Trigger Rate Anomaly Detection with Conditional Variational Autoencoders at the CMS Experiment

Adrian Alan Pol
CERN, Université Paris-Saclay

Victor Berger
Université Paris-Saclay

Gianluca Cerminara
CERN

Cecile Germain
Université Paris-Saclay

Maurizio Pierini
CERN

Abstract

Exploiting the rapid advances in probabilistic inference, in particular variational autoencoders (VAEs) for machine learning (ML) anomaly detection (AD) tasks, remains an open research question. In this work, we use the deep conditional variational autoencoders (CVAE), and we define an original loss function together with a metric that targets AD for hierarchically structured data. Our target application is a real world problem: monitoring the trigger system which is a component of many particle physics experiments at the CERN Large Hadron Collider (LHC). Experiments show the superior performance of this method over vanilla VAEs.

1 Introduction

AD is called to evolve significantly due to two factors: the explosion of interest in representation learning and the rapid advances in inference and learning algorithms for deep generative models. Particularly relevant is the variational learning framework of deep directed graphical model with Gaussian latent variables i.e. VAE, [1, 2].

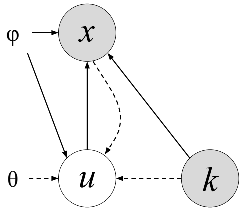

Figure 1: CVAE as a directed graph. Solid lines denote the generative model $p_\theta(x|u,k)p_\theta(u)$. Dashed lines denote variational approximation $q_\phi(u|x,k)$. Both variational parameters θ and generative parameters ϕ are learned jointly.

This work is originally motivated by a real world problem: improving AD for the trigger system, this is the first stage of event selection in many experiments at the CERN LHC. To be acceptable in this high-end production context, any method must abide to stringent constraints: performance, simplicity and robustness for long-term maintainability. Because of the nature of our target application, the algorithm has to be conditional. In layman terms, some of the structure of the model is known and associated observables are available. This setup points towards CVAE architectures [3]. CVAE (see Figure 1) is a conditional directed graphical model where input observations modulate the prior on latent variables in order to model the distribution of high-dimensional output space as a generative model conditioned on the input observation. Our overall contribution is to show that regular CVAE architectures can be exploited for general ML and AD tasks. We overcome the fundamental and technical obstacles to this goal by designing a new loss function targeting reconstruction resolution, and a new anomaly metric.

Problem Statement We are operating in a semi-supervised setup, where the examples of anomalous instances are not available. However, we know the design of the system and we directly observe some factors of variation in data. The observable x is a function of k (*known*) and u (*unknown*)

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

latent vectors: $x = f(k, u)$. For a collection of samples $x = [x_1, x_2, ..., x_n]$ we are interested in highlighting instances where we observe big change on a single feature, we later call **Type A** anomaly, and small but systematic changes on a group of features in the same configuration group (generated using the same k, as we further explain in Section 3), called **Type B** anomaly. Samples with a small severity of a problem on a group of uncorrelated features should be considered as an inlier, caused by expected statistical fluctuations. In summary, an algorithm needs to exploit a known causal structure in data, spot both types of problems, generalize to unseen cases and use data instead of relying on feature engineering. Inference time is negligible in the context of the target application.

Motivation This work emerges directly from the explicit urgency of extending monitoring of the CMS [4] experiment. The CMS experiment at CERN LHC [5] operates at the remarkable rate of 40 million particle collisions (*events*) per second. Each event corresponds to around 1 MB of data in unprocessed form. Due to understandable storage constrains and technological limitations (e.g. fast enough read-out electronics), the experiment is required to reduce the number of recorded data to 1 kHz. To this purpose, a hierarchical set of algorithms collectively referred to as the *trigger system* is used to process and filter the incoming data stream which is the start of the physics event selection process. Trigger algorithms [6] are designed to reduce the event rate while preserving the physics reach of the experiment. The CMS trigger system is structured in two stages using increasingly complex information and more refined algorithms. The **Level 1** (L1) Trigger, implemented on custom electronics reduces the 40 MHz input to a 100 kHz rate. **High Level Trigger** (HLT), a collision reconstruction software running on a computer farm, which scales the 100 kHz rate output of L1 Trigger down to 1 kHz. Both the L1 and the HLT systems implement a set of rules to perform the selection (called *paths*). The HLT ones are seeded by the events selected by a set of L1 Trigger paths.

