AnoGen: Deep Anomaly Generator

Nikolay Laptev
Facebook
Menlo Park, CA
nlaptev@fb.com

ABSTRACT

Validating and testing a machine learning model is a critical stage in model development. For time-series anomaly detection, validation and testing is challenging because of the lack of labeled data and the difficulty of generating a realistic time-series with anomalies. Motivated by the continued success of Variational Auto-Encoders (VAE) and Generative Adversarial Networks (GANs) to produce realistic-looking data we provide a platform to generate a realistic time-series with anomalies called AnoGen. Our contribution includes a sampling technique that allows us to sample from the latent z space of a trained variational auto-encoder to deterministically generate a realistic time-series with anomalies.

1. INTRODUCTION

While rapid advances in computing hardware and software have led to powerful applications, still hundreds of software bugs and hardware failures continue to happen in a large cluster compromising system reliability. Non-stop systems have a strict uptime requirement and continuous monitoring of these systems is critical. From the data analysis point of view, this means non-stop monitoring of large volume of time-series data in order to detect potential faults or anomalies. Due to the large scale of the problem, human monitoring of this data is practically infeasible which leads us to anomaly detection using Machine Learning and Data Mining techniques. An anomaly, or an outlier, is a data point which is significantly different from the rest of the data. Generally, the data in most applications is created by one or more generating processes that reflect the functionality of a system.

When the underlying generating process behaves in an unusual way, it creates outliers. Fast and efficient identification of these outliers is useful for many applications including: intrusion detection, credit card fraud, sensor events, medical diagnoses, law enforcement and others [1].

Current anomaly detection systems suffer from the lack of training data and consequently from a large number of false positives which prohibit the usefulness of these systems in practice. Use-case, or category specific, anomaly detection models [2] may enjoy a low false positive rate for a specific application, but when the characteristics of the time-series change, these techniques perform poorly without proper retraining.

One way to make the iteration of anomaly model development quicker is to create a synthetic anomaly generator that represents the distribution and the types of time-series and anomalies one would see in the production environment.

Thus, at Facebook we created a system called AnoGen which enables us to generate synthetic time-series with anomalies at scale. AnoGen leverages recent work on Variational Autoencoders (VAE) to learn the distribution of the time-series and then to sample from that distribution in such a way so as to generate realistic time-series with anomalies at predictable points.

DOI: 10.475/123_4

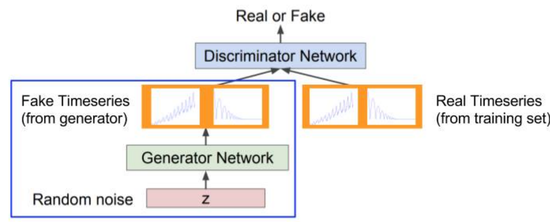

Figure 1: GAN Training

2. RELATED WORK

A generative adversarial network (GAN) [4] consists of a discriminator and a generator playing a two-player minimax game, wherein the generator aims to generate samples that resemble those in the training data whereas the discriminator tries to distinguish between the two as narrated in [3]. Training GANs, however, is challenging as it can be easily trapped into the mode collapsing problem where the generator only concentrates on producing samples lying on a few modes instead of the whole data space. Furthermore, GANs do not allow a simple way to sample from the resulting distribution in order to generate synthetic time-series. Figure 1 depicts this interaction where the discriminator is trained by getting input from a real training set (label = real) and from the generator (label = fake). The generator creates the time-series by performing a nonlinear transform from z. After training, the generator can create new time-series by sampling from z [4].

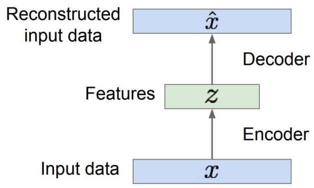

Figure 2: Autoencoder Training

A simpler model that also generates high quality results is called a variational auto-encoder and is based on the auto-encoder architecture (see Figure 2). Variational Auto Encoders (VAEs) were introduced in [5] as generative analogues to the standard deterministic auto encoder. As with deterministic auto encoders, VAEs pair a bottom-up inference network called an encoder with a top-down generative network called a decoder.

