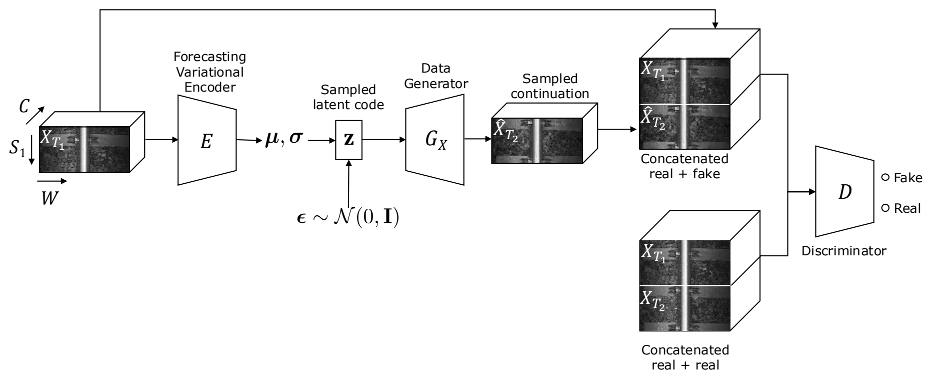
Article:

Xing, Zeyu, Mehmood, Owais and Smith, William Alfred Peter orcid.org/0000-0002-6047-0413 (2025) Unsupervised anomaly detection with a temporal continuation, confidence-aware VAE-GAN. Pattern Recognition. 111699. ISSN 0031-3203

https://doi.org/10.1016/j.patcog.2025.111699



Graphical Abstract

Unsupervised anomaly detection with a temporal continuation, confidence-aware VAE-GAN

Zeyu Xing, Owais Mehmood, William A. P. Smith

Highlights

Unsupervised anomaly detection with a temporal continuation, confidence-aware VAE-GAN

Zeyu Xing, Owais Mehmood, William A. P. Smith

- Propose an unsupervised, zero-shot anomaly detection method for spatiotemporal signals, separating anomalies in predictable regions from unimportant stochastic variations
- Our method is based on using a *forecasting VAE-GAN* to learn the space of plausible continuations of a temporal sequence
- We make the model *confidence-aware* by also learning to predict the pointwise confidence of the reconstruction, allowing us to separate structural from stochastic uncertainty
- Achieve state-of-the-art performance on the ECG5000 [1, 2] and MIT-BIH [3] time series anomaly detection datasets

Unsupervised anomaly detection with a temporal continuation, confidence-aware VAE-GAN

Zeyu Xing[a], Owais Mehmood[b], William A. P. Smith[a]

[a]Department of Computer Science, University of York, York, UK
[b]Omnicom Balfour Beatty, York, UK

Abstract

We propose an unsupervised approach to anomaly detection in data with a temporal dimension. We adapt the VAE-GAN architecture to learn the proxy task of temporal sequence continuation. Rather than reconstructing the input, our variational decoder decodes to a forecast of the future sequence. In order to separate structural uncertainty (which our model can reconstruct by fitting to observed data) from stochastic uncertainty (which it cannot) we introduce an additional decoder that outputs the pointwise confidence of the prediction, after the optimal latent-variable has been found. We can use this for zero-shot anomaly detection, separating anomalies from stochastic variation that cannot be modelled, without any examples. This is important for domains in which anomalies are so rare that it is not possible or meaningful to train a supervised model. As an example of such a domain, we introduce a new dataset comprising linescan imagery of railway lines which we use to illustrate our methods. We also achieve state-of-the-art performance on the ECG5000 and MIT-BIH time series anomaly detection datasets. We make an implementation of our method available at `https://github.com/YorkXingZeyu/ECG-VAEGAN-Project`.

Keywords: time series anomaly detection, unsupervised anomaly detection, variational autoencoder, VAE-GAN

1. Introduction

Temporal sequential data arises across a whole host of data modalities from time series to video to audio. For such data, sequence continuation, completion, interpolation or reordering are emerging as promising proxy tasks for self-supervised feature learning. The overarching premise is that, in order to reason about the future, ordering or interpolation, it is necessary to learn a model that extracts not only low level features but high level concepts as abstract as physical laws (for example, predicting that a falling ball will bounce). Temporal sequences can be further subdivided into those where the observations are overlapping and non-overlapping. Overlapping sequences may observe the same part of the world at different times. For example, adjacent video frames are likely to contain many of the same scene components. Such sequences can be handled in a special way by explicitly modelling the relationship between the same points at different times, for example using optical flow motion fields. This makes the task of future prediction easier since it can, at least partly, be posed as motion prediction of observed scene components.

In this paper, we focus on *non-overlapping* temporal sequences. Examples include audio streams, time series data and linescan images from pushbroom cameras. We propose a generative framework for self-supervised feature learning and anomaly detection based on continuation of such sequences. We use a VAE-GAN [4] as our underlying architecture. The GAN discriminator component ensures that continuations are natural and realistic, by encouraging them to follow the distribution of real complete sequences. This avoids blurring multiple possible futures together. The VAE latent variational variable model captures the stochasticity of future prediction. The distribution mean computed by our variational encoder can be seen as capturing the predictable elements of the future which depend only on the observed portion of the data. The random sampling process from the resulting latent distribution can be seen as exploring possible futures. Within this space we expect to be able to reconstruct structural aspects of the actual future but not stochastic ones. For example (see Figure 5), if we observe a section of a linescan image of a railway line, this constrains the positioning of the next sleeper to a small range of possibilities (structural uncertainty) but the exact configuration of the ballast stones cannot be meaningfully constrained (stochastic uncertainty). We therefore augment the VAE-GAN model with an additional decoder that predicts spatially varying confidence, i.e. the remaining pointwise similarity once the optimal sample from the latent space has been found. Using the same example, we expect high confidence to be assigned to sleepers and low confidence to the ballast. Once trained, our model learns an efficient encoder of the observed data that can be used as a pretrained backbone for downstream tasks. However, the model can additionally be used for *unsupervised* anomaly detection. Where a high confidence region cannot be reconstructed accurately, we can assume the feature is anomalous. It is on this task that we evaluate our proposed model.

While *supervised* anomaly detection methods provide state-of-the-art performance in some domains, for some problems anomalies are so rare that a supervised approach is not possible. For example, in rail surveying, we would like to detect anomalies that have never been observed before. Severe anomalies such as cracks in the railhead are so rare that only single examples may be observed over a period of decades. Posing this as a supervised or weakly supervised problem leads to such severe class imbalance that such approaches fail to learn any meaningful features. On the other hand, unsupervised approaches can use the abundance of non-anomalous data to learn a rich model of normal appearance and treat anomaly detection as the problem of detecting out-of-distribution features. It is this problem setting that we address with the particularly challenging case of also learning to ignore uninteresting stochastic variations.

Our contributions are as follows:

1. We propose an unsupervised, zero-shot anomaly detection method for spatiotemporal signals, separating anomalies in predictable regions from unimportant stochastic variations;
2. Our method is based on using a *forecasting VAE-GAN* to learn the space of plausible continuations of a temporal sequence;
3. We make the model *confidence-aware* by also learning to predict the pointwise confidence of the reconstruction, allowing us to separate structural from stochastic uncertainty in a self-supervised manner;
4. We achieve state-of-the-art performance on the ECG5000 [1, 2] and MIT-BIH [3] time series anomaly detection datasets while also showing application to linescan imagery on a new rail track surveying dataset.

While our method is general and could, in principle, be applied to temporal sequential data from any domain, our evaluation focusses on linescan images and time series data (specifically electrocardiogram traces).

