Unsupervised Anomaly Detection in Multivariate Time Series across Heterogeneous Domains

Vincent Jacob
Ecole Polytechnique
Palaiseau, France
vincent.jacob@polytechnique.edu

Yanlei Diao
Ecole Polytechnique
Palaiseau, France
yanlei.diao@polytechnique.edu

ABSTRACT

The widespread adoption of digital services, along with the scale and complexity at which they operate, has made incidents in IT operations increasingly more likely, diverse, and impactful. This has led to the rapid development of a central aspect of "Artificial Intelligence for IT Operations" (AIOps), focusing on detecting anomalies in vast amounts of multivariate time series data generated by service entities. In this paper, we begin by introducing a unifying framework for benchmarking unsupervised anomaly detection (AD) methods, and highlight the problem of shifts in normal behaviors that can occur in practical AIOps scenarios. To tackle anomaly detection under domain shift, we then cast the problem in the framework of domain generalization and propose a novel approach, Domain-Invariant VAE for Anomaly Detection (DIVAD), to learn domain-invariant representations for unsupervised anomaly detection. Our evaluation results using the Exathlon benchmark show that the two main DIVAD variants significantly outperform the best unsupervised AD method in maximum performance, with 20% and 15% improvements in maximum peak F1-scores, respectively. Evaluation using the Application Server Dataset further demonstrates the broader applicability of our domain generalization methods.

PVLDB Reference Format:
Vincent Jacob and Yanlei Diao. Unsupervised Anomaly Detection in Multivariate Time Series across Heterogeneous Domains. PVLDB, 18(6): 1691 - 1704, 2025.
doi:10.14778/3725688.3725699

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/exathlonbenchmark/divad.

1 INTRODUCTION

Time series anomaly detection has been studied intensively due to its broad application to domains such as financial market analysis, system diagnosis, and mechanical systems [10, 16]. Recently, it has been increasingly adopted in an emerging domain known as "Artificial Intelligence for IT operations" (AIOps) [15], which proposes to use AI to automate and optimize large-scale IT operations [51]. Not long ago, the role of IT was to support the business. Today as digital services and applications become the primary way that enterprises serve and interact with customers, IT *is* the business – almost every business depends on the continuous performance and innovation of its digital services.

With this paradigm shift, incidents in IT operations have become more impactful, inducing ever-increasing financial costs, both directly through service-level agreements made with customers and indirectly through brand image deterioration. Concurrently, the popularity of such services, along with their widespread migration to the cloud, has greatly increased the scale and complexity at which they operate, relying on more resources to process larger volumes of data at high speed. This evolution has made incidents more frequent, costly, diverse, and difficult for engineers to manually anticipate and diagnose, thus calling for more automated solutions.

To respond to such needs, this paper focuses on *anomaly detection in multivariate time series* that suits the challenges in AIOps. More specifically, a large set of multivariate time series are generated from the periodic monitoring of service entities, and "anomalies" are reported as patterns in data that deviate from a given notion of *normal behavior* [9].

Challenges. Detecting anomalies in AIOps presents a set of technical challenges [51]. (CH1) The *scarcity of anomaly labels* is due to the lack of domain knowledge of IT operations to reliably label anomalies, and the labor-intensive process of examining large amounts of time series data. (CH2) The *high dimensionality of recorded time series*, both in terms of time and feature dimensions, is common in AIOps due to the collection of numerous metrics at high frequency across a large number of entities. (CH3) The *complexity and variety in normal behaviors* arise because multiple, complex entities are monitored at scale in different contexts. (CH4) The *shifts in normal behaviors* further arise due to potentially frequent changes in software/service components, hardware components, or operation contexts of the monitored entities. The recent Exathlon benchmark [22] exhibits significant shifts in normal behaviors across traces collected from different runs of Spark streaming applications. Similarly, the Application Server Dataset (ASD) [27] exhibits shifts in behaviors of different servers. In these cases, the shifts in normal behaviors are so significant that they appear to be samples collected from different *domains* or *contexts*.

A large number of anomaly detection (AD) methods for multivariate time series have been developed, as categorized recently by Schmidl et al. [40]. CH1 has been typically addressed through the development of *unsupervised* AD methods, assuming no label information for training, and *semi-supervised* methods that assume (possibly noisy) labels for the normal class only [9]. In this paper, we jointly refer to them as "unsupervised" methods, trained on mostly-normal data and evaluated on a labeled test set. Concurrently, the advent of *deep learning* (DL) [25] has been instrumental in partly

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 18, No. 6 ISSN 2150-8097.
doi:10.14778/3725688.3725699

addressing CH2 and CH3, offering the ability to learn succinct yet effective representations of high-dimensional data while capturing both temporal (i.e., intra-feature) and spatial (i.e., inter-feature) dependencies in multivariate time series [27, 45, 50]. Despite covering a wide range of assumptions about both normal data and anomalies, all these methods are vulnerable to CH4, by assuming a similar distribution of training and test normal data, which makes them of limited use in the new AIOps scenario.

This paper tackles the last challenge (CH4), compounded by other challenges (CH1-Ch3), through the framework of **domain generalization** (DG). In this framework, data samples are collected from multiple, distinct *domains* (the normal contexts here), with certain characteristics of the observed data being determined by the domain, and others being independent from it. This amounts to associating shifts in normal behavior to the concept of *domain shift*, and aiming to build models from a set of training (or *source*) domains that can generalize to another set of test (or *target*) domains. Most of existing DG methods were proposed for image classification, categorized as based on *explicit feature alignment*, *domain-adversarial learning* or *feature disentanglement* [48, 53]. In practice, adversarial methods can suffer from instabilities that make them hard to reproduce [24, 37], and explicit feature alignment become very costly as the number of source domains increases, like in AIOps. For these reasons, this paper focuses on feature disentanglement [20, 33, 34], where methods seek to decompose the input data into *domain-shared* and *domain-invariant* features. The existing methods, designed for image classification, are not applicable to unsupervised time series anomaly detection (with no labels). Further, recent efforts on domain generalization for time series AD handle only univariate sound waves with various labeling assumptions [11], making them unsuitable for the AIOps setting.

Contributions. In this paper, we present the first *multivariate time series anomaly detection approach that generalizes across heterogeneous domains*. Given that this topic has been underaddressed in the anomaly detection literature, we conduct an in-depth study to characterize the problem of normal behavior shifts using the recent Exathlon benchmark [22] (which was motivated by AIOps use cases) and to highlight the performance issues of existing unsupervised anomaly detection methods under domain shift. We then address this challenge by proposing a novel approach based on domain generalization and feature disentanglement, custom-designed for unsupervised time series anomaly detection. More specifically, our paper makes the following contributions:

- We introduce a unifying framework for benchmarking unsupervised anomaly detection methods, and highlight the domain shift problem in AIOps scenarios (Section 3).
- To tackle the problem of domain shift, we develop a theoretical formulation of unsupervised anomaly detection in the framework of *domain generalization* (Section 4).
- In this proposed framework, we develop a novel approach, called Domain-Invariant VAE for Anomaly Detection (DIVAD), with a set of variants to learn domain-invariant representations, thereby enabling effective anomaly detection in unseen domains (Section 5).

Our evaluation using the Exathlon benchmark shows that our two main DIVAD variants can significantly outperform the best unsupervised AD method in maximum performance, with 20% and 15% improvements in maximum peak F1-scores (0.79 and 0.76 over 0.66), respectively. Our evaluation also applies DIVAD to the Application Server Dataset (ASD) [27], reflecting a similar use case, and shows that its explicit domain generalization can be more broadly applicable and useful in this second use case.

The code for our DIVAD method and experiments is available at https://github.com/exathlonbenchmark/divad.

2 RELATED WORK

Anomaly Detection in Multivariate Time Series. Numerous unsupervised anomaly detection methods in multivariate time series have been proposed over the years [9, 10, 16]. Schmidl et al. [40] recently introduced a taxonomy based on the way the methods derive their *anomaly scores* for data samples (the higher the score, the more deemed anomalous by the method). The only category we do not consider in this work is *distance methods*, which typically do not scale well with the large dimensionality of AIOps.

Forecasting methods define anomaly scores of data samples as forecasting errors, based on the distance between the forecast and actual value(s) of one or multiple data point(s) in a context window of length L. LSTM-AD [31] is the most popular forecasting method. It trains a stacked LSTM network to predict the next l data records from the first $L - l$ of a window. It then fits a multivariate Gaussian distribution to the error vectors it produced in a validation set, and defines the anomaly score of a record as the negative log-likelihood of its error with respect to this distribution.

Reconstruction methods score data samples based on their reconstruction errors from a transformed space. Principal Component Analysis (PCA) [2] and Autoencoder (AE) [18, 39] are representative shallow and deep reconstruction methods, respectively. PCA's transformation is a projection on the linear hyperplane formed by the principal components of the data, while AE's is a non-linear mapping to a latent encoding learned by a neural network that was trained to reconstruct data from it. More recently, Multi-Scale Convolutional Recurrent Encoder-Decoder (MSCRED) [50] turns a multivariate time series into multi-scale signature matrices characterizing system status at different time steps, and learns to reconstruct them using convolutional encoder-decoder and attention-based ConvLSTM networks. TranAD [45] relies on two transformer-based encoder-decoder networks, with the first encoder considering the current input window, and the second one considering a larger *context* of past data in the window's sequence. It defines the anomaly score of an input window as the average of its reconstruction errors coming from two decoders and inference phases, with the second phase using the reconstruction error from the first phase as a focus score to detect anomalies at a finer level.

Encoding methods score data samples based on their deviation within a transformed space. Deep SVDD [38] is the most popular recent encoding method, training a neural network to map the input data to a latent representation enclosed in a small hypersphere, and defining anomaly scores of test samples as their squared distance from this hypershere's centroid. More recently, DCDetector [49] uses a dual-view attention structure based on contrastive learning to derive representations where differences between normal points and anomalies are amplified, subdividing windows into adjacent "patches", with one view modeling relationships within patches

and the other across patches. To derive anomaly scores, it uses the insight that normal points tend to be similarly correlated for both views, while anomalies tend to be more correlated to their adjacent points than to the rest of the window.

Distribution methods define anomaly scores of data samples as their deviation from an estimated distribution of the data. The Mahalanobis method [2, 42] and Variational Autoencoder (VAE) [6] are representative shallow and deep distribution methods, respectively. The Mahalanobis method estimates the data distribution as a multivariate Gaussian, and defines the anomaly score of a test vector as its squared Mahalanobis distance from it. VAE estimates it using a variational autoencoder, with the anomaly score of a test point derived by drawing multiple samples from its probabilistic encoder, and averaging the negative log-likelihood of the reconstructions obtained from each of these samples. A more recent method is OmniAnomaly [43]. It estimates the distribution of multivariate windows with a stochastic recurrent neural network, explicitly modeling temporal dependencies among variables through a combination of GRU and VAE. It then defines a test window's anomaly score as the negative log-likelihood of its reconstruction.

