers in the encoder, the decoder, and the latent space dimension has not had a strong effect on the result

5.1. Comparison and best model performance

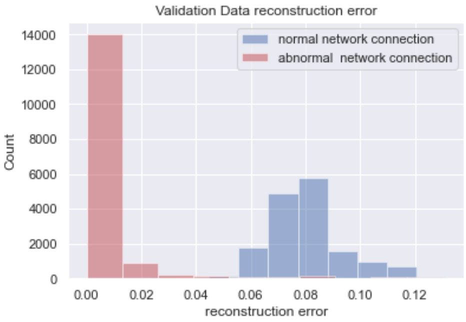


Fig 5: The reconstruction error of the best model (α-VAE with two encoder and decoder layers and 4D latent space) over the validation data

	Accuracy	AUC	Number of encoder and decoder layers	Latent space dimension
VAE	0.854	0.854	2	5
β-VAE	0.854	0.854	2	5
α-VAE	**0.98**	**0.98**	**4**	**2**

Table 1– Accuracy score and AUC score of the best model in each VAE architecture

As shown in Fig 5, The best model managed to achieve a good separation between the normal network connection and abnormal network connection**.** In table 2, we can observe the α-VAE architecture achieves the best performance in the anomaly detection task. The α-VAE has 10% higher accuracy than both VAE and β-VAE.

6. Conclusions

We present a simple implementation of using VAE for semi-supervised anomaly detection. Using only the reconstruction error as a classification rule, we achieve 85-98% accuracy. The proposed α -VAE achieves the best performance in our experiments. The α -VAE achieves accuracy, and AUC scores close to those reported in perverse studies. α -VAE is more suitable for our anomaly detection method as it gives much more influence on the reconstruction loss in the training process, leading to a much more efficient separation between normal and anomaly data. In our opinion, if we also use the latent space representation in our classification method, the β-VAE will achieve the best performance. Unfortunately, we can't observe any effect of the number of layers in the encoder and decoder and the latent space dimension on the accuracy and AUC as we expected. We assume that those parameters will have much more influence on the result in other types of data such as images and much more high dimension data.

References

1. Chandola, V., Banerjee, A. and Kumar, V., 2009. Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), pp.1-58.
2. Görnitz, N., Kloft, M., Rieck, K. and Brefeld, U., 2013. Toward supervised anomaly detection. Journal of Artificial Intelligence Research, 46, pp.235-262.
3. Akcay, S., Atapour-Abarghouei, A. and Breckon, T.P., 2018, December. Ganomaly: Semi-supervised anomaly detection via adversarial training. In Asian conference on computer vision (pp. 622-637). Springer, Cham.
4. Eskin, E., Arnold, A., Prerau, M., Portnoy, L. and Stolfo, S., 2002. A geometric framework for unsupervised anomaly detection. In Applications of data mining in computer security (pp. 77-101). Springer, Boston, MA.
5. Buttenfield, B.P., 1993. Representing data quality. Cartographica: The International Journal for Geographic Information and Geovisualization, 30(2-3), pp.1-7.
6. Murtaza, S.S., Khreich, W., Hamou-Lhadj, A. and Couture, M., 2013, November. A host-based anomaly detection approach by representing system calls as states of kernel modules. In 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE) (pp. 431-440). IEEE.
7. Zenati, H., Foo, C.S., Lecouat, B., Manek, G. and Chandrasekhar, V.R., 2018. Efficient gan-based anomaly detection. arXiv preprint arXiv:1802.06222.
8. Bank, D., Koenigstein, N. and Giryes, R., 2020. Autoencoders. arXiv preprint arXiv:2003.05991.
9. Doersch, C., 2016. Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.
10. Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. Journal of the American statistical Association, 112(518), 859-877

11. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., ... & Lerchner, A. (2016). beta-vae: Learning basic visual concepts with a constrained variational framework.
12. He, J., Spokoyny, D., Neubig, G., & Berg-Kirkpatrick, T. (2019). Lagging inference networks and posterior collapse in variational autoencoders. arXiv preprint arXiv:1901.05534.
13. Yan, C., Wang, S., Yang, J., Xu, T., & Huang, J. (2020, September). Re-balancing variational autoencoder loss for molecule sequence generation. In Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (pp. 1-7).
14. Sun, J., Wang, X., Xiong, N. and Shao, J., 2018. Learning sparse representation with variational auto-encoder for anomaly detection. IEEE Access, 6, pp.33353-33361.
15. Nguyen, Q.P., Lim, K.W., DIVAKARAN, D., LOW, K. and CHAN, M., 2019. A gradient-based explainable variational autoencoder for network anomaly detection. In 2019 IEEE Conference on Communications and Network Security (CNS) (pp. 91-99).
16. Gonzalez, D., Patricio, M.A., Berlanga, A. and Molina, J.M., 2021. Variational autoencoders for anomaly detection in the behaviour of the elderly using electricity consumption data. Expert Systems, p.e12744.
17. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
18. https://github.com/maorb91/DeepLearning
19. https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html
20. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html