2. Related work

2.1. Self-supervised and generative models

Most commonly, self-supervised learning refers to *feature learning* [5]. Here, self-supervision is used for pretraining only, to discover useful representations of data that are subsequently fine-tuned for other tasks. Examples of proxy tasks that have been used for this purpose include predicting relative position of two regions in the same image [6], colourisation [7], orientation prediction [8], video frame ordering [9], video playback direction [10] and cycle-consistent point tracking [11]. Although these methods obviate the need for supervision, they only provide a route to *feature* learning - i.e. they do not solve any useful task directly, just provide learnt features that can be used for subsequent fine-tuning for a specific task. Another class of approaches use generative models such as GANs. Here, a discriminator or critic provides a supervision signal from an unlabelled dataset while some component of the model learns to extract useful features. Bi-directional GAN (BiGAN) [12] is a variant of a conventional GAN in which an encoder is additionally learnt that maps from the data space to the latent space. Our approach also learns to encode from data space to latent space but this forms only the conditioning signal of our generator, like a conditional GAN [13] and, rather than learning to reconstruct data from the latent space, we learn to predict temporal sequence continuations along with confidence in our prediction. Kingma and Welling [14] introduced the Variational Autoencoder (VAE), a powerful generative model that combines variational inference with autoencoders. This method has proven effective in generating realistic data and learning latent representations. Similarly, He at al. [15] introduced Masked Autoencoders (MAEs) which have demonstrated their scalability and effectiveness in vision learning by leveraging masked signal modelling to improve the representation learning capability of autoencoders. VAE-GANs have been used for stochastic future video frame prediction [16], however we are the first to tackle the problem of non-overlapping sequential data and to introduce estimation of the spatially-varying confidence of the future prediction. Recent advancements in self-supervised learning, such as [17] introduced the Joint-Embedding Predictive Architecture (JEPA) which has further improved visual representation learning by predicting the embedding of masked or missing portions of images. While VAEs have been widely adopted for various applications, such as image generation and data compression, we extend these concepts to tackle the problem of non-overlapping sequential data and introduce the estimation of spatially-varying confidence for future predictions.

2.2. Temporal models

Generative modelling for self-supervised learning has been applied to a number of different data modalities. For time series samples, self-supervision can learn the underlying structural features of unlabeled time series by exploring the inter-sample relationship and intra-time relationship of time series [18]. When dealing with audio or speech data, it is often necessary to convert them into feature vectors [19]. Giri et al. [20] use self-supervised learning to learn a compact representation of normal data using self-supervised classification of metadata based on audio files, to detect anomalies in sound data. At the same time, self-supervised pretraining for Automated Speech Recognition (ASR) also makes great progress in processing audio data [21]. Later, based on ASR and to supplement the ability to compare learning in self-supervision, [22] proposed to co-learn the presentation from different models of speech and literacy during pre-training.

For video data, the first is Arrow of Time, which will help tell whether a video is running forward or backward [23]. Since video data cannot be captured simply through a two-dimensional CNN, some researchers propose to use three-dimensional CNN to solve space-Time cubic puzzles of videos [24]. Long short term memory (LSTM) networks tend to be used when processing such temporal data. [25] tried to use LSTM to learn the representation of time series, using the encoder-decoder LSTM model to rearrange the shuffled input sequence in the correct order. Tao et al. [26] propose the pretext-Contrastive Learning (PCL) model on the basis of pretext-task and comparison learning and applied it to self-supervised video feature learning. Similarly, the Video-based Temporal-Discriminative Learning (VTDL) framework is used to process unlabelled video data [27]. For the video future prediction task, the purpose is to predict the future frame sequence or the future frame sequence feature. The idea is to make predictions by parsing a given finite number of video frames [28]. For models, the hope is that they can learn the dynamics of these known sequences of frames, the more famous of which is the LSTM [29], and many methods have been proposed afterwards [30, 31, 32, 33, 34]. Many algorithms use LSTM to deal with time dynamic problems in video [31, 33, 34]. These methods can be used in some self-supervised feature learning tasks, and the advantage is that no manual labelling of data is required. MCnet [34] has two encoders that learn the spatial features of the image and the temporal dynamics of the video. They output temporal and spatial characteristics of the data, which are fed into the decoder to predict future videos.

In the exploration of time series anomaly detection, many outstanding methods have achieved remarkable results. Giannoulis et al. [35] presents Ditan, a deep-learning domain-agnostic framework tailored for the detection and interpretation of anomalies in multivariate time series data. The framework employs Convolutional Neural Networks (CNNs) to extract local features from time series data and LSTMs to capture long-term dependencies in the sequences to identify temporal patterns and anomalies across various datasets and applications, demonstrating its adaptability and effectiveness in handling diverse time series anomaly detection tasks. Audibert et al. [36] explore the role of deep neural networks in multivariate time series anomaly detection. The study utilizes deep learning techniques such as CNNs, Recurrent Neural Networks (RNNs) and their variant LSTM networks, as well as Autoencoders. These models effectively capture complex temporal dependencies and patterns, significantly improving the performance of anomaly detection in complex multivariate time series. Mokoena et al. [37] address the challenge of explaining anomalies detected in time series data using a method called sequential explanations. It underscores the importance of not just identifying anomalies but also understanding their underlying causes. The proposed method provides detailed, step-by-step explanations for detected anomalies, enhancing interpretability and aiding in more informed decision-making for time series anomaly detection.

Pereira and Silveira [38] explore learning representations from healthcare time series data for unsupervised anomaly detection. The study utilizes Autoencoders to learn normal data patterns and detect reconstruction errors, CNNs to extract local features and patterns, and RNNs along with LSTM networks to capture temporal dependencies. By combining these models, the research effectively extracts meaningful features from complex healthcare time series data to achieve efficient unsupervised anomaly detection.

2.3. Anomaly detection

Significant advancements have also been made in the exploration of unsupervised and semi-supervised learning methods. Yang et al. [39] propose an unsupervised anomaly detection and segmentation method by learning deep feature correspondence. The method effectively detects and segments anomalies without the need for labeled data, using deep neural networks to automatically extract relevant features from the input data. The approach demonstrated superior performance across multiple real-world datasets. Zhang et al. [40] introduces a novel deep anomaly detection method combining self-supervised learning and adversarial training. By employing Generative Adversarial Networks (GANs), the model is able to self-supervise during the training process, thereby improving the accuracy and robustness of anomaly detection. Experimental results show that this method significantly enhances detection performance across various datasets. Zavrtanik et al. [41] presents a visual anomaly detection method based on image inpainting, utilizing inpainting techniques to detect and localize anomalous regions. By comparing the normal parts of an image with the inpainted version, the method effectively identifies and marks anomalies. It demonstrated high efficiency and accuracy in multiple visual detection tasks. Similar to our approach, Zhou et al. [42] use an autoencoder but propose to use the latent representation itself as part of the anomaly detection process. However this approach requires weak supervision whereas ours is completely unsupervised.

Akcay et al. [43] introduce GANomaly, a semi-supervised anomaly detection method using adversarial training. GANomaly employs a combination of a generator and discriminator within a GAN framework to learn the underlying data distribution and identify anomalies. The method shows strong performance in various computer vision tasks, such as image-based anomaly detection, by effectively learning to differentiate between normal and abnormal data patterns. BeatGAN [44] also uses a generative model for anomaly detection. Like GANomaly, the idea is to learn the distribution of normal data and detect anomalies as hard-to-reconstruct data samples. However, unlike our model, they model and reconstruct the whole signal whereas we learn to forecast the continuation of a given signal segment. We believe this proxy task of temporal continuation leads to a better model of the underlying features of the data.