Isolation tree methods score data samples based on their "isolation level" from the rest of the data. Isolation forest [28] is the most popular isolation tree method. It trains an ensemble of trees to isolate the samples in the training data, and defines the anomaly score of a test instance as inversely proportional to the average path length required to reach it using the trees.

Overall, these methods cover a wide range of assumptions about both normal data and anomalies. As we show in this paper, *by assuming a similar distribution of training and test normal data, all of them are vulnerable to shifts in normal behavior*, limiting their applicability in our AIOps scenario.

Domain Generalization. Domain generalization (DG) has been mainly studied in the context of *image classification*, with the domains usually corresponding to the way images are represented or drawn. DG methods can broadly be categorized as based on *explicit feature alignment*, *domain-adversarial learning* or *feature disentanglement* [48, 53]. Explicit feature alignment methods seek to learn data representations where feature distribution divergence is explicitly minimized across domains, with divergence metrics including the Wasserstein distance or Kullback-Leibler divergence [48, 53]. Rather than using such divergence metrics directly, domain-adversarial learning methods seek to minimize domain distribution discrepancy through a minimax two-player game, where the goal is to make the features confuse a domain discriminator [14], usually implemented as a domain classifier [3, 26, 32, 41]. Such adversarial methods can suffer from instabilities that make them hard to reproduce [24, 37], while explicit feature alignment can become very costly as the number of source domains increases, like in AIOps. For these reasons, this work considers domain generalization based on feature disentanglement [20, 33, 34], where methods seek to decompose the input data into *domain-specific* and *domain-invariant* features, and perform their tasks in domain-invariant space.

Our work is specifically related to Domain-Invariant Variational Autoencoders (DIVA) [20], designed for image classification. It uses variational autoencoders (VAE) to decompose input data into domain-specific, class-specific, and residual latent factors, conditioning the distributions of its domain-specific and class-specific factors on the training domain and class, respectively, and enforcing this conditioning by using classification heads to predict the domain and class from the corresponding embeddings. It then uses its class-related classifier to derive its predictions for the test images. Because of this class supervision, this method cannot be applied to our unsupervised AD setting.

Domain generalization for time series AD recently gained attention through anomalous sound detection and the DCASE2022 Challenge, where the task was to identify whether a machine was normal or anomalous using only normal sound data under domain-shifted conditions [11]. The methods proposed however modeled single-channel (univariate) sound waves, while also assuming labels such as the machine state, the type of machine, domain shift or noise considered to train domain-invariant or disentangled representations [47]. This univariate aspect, coupled with these simplifying assumptions, makes such methods unsuitable for our AIOps setting.

Data Drift Detection. Many techniques exist for data drift detection [13]. However, popular methods such as using the Kolmogorov-Smirnov distance [8, 12] require a significant amount of drifted data to detect a distribution change accurately. Anomaly detection under domain shift is essentially a different problem, where anomalies must be detected with low latency as they arise, although the normal behaviors in the current domain may appear to be drawn from a different context from those seen in training data.

3 GENERAL AD FRAMEWORK

In this section, we present the unsupervised anomaly detection (AD) problem, propose a unifying framework to encompass AD approaches in evaluation, and highlight the presence of domain shift in the current framework using the Exathlon [22] benchmark.

3.1 Unsupervised Anomaly Detection

We first introduce the notation of the paper and define the AD problem in the unsupervised setting. More specifically, we consider N_1 training sequences and N_2 test sequences:

$$\mathcal{S}_{\text{train}} = (\boldsymbol{S}^{(1)}, \ldots, \boldsymbol{S}^{(N_1)}) \ , \ \mathcal{S}_{\text{test}} = (\boldsymbol{S}^{(N_1+1)}, \ldots, \boldsymbol{S}^{(N_1+N_2)}),$$

where each $\boldsymbol{S}^{(i)}$ consists of T ordered data records of dimension M. To simplify the notation, our problem definition uses T to denote the (same) length of all sequences, while our techniques do not make this assumption and can handle variable-length sequences.

For each *test* sequence, we consider a sequence of *anomaly labels*:

$$\mathcal{Y}_{\text{test}} = \{\boldsymbol{y}^{(N_1+1)}, \ldots, \boldsymbol{y}^{(N_1+N_2)}\},$$

with $\boldsymbol{y}^{(i)} \in \{0, 1\}^T$, such that:

$$\begin{cases} y_t^{(i)} = 1 & \text{if the record at index } t \text{ in sequence } i \text{ is } \textit{anomalous}, \\ y_t^{(i)} = 0 & \text{otherwise (i.e., the record is } \textit{normal}). \end{cases}$$

Our goal is to build an anomaly detection model as follows.

Definition 1. *An anomaly detection model is a* record scoring function $g : \mathbb{R}^{T \times M} \rightarrow \mathbb{R}^T$*, mapping a sequence* $\boldsymbol{S}$ *to a sequence of real-valued record-wise anomaly scores* $g(\boldsymbol{S})$*, which assigns higher anomaly scores to anomalous records than to normal records in test sequences. That is,* $g(\boldsymbol{S}^{(i)})_{t_1} > g(\boldsymbol{S}^{(j)})_{t_2}$, $\forall i, j \in [N_1 \ldots N_1 + N_2]$, $t_1, t_2 \in [1 \ldots T]$ *s.t.* $y_{t_1}^{(i)} = 1 \wedge y_{t_2}^{(j)} = 0$.

This record scoring function should further be constructed in a setting of *offline training* and *online inference*. More precisely, it means that training has to be performed offline on $\mathcal{S}_{\text{train}}$, and inference must be performed online on $\mathcal{S}_{\text{test}}$ by considering only the data preceding a given record at time index t:

$$g(\boldsymbol{S}^{(i)})_t = g(\boldsymbol{S}^{(i)}_{1:t})_t \ , \ \forall i \in [N_1 \,..\, N_1 + N_2] \ , \ t \in [1 \,..\, T].$$

Due to this requirement, we refer to the anomaly detection methods based on g as *online scorers*.

3.2 Unifying Anomaly Detection Framework

We next propose a framework unifying AD approaches under a common evaluation structure. In this framework, each online scorer relies on a *windowing operator* W_L that extracts sliding windows, or *samples*, of length $L > 0$ from a given sequence i:

$$W_L(\boldsymbol{S}^{(i)}) = \{\boldsymbol{S}^{(i)}_{t-L+1:t}\}_{t=L}^{T} =: \{\boldsymbol{x}_t^{(i)}\}_{t=L}^{T},$$

with $\boldsymbol{x}_t^{(i)} \in \mathbb{R}^{L \times M}$. Then the training set is composed of the samples extracted from all the training sequences:

$$\mathcal{D}_{\text{train}} := \bigcup_{i \in [1..N_1]} \left\{ W_L(\boldsymbol{S}^{(i)}) \right\}.$$

Definition 2. *A* window scorer, *trained on D_{train}, is a* window scoring function *that assigns an anomaly score to a given window, proportional to its abnormality for the method, $g_W : \mathbb{R}^{L \times M} \to \mathbb{R}$, $\boldsymbol{x} \mapsto g_W(\boldsymbol{x})$.*

We encapsulate each individual AD method within a window scorer, and then propose a universal *online scorer* constructed from the window scorer. Given a test sequence $\boldsymbol{S}$ and a smoothing factor $\gamma \in [0, 1)$, the online scorer assigns anomaly scores as follows:

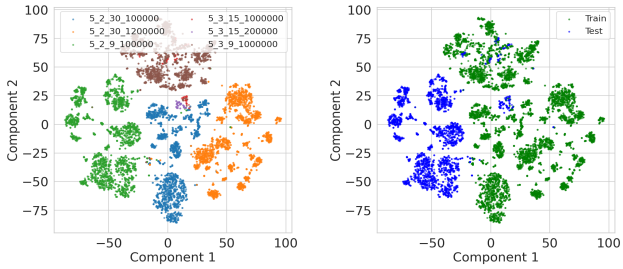
$$g(\boldsymbol{S}; L, \gamma)_t = \begin{cases} -\infty & \text{if } t < L, \\ (1-\gamma)\hat{y}_L =: m_L & \text{if } t = L, \\ \frac{\gamma m_{t-1} + (1-\gamma)\,\hat{y}_t}{(1-\gamma^{t+1})} =: m_t & \text{if } t > L. \end{cases}$$

$$\hat{y}_t = g_W(\boldsymbol{S}_{t-L+1:t}) \ , \ \forall t \in [L \,..\, T].$$

In other words, for a test sequence, we assign an anomaly score to the current sliding window of length L using g_W (which is fixed and trained offline). We then define this window score as the anomaly score of its last record (i.e., the one just received in an online setting). To allow additional control on the tradeoff between the "stability" and "reactivity" of the record scoring function, we further apply an *exponentially weighted moving average* with smoothing hyperparameter γ to the anomaly scores. This produces the final output of the record scoring function g for a test sequence, prepended with infinitely low anomaly scores for the timestamps before its first full window of length L. In this framework, both L and γ are hyperparameters to set for every anomaly detection method.

3.3 Domain Shift in Exathlon

Inspired by real-world AIOps use cases, the Exathlon benchmark [22] is one of the most challenging benchmarks for anomaly detection due to the high-dimensionality and complex, diverse behaviors in its dataset, as reported in a recent comprehensive experimental study of AD methods [40].



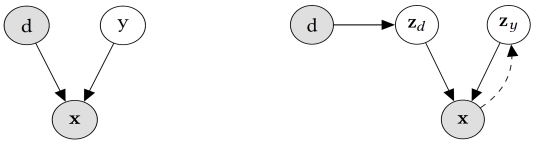
(a) Records colored by context. **(b) Records colored by dataset.**

Figure 1: t-SNE scatter plots of application 2's normal data, undersampled to 10,000 data records balanced by context.

3.3.1 Background on Exathlon. Exathlon [22] has been systematically constructed based on repeated executions of distributed Spark streaming *applications* in a cluster under different Spark *settings* and *input rates*. The dataset includes 93 repeated executions of 10 Spark applications, with one *trace* collected for each execution containing 2,283 raw features, resulting in a total size of 24.6GB. While 59 traces were collected in normal execution (*normal traces*), 34 other traces were disturbed manually by injecting anomalous events (*disturbed traces*). There are 6 different classes of anomalous events (e.g., misbehaving inputs, resource contention, process failures), with a total of 97 anomalous instances. For each of these anomalies, Exathlon provides the ground truth label for the interval spanning the root cause event and its lasting effect, enabling accurate evaluation of AD methods. In addition to anomalous instances, both the normal and disturbed traces contain enough variety (e.g., Spark's checkpointing activities) to capture diverse normal behaviors.