Under typical running conditions, the trigger system regulates the huge data deluge coming from the observed collisions. The quality of the recorded data is guaranteed by monitoring the trigger rates. The event acceptance rate is affected in presence of number of issues e.g. detector malfunctions. Depending on the nature of the problem, the rate associated to specific paths could change to unacceptable levels. Critical cases include dropping to zero or increasing to extreme values. In those cases, the system should alert the shift crew, calling for a problem diagnosis and intervention.

HLT paths are often very strongly correlated. This is due to the fact that groups of paths select similar physics objects (thus selecting the same event) and/or are seeded by the same set of L1 Trigger paths. While critical levels of rate deviations for singular paths should be treated as an anomaly, smaller ones, on a number of random trigger paths, are likely a result of statistical fluctuations. On the other hand, an observable coherent drift (even small) on a group of trigger paths related by similar physics or making use of the same hardware infrastructure, is an indication of a likely fault present in the trigger system or hardware components. We explore this hierarchical structure in our algorithm. Each HLT path has a direct link to a set of L1 Trigger paths through specified configuration as shown in Figure 2. Hence, the HLT system performance is directly linked with the status of L1 Trigger.

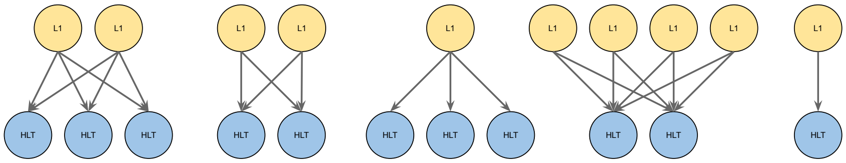

Figure 2: Schematic graph inspired by the trigger system configuration. Blue nodes represent HLT paths while yellow L1 Trigger paths. Each link is unidirectional starting from yellow nodes. The graph has a few hundred nodes spread approximately equally between HLT and L1 Triggers paths.

2 Background and Proposed Method

Variational Autoencoders. VAEs ([1, 2]) are a class of likelihood-based directed graphical generative models, maximizing the likelihood of the training data according to the generative model $p_\theta(x)$ augmented by the introduction of a latent variable z: $p_\theta(x) = \int p_\theta(x|z)p(z)dz$. The VAEs parameters are efficiently trained using an inference distribution $q_\phi(z|x)$ in a fashion very similar to autoencoders, using stochastic gradient variational Bayes framework. The recognition model $q_\phi(z|x)$

is included to approximate the true posterior $p_\theta(z|x)$). The training loss of the VAE is defined as:

$$\mathbb{E}_{z\sim q}[-\log p_\theta(x|z)] + \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x)||p(z)) \tag{1}$$

Optimal resolution. Typically VAEs model the reconstruction as a mean squared error (MSE) between the data x and the output of the decoder. However, this is equivalent to setting the observation model $p_\theta(x|z)$ as a normal distribution of fixed variance $\sigma = 1$. We argue that fixing the variance this way can be detrimental to learning as it puts a limit on the accessible resolution for the decoder. This defines the generative model as having a fixed noise of variance 1 on its output, making it impossible for it to accurately model patterns with a characteristic amplitude smaller than that. However, unless *a priori* knowledge suggests it, there is no guarantee that all patterns of interest would have such a large characteristic amplitude. Rather than adding a weighting term between the two parts of the loss like has often been done (e.g. [7]) we rather let the model learn the variance of the output of the decoder feature-wise (i running as the dimensionality of the data vectors x):

$$-\log p_\theta(x|z) = \sum_i \frac{(x_i-\mu_i)^2}{2\sigma_i^2} + \log\left(\sqrt{2\pi}\sigma_i\right) \tag{2}$$

Learning the variance of the MSE reconstruction allows the model to find the optimal resolution for the reconstruction of each feature of the data, separating the intrinsic noise from the correlations.