Like in our work, Tang et al. [45] consider linescan image data. However, they do not treat the data as temporal, instead working with fixed size images in which the temporal dimension is a second spatial dimension. They tackle the supervised anomaly detection problem for industrial inspection using a skip autoencoder and deep feature extractor. The skip autoencoder captures multi-scale features by incorporating skip connections, while the deep feature extractor enhances the representation of the input data. This combination significantly improves the accuracy of anomaly detection in industrial settings, demonstrating robustness in identifying defects in complex environments.

Specifically related to anomaly detection in rail images, Liu et al. [46] present a machine vision-based method for inspecting rail fastener defects across multiple railways. The approach utilizes image processing and deep learning techniques to automatically detect and classify defects in rail fasteners, ensuring the safety and reliability of railway infrastructure. The proposed method achieves high precision and efficiency, making it suitable for large-scale railway maintenance applications.

Modern approaches to time series anomaly detection were recently surveyed by Zamanzadeh et al. [47]. We conclude the literature review by mentioning the most recent and relevant methods. Wang et al. [48] propose a VAE that is conditioned on both global and local frequency features. This improves reconstruction normal data significantly. Kang and Kang [49] use a transformer to model temporal dependencies and relationships among variables via self attention across these two dimensions. Miao et al. [50] combine GAN losses with a transformer-based autoencoder while incorporating a contrastive loss into the discriminator which helps improve generalisation of the normal model. CARLA [51] also uses a contrastive loss but proposes to inject anomalies to create negative samples for contrastive learning. Kim et al. [52] consider the

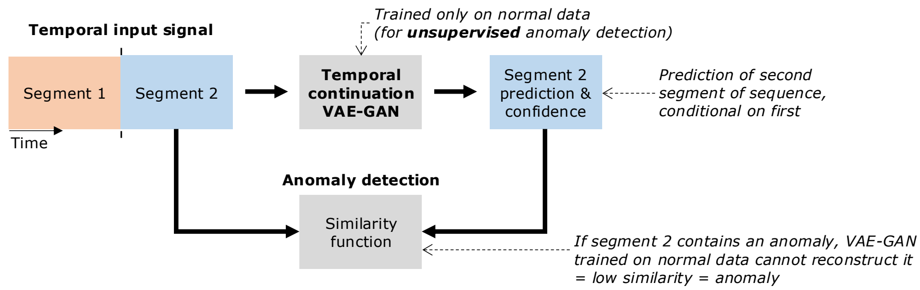


Figure 1: Overview of unsupervised anomaly detection method using a temporal continuation VAE-GAN. An input signal with a temporal dimension is divided into two segments. The temporal continuation VAE-GAN predicts the second segment, conditional on the first. This VAE-GAN is trained to learn the space of normal signals, including the subspace of plausible continuations and a pointwise confidence estimate to distinguish structural uncertainties (which we expect the model to be able to capture) from stochastic uncertainties (which we do not). If the second segment contains an anomaly, we do not expect the VAE-GAN to be able to accurately reconstruct it and this dissimilarity should be measurable and indicative of an anomaly. Since the VAE-GAN only needs to see normal data, this provides a means to perform unsupervised anomaly detection.

problem of test-time adaptation when a learnt normal model must deal with distributional shift at test-time. Other generative architectures have also been considered. Zhou et al. [53] use normalising flows as a generative model for both anomaly detection and localisation. Yao et al. [54] use a diffusion model to remove anomalies. However, they propose to adapt the level of noise such that it is appropriate to the scale of anomaly. Dai et al. [55] also use a diffusion model but for generating synthetic anomalies without a prior. Finally, Liu et al. [56] tackle the problem of unsupervised anomaly detection in the context of continual learning. Here, the task is to incrementally learn different anomalies without forgetting those learnt earlier.

3. Method

Our goal is to learn the space of normal variation of a temporal signal. We pose this in terms of estimating the temporal continuation of a given signal segment. However, rather than estimate a single point estimate, we predict the subspace of possible continuations. This provides a route to unsupervised anomaly detection since we can measure

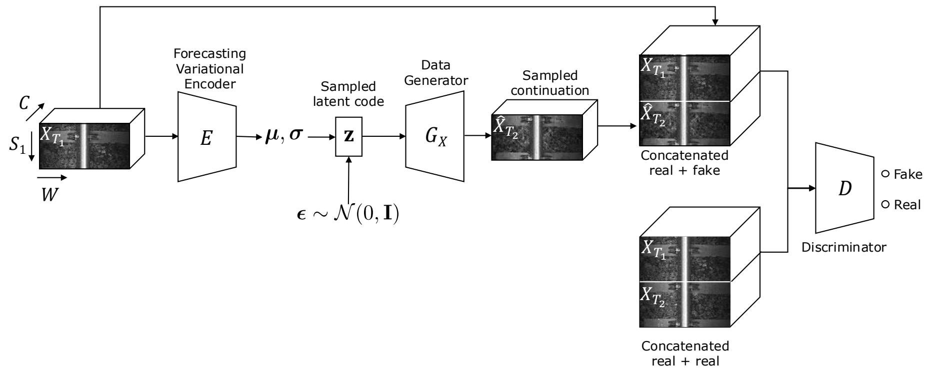


Figure 2: The temporal continuation VAE-GAN architecture. From the observed part of the time series X_{T_1}, the forecasting variational encoder E computes the parameters of a latent distribution, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. The data generator, G_X, decodes a sample from this latent distribution, $\mathbf{z}$, into a prediction of the following time-steps $\hat{X}_{T_2}$. The discriminator D is given real, cat(X_{T_1}, X_{T_2}), or fake, cat$(X_{T_1}, \hat{X}_{T_2})$, concatenated time series and seeks to distinguish them.

how the true continuation differs from the predicted subspace (see the overview in Figure 1). Our underlying model is a temporal continuation VAE-GAN (see Section 3.2). This model learns to encode a given segment of signal to the subspace of possible continuations, represented as a mean and variance of a latent representation. Sampling from this distribution and decoding provides a possible continuation. Our model also learns to predict a pointwise confidence map so that it learns in an unsupervised manner which regions of the continuation are predicted with high confidence (see Section 3.3). It is in these regions that we expect to be able to reliably detect anomalies. The confidence map represents the predicted pointwise confidence of the continuation *after the optimal latent representation has been found*. This optimal representation is found in practice via a process of analysis-by-synthesis to fit the model (see Section 3.4). Our model is trained with several losses described in Section 3.6. Specifically, the objective is that the predicted subspace contains the true continuation observed in the training data and that the latent space is well-behaved (achieved using conventional VAE-GAN losses) but also that the subspace of continuations is diverse and not overfitted to the particular observed continuations.

Intuitively, our model allows us to answer the question: “Given the first part of a temporal sequence, what possible continuations do we expect to see?" Then, given an actual continuation, we can ask: "How far does the actual continuation lie from the subspace of possible continuations that the model predicted?" Finally, our confidence map allows us to ask: "How confident is the model in its prediction at each output point?" Together, the answers to the second and third question allows us to detect anomalies when we see a features in a continuation that our model cannot predict yet our model is confident in the prediction of those features.

3.1. Problem statement

Consider a signal with zero or more spatial dimensions, one or more channels and a temporal dimension that is observed at S_1 evenly spaced time steps. We represent this observation by the tensor $X_{T_1} \in \mathbb{R}^{W \times C \times S_1}$, where C is the number of channels and the spatial dimension W may be expanded or dropped as appropriate to the particular signal. We are interested in the task of predicting the signal at the following S_2 time steps, i.e. predicting the tensor $X_{T_2} \in \mathbb{R}^{W \times C \times S_2}$ given X_{T_1}. Hat denotes an estimated quantity, e.g. $\hat{X}_{T_2}$ is the prediction of the true X_{T_2}.