3.3.2 Analysis of Domain Shift. Given the limited number of Spark *applications* (or *entities* [52]) in the Exathlon dataset, our study focuses on the ability of an AD method to generalize, not to new applications, but rather to the new *contexts* of the Spark applications. For this reason, the **domain** of a trace is defined as its *context*, characterized by the following factors: (i) The Spark *settings* for each application run includes its processing period (i.e., batch interval or window slide), set to a specific value for the application, the number of active executors and "memory profile" (i.e., maximum memory set for the driver block manager, executors JVM, and garbage collection). The last two aspects had either a direct or indirect impact on a lot of features (e.g., executors memory usage). (ii) The *input rate* is the rate at which data records were sent to the application, which had a direct effect on many recorded features (e.g., last completed batch processing delay). The "normal behavior" in a trace is, therefore, mainly determined by its trace characteristics, defined as the combination of its *entity* and *domain/context*.

To illustrate the *diversity* and *shift* in domains/contexts, Figure 1 shows t-SNE scatter plots [46] of application 2's normal data, undersampled to 10,000 data records. The *diversity* is shown in Figure 1a, where data records are colored by context (where context labels include the processing period, number of Spark executors, maximum executors memory, and data input rate). We see that different contexts appear as distinct clusters, constituting a *multimodal* distribution for data records. Figure 1b illustrates the *shift* in context: the different contexts induce a distribution shift from the training to the



(a) Observed variable x depends on its domain d and latent (unobserved) class y.

(b) x is caused by independent domain-specific z_d and domain-independent z_y.

Figure 2: Generative models. For (b), constructing f_y amounts to *inferring* z_y from x (dashed arrow).

test data, even within normal records of the same application—we refer to this phenomenon as the *domain shift* problem.

4 PROBLEM OF DOMAIN SHIFT

In this section, we formally define the problem of anomaly detection under domain shift. We took inspiration from one of the first adopted definitions of an *anomaly* proposed by Douglas M. Hawkins in 1980, describing it as "an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism" [17].

This definition naturally suggests addressing our AD problem from a *generative* perspective, assuming data samples were generated from a distribution $p_{\text{data}}(\mathbf{x}, \text{y})$[1], with *normal* samples generated from $p_{\text{data}}(\mathbf{x}|\text{y}=0)$. Our general goal then translates to constructing a **model** $p_{\boldsymbol{\theta}}(\mathbf{x})$ of $p_{\text{data}}(\mathbf{x}|\text{y}=0)$, parameterized by $\boldsymbol{\theta} \in \Theta$. This yields a natural definition for the **anomaly score** of a test sample, as its negative log-likelihood with respect to this model:

$$g_W(\boldsymbol{x}; \boldsymbol{\theta}) := -\log p_{\boldsymbol{\theta}}(\mathbf{x} = \boldsymbol{x})$$

Since normal samples from the training and test sets are assumed generated from $p_{\text{data}}(\mathbf{x}|\text{y}=0) \approx p_{\boldsymbol{\theta}}(\mathbf{x})$, we would indeed expect them to have a higher likelihood under this model than anomalous samples, generated from $p_{\text{data}}(\mathbf{x}|\text{y}=1) \neq p_{\boldsymbol{\theta}}(\mathbf{x})$.

A unique aspect of AIOps scenarios, however, is that the distribution generating an observed sample can be conditioned not only on its class, but also on the specific *sequence* this sample was extracted from. In particular, each sequence corresponds to a **domain/context** that impacts the distribution of observed data, even for the same entity being recorded. These domains can be included in our generative model, by assuming that the selection of a sequence i corresponds to the realization d_i of a discrete random domain variable $\text{d} \sim p_{\text{data}}(\text{d})$ with infinite support[2]. In this setting, the samples of class $c \in \{0, 1\}$ from sequence i can be seen as independently drawn from a sequence-induced, or *domain* distribution:

$$p_i(\mathbf{x}|\text{y}=c) = p_{\text{data}}(\mathbf{x}|\text{y}=c, \text{d}=d_i).$$

This amounts to assuming the distribution of $\mathbf{x}$ is conditioned on the two independent variables d and y, the former determining the domain the sample originates from, and the latter determining whether the sample is normal or anomalous. We illustrate the corresponding generative model in Figure 2a. Under this model, the

[1]To simplify the notation, we use y to refer to the label of an input *sample* in the following: $y = 0$ if the sample is (fully) normal, $y = 1$ otherwise.

[2]We use d_i (as opposed to i) to reflect the fact that multiple sequences can correspond to the same context, and thus value d (i.e., we can have $d_i = d_j$ for $i \neq j$).

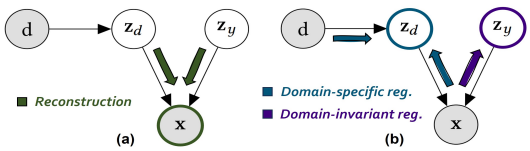


Figure 3: Illustration of the (a) reconstruction and (b) regularization terms in the ELBO objective (Eq. 1).

data-generating distribution of normal samples can be expressed as the countable mixture of all possible domain distributions:

$$p_{\text{data}}(\mathbf{x}|\text{y}=0) = \sum_{d=1}^{\infty} p_{\text{data}}(\mathbf{x}|\text{y}=0, \text{d}=d) p_{\text{data}}(\text{d}=d).$$

Definition 3 (Domain Shift Challenge). *Directly applying traditional generative methods in an unsupervised setting amounts to making $p_{\boldsymbol{\theta}}(\mathbf{x})$ estimate the data-generating distribution of the normal **training** samples (with d_i's fixed and all samples equally-likely to come from every sequence i):*

$$p_{\text{train}}(\mathbf{x}|\text{y}=0) = \frac{1}{N_1} \sum_{i=1}^{N_1} p_{\text{data}}(\mathbf{x}|\text{y}=0, \text{d}=d_i),$$

*which, given the infinitude of possible domains, is likely to differ from the data-generating distribution of the normal **test** samples:*

$$p_{\text{test}}(\mathbf{x}|\text{y}=0) = \frac{1}{N_2} \sum_{i=N_1+1}^{N_1+N_2} p_{\text{data}}(\mathbf{x}|\text{y}=0, \text{d}=d_i),$$

with $\{d_i\}_{i=1}^{N_1} \neq \{d_i\}_{i=N_1+1}^{N_1+N_2}$. This mismatch induces a domain shift challenge, *characterized by test normal samples $\boldsymbol{x}_0 \sim p_{\text{test}}(\mathbf{x}|\text{y}=0)$ and test anomalous samples $\boldsymbol{x}_1 \sim p_{\text{test}}(\mathbf{x}|\text{y}=1)$ being both unlikely in uncontrollable ways under $p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p_{\text{train}}(\mathbf{x}|\text{y}=0)$, which hinders anomaly detection performance.*

A suitable framework to address this domain shift challenge is *domain generalization* [48, 53]. In this framework, the domains sampled for training are referred to as *source domains*, while those sampled at test time are called *target domains*.

Definition 4 (Anomaly Detection with Domain Generalization). *Our problem can be framed as* building an AD model from the source domains that generalizes to the target domains. *We do so by assuming that the* observed *variable* $\mathbf{x}$ *can be mapped via f_y to a* latent *representation $\mathbf{z}_y$, whose distribution is discriminative with respect to the class* y *(i.e., normal vs. anomalous) and at the same time, independent from the domain* d. *Our goal can be formulated as: (1) Finding such a mapping $f_y(\mathbf{x}) = \mathbf{z}_y$; (2) Constructing $p_{\boldsymbol{\theta}}(\mathbf{x})$ to estimate $p_{\text{train}}(f_y(\mathbf{x})|\text{y}=0)$ instead of $p_{\text{train}}(\mathbf{x}|\text{y}=0)$.*

Since $f_y(\mathbf{x}) = \mathbf{z}_y$ is independent from d, we then have:

$$\begin{aligned} p_{\text{train}}(\mathbf{z}_y|\text{y}=0) &= \frac{1}{N_1} \sum_{i=1}^{N_1} p(\mathbf{z}_y|\text{y}=0, \text{d}=d_i) \\ &= p(\mathbf{z}_y|\text{y}=0) = p_{\text{test}}(\mathbf{z}_y|\text{y}=0), \end{aligned}$$

which means that, under $p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p_{\text{train}}(\mathbf{z}_y|\text{y}=0) = p_{\text{test}}(\mathbf{z}_y|\text{y}=0)$, the **normal test samples $\boldsymbol{x}_0$ should be more likely than the test anomalies $\boldsymbol{x}_1$**, hence addressing the domain shift challenge.

5 THE DIVAD METHOD

In this section, we introduce a new approach to anomaly detection under domain shift. At a high level, our central assumption is that *anomalies should have a sensible impact on the properties of the input samples that are invariant with respect to the domain.* In an AIOps scenario, this means that, although some aspects of a running process may vary from domain to domain (e.g., its memory used or processing delay), others typically remain constant and characterize its "normal" behavior (e.g., its scheduling delay, processing delay per input record, or any other Key Performance Indicator (KPI) [52] that behaves similarly across contexts). These domain-invariant, normal-specific characteristics tend to reflect whether the process *functions properly*, while domain-specific characteristics simply manifest *different modes of normal operation.*

To realize this intuition, we propose Domain-Invariant VAE for Anomaly Detection, or DIVAD. This method embodies (i) a new generative model with feature disentanglement to decompose the input data into domain-invariant and domain-specific factors, (ii) an effective training approach in the VAE framework, with a custom training objective for our unique AD model, and (iii) different alternatives for model inference, deriving anomaly scores based on the training distribution of domain-invariant factors only.

5.1 Modeling

Based on the notion of *feature disentanglement* [48, 53], our model assumes that the observed variable $\mathbf{x}$ is caused by two *independent* latent factors $\mathbf{z}_d$ and $\mathbf{z}_y$: (i) $\mathbf{z}_d$ is conditioned on the observed domain d, and (ii) $\mathbf{z}_y$ is assumed independent from it and can be used to detect anomalies in test samples. The corresponding generative model is shown in Figure 2b. Assuming that the model is parameterized by *model parameters* $\boldsymbol{\theta} \in \Theta$, the marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathrm{d})$ can be derived based on the structure of the generative model:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathrm{d}) &= \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_d, \mathbf{z}_y|\mathrm{d}) d\mathbf{z}_d d\mathbf{z}_y \\ &= \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_y, \mathbf{z}_d) p_{\boldsymbol{\theta}}(\mathbf{z}_d|\mathrm{d}) p(\mathbf{z}_y) d\mathbf{z}_d d\mathbf{z}_y. \end{aligned}$$

5.2 Model Training

We seek to learn the model parameters of $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathrm{d})$ through maximum likelihood estimation. Since computing $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathrm{d})$ directly is intractable, we leverage a variational autoencoder (**VAE**) **framework** [23, 35], considering *variational parameters* $\boldsymbol{\phi}_d, \boldsymbol{\phi}_y \in \Phi$, and optimizing the following evidence lower bound (**ELBO**) instead, known to be a lower bound on $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathrm{d})$ and lead to effective learning of its model parameters:

$$\begin{aligned} &\mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{x}, d; \boldsymbol{\theta}_{yd}, \boldsymbol{\theta}_d, \boldsymbol{\phi}_d, \boldsymbol{\phi}_y) = \\ &\mathbb{E}_{q_{\boldsymbol{\phi}_d}(\mathbf{z}_d|\boldsymbol{x}) q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}_{yd}}(\boldsymbol{x}|\mathbf{z}_d, \mathbf{z}_y)] \\ &- \beta D_{\mathrm{KL}}(q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\boldsymbol{x}) \| p(\mathbf{z}_y)) - \beta D_{\mathrm{KL}}(q_{\boldsymbol{\phi}_d}(\mathbf{z}_d|\boldsymbol{x}) \| p_{\boldsymbol{\theta}_d}(\mathbf{z}_d|d)). \end{aligned} \tag{1}$$

where the KL divergence terms are weighted by a factor β [19, 20].