Setup Description In our setup we have three types of variables, see Figure 1: for random observable variables x, u (*unknown*, unobserved) and k (*known*, observed) are independent random latent variables. The conditional likelihood function $p_\theta(x|u,k)$ is formed by a non-linear transformation, with parameters θ. ϕ is another non-linear function that approximates inference posterior $q_\phi(u|k,x) = N(\mu, \sigma I)$. The latent variables u allow for modeling multiple modes in conditional distribution of x given k making the model sufficient for modeling one-to-many mapping:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|k,x)}[\log p_\theta(x|z,k)] - \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x,k)||p(z)) \tag{3}$$

where z intends to capture variable u. To approximate ϕ and θ we minimize the following loss:

$$\mathcal{L}_{\mathrm{CVAE}}(x,k,\theta,\phi) = \sum_i \frac{(x_i-\mu_i)^2}{2\sigma_i^2} + \log\left(\sqrt{2\pi}\sigma_i\right) + \mathbb{D}_{\mathrm{KL}}(q_\phi(z|x,k)||p(z)). \tag{4}$$

An AD metric with CVAE. For a given datapoint (x,k), the evaluation of the loss of the CVAE $\mathcal{L}(x,k)$ is an upper-bound approximation of $-\log p_\theta(x|k)$. The CVAE thus provides a model that naturally estimates how anomalous x is given k, rather than how anomalous the couple (x,k) is. This means that a rare value of k associated with a proper value of x will be treated as non-anomalous.

The equation 4 can be broken up to target two independent problems (Type A and B). AD for Type A uses an average infinity norm of the reconstruction loss $||\frac{1}{\sigma}(x-\hat{x})^2||_\infty$ ($\hat{x}$ as the reconstructed mean and σ as the reconstructed variance of decoder output), performing multiple sampling of z (we arbitrarily choose 30). Type B AD uses mean $\mathbb{D}_{\mathrm{KL}}$ of z. Because of two separate failure scenarios, we do not combine the metrics in one overall score but rather use logical `OR` to spot anomalies. In the first case we are interested in identifying an anomaly on a single feature. Typically used, MSE would likely be an incorrect choice when most of the features do not manifest abnormalities and lower the anomaly score. In the second case we expect μ_z to land on the tail of the distribution for anomalous cases. As argued in [8] the $\mathbb{D}_{\mathrm{KL}}$ measures the amount of additional information needed to represent the posterior distribution given the prior over the latent variable being used to explain the current observation. The lower the absolute value of $\mathbb{D}_{\mathrm{KL}}$ the more predictable state is observed. Finally, VAE framework guarantees that the method generalizes to unseen observations [9].

3 Experiments

Models are built upon CVAE focusing on distribution of output variables for AD tasks. We use Keras [10] with TensorFlow [11] backend and Adam [12] optimizer with early stopping [13] criterion.

Synthetic Problem. The synthetic dataset uses normally distributed ($\mu = 0$, $\sigma = 1$), continuous and independent latent variables u and k. Observable x is simply a product of u, k and additional

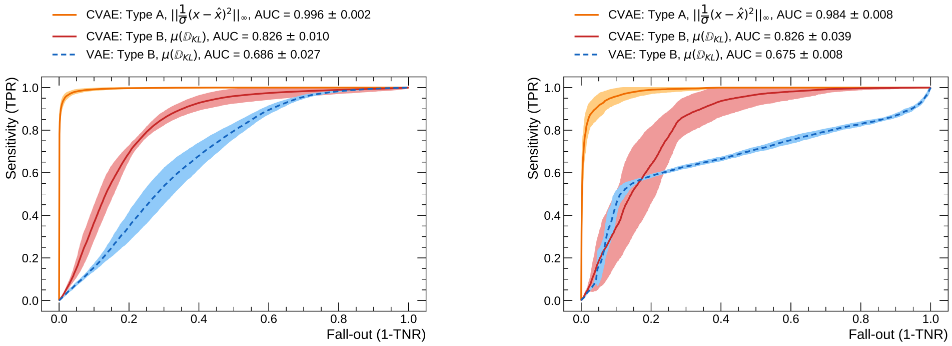

Figure 3: The ROC curves for two AD problems using synthetic (left) and CMS trigger rates test dataset (right). The bands correspond to σ computed after running the experiment 5 times.