3.2. Temporal Continuation VAE-GAN

The first component of our model is a VAE-GAN, as shown in Figure 2. However, unlike a conventional VAE-GAN, we do not seek to autoencode, i.e. to reconstruct samples similar to the input. Instead, we decode to a continuation of the temporal sequence. Therefore, the job of the encoder is to find latent distribution parameters that model the space of possible continuations. We do not use a 'content' (or 'data') loss that directly penalises differences between X_{T_2} and $\hat{X}_{T_2}$ as in a VAE or autoencoder. Instead, we require only that the continuation is plausible (as measured by the discriminator) as in a GAN. The discriminator sees the concatenation of the observed part of the sequence and its predicted continuation and can therefore judge whether the continuation is plausible given the observation. The VAE-GAN part of our model comprises the following components.

Forecasting Variational Encoder. The forecasting variational encoder is a pair of functions $\boldsymbol{\mu}, \boldsymbol{\sigma} : \mathbb{R}^{C \times S_1 \times W} \rightarrow \mathbb{R}^d$ such that $\boldsymbol{\mu}(X_{T_1}), \boldsymbol{\sigma}(X_{T_1})$ provides the parameters of the normal distribution corresponding to the embedding of X_{T_1} into a d-dimensional space. The mean, $\boldsymbol{\mu}(X_{T_1})$, of this distribution encodes the predictable aspects of X_{T_2}, while $\boldsymbol{\sigma}(X_{T_1})$ describes the shape of the distribution characterising the uncertain aspects.

Data generator. Unlike in a conventional VAE or VAE-GAN, our generator (or decoder) does not seek to reconstruct the original input data. Instead, it predicts the temporal continuation of the input data. We call this our data generator to distinguish it from the confidence generator later. The data generator is a function $G_X : \mathbb{R}^d \rightarrow \mathbb{R}^{C \times S_2 \times W}$ such that $\hat{X}_{T_2} = G_X(\mathbf{z})$ is a prediction of X_{T_2} conditioned on latent vector $\mathbf{z}(X_{T_1}, \boldsymbol{\epsilon}) = \boldsymbol{\mu}(X_{T_1}) + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}(X_{T_1})$ where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ is random noise drawn from a normal distribution. The idea is that $\boldsymbol{\mu}$ should encode the predictable aspects of X_{T_2} while $\boldsymbol{\epsilon}$ provides a space in which to explore the structurally or stochastically uncertain aspects.

Discriminator. The discriminator is a function $D : \mathbb{R}^{C \times S_1 + S_2 \times W} \rightarrow [0, 1]$ that is given a concatenation of the observed X_{T_1} and either the true (X_{T_2}) or predicted ($\hat{X}_{T_2}$) continuation and returns the probability that the concatenated observation is drawn from the true data distribution. i.e. $D\left(\text{cat}(X_{T_1}, X_{T_2})\right)$ aims to predict whether $\text{cat}(X_{T_1}, X_{T_2})$ is real or fake, where cat concatenates tensors along the temporal dimension.

3.3. Confidence prediction and model fitting

We further augment our Temporal Continuation VAE-GAN with a means to predict confidence in the continuation for each spatiotemporal location. This is important for distinguishing between anomalous deviations from normality and stochastic variations that we do not expect the model to be able to reconstruct. We supervise the confidence prediction based on the actual error between the true continuation and the *best possible fit* of the model to the continuation. Concretely, when given access to the true X_{T_2}, we optimise the noise $\boldsymbol{\epsilon}$ to minimise the error between X_{T_2} and $\hat{X}_{T_2}$. The remaining error represents the inability of the model to explain all of X_{T_2} and we use this to supervise the confidence map.

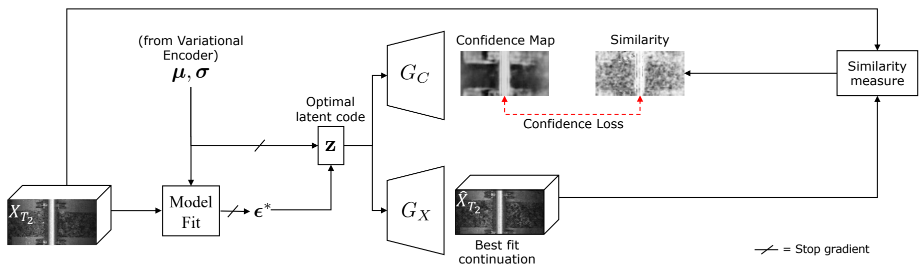


Figure 3: The confidence generator, G_C, predicts a confidence map from an optimal latent code. This should correspond to the spatial similarity between X_{T_2} and $\hat{X}_{T_2}$ when the optimal $\boldsymbol{\epsilon}^*$ has been found via a model fitting procedure (through which we do not propagate gradients) that minimises the reconstruction error.

Confidence generator. The confidence generator is a function $G_C : \mathbb{R}^d \rightarrow [0, 1]^{S_2 \times W}$ such that $\mathbf{C}_m = G_C(\mathbf{z}(X_{T_1}, \boldsymbol{\epsilon}^*))$ is a single channel confidence map of the same spatiotemporal dimension as X_{T_2}. Entries in $\mathbf{C}_m$ represent the confidence (a value in the range $0 \ldots 1$) of the prediction of X_{T_2} at the corresponding spatiotemporal location. The intention is that this confidence value reflects the similarity between X_{T_2} and $\hat{X}_{T_2}$ when the latent vector with optimal $\boldsymbol{\epsilon}$ is passed to G_X (see Model Fitting below). The definition of similarity depends upon the choice of similarity measure used in the confidence loss.

3.4. Model fitting

Suppose we are given both an observed X_{T_1} and the true continuation X_{T_2}. We want to find the best representation within our model of this observation, i.e. to fit the model. This entails finding the optimal $\boldsymbol{\epsilon}^*$ such that $G_X(\mathbf{z}(X_{T_1}, \boldsymbol{\epsilon}^*))$ best fits the true continuation X_{T_2}. We solve the analysis-by-synthesis optimisation problem:

$$\boldsymbol{\epsilon}^* = \arg\min_{\boldsymbol{\epsilon}} \|G_X(\boldsymbol{\mu}(X_{T_1}) + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}(X_{T_1})) - X_{T_2}\|_1. \tag{1}$$

This seeks to minimise the L_1 difference between actual and synthesised X_{T_2}. We use this optimal model fit to compute the similarities that are used to train the confidence generator. Specifically, we solve the optimisation problem using gradient descent for a fixed number of iterations within the outer training loop.

3.5. Training the confidence generator

The confidence generator is trained using model fitting as shown in Figure 3. The model fitting process is used as an oracle that provides the optimal latent code corresponding to the best fit continuation. The difference between the best fit and true continuations is determined using a data-specific similarity measure. The confidence generator is supervised to predict confidence maps that are close to the true similarity. We do not propagate gradients from the confidence generator or through the model fitting optimisation process into the variational encoder. So the confidence generator can either be trained independently of the temporal continuation VAE-GAN or in parallel with it.

3.6. Losses

The goal of our Temporal Continuation VAE-GAN is to learn the space of plausible continuations, conditioned on the observation X_{T_1}. Training only with a reconstruction or content loss as in a VAE encourages overfitting and collapse of the latent space to predict only the true continuation without any diversity. Instead, we use a discriminator and GAN loss to ensure that all continuations are plausible and a diversity loss to ensure the latent distributions capture meaningful and significant variation. Our assumption is that, if both of these are satisfied, then the true continuation lies somewhere in the latent space. In addition, as in a VAE we impose a prior regularisation loss on the predicted distributions using the KL divergence. This encourages a well-behaved latent space in which the model fitting optimisation can smoothly converge to a good solution. We now describe the various losses used during training.