VAEs employ a probabilistic interpretation of these encoder and decoder networks. VAEs assume that the data set $\{x^{(i)}\}_{i=1}^{N}$ is composed of N i.i.d. samples of some variable x. Further, VAEs assume that the data were generated by a random process with continuous latent variable z and x was generated by some conditional distribution $p_\theta(x|z)$, where p_θ is a probability distribution with parameters θ. This provides a probabilistic interpretation of the decoder network, where given a latent variable or 'code' z we generate a sample x in the data space. Similarly, the role of the encoder would be to take a sample x from data space and give us a latent z sampled from the posterior density distribution $p_\theta(z|x)$.

In order to learn an encoder-decoder network pair, VAEs learn $q_\phi(z|x)$ which approximates the true, intractable posterior distribution. Training a VAE will amount to jointly learning these these parameters.

The training objective of VAEs is a tractable lower bound to the log-likelihood:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\right] = -(x) \quad (1)$$

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

Where D_{KL} is the Kullback-Leibler divergence. The reconstruction error term $\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right]$ is present in deterministic auto encoders (see Figure 2), and represents the likelihood that the input data would be reconstructed by the model. The variational regularization term $D_{KL}\left(q_\phi(z|x)||p_\theta(z)\right)$ represents the KL-divergence between the encoder-induced latent distribution and the true prior on the latent distribution. This term encourages the approximate posterior $q_\phi(z|x)$ to be close to $p_\theta(z)$.

In Variational auto-encoder, we sample z from a normal distribution parametrized by the mean and the variance. After training the model, we could generate a new time-series by sampling from latent space z. The diagram of the training process is shown in Figure 3.

3. DEEP TIME-SERIES ANOMALY

The naive way of generating an anomaly is by picking from some distribution D a set of anomaly features F which could be the magnitude m and duration d of the anomaly.

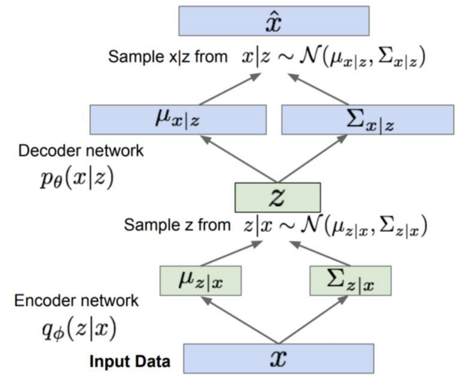

Figure 3: VAE Training

While generating anomaly by defining the explicit features is straight forward, it restricts the types of anomalies we can create and often leads to the problem of over-fitting our models to the synthetic training set, which usually performs poorly on the test set because the synthetic and test sets are rarely similar.

AnoGen learns the time-series normal and abnormal distributions using the variational autoencoder. To generate anomalies using the variational auto-encoder we sample from the outlier region of latent variable z. Figure 4 shows a sample z space reduced to 2D (the latent space size we used during training was 16).

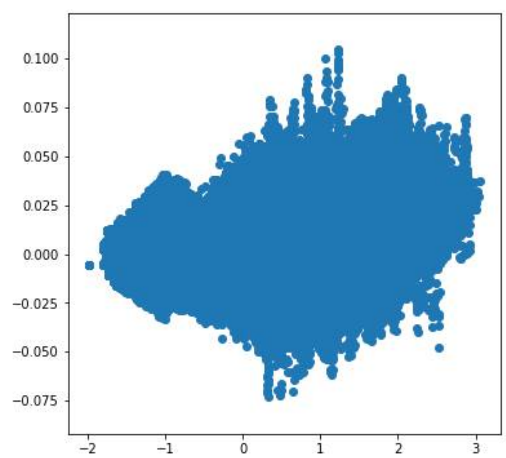

Figure 4: Z Space from which we sample

By sampling from the outlier region at predefined intervals, we are able to generate anomalies at deterministic locations. The sample algorithm is summarized in Algorithm 1.

4. EXPERIMENTS

In practice, for time-series data, the results produced by the generative adversarial model and by the variational auto-encoder are similar with the variational auto-encoder being significantly faster and easier to train. The training was done on about $100K$ series, producing the sample time-series shown in Figure 5.