noise ϵ given configuration constraints: $x_j = f_j(\vec{u}) \cdot \sum_{i=0}^{m} \mathbf{S}_{ji} k_i + \epsilon$, where j is a feature index for $\vec{x}$ in $\mathbb{R}^n$. A binary matrix $\mathbf{S}$ describes which k is used to compute feature j and function $f(\vec{u})$ describes which u enters the product that defines each feature j: $f_j(\vec{u}) = \prod_o u_o$. S and $f(\vec{u})$ stay unchanged across each sample in the dataset but the values of k and u do change. For simplicity, we ensure that each j depends only on one k and the dependence is equally distributed. For instance, the first column x_0 can use k_0, u_1 and u_4: $x_0 = k_0 u_1 u_4$, $x_{99} = k_4 u_0$ etc. We generate samples with x being 100-dimensional ($n = 100$) and $m = o = 5$. For testing we generate samples according to the table:

Test set	**Description**
Type A Inlier	Generated in the same process as training data
Type A Anomaly	5σ change on ϵ for a random feature
Type B Inlier	3σ change on ϵ for a random set of $\frac{n}{m}$ features
Type B Anomaly	3σ change on ϵ for a random feature cluster

The choice of 5σ and 3σ comes from legacy requirements of our target application. The AD is performed by comparing output of the decoder with the input for problems observed only on one of the features (Type A), or comparing $\mathbb{D}_{\mathrm{KL}}$ yield for samples with problems present on all features belonging to the same causal group (using the same k column as input) i.e. Type B. The ROC curves for both of the problems are shown in Figure 3. Given the high order of the deviation on Type A anomalies, the model easily spots them. In context of hierarchical structures, an algorithm needs to model a mapping from single input to multiple outputs. Type B detection results show that CVAE is outperforming VAE baseline and confirming it is suitable for such task.

CMS Trigger Rate Monitoring. We treat HLT rates as x and L1 Trigger rates as k. Our prototype uses 4 L1 Trigger paths that seed 6 unique HLT paths each. We extract rates only from samples where all chosen paths are present in the configuration. We end up with 102895 samples which are then split into training, validation, and test sets. Our test set has 2800 samples. We consider hypothetical situations that are likely to happen in the production environment. We generate four synthetic test datasets manipulating our test set in similar manner to the synthetic dataset. We detect isolated problems on one of the HLT paths (Type A) and problems present across HLT paths seeding the same L1 Trigger path (Type B). We report the results in Figure 3. The performance of the algorithm on CMS dataset is matching the performance we reported for the synthetic one. The CMS experiment currently does not provide any tools to track problems falling into Type B category. Given a good performance of the proposed method, we believe that the solution could be considered for deployment, provided further tests and refinements in the production environment.

4 Conclusions and Future Work

This paper shows how anomalies can be identified using CVAE. We have considered the specific case of CMS trigger rate monitoring to extend current functionality and showed good detection performance. We did not perform a hyper-parameter scan, thus we expect the results to improve if further optimized. Subsequent studies foresee using a full configuration of the CMS trigger system.

Acknowledgments

We thank the CMS collaboration for providing the dataset used in this study. We acknowledge the support of the CMS CERN group for providing the computing resources to train our models. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement n^o 772369).

References

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

[2] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, 2014.

[3] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[4] Serguei Chatrchyan et al. The CMS experiment at the CERN LHC. *JINST*, 3:S08004, 2008.

[5] The LHC Study Group. The Large Hadron Collider, conceptual design. Technical report, CERN/AC/95-05 (LHC) Geneva, 1995.

[6] Vardan Khachatryan et al. The CMS trigger system. *JINST*, 12(01):P01020, 2017.

[7] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

[8] Mevlana Gemici, Chia-Chun Hung, Adam Santoro, Greg Wayne, Shakir Mohamed, Danilo J Rezende, David Amos, and Timothy Lillicrap. Generative temporal models with memory. *arXiv preprint arXiv:1702.04649*, 2017.

[9] Durk 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.

[10] François Chollet et al. Keras, 2015.

[11] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] 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. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.