Generator loss. We use a binary cross entropy loss for the discriminator. Although other GAN losses could be used (we experimented with the WGAN) we found this simple loss to work well for our applications. To update the generator, we compute this with inverted labels, i.e. seek to maximise the probability of being real for a batch of N fake images:

$$\mathcal{L}_{\text{gen}} = -\sum_{i=1}^{N} \log D(\text{cat}[X^i_{T_1}, G_X(\mathbf{z}(X^i_{T_1}, \boldsymbol{\epsilon}))]). \quad (2)$$

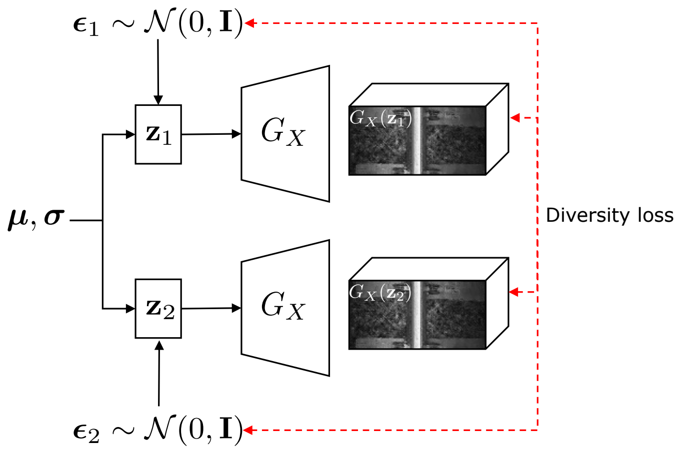


Figure 4: Each time the forecasting variational encoder estimates the latent distribution, we draw two different samples. The diversity loss encourages that, when the samples are further apart, so should the decoded continuations be further apart.

Discriminator loss. To update the discriminator, we compute binary cross entropy loss for a batch of correctly labelled fake and real images:

$$\mathcal{L}_{\text{dis}} = -\sum_{i=1}^{N} \log D(\text{cat}[X^i_{T_1}, X^i_{T_2}]) + \log(1 - D(\text{cat}[X^i_{T_1}, G_X(\mathbf{z}(X^i_{T_1}, \boldsymbol{\epsilon}))])). \tag{3}$$

Prior loss. We use the KL divergence as a prior loss to encourage the latent distribution for every input to be close to a standard normal distribution:

$$\mathcal{L}_{\text{KL}} = \sum_{i=1}^{N} \sum_{j=1}^{d} \mu_j(X^i_{T_1})^2 + \sigma_j(X^i_{T_1})^2 - \log \sigma_j(X^i_{T_1}) - 1 \tag{4}$$

Diversity loss. This loss encourages diversity within the latent space, i.e. that a large change in $\mathbf{z}$ should correspond to a large change in $\hat{X}_{T_2}$. We ensure this using the diversity loss proposed by [57]:

$$\mathcal{L}_{\text{diversity}} = \sum_{i=1}^{N} \frac{\|\mathbf{z}(X^i_{T_1}, \boldsymbol{\epsilon}_1) - \mathbf{z}(X^i_{T_1}, \boldsymbol{\epsilon}_2)\|_1}{\|G_X(\mathbf{z}(X^i_{T_1}, \boldsymbol{\epsilon}_1)) - G_X(\mathbf{z}(X^i_{T_1}, \boldsymbol{\epsilon}_2))\|_1 + \varepsilon}, \tag{5}$$

where ε is a small constant for numerical stability, and $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ are two random samples. See Figure 4. This loss was previously used in the context of GANs and has not been used in the context of VAE-GANs or temporal continuation previously.

Algorithm 1 Training our proposed model

```
while not converged do
    Sample random batch of real images (X^1_{T_1}, ..., X^B_{T_1})
    # Phase 1: encourage G_X and E to produce
    # images that are more realistic and diverse
    Generate two continuations for each real image:
        G_X(z(X^i_{T_1}, ε_1)) and G_X(z(X^i_{T_1}, ε_2))
    Compute L_gen, L_diversity and L_KL and backpropagate into G_X and G_E
    # Phase 2: encourage G_C to predict confidence
    # consistent with similarity using optimal z
    Fit the model to the current target images by solving (1)
    Compute L_confidence and backpropagate into G_C
    # Phase 3: improve discriminator D
    # to better detect fake images
    Compute L_dis and backpropagate into discriminator D
    Take gradient descent step
    Zero gradients
end while
```

Confidence loss. The confidence loss measures the L_1 error between the confidence map predicted by G_C and the true similarity map $\mathbf{S} \in [0,1]^{S_2 \times W}$:

$$\mathcal{L}_{\text{confidence}} = \sum_{i=1}^{N} \left\| G_C(\mathbf{z}(X^i_{T_1}, \boldsymbol{\epsilon}^*) - \mathbf{S}^i \right\|_1 \tag{6}$$

where $\mathbf{S}^i = s(G_X(\mathbf{z}(X^i_{T_1}, \boldsymbol{\epsilon}^*), X^i_{T_2})$ is computed according to some similarity function $s : \mathbb{R}^{S_2 \times W} \times \mathbb{R}^{S_2 \times W} \to [0,1]^{S_2 \times W}$. There are many ways we might choose to define similarity depending on the nature of the data. We specify what was used for each dataset below.

3.7. Implementation

Each iteration of our training pipeline comprises three phases, as shown in Algorithm 1. In each phase, losses that relate to different components of our model are calculated and backpropagated before a gradient descent step is taken on the accumulated gradients. We use the RMSProp optimiser. We implement our generators and discriminator as convolutional neural networks, though this choice is orthogonal to our overall idea and any architecture (such as a transformer) could be used. We follow the DCGAN architecture for each component, adapting filter sizes to accommodate spatial input size of $S_1 \times W$ for the encoder, $S_2 \times W$ for the generator and $S_1 + S_2 \times W$ for the discriminator. Our generators use batchnorm and ReLU activation with tanh activation at the output layer while our discriminator uses batchnorm, LeakyReLU activation and sigmoid activation for the output.

3.8. Unsupervised anomaly detection

Assuming that our model has been trained only on normal data (i.e. excluding anomalies) then, given real observation X_{T_1} and its true continuation X_{T_2}, we can use our model to assess whether X_{T_2} contains any anomalies. The difference between $\hat{X}_{T_2} = G_X(\mathbf{z}(X_{T_1}), \boldsymbol{\epsilon}^*)$ and X_{T_2} indicates which parts of the real continuation were difficult for the model to reconstruct. However, we know that our prediction will only be reliable in non-stochastic regions of the continuation, i.e. where the model is confident. We can therefore produce an anomaly map, $\mathbf{A}$, that scales errors by their corresponding confidence:

$$\mathbf{A} = G_C(\mathbf{z}(X_{T_1}, \boldsymbol{\epsilon}^*)) \odot e(G_X(\mathbf{z}(X_{T_1}, \boldsymbol{\epsilon}^*)), X_{T_2}), \tag{7}$$

where $e(\cdot, \cdot)$ is a data-specific error function. Large values in this map, indicate regions where the model is confident in its prediction but the prediction is very different to the data - i.e. an anomaly. For anomaly detection, we threshold these anomaly maps, count the number of anomalous-labelled points and then threshold the count to classify data as anomalous.