Figure 3 illustrates the effects of the different terms in the above training objective. The first term of Eq. 1 involves the *likelihood* $p_{\boldsymbol{\theta}_{yd}}(\boldsymbol{x}|\mathbf{z}_d, \mathbf{z}_y)$. It measures DIVAD's ability to *reconstruct* an input from its latent factors, $\mathbf{z}_d$ and $\mathbf{z}_y$, as shown in Figure 3a. The second and third terms act as domain-invariant and domain-specific *regularizers*, pushing the *variational posteriors* $q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\boldsymbol{x})$ and $q_{\boldsymbol{\phi}_d}(\mathbf{z}_d|\boldsymbol{x})$ toward their *priors* $p(\mathbf{z}_y)$ and $p_{\boldsymbol{\theta}_d}(\mathbf{z}_d|d)$, respectively, as illustrated by Figure 3b. The prior $p(\mathbf{z}_y)$ will be used for anomaly scoring and further detailed in Section 5.3. The remaining distributions are learned using neural networks:

$$\begin{aligned} p_{\boldsymbol{\theta}_{yd}}(\mathbf{x}|\mathbf{z}_d, \mathbf{z}_y) &= \mathcal{N}(\mathrm{NN}_{\boldsymbol{\theta}_{yd}}(\mathbf{z}_d, \mathbf{z}_y), \mathrm{NN}_{\boldsymbol{\theta}_{yd}}(\mathbf{z}_d, \mathbf{z}_y)) \\ p_{\boldsymbol{\theta}_d}(\mathbf{z}_d|\mathrm{d}) &= \mathcal{N}(\mathrm{NN}_{\boldsymbol{\theta}_d}(\mathrm{d}), \mathrm{NN}_{\boldsymbol{\theta}_d}(\mathrm{d})) \\ q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\mathbf{x}) &= \mathcal{N}(\mathrm{NN}_{\boldsymbol{\phi}_y}(\mathbf{x}), \mathrm{NN}_{\boldsymbol{\phi}_y}(\mathbf{x})) \\ q_{\boldsymbol{\phi}_d}(\mathbf{z}_d|\mathbf{x}) &= \mathcal{N}(\mathrm{NN}_{\boldsymbol{\phi}_d}(\mathbf{x}), \mathrm{NN}_{\boldsymbol{\phi}_d}(\mathbf{x})), \end{aligned}$$

where $\mathcal{N}$ denotes a Gaussian distribution with mean and variance each modeled by $\mathrm{NN}_{\boldsymbol{\theta}}(\cdot)$, a neural network with parameters $\boldsymbol{\theta}$.

The conditional prior $p_{\boldsymbol{\theta}_d}(\mathbf{z}_d|\mathrm{d})$ has the effect of making $\mathbf{z}_d$ more dependent on d, by ensuring that signals from d are incorporated into $\mathbf{z}_d$ (and thus facilitating the classification of d given $\mathbf{z}_d$). To further facilitate this domain classification, we add to maximum likelihood the following **domain classification** objective:

$$\mathcal{L}_d(\boldsymbol{x}, d; \boldsymbol{\phi}_d, \boldsymbol{\omega}_d) = \mathbb{E}_{q_{\boldsymbol{\phi}_d}(\mathbf{z}_d|\boldsymbol{x})} \log q_{\boldsymbol{\omega}_d}(d|\mathbf{z}_d),$$

with $\boldsymbol{\omega}_d \in \Omega$ the *domain classifier parameters*. This objective amounts to training a domain classification head, by minimizing the cross-entropy loss based on the source domain labels.

We perform gradient ascent on the final maximization objective:

$$\begin{aligned} &\mathcal{L}(\boldsymbol{x}, d; \boldsymbol{\theta}_{yd}, \boldsymbol{\theta}_d, \boldsymbol{\phi}_d, \boldsymbol{\phi}_y, \boldsymbol{\omega}_d) = \\ &\qquad \mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{x}, d; \boldsymbol{\theta}_{yd}, \boldsymbol{\theta}_d, \boldsymbol{\phi}_d, \boldsymbol{\phi}_y) + \alpha_d \mathcal{L}_d(\boldsymbol{x}, d; \boldsymbol{\phi}_d, \boldsymbol{\omega}_d), \end{aligned}$$

where $\alpha_d \in \mathbb{R}$ is a tradeoff hyperparameter balancing maximum likelihood estimation and domain classification. We do not share the parameters of our encoder networks $\mathrm{NN}_{\boldsymbol{\phi}_y}$ and $\mathrm{NN}_{\boldsymbol{\phi}_d}$, but instead consider a multi-encoder architecture.

DIVAD is similar in spirit to *Domain-Invariant Variational Autoencoders* (DIVA) [20], proposed for image classification. However, our new problem setting of unsupervised anomaly detection leads to major differences from classification-based DIVA. First, by not relying on training class labels, DIVAD fuses DIVA's class-conditioned and residual latent factors $\mathbf{z}_y$ and $\mathbf{z}_x$ into a single, unconditioned domain-invariant factor $\mathbf{z}_y$, considering a conditioning and auxiliary classification objective only for the domain-specific factor $\mathbf{z}_d$. Second, rather than relying on an explicit classifier on top of the class-specific factor $\mathbf{z}_y$, DIVAD derives its anomaly scores from these factors' training distribution, modeled with the flexibility described in the following section.

5.3 Model Inference

Based on Definition 4, our inference goals are to: (1) find a mapping $f_y(\mathbf{x}) = \mathbf{z}_y$ from the input to domain-invariant space, and (2) model the training distribution of $\mathbf{z}_y$ to derive our anomaly scores.

For task (1), a known result from VAE [23, 35] is that after training based on Eq. 1, the variational posterior $q_{\phi_y}(\mathbf{z}_y|\mathbf{x})$ should approximate the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}_y|\mathbf{x})$ (dashed arrow in Figure 2b). We can therefore use it to construct our mapping f_y:

$$\mathbf{z}_y = f_y(\mathbf{x}) \sim q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\mathbf{x}) \approx p_{\boldsymbol{\theta}}(\mathbf{z}_y|\mathbf{x}).$$

For task (2), we propose two alternatives below to model the training distribution of $\mathbf{z}_y$: prior and aggregated posterior estimate.

5.3.1 Scoring from Prior. In the first alternative, we derive the **anomaly score** $g_W(\boldsymbol{x})$ of a sample $\boldsymbol{x}$ as the negative log-likelihood of $f_y(\boldsymbol{x})$ with respect to the prior $p(\mathbf{z}_y)$:

$$\boxed{g_W(\boldsymbol{x}) := -\log p(\mathbf{z}_y = f_y(\boldsymbol{x}))} \tag{2}$$

The rationale behind this method is the following: First, we observe that maximizing Eq. 1 on average on the training set amounts to maximizing the regularization term of the ELBO w.r.t. $\mathbf{z}_y$:

$$\Omega_{\boldsymbol{\phi}_y} := -\mathbb{E}_{p_{\text{train}}(\mathbf{x})} D_{\text{KL}}(q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\mathbf{x}) \| p(\mathbf{z}_y)).$$

However, based on [44], we have the result:

$$\Omega_{\boldsymbol{\phi}_y} = \int \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\boldsymbol{x}_i) \log p(\mathbf{z}_y) d\mathbf{z}_y - \int \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\boldsymbol{x}_i) \log q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\boldsymbol{x}_i) d\mathbf{z}_y,$$

where N_{train} is the number of training samples. By considering:

$$q_{\boldsymbol{\phi}_y}(\mathbf{z}_y) = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\boldsymbol{x}_i),$$

the **marginal**, or **aggregated posterior** [4, 30] (here the empirical distribution of encoded, presumably domain-invariant, samples), we therefore have:

$$\Omega_{\boldsymbol{\phi}_y} = \int q_{\boldsymbol{\phi}_y}(\mathbf{z}_y) \log p(\mathbf{z}_y) d\mathbf{z}_y - \int \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\boldsymbol{x}_i) \log q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\boldsymbol{x}_i) d\mathbf{z}_y$$
$$\Omega_{\boldsymbol{\phi}_y} = -H(q_{\boldsymbol{\phi}_y}(\mathbf{z}_y), p(\mathbf{z}_y)) + H(q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\mathbf{x})),$$

where $H(q_{\boldsymbol{\phi}_y}(\mathbf{z}_y), p(\mathbf{z}_y))$ is the cross-entropy between the aggregated posterior and the prior, and $H(q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\mathbf{x}))$ is the conditional entropy of $q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\mathbf{x})$ with the empirical distribution $\hat{p}_{\text{train}}(\mathbf{x})$ [44]. As we can see, **the maximization process of the ELBO has the effect of trying to make the aggregated posterior $q_{\boldsymbol{\phi}_y}(\mathbf{z}_y)$ match the prior $p(\mathbf{z}_y)$**, which *a priori* motivates the choice above of using the prior to derive anomaly scores.

Fixed Standard Gaussian Prior. We first consider the default $\mathbf{z}_y$ prior in VAEs, a fixed standard Gaussian:

$$p(\mathbf{z}_y) = \mathcal{N}(\mathbf{0}, \boldsymbol{I}),$$

and refer to the method that uses this $\mathbf{z}_y$ prior and scores with Eq. 2 as **DIVAD-G**. A limitation of DIVAD-G is that, although the aggregated posterior and prior *should* be brought closer when maximizing the ELBO, they usually do not end up matching in practice at the end of training [4, 36]. This phenomenon is sometimes described as "*holes in the aggregated posterior*", referring to the regions of the latent space that have high density under the prior but very low density under the aggregated posterior [7].