Data: Init
Result: Time-series with anomalies
while *Length of result time-series not met* **do**
  read current idx; **if** *idx is not anomaly* **then**
    sample from normal space and add point to result;
  else
    sample from anomalous z region and add to result;
  end
end

Algorithm 1: Anomalous time-series generation with VAE.

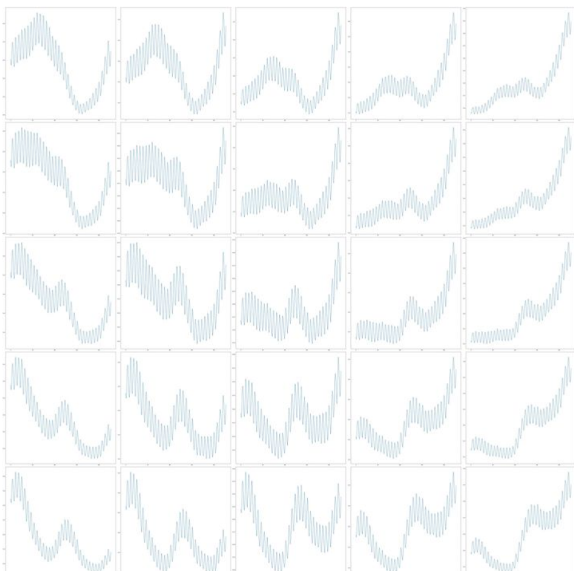

Figure 5: Sample generated time-series

By sampling from the outlier region of z one can generate the types of time-series behavior that is rare according to the model. An example of the resulting time-series with an anomaly in the middle of the time-series is shown in Figure 6.

For our experiments, we use AnoGen to generate training data for an Anomaly Detection model. For simplicity, we abstract the machine learning model used for Anomaly Detection as a simple binary classifier that for every time-step t outputs if a given point is $1, 0$ indicating anomaly or benign point respectively. More detailed experiments of different types of models is part of the longer version of the paper. The baseline method for generating training data is a naive approach where an anomaly is introduced in a time-series as spikes of certain magnitude m, variance v and duration d. Note that parameters m, v, d are chosen from a predefined distribution D. The overall results are presented in Table 1. The metrics *precision* and *recall* indicate the performance of the Anomaly Detection model trained using the baseline and using the AnoGen methods. By leveraging the true time-series, AnoGen was able to capture representative series behavior which was used to train the model to significantly outperform the same Anomaly Detection model trained using the baseline method on the manually labeled

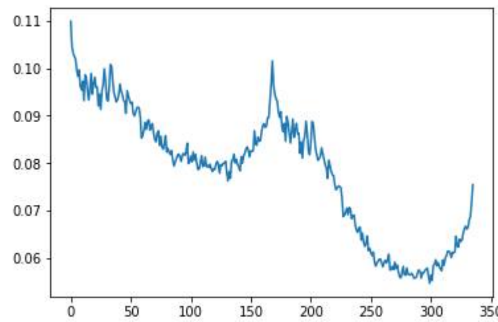

Figure 6: Sample anomaly

Synthetic Data Source	Precision	Recall
AnoGen Training Data	0.75	0.86
Baseline Training Data	0.37	0.78

Table 1: Resulting Anomaly ML Model performance when using baseline synthetic vs. AnoGen training data.

test set. This is the first step towards alleviating the reliance on manually curated data for anomaly detection.

5. CONCLUSION

Iteratively improving an anomaly detection model is difficult due to the lack of labeled data. Using pure synthetic time-series and anomaly data for training a machine learning model may provide suboptimal results for anomaly detection. In this paper, we introduced AnoGen, a system that uses a Variational Autoencoder to learn the latent space representation of real timeseries to generate a representative time-series with anomalies by sampling from the learned latent space. Our results indicate superior performance for training an Anomaly Detection machine learning model. This is an important first step towards reducing our reliance on manually curated time-series data for anomaly detection model training.

6. REFERENCES

[1] C. C. Aggarwal. *Outlier Analysis*. Springer Publishing Company, Incorporated, 2013.

[2] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[4] Q. Hoang, T. D. Nguyen, T. Le, and D. Q. Phung. Multi-generator generative adversarial nets. *CoRR*, abs/1708.02556, 2017.

[5] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.