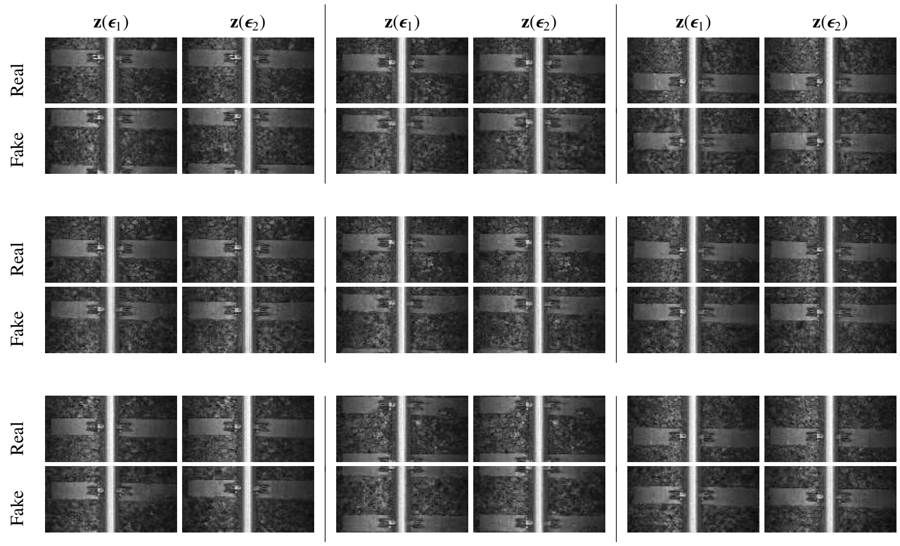


Figure 5: Illustration of quality of temporal continuation and diversity. Each 2×2 block of images shows the same X_{T_1} (observed real image) in the top row and two different $\hat{X}_{T_2}$ (fake images) in the bottom row, produced by two different random samples from the latent space. Both should provide plausible continuations of the real image while also showing diversity between the two samples either in stochastic elements (such as the ballast in the background) or structural elements (such as the precise positioning of the sleepers or clips).

4. Results

4.1. Datasets

We provide experimental results on three different datasets across two modalities to demonstrate the performance of our method. Testing our approach on other modalities of data such as audio or time series from a source other than ECG is left to future work.

We use a dataset of grayscale ($C = 1$) railtrack images in order to qualitatively evaluate the behaviour of our model. This is captured with a linescan camera mounted on the underside of a track inspection car, the vision system illuminates the track with a series of LED wire lights and gets images of the track and its surroundings as the car moves along it at speeds of up to 125mph. The resolution of the original images is $W =$ 2,048, $H =$ 15,000 where H corresponds to the temporal dimension. We take crops of size 2048×2048 via sliding a window vertically along the images with a step size of

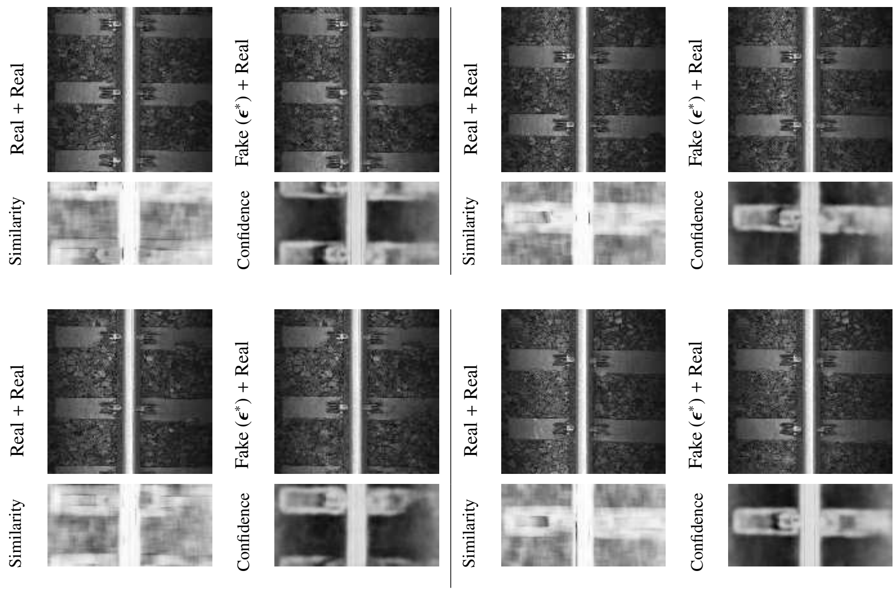


Figure 6: Model fitting and confidence prediction. For each example (comprising two rows and two columns), the first row of the first column shows an observed image X_{T_1} and its true continuation X_{T_2}. The first row of the second column shows the observed image X_{T_1} and its predicted continuation $\hat{X}_{T_2}$ using the optimal $\boldsymbol{\epsilon}^*$ after fitting the model to the observed X_{T_2}. The similarity between X_{T_2} and $\hat{X}_{T_2}$ (according to the structural similarity index) is shown in the second row of the first column while the estimated confidence, having seen only X_{T_1} is shown in the second row of the second column.

100 pixels. We then downsample the images to size 128×128 for $W = 128$ and split equally into size $S_1 = S_2 = 64$. From 20 linescan images, this leads to a dataset of 10k 128×128 images. It is assumed that there are no anomalies within this training set and we use no labels. For this dataset, we use the structural similarity [58] to supervise confidence maps, i.e. $s(x, y) = \text{SSIM}(x, y)$. This measures similarity over a local region at each point, rather than only pixel-wise similarity. This is helpful in reflecting low confidence in stochastic regions. For anomaly detection we use negated similarity as our error measure, i.e. $e(x, y) = 1 - s(x, y)$.

We provide quantitative evaluation on the ECG5000 benchmark. This is a time series anomaly detection benchmark. It forms part of the UCR time series archive [1] and comprises 5,000 electrocardiogram (ECG) single channel ($C = 1$) traces from the dataset originally collected by [2]. Each trace consists of a total of $W = 140$ uniform

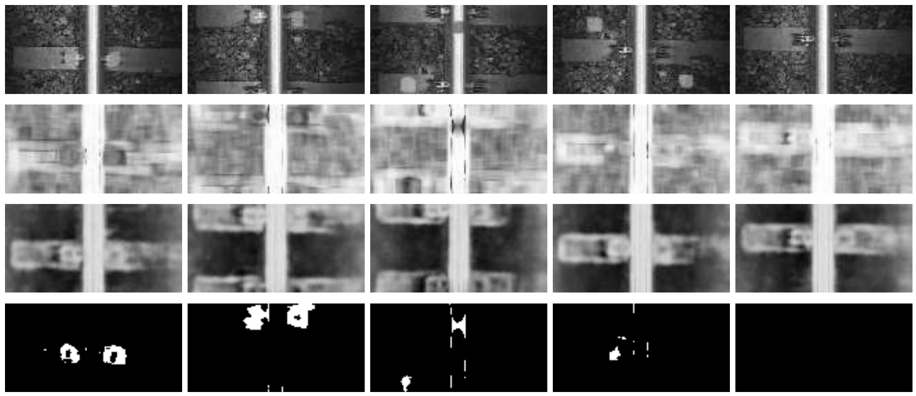


Figure 7: Anomaly detection on synthetic examples. From top to bottom: real image with synthetic anomaly, similarity $s(G_X(\mathbf{z}(X_{T_1}, \epsilon^*)), X_{T_2})$, confidence map and thresholded error map. The first three examples show anomalies on the clips, sleeper and rail, the fourth shows an anomaly on the ballast and the fifth no anomaly.

time steps corresponding to one heartbeat from a patient with congestive heart failure. We use $S_1 = 76$ time steps for the observed portion and $S_2 = 64$ time steps for the predicted portion. We supervise the confidence generator with L1 error, i.e. $s(x, y) = \text{abs}(x - y)$, hence our confidence generator is actually predicting error. This means that when we perform anomaly detection we can directly use the scaled "confidence" value as error: $e(x, y) = w \cdot s(x, y)$, where w is a scalar weight parameter.