Learned Gaussian Mixture Prior. A method that has been shown to (at least partly) address the problem of aggregated posterior holes is to replace the fixed $\mathbf{z}_y$ prior with a *learnable prior* $p_{\boldsymbol{\lambda}}(\mathbf{z}_y)$ [7, 44], and hence have the maximization process update both the aggregated posterior and the prior. If sufficiently expressive, the prior can serve as a good approximation $\hat{q}_{\boldsymbol{\phi}_y}(\mathbf{z}_y)$ of the aggregated posterior at the end of training, which makes it safer to use for anomaly scoring:

$$\boxed{g_W(\boldsymbol{x}) := -\log p_{\boldsymbol{\lambda}}(\mathbf{z}_y = f_y(\boldsymbol{x}))} \tag{3}$$

In a way, considering a learnable prior amounts to explicitly performing a joint density estimation of the marginal likelihood and aggregated posterior. With sufficiently expressive priors, this joint estimation also has the effect of *putting less constraints on the aggregated posterior*, letting it capture normal clusters with more variance and arbitrary shapes. This is particularly useful in AD, where the "normal" class can refer to a variety of different behaviors (even for the same entity). In practice, any density estimator $p_{\boldsymbol{\lambda}}(\mathbf{z}_y)$ can be used to model the aggregated posterior. In this work, we consider a Gaussian Mixture (GM) distribution with K components:

$$p_{\boldsymbol{\lambda}}(\mathbf{z}_y) = \sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2),$$

with $\boldsymbol{\lambda} = \{w_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2\}_{k=1}^K$ randomly initialized and trained along with the other parameters. We refer to the method that uses this $\mathbf{z}_y$ prior and scores with Eq. 3 as **DIVAD-GM**.

5.3.2 Scoring from Aggregated Posterior Estimate. An alternative (or complementary) solution to the problem of aggregated posterior holes is to perform the density estimation of the aggregated posterior $\hat{q}_{\boldsymbol{\phi}_y}(\mathbf{z}_y)$ separately, and then define the anomaly score with respect to this estimate instead of the prior:

$$\boxed{g_W(\boldsymbol{x}) := -\log \hat{q}_{\phi_y}(\mathbf{z}_y = f_y(\boldsymbol{x}))} \tag{4}$$

In the following, we consider this alternative in addition to the prior-based scoring for both DIVAD-G and DIVAD-GM. For DIVAD-G, the aggregated posterior is estimated by fitting a multivariate Gaussian distribution to the training samples in latent space. For DIVAD-GM, it is estimated by fitting to them a Gaussian Mixture model with the same number of components K as the prior.

5.4 Putting It All Together

We illustrate the multi-encoder architecture of our DIVAD method in Figure 4, shown here for the learned Gaussian Mixture prior detailed in Section 5.3. From this figure, we can see that encoder networks $\text{NN}_{\boldsymbol{\phi}_d}$ and $\text{NN}_{\boldsymbol{\phi}_y}$ take the same sample $\boldsymbol{x}$ as input to output the mean and variance parameters of multivariate Gaussians $q_{\boldsymbol{\phi}_d}(\mathbf{z}_d|\mathbf{x})$ and $q_{\boldsymbol{\phi}_y}(\mathbf{z}_y|\mathbf{x})$, respectively. These parameters are first used to compute the KL divergence terms of Equation 1, with the parameters of the conditional prior $p_{\boldsymbol{\theta}_d}(\mathbf{z}_d|\text{d})$ outputted by a network $\text{NN}_{\boldsymbol{\theta}_d}$ from the domain d of $\boldsymbol{x}$, and the parameters of $p_{\boldsymbol{\lambda}}(\mathbf{z}_y)$ learned as described in Section 5.3. They are then used to sample the corresponding domain and class encodings of $\boldsymbol{x}$: $\boldsymbol{z}_d$ and $\boldsymbol{z}_y$. These encodings, considered here of same dimension M', are further concatenated to form the input of the decoder $\text{NN}_{\boldsymbol{\theta}_{yd}}$, outputting the parameters of the multivariate Gaussian $p_{\boldsymbol{\theta}_{yd}}(\mathbf{x}|\mathbf{z}_d, \mathbf{z}_y)$, from which the likelihood (or *reconstruction*) term of Equation 1 is computed. The bottom right of the figure finally shows the domain classification head, $\text{NN}_{\boldsymbol{\omega}_d}$, which takes the domain encoding $\boldsymbol{z}_d$ of $\boldsymbol{x}$ as input, and outputs the parameters of the Categorical $q_{\boldsymbol{\omega}_d}(\text{d}|\mathbf{z}_d)$, used to compute the domain classification objective $\mathcal{L}_d$.

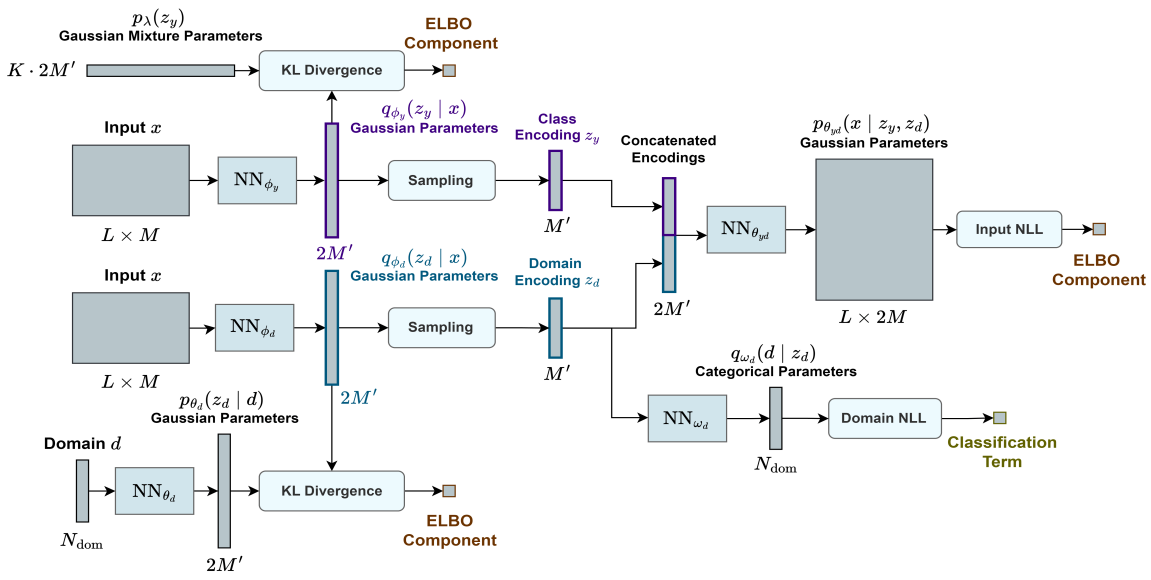


Figure 4: Multi-encoder architecture of our DIVAD-GM models, with N_{dom} the number of source domains (DIVAD-G models use a similar architecture, with the learned Gaussian Mixture parameters replaced with fixed Gaussian parameters).

Regarding computational cost, assuming that the encoding dimension M' and number of source domains N_{dom} are negligible compared to the input dimension $L \cdot M$ (i.e., the network is dominated by its encoder-decoder architecture), DIVAD has the same asymptotic training time as a regular VAE. In practice, DIVAD requires more training resources than VAE, as it uses (i) 2 encoder networks, resulting in twice the encodings and encoder gradients to compute, and (ii) 2 encodings as input to the decoder, leading to more parameters for the first decoder layer. During inference, DIVAD incurs about half the cost of a VAE, since its anomaly scoring involves only a forward pass through a single encoder, compared to a complete input reconstruction for the VAE. Finally, using DIVAD-GM over DIVAD-G incurs modest increase in both training and inference costs, with (limited) $K \cdot 2M'$ prior parameters to learn ($2M'$ for each GM component), and K components, instead of 1, to consider when evaluating likelihoods with respect to the prior.

6 EXPERIMENTS

In this section, we evaluate the anomaly detection performance of DIVAD against existing AD methods using both the Exathlon benchmark [22] and Application Server Dataset (ASD) [27].

6.1 Experimental Setup

Our experimental setup involves different steps of the Exathlon pipeline [22]. In data preprocessing, we excluded applications 7 and 8, for which there are no disturbed and normal traces, respectively. In feature engineering, we dropped the features constant throughout the whole dataset and took the average of Spark executor features to reduce dimensionality. These steps result in $M = 237$ features to use by the AD methods.

Data Partitioning. To build a single AD model for all Spark applications, we ensure that the 8 Spark applications are represented in both the training and test sets. The training set includes normal traces and some disturbed traces to increase the variety in application *settings* and *input rates*. After data partitioning, our test sequences contain 15 Bursty Input (**T1**) anomalies; 5 Bursty Input Until Crash (**T2**) anomalies; 6 Stalled Input (**T3**) anomalies; 7 CPU Contention (**T4**) anomalies; 5 Driver Failure (**T5**) anomalies; 5 Executor Failure (**T6**) anomalies.

Training and Inference. All deep learning methods use the same random 20% of training data as validation. By default, they are trained for 300 epochs, using a Stochastic Gradient Descent (SGD) strategy, mini-batches of size B, the AdamW optimizer [29], a weight decay coefficient of 0.01, early stopping and checkpointing on the validation loss with a patience of 100 epochs.

The hyperparameters are treated by following recommended practices [1, 5]. For *architecture parameters*, we start with the default architecture setting of each AD method as suggested in its original paper, and vary the number of hidden units, number of layers or latent dimension by a factor of 2-4, resulting in n_1 architectures per method (we generate more model variants for shallow methods, leading to larger values of n_1). Then, the *learning rate* is tuned for each architecture, considering $\eta \in \{1e{-}5, 3e{-}5, 1e{-}4, 3e{-}4\}$, and selecting the value that yields the lowest validation loss (i.e., the best *modeling* performance [5]). The *batch size* is method-dependent and set to the value used in the method's original paper (or if absent, to 32 by default). This entails n_1 trained models per method, each with its "best" learning rate and recommended batch size. At inference time (running each model on the test sequences), we derive the record scoring function g using a grid of 12 *anomaly score smoothing factors* γ, leading to $12n_1$ runs per AD method. We further filter out the architectures whose runs give overall poor performance, resulting in $12n_2$ runs, with $n_2 \leq n_1$, per AD method.

We evaluate AD methods based on their *point-based* AD performance, using the **peak F1-score** metric of the Exathlon benchmark

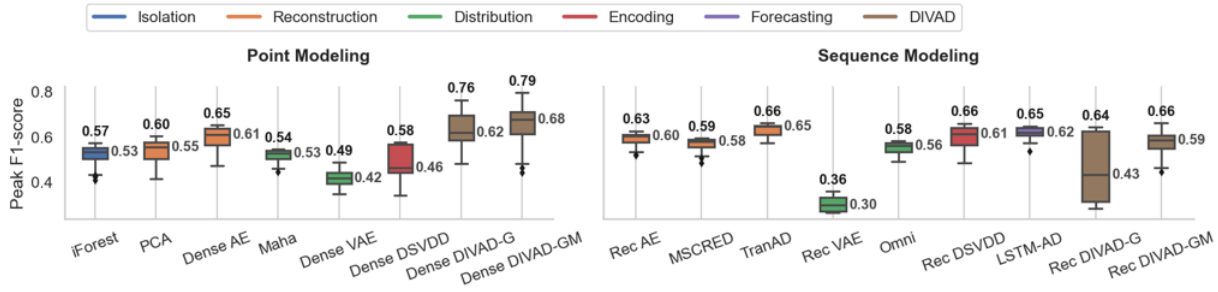


Figure 5: Box plots of peak F1-scores achieved by the existing and DIVAD methods, separated by modeling strategy (point vs. sequence) and colored by method category (from Schmidl et al. [40] plus DIVAD).