We also use the MIT-BIH Arrhythmia Database [3] for quantitative evaluation. This is a widely used reference dataset for ECG signal analysis. This data set contains multichannel ECG recordings with detailed annotations for each heartbeat, identifying beat types and rhythm information. Each recording is sampled at 360 Hz and represents a complete ECG trace of a patient. We use timesteps $S_1 = S_2 = 64$ and again use L1 error to supervise the correspondence generator.

4.2. Qualitative analysis

We begin by providing qualitative analysis of the behaviour of our model on the railway dataset.

In Figure 5 we show the ability of the model to generate plausible and diverse samples. For each example, we take a single real $X^i_{T_1}$ and generate fake continuations $G_X(\mathbf{z}(X^i_{T_1}, \epsilon_1))$ and $G_X(\mathbf{z}(X^i_{T_1}, \epsilon_2))$ where ϵ_1 and ϵ_2 are two different random samples.

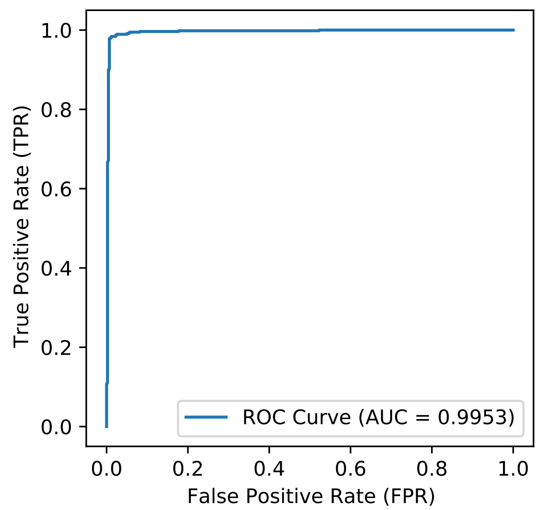
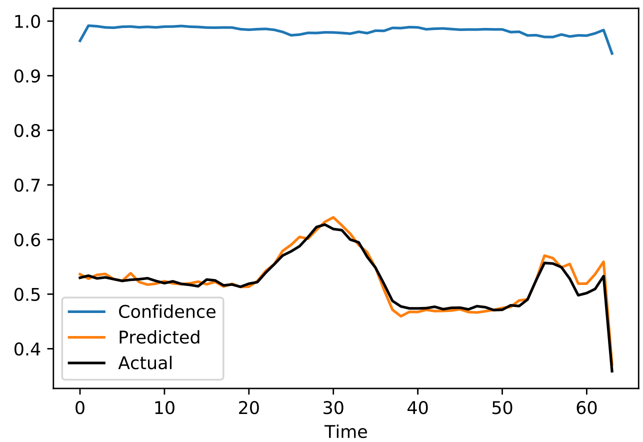


Figure 8: ROC curve for the ECG5000 dataset [1, 2].

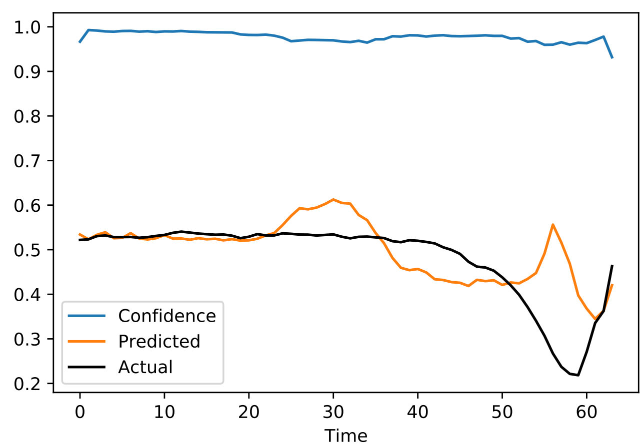
The generator is able to create images with the right structure (e.g. spacing between sleepers) and detail while using different random samples leads to slight changes in stochastic and structural elements. This illustrates that our model not only learns to decode the latent space to a plausible continuation but also that it learns a subspace of variation for the possible continuations.

In Figure 6 we illustrate fitting our model to observed data. Given a real observed X_{T_1}, we optimise $\boldsymbol{\epsilon}$ in order to minimise the error to the real observed X_{T_2} by solving the optimisation problem in (1). Note that this successfully adjusts structural elements of the fake image such that the main features align well. The similarity maps show which regions are reconstructed accurately (white means perfect local similarity). The predicted confidence map shows the model prediction of which regions in the image the model will be able to generalise to well. This includes the rail itself, the sleeper and certain elements of the clamp while it has low confidence for the stochastic background as expected. We emphasise that this separation of learnable structural uncertainty from unlearnable stochastic uncertainty is learnt without supervision. The structural elements are effectively 'detected' by the fact that they can be reliably modelled.

To qualitatively evaluate anomaly detection, we manually painted anomalies onto



(a)



(b)

Figure 9: ECG traces for a normal (a) and abnormal (b) heartbeat.

the rail, sleeper, clamp and background ballast. In Figure 7 we show qualitative examples of anomaly detection on these images. In the top row, our synthetic anomalies are visible as gray blobs. In the second row, the raw similarity between the reconstructed and observed images does show low similarity in the anomaly regions but also in the stochastic parts of the image. In the third row, the confidence map predicted by our model allows suppression of dissimilarity in regions of low confidence. The resulting

Source	[S]upervised/ [U]nsupervised	AUC	Acc	F1
Ours	U	**0.9953**	**0.9860**	**0.9875**
Pereira and Silveira [38]	S	0.9836	0.9843	0.9844
	U	0.9819	0.9596	0.9522
Lei et al. [59]	S	0.9100	-	-
Karim et al. [60]	S	-	0.9496	-
Malhotra et al. [61]	S	-	0.9340	-
Liu et al. [62]	U	-	-	0.8084

Table 1: Quantitative anomaly detection results on the ECG5000 dataset.

anomaly maps in the bottom row detect only badly reconstructed regions in areas of high confidence. This is crucial to limit false positives.

4.3. Quantitative evaluation

Although the ECG5000 dataset was originally used for five-way classification (normal plus four abnormalities), this dataset is now widely used for time series anomaly evaluation (normal versus any abnormality). We follow [38] and divide the dataset randomly into 80% training and 20% testing. We train the model using only the normal portion of the training set (i.e. excluding anomalies), comprising 2,359 traces in total. We then test on the whole of the test set which comprises 1,000 traces in total, 560 of which are normal and 440 of which are anomalous. Note that we operate in an unsupervised setting: we never see abnormal traces at training time. We show our ROC curve in Figure 8 and quantitative results in Table 1. Our approach outperforms all previous unsupervised methods and even outperforms the best supervised method on both area under curve and accuracy. In Figure 9 we show qualitative results for a normal (a) and anomalous (b) trace. We plot the true X_{T_2} (black), best-fit continuation $\hat{X}_{T_1} = G_X(\mathbf{z}(X_{T_1}, \boldsymbol{\epsilon}^*))$ (orange) and predicted error (i.e. one minus predicted confidence). The model can fit the normal trace well but cannot explain the anomalous trace. In other words, conditional on the first observed segment, the anomalous

Model	[S]upervised/ [U]nsupervised	AUC	F1 (%)	Accuracy (%)	Recall (%)	Precision (%)
Stacked LSTM [63]	U		81.0	-	87.0	82.0
LSTM with MLP [64]	S		87.0	95.0	75.0	-
VAE [65]	U		76.6	87.8	-	-
Transformer [66]	U	0.93	92.3	89.5	**98.2**	87.1
Our work	U	0.93	**93.2**	**90.1**	95.1	**91.4**

Table 2: Quantitative anomaly detection results on the MIT-BIH dataset.

second segment does not lie within the subspace forecast by our model. The predicted error is relatively flat though increases sharply towards the end where there is often a lot of variability in the training data. The fact that our model knows its prediction in this region is unreliable means differences here can be ignored - i.e. they cannot be confidently labelled as anomalies.