(i.e., the "best"-possible F1-score on the Precision-Recall curve). When computing F1-scores, we average Recall values across different event types. We finally summarize the performance of each AD method, in terms of peak F1-score, using a box-plot over its $12n_2$ model runs. Additional details are available in [21].

Representative AD Methods. Our analysis compares DIVAD to 13 unsupervised AD methods, either performing a *point modeling* (i.e., window length $L = 1$) or *sequence modeling* (window length $L > 1$) of the data. The unsupervised AD methods are further grouped based on their anomaly scoring strategy, as per the taxonomy of Schmidl et al. [40] discussed in Section 2.

We include the following **point modeling** AD methods in our study (details about these methods and hyperparameter grids considered are given in [21]): **(1)** Isolation forest [28] (**iForest**), as the most popular isolation tree method; **(2-3)** Principal Component Analysis (**PCA**) [42] and Dense Autoencoder (**Dense AE**) [18, 39], as representative and popular shallow and deep reconstruction methods, respectively; **(4)** Dense Deep SVDD [38] (**Dense DSVDD**), as a recent and popular encoding method; **(5-6)** Mahalanobis [2, 42] (**Maha**) and Dense Variational Autoencoder (**Dense VAE**) [6], as representative shallow and deep distribution methods, respectively.

We include the following **sequence modeling** methods (with further details given in [21]): **(7-9)** Recurrent Autoencoder [18, 39] (**Rec AE**), **MSCRED** [50] and **TranAD** [45], as the sequence modeling version of Dense AE and more recent reconstruction methods, respectively; **(10) LSTM-AD** [31], as the most popular forecasting method; **(11)** Recurrent Deep SVDD [38] (**Rec DSVDD**) as the sequence modeling version of the Dense DSVDD encoding method; **(12-13)** Recurrent VAE (**Rec VAE**) and OmniAnomaly [43] (**Omni**) as the sequence modeling version of Dense VAE and a more recent distribution method, respectively. We also tried to include the more recent encoding method DCDetector [49], but did not retain it due to its poor performance on our dataset.

DIVAD Variants. This study considers 4 DIVAD variants, either performing a point modeling or sequence modeling of the data. The point modeling variants, referred to as **Dense DIVAD-G** and **Dense DIVAD-GM**, use a fully-connected architecture for the encoders NN_{ϕ_d}, NN_{ϕ_y} and decoder $\text{NN}_{\theta_{yd}}$. The sequence modeling variants, referred to as **Rec DIVAD-G** and **Rec DIVAD-GM**, use recurrent architectures instead (more details are in [21]). We employ the same hyperparameter selection strategy for DIVAD variants as the other DL methods, considering KL divergence weights $\beta \in \{1, 5\}$ (i.e., a regular VAE and β-VAE [19] framework with increased latent space regularization), and a domain classification weight $\alpha_d = 100,000$ set based on the scale we observed for the losses $\mathcal{L}_{\text{ELBO}}$ and $\mathcal{L}_d$ in initial experiments. We study the sensitivity of DIVAD to those two hyperparameters in Section 6.2.6.

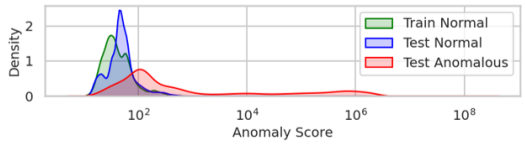
6.2 Results and Analyses using Exathlon

For the Exathlon benchmark (detailed in §3), Figure 5 shows the box plots of the peak F1-scores achieved by the DIVAD variants and 13 AD methods across their hyperparameter values. It separates point from sequence modeling methods into two subplots with a shared y-axis, with boxes colored based on the method category.

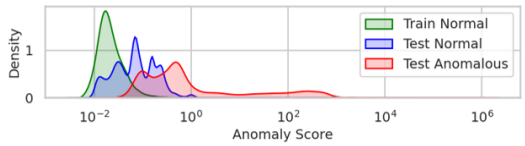
6.2.1 Existing AD Methods. Our main observations about existing AD methods are the following: (1) The best performance achieved is the maximum peak F1-score of 0.66 by TranAD, which is not highly accurate. (2) Across different categories, *reconstruction methods performed the best* (with a maximum peak F1-score of 0.66 by TranAD) *while distribution methods performed the worst* on average (with a maximum peak F1-score of 0.58 by OmniAnomaly). These results are also consistent with the study of [45], which reported that TranAD outperformed OmniAnomaly and MSCRED. (3) The use of deep learning was beneficial among reconstruction methods, while it tended to degrade performance for distribution methods—our subsequent domain shift analysis will explain this behavior.

6.2.2 Analysis of Domain Shift. Figure 6 illustrates the impact of domain shift on AD methods by showing the Kernel Density Estimate (KDE) plots of the anomaly scores they assigned to the training normal, test normal, and test anomalous records. On these plots, the separation between the anomaly scores assigned to the test normal and test anomalous records (i.e., between the **blue** and **red** KDEs) directly relates to the AD performance of a method. As illustrated for the point modeling reconstruction and distribution methods, all the methods have the test normal scores and test anomalous scores overlapping, hence the limited detection accuracy.

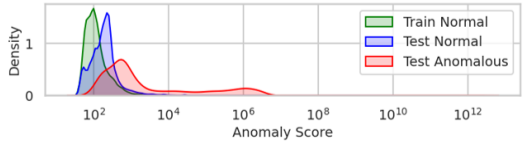
Furthermore, the overlap between the scores of training normal and test normal records (i.e., **green** and **blue** KDEs) reflects its "robustness" to the domain shift from training to test data. (1) Comparing Figures 6c to 6a and 6d to 6b, respectively, we can see that distribution methods, by modeling the training distribution more *explicitly*, tended to produce more similar anomaly scores across the training normal records (i.e., tighter **green** KDEs). However, this tighter modeling of the training distribution also made these methods more sensitive to domain shift, deeming test normal and test anomalous records "similarly anomalous" (i.e., high **blue** and



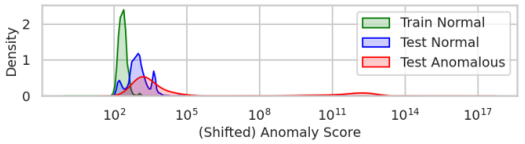
(a) PCA (shallow reconstruction method).



(b) Dense AE (deep reconstruction method).



(c) Maha (shallow distribution method).



(d) Dense VAE (deep distribution method).

Figure 6: Kernel Density Estimate (KDE) plots of the anomaly scores assigned by reconstruction and distribution methods to training normal, test normal and test anomalous records.

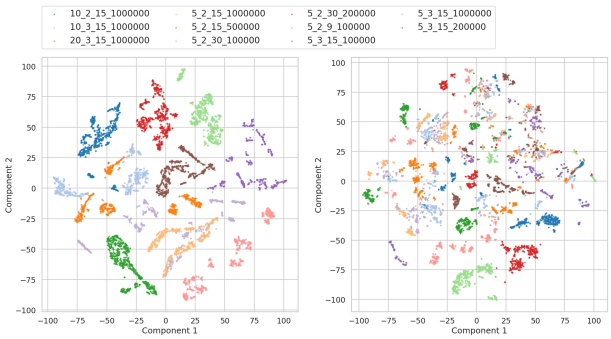


Figure 7: t-SNE scatter plots of Dense DIVAD-GM's domain-specific (left) and domain-invariant (right) encodings of test normal records, undersampled to 10,000 records, balanced and colored by domain.

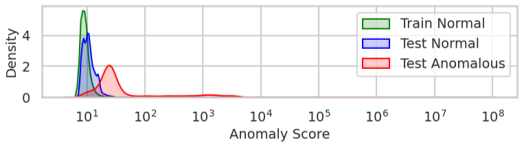


Figure 8: KDE plots of the anomaly scores assigned by Dense DIVAD-GM to training normal, test normal, and test anomalous records.

red KDEs overlap), which hindered their performance. (2) Comparing Figures 6b to 6a and 6d to 6c, we see that deep methods achieved a better separation between the training normal and test anomalous records, by modeling the training data at a finer level than shallow methods. At the same time, they suffer from a larger separation between the training normal and test normal records, indicating their sensitivity to domain shifts. In both cases, **the more a method precisely and explicitly models the training data, the more vulnerable it is to the domain shift challenge**. We made a similar observation for distribution-based OmniAnomaly: its anomaly scores assigned to training normal records and test anomalies overlapped significantly, indicating a shortcoming in normal data modeling (more details are given in [21]).

6.2.3 DIVAD vs. Existing AD Methods. We now examine DIVAD's performance. First, Figure 5 shows that **Dense DIVAD-GM and Dense DIVAD-G significantly outperform the SOTA method TranAD in maximum performance**, achieving 20% and 15% improvements in maximum peak F1-scores (0.79 and 0.76 over 0.66), respectively. Between the variants, using a learned Gaussian Mixture prior (DIVAD-GM) instead of a fixed Gaussian prior (DIVAD-G) is beneficial in improving both the maximum and median peak F1-scores for the point and sequence modeling variants.

Second, **the higher performance of Dense DIVAD-GM can be directly attributed to its accurate domain generalization**. To illustrate this, Figure 7 shows t-SNE scatter plots of the domain-specific and domain-invariant encodings it produced for test normal records (sampled from $q_{\phi_d}(\mathbf{z}_d|\mathbf{x})$ and $q_{\phi_y}(\mathbf{z}_y|\mathbf{x})$, respectively), undersampled to 10,000 data records, balanced and colored by domain.

We can see that the mapping learned by Dense DIVAD-GM from the *input* to its *domain-specific* space produced the distinct domain clusters expected, while the mapping learned from the *input* to its *domain-invariant* space produced more scattered encodings.

Furthermore, Figure 8 shows the KDE plots of the anomaly scores assigned by the best-performing Dense DIVAD-GM to the training normal, test normal and test anomalous records. We see that the explicit modeling of the training data distribution by Dense DIVAD-GM led to a similar benefit as Dense VAE (see Figure 6d), with a low variance in the anomaly scores assigned to the training normal records. Contrary to Dense VAE, Dense DIVAD-GM performed this precise density estimation in a *domain-invariant* space (where distribution shifts were drastically reduced), which made it generalize to *test* normal records as well (i.e., better *aligned* and similarly narrow **green** and **blue** KDEs). As such, Dense DIVAD-GM could generally view test anomalies as "more abnormal" than test normal records, which led to the better performance.

6.2.4 DIVAD Variants. Another observation we can make from Figure 5 is that *point modeling DIVAD variants could outperform TranAD for our dataset and experimental setup, while sequence modeling variants could not.* Point modeling variants being sufficient here can be explained by the dataset's event types being *mostly reflected as* contextual *anomalies* given our features (i.e., data records that are anomalous in a given context/*domain*, but normal in some others). Figure 9 illustrates this by showing KDE plots of the *last completed batch processing delay* feature over normal data and T1 (Bursty Input) events. The top plot shows the distributions for a given test T1 trace, while the bottom plot shows them for the remaining data, using the same x-axis in log scale. We can see that, although T1 events induce higher processing delays than normal *within the context of a trace*, these "higher" values actually appear

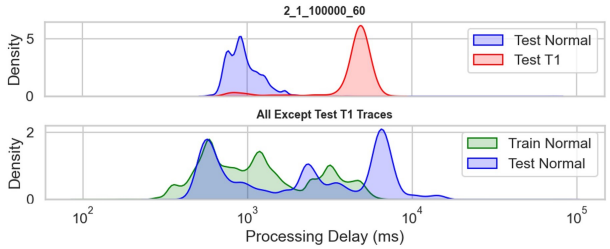


Figure 9: KDE plots of the last completed batch processing delay for training normal data, test normal data, and test anomalous data in a Bursty Input (T1) trace (top) and non-T1 traces (bottom).

normal with respect to the training and test normal data globally, in particular, in some other *contexts/domains*. When viewed in the domain-invariant spaces of our DIVAD methods, such contextual anomalies could typically be turned into *point* anomalies (data records deviating from the rest of the data, no matter the context). Referring back to the central assumption of DIVAD, considering *feature combinations* at *single time steps* at a time was here sufficient for the point modeling methods to learn domain-invariant patterns, given that most anomalies in Exathlon are of the contextual type.

The lower performance observed for sequence modeling DIVAD variants could be explained by the *heightened challenge of learning domain-invariant patterns in the sequential setting*. While leveraging sequential information can be useful in theory, identifying domain-invariant *shapes* within and across $M = 237$ time series constitutes a harder task than relying on simple feature combinations at given time steps for our dataset and setup. This can be verified using the anomaly score distributions, with a higher overlap between the test normal and abnormal records explaining the lower performance.

6.2.5 Sensitivity to Anomaly Scoring Strategy. Figure 10 presents a sensitivity analysis of the anomaly scoring strategy used by our DIVAD methods. It shows the box plots of peak F1-scores achieved by each DIVAD variant and anomaly scoring strategy, with "(P)" indicating the scoring is based on the class encoding *prior* (fixed Gaussian for DIVAD-G, learned Gaussian Mixture for DIVAD-GM), and "(AP)" indicating the scoring is based on the class encoding *aggregated posterior* (estimated as a Gaussian for DIVAD-G, and as a Gaussian Mixture with K components for DIVAD-GM). As we can see from this figure, sequence modeling DIVAD methods again performed worse than the point modeling variants in both median and maximum peak F1-scores no matter the scoring strategy used. Like expected, **deriving the anomaly scores from an aggregated posterior estimate instead of the prior was significantly beneficial for both DIVAD-G methods**, which, by relying on a fixed Gaussian prior, are particularly subject to the issue of "holes in the aggregated posterior" discussed in Section 5.3. By relying on a *more expressive* and *learned* class encoding prior, **DIVAD-GM was less sensitive to the type of scoring strategy used**, with the scoring based on the prior performing better in point modeling, and the one based on the aggregated posterior performing better in sequence modeling. This observation is consistent with our expectations, and motivated our choice of including both scoring strategies into the hyperparameters grid of DIVAD-GM in our study.

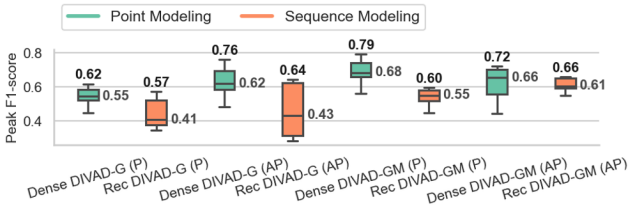


Figure 10: Box plots of peak F1-scores achieved by each DIVAD variant and anomaly scoring strategy (class encoding prior (P) vs. aggregated posterior (AP)), colored by modeling strategy.

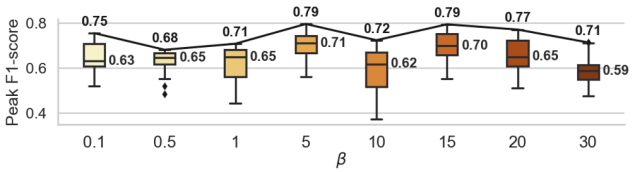


Figure 11: Box plots of peak F1-scores achieved by Dense DIVAD-GM for different KL divergence weights β.

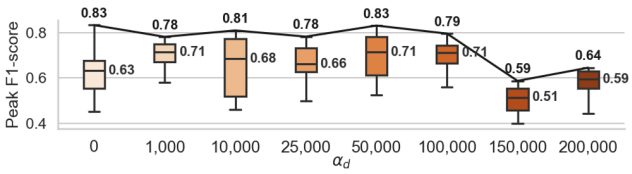


Figure 12: Box plots of peak F1-scores achieved by Dense DIVAD-GM for different domain classification weights α_d.

6.2.6 Sensitivity to Hyperparameters. Figure 11 shows the box plots of peak F1-scores achieved by Dense DIVAD-GM across different KL divergence weights β. From this figure, we can see that finding an optimal β value improves both the maximum and median performance significantly (by up to 16% and 20%, respectively, from the worst value). The benefit of Dense DIVAD-GM over other AD methods, however, remains robust across all β values tested (recall that the best peak F1-score of other AD methods is 0.66).

Figure 12 shows the peak F1-scores achieved by Dense DIVAD-GM across different domain classification weights α_d. We see that its maximum peak F1-score is robust across low and medium α_d values, with some even yielding better results than the value of 100,000 selected for our study. This figure also shows that obtaining the best performance is possible even without domain classification head NN_{ω_d} (i.e., setting $\alpha_d = 0$). However, enforcing domain information via NN_{ω_d} helps reduce Dense DIVAD-GM's sensitivity to other hyperparameters, enabling it to outperform existing methods more *consistently*, with significantly higher median and upper quartile peak F1-scores for suitable α_d values than for $\alpha_d = 0$.

6.2.7 Training and Inference Times. Table 1 shows the average time of training and inference steps (one step per mini-batch of size $B = 32$) for the VAE and DIVAD variants on an NVIDIA A100 80GB PCIe, with hyperparameters adjusted to make DIVAD and VAE directly comparable (more details are in [21]). This table shows the expected trend for training: DIVAD's training steps take about

Table 1: Training and inference times for DIVAD and VAE.

Method	Training Step (ms)	Inference Step (ms)
Dense VAE [6]	3.3	19.4
Dense DIVAD-G	7.3	9.1
Dense DIVAD-GM	9.7	8.9
Rec VAE [6]	7.2	28.1
Rec DIVAD-G	11.6	12.9
Rec DIVAD-GM	13.3	12.4

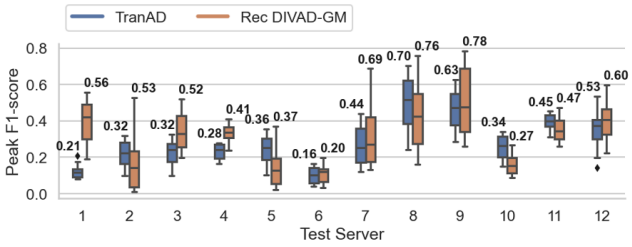
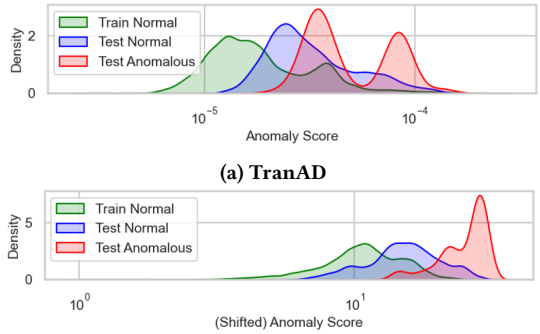


Figure 13: Box plots of peak F1-scores achieved by TranAD and Rec DIVAD-GM for ASD, using each server as a test set.



(a) TranAD

(b) Rec DIVAD-GM.

Figure 14: KDE plots of the anomaly scores assigned by TranAD and Rec DIVAD-GM to ASD's data records, using server 1 as test.

twice the time of VAE's, and DIVAD-GM takes 16.5% longer than DIVAD-G on average for a given architecture. During inference, Table 1 confirms that DIVAD takes less than half the time of VAE, with no significant difference between DIVAD-G and DIVAD-GM.

6.3 Broader Applicability: ASD Use Case

We now study the broader applicability of our DIVAD framework using the Application Server Dataset (ASD) [27]. This dataset, collected from a large Internet company, consists of 12 *traces*, each of which recorded the status of a group of services running on a separate *server*, using 19 metrics every five minutes. The goal is to detect the labeled anomaly ranges located at the end of the traces. The anomaly ratio is 4.61%, with minimum, median and maximum anomaly lengths of 3, 18 and 235 data records, respectively.

In this study, we use ASD to assess the extent to which our DIVAD framework can learn *server-invariant* normal patterns to detect anomalies in a new, *unseen* test server. As such, our experimental setup considers 11 out of the 12 traces as training (without the anomalies) for a single model instance, and the remaining trace as test. For these 12 runs, we report the performance of TranAD (the best-performing existing method) and Rec DIVAD-GM. This time, Rec DIVAD-GM indeed outperformed Dense DIVAD-GM in our experiments, most likely due to (i) a higher presence of *collective* anomalies in ASD (i.e., data records that are anomalous collectively, but not individually), and (ii) the lower number of features $M = 19$, making it easier to identify meaningful domain-invariant shapes among them. We consider a window length $L = 20$ for both methods, and the same model training and selection strategy as in the previous study. More details can be found in [21].

Figure 13 presents the results of our 12 tests, showing the box plots of peak F1-scores achieved by TranAD and Rec DIVAD-GM across their hyperparameter values for each test server. We see that Rec DIVAD-GM outperforms TranAD in maximum peak F1-score for 11 out of 12 test servers (i.e., 92% of the cases), improving the maximum performance by more than 10% for 8 of them. These results also show that the *median* performance is improved by Rec DIVAD-GM for 7 out of the 12 test servers, indicating the sensitivity of DIVAD with respect to hyperparameters, and the necessity of properly tuning them to benefit from a performance gain.

Figures 14a and 14b show the KDE plots of the anomaly scores assigned by the best-performing TranAD and Rec DIVAD-GM to training normal, test normal, and test anomalous records when using server 1 as a test set (i.e., the setup for which Rec DIVAD-GM improved the performance the most, by 167%). From Figure 14a, we can see that the low performance of TranAD was primarily to the lower mode of its distribution of anomaly scores assigned to anomalies, which had a significant overlap, and thus were considered "similarly abnormal", to some test normal data. As shown in Figure 14b, Rec DIVAD-GM was able to alleviate this issue, producing much less overlap between this lower mode and the rest of test normal data, which explains the performance gain.

7 CONCLUSIONS AND FUTURE DIRECTIONS

This paper presented a unified framework for benchmarking anomaly detection (AD) methods, and highlighted the problem of *shifts* in normal behavior in practical AIOps scenarios. We then formally formulated the AD problem under domain shift and proposed a new approach, Domain-Invariant VAE for Anomaly Detection (DIVAD), to learn domain-invariant representations for effective anomaly detection in unseen domains. Evaluation results show that the two main DIVAD variants significantly outperform the best unsupervised AD method using the Exathlon benchmark, with 15-20% improvements in maximum peak F1-scores, and can be applied to the Application Server Dataset to demonstrate broader applicability.

Our future research directions include a *weakly-supervised* extension of DIVAD, combining its explicit modeling of normal behavior shifts with a higher robustness to removing anomaly signals enabled by a few training anomalies, and enhancing the model with *explainability*, indicating the reasons behind anomalies, which will be key to widespread adoption in real-world use cases.

ACKNOWLEDGMENTS

This work was supported by the European Research Council (ERC) Horizon 2020 research and innovation programme (grant n725561).

REFERENCES

[1] Charu C. Aggarwal. 2017. *An Introduction to Outlier Analysis.* Springer International Publishing, Cham, 1–34. https://doi.org/10.1007/978-3-319-47578-3_1

[2] Charu C. Aggarwal. 2017. *Linear Models for Outlier Detection.* Springer International Publishing, Cham, 65–110. https://doi.org/10.1007/978-3-319-47578-3_3

[3] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. 2020. Adversarial Invariant Feature Learning with Accuracy Constraint for Domain Generalization. In *Machine Learning and Knowledge Discovery in Databases*, Ulf Brefeld, Elisa Fromont, Andreas Hotho, Arno Knobbe, Marloes Maathuis, and Céline Robardet (Eds.). Springer International Publishing, Cham, 315–331.

[4] Alexander A. Alemi, Ben Poole, Ian Fischer, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. 2018. Fixing a Broken ELBO. arXiv:1711.00464 [cs.LG]

[5] Sarah Alnegheimish, Dongyu Liu, Carles Sala, Laure Berti-Equille, and Kalyan Veeramachaneni. 2022. Sintel: A Machine Learning Framework to Extract Insights from Signals. In *Proceedings of the 2022 International Conference on Management of Data* (Philadelphia, PA, USA) *(SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 1855–1865. https://doi.org/10.1145/3514221.3517910

[6] Jinwon An and Sungzoon Cho. 2015. Variational Autoencoder based Anomaly Detection using Reconstruction Probability. In *Special Lecture on IE*. https://api.semanticscholar.org/CorpusID:36663713

[7] Matthias Bauer and Andriy Mnih. 2019. Resampled Priors for Variational Autoencoders. arXiv:1810.11428 [stat.ML]

[8] Li Bu, Dongbin Zhao, and Cesare Alippi. 2017. An incremental change detection test based on density difference estimation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47, 10 (2017), 2714–2726.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly Detection: A Survey. *ACM Comput. Surv.* 41, 3, Article 15 (jul 2009), 58 pages. https://doi.org/10.1145/1541880.1541882

[10] Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu C. Aggarwal, and Mahsa Salehi. 2022. Deep Learning for Time Series Anomaly Detection: A Survey. arXiv:2211.05244 [cs.LG]

[11] Kota Dohi, Keisuke Imoto, Noboru Harada, Daisuke Niizumi, Yuma Koizumi, Tomoya Nishida, Harsh Purohit, Ryo Tanabe, Takashi Endo, Masaaki Yamamoto, and Yohei Kawaguchi. 2022. Description and Discussion on DCASE 2022 Challenge Task 2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques. In *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*. Nancy, France, 1–5.

[12] Denis Moreira Dos Reis, Peter Flach, Stan Matwin, and Gustavo Batista. 2016. Fast unsupervised online drift detection using incremental kolmogorov-smirnov test. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1545–1554.

[13] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46, 4 (2014), 1–37.

[14] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* 17, 1 (jan 2016), 2096–2030.

[15] Gartner. 2018. Market Guide for AIOps Platforms. https://www.gartner.com/en/documents/3772124. Accessed: 2025-04-02.

[16] Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. 2014. Outlier Detection for Temporal Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 26, 9 (2014), 2250–2267.

[17] D. M. Hawkins. 1980. *Identification of outliers.* Chapman and Hall, London [u.a.].

[18] Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. 2002. Outlier Detection Using Replicator Neural Networks. In *Data Warehousing and Knowledge Discovery*, Yahiko Kambayashi, Werner Winiwarter, and Masatoshi Arikawa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 170–180.

[19] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations*. https://api.semanticscholar.org/CorpusID:46798026

[20] Maximilian Ilse, Jakub M. Tomczak, Christos Louizos, and Max Welling. 2020. DIVA: Domain Invariant Variational Autoencoders. In *Proceedings of the Third Conference on Medical Imaging with Deep Learning (Proceedings of Machine Learning Research)*, Vol. 121. PMLR, 322–348. https://proceedings.mlr.press/v121/ilse20a.html

[21] Vincent Jacob and Yanlei Diao. 2025. Unsupervised Anomaly Detection in Multivariate Time Series across Heterogeneous Domains. arXiv:2503.23060 [cs.LG] https://arxiv.org/abs/2503.23060

[22] Vincent Jacob, Fei Song, Arnaud Stiegler, Bijan Rad, Yanlei Diao, and Nesime Tatbul. 2021. Exathlon: A Benchmark for Explainable Anomaly Detection over Time Series. *Proc. VLDB Endow.* 14, 11 (jul 2021), 2613–2626. https://doi.org/10.14778/3476249.3476307

[23] Diederik P Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. arXiv:1312.6114 [stat.ML]

[24] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. 2017. On Convergence and Stability of GANs. arXiv:1705.07215 [cs.AI] https://arxiv.org/abs/1705.07215

[25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[26] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. 2018. Deep Domain Generalization via Conditional Invariant Adversarial Networks. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer International Publishing, Cham, 647–663.

[27] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. 2021. Multivariate Time Series Anomaly Detection and Interpretation Using Hierarchical Inter-Metric and Temporal Embedding. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 3220–3230. https://doi.org/10.1145/3447548.3467075

[28] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*. 413–422. https://doi.org/10.1109/ICDM.2008.17

[29] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Bkg6RiCqY7

[30] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2016. Adversarial Autoencoders. arXiv:1511.05644 [cs.LG]

[31] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *23rd European Symposium on Artificial Neural Networks, ESANN 2015, Bruges, Belgium, April 22-24, 2015*. https://www.esann.org/sites/default/files/proceedings/legacy/es2015-56.pdf

[32] Toshihiko Matsuura and Tatsuya Harada. 2020. Domain Generalization Using a Mixture of Multiple Latent Domains. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 07 (Apr. 2020), 11749–11756. https://doi.org/10.1609/aaai.v34i07.6846

[33] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. 2020. Efficient Domain Generalization via Common-Specific Low-Rank Decomposition. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research)*, Vol. 119. PMLR, 7728–7738. http://proceedings.mlr.press/v119/piratla20a.html

[34] Fengchun Qiao, Long Zhao, and Xi Peng. 2020. Learning to Learn Single Domain Generalization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 12553–12562. https://doi.org/10.1109/CVPR42600.2020.01257

[35] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *Proceedings of the 31st International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Eric P. Xing and Tony Jebara (Eds.), Vol. 32. PMLR, Bejing, China, 1278–1286. https://proceedings.mlr.press/v32/rezende14.html

[36] Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. 2019. Distribution Matching in Variational Inference. arXiv:1802.06847 [stat.ML]

[37] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. 2017. Stabilizing Training of Generative Adversarial Networks through Regularization. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/7bccfde7714a1ebadf06c5f4cea752c1-Paper.pdf

[38] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep One-Class Classification. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, 4393–4402. https://proceedings.mlr.press/v80/ruff18a.html

[39] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis* (Gold Coast, Australia QLD, Australia) *(MLSDA'14)*. Association for Computing Machinery, New York, NY, USA, 4–11. https://doi.org/10.1145/2689746.2689747

[40] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly Detection in Time Series: A Comprehensive Evaluation. *Proc. VLDB Endow.* 15, 9 (may 2022), 1779–1797. https://doi.org/10.14778/3538598.3538602

[41] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C. Yuen. 2019. Multi-Adversarial Discriminative Deep Domain Generalization for Face Presentation Attack Detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10015–10023. https://doi.org/10.1109/CVPR.2019.01026

[42] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and Liwu Chang. 2003. A Novel Anomaly Detection Scheme Based on Principal Component Classifier. In *Proceedings of International Conference on Data Mining*.

[43] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2828–2837. https://doi.org/10.1145/3292500.3330672

[44] Jakub M. Tomczak. 2021. Priors in VAEs. https://jmtomczak.github.io/blog/7/7_priors.html. Accessed: 2025-04-02.

[45] Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. 2022. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *Proc. VLDB Endow.* 15, 6 (feb 2022), 1201–1214. https://doi.org/10.14778/3514061.3514067

[46] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[47] Satvik Venkatesh, Gordon Wichern, Aswin Subramanian, and Jonathan Le Roux. 2022. *Disentangled surrogate task learning for improved domain generalization in unsupervised anomalous sound detection.* Technical Report. DCASE2022 Challenge.

[48] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S. Yu. 2023. Generalizing to Unseen Domains: A Survey on Domain Generalization. *IEEE Transactions on Knowledge and Data Engineering* 35, 8 (2023), 8052–8072. https://doi.org/10.1109/TKDE.2022.3178128

[49] Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. 2023. DCdetector: Dual Attention Contrastive Representation Learning for Time Series Anomaly Detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 3033–3045. https://doi.org/10.1145/3580305.3599295

[50] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence* (Honolulu, Hawaii, USA) *(AAAI'19/IAAI'19/EAAI'19)*. AAAI Press, Article 174, 8 pages. https://doi.org/10.1609/aaai.v33i01.33011409

[51] Zhenyu Zhong, Qiliang Fan, Jiacheng Zhang, Minghua Ma, Shenglin Zhang, Yongqian Sun, Qingwei Lin, Yuzhi Zhang, and Dan Pei. 2023. A Survey of Time Series Anomaly Detection Methods in the AIOps Domain. arXiv:2308.00393 [cs.LG]

[52] Zhenyu Zhong, Qiliang Fan, Jiacheng Zhang, Minghua Ma, Shenglin Zhang, Yongqian Sun, Qingwei Lin, Yuzhi Zhang, and Dan Pei. 2023. A Survey of Time Series Anomaly Detection Methods in the AIOps Domain. arXiv:2308.00393 [cs.LG]

[53] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. 2023. Domain Generalization: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 4 (April 2023), 4396–4415. https://doi.org/10.1109/TPAMI.2022.3195549