For the MIT-BIH dataset [3] we again divide into training and testing sets, where the training set consists only of normal beats, and the testing set included both normal and abnormal beats. We follow standard practices for the preprocessing and error evaluation for this dataset [66]. Segmentation of the continuous traces into training and testing samples was based on the annotated R-peak positions, with each heartbeat segment spanning from one R-peak to the next. To ensure consistent segment length, signals from both channels were resampled to a fixed length of 128 time steps per segment. Additionally, to reduce the impact of amplitude variations, the signal data from all channels were normalized using the 3rd and 97th percentiles as the range for scaling. Specifically, spectral error was employed as the core metric to measure the difference between predicted and actual signals. The spectral error was calculated by performing a Fast Fourier Transform (FFT) on the residuals of the first channel and applying our confidence-weighted scaling. As shown in Table 2, our method achieves good performance in terms of F1 score, recall, and precision, particularly in the unsupervised anomaly detection task. Compared to existing unsupervised methods and even some supervised approaches, our method set a new benchmark for the MIT-BIH dataset.

Dataset	Condition	F1 (%)	AUC (%)	Accuracy (%)
ECG5000	Baseline	98.3	99.56	98.1
	No confidence map	97.9	99.56	97.7
	No optimization of $\boldsymbol{\epsilon}$	98.3	99.4	98.1
	No diversity loss	97.6	99.5	97.3
MIT-BIH	Baseline	93.2	92.6	90.1
	No confidence map	93.2	92.6	90.1
	No optimization of $\boldsymbol{\epsilon}$	89.9	86.9	84.8
	No diversity loss	89.6	84.3	86.1

Table 3: Ablation experiments Results on ECG5000 and MIT-BIH Datasets.

4.4. Ablation study

Finally, we conducted an ablation study of the three key ingredients of our method using the ECG5000 and MIT-BIH datasets. Specifically, we evaluate the impact of: confidence map, model fitting (i.e. optimization of $\boldsymbol{\epsilon}$), and diversity loss. The results, summarized in Table 3, highlight the varying impact of these components on model performance. In the ECG5000 data set, incorporating the confidence map led to notable improvements in the F1 score and accuracy, while the AUC remained consistent. However, on the MIT-BIH dataset, the inclusion of the confidence map did not show any measurable effect, as all metrics remained identical with or without it. This suggests that the utility of the confidence map may vary depending on the dataset characteristics or noise levels. The optimization of the latent variable $\boldsymbol{\epsilon}$ exhibited a more consistent influence. On ECG5000, it slightly enhanced AUC while maintaining F1 and accuracy scores. On MIT-BIH, the impact was more pronounced, with F1 score, AUC and accuracy all improving substantially. These results indicate that $\boldsymbol{\epsilon}$-optimization significantly contributes to the model's ability to generalize, particularly on datasets with diverse and complex patterns like MIT-BIH. The diversity loss consistently improved model performance on both datasets. This demonstrates the robustness of diversity loss in improving the detection of anomalous samples and reducing overfitting across

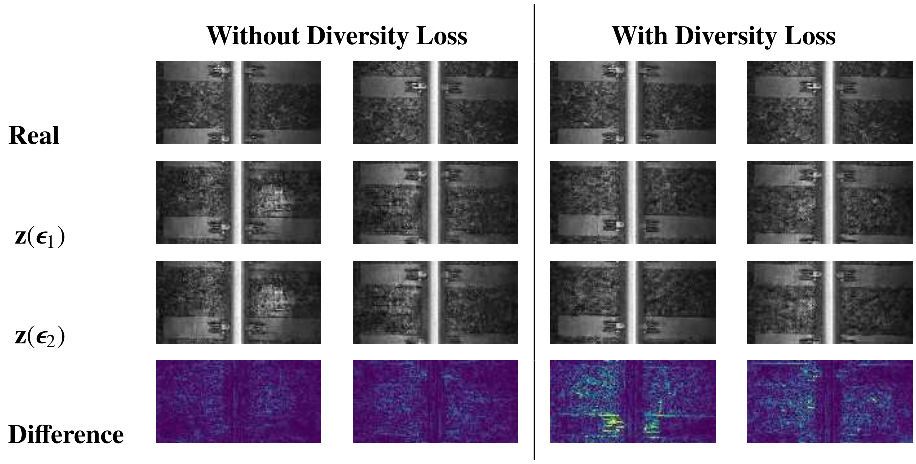


Figure 10: Impact of diversity loss: In the first row we show an observed real X_{T_1}. In the second and third rows we show two possible continuations in which we use different random noise samples $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$. In the fourth row we show a heat map visualisation of the absolute difference between the second and third rows (same colour map scale used for all four images). In the first two columns the results are an ablation in which diversity loss is not used during training. In the last two columns diversity loss is used during training.

datasets.

In summary, while the impact of the confidence map appears dataset-dependent, the optimization of $\boldsymbol{\epsilon}$ and the diversity loss consistently enhance model performance. $\boldsymbol{\epsilon}$-optimization is particularly effective in improving generalization on complex datasets, and the diversity loss contributes significantly to anomaly detection and classification robustness. These findings validate the necessity of these components in achieving state-of-the-art results on anomaly detection tasks.

Finally, in Figure 10 we show a qualitative illustration of the impact of the diversity loss on the railway image dataset. Without diversity loss we can see that the model has learnt limited dependence on $\boldsymbol{\epsilon}$ (the two generated images are very similar and the difference map shows little change anywhere). With the diversity loss, more variation for different $\boldsymbol{\epsilon}$ is evident and this is visible in the difference map. This shows both structural variation (around the clamps) and stochastic variation in the ballast. Interestingly, we also observe that diversity loss improves the model more generally. Without diversity loss the generations exhibit artefacts which are not present with it. This is because diversity loss avoids overfitting to the particular continuation observed in the training data and encourages learning a smooth subspace of plausible continuations.

5. Conclusions

We have shown that we can learn a generative model for stochastic continuation of non-overlapping temporal sequences. Our unsupervised method automatically learns which parts of the continuation are predictable and which are stochastic. This provides a route to unsupervised anomaly detection.

From a practical perspective, deploying our approach in real-world settings for a new data domain entails only the following steps. First, choose an appropriate architecture for the encoder and data/confidence generators. Any off-the-shelf architecture that is widely used for the data modality could be used here. Second, choose an appropriate similarity or dissimilarity measure to supervise the confidence generator. Again, any standard metric such as SSIM for images or MSE for time series signals could be used. Finally, adjust the key hyperparameters of latent space dimension and segment sizes (often $S_1 = S_2$ will prove the best choice, giving an equal balance between the size of input data and model prediction). In terms of computational cost, in the simplest case our method only requires a forward pass through the encoder and generator networks and evaluation of the anomaly map metric. For slightly improved performance, optional iterative optimisation of $\boldsymbol{e}psilon$ requires a fixed number of gradient descent steps.

There are a number of limitations to our approach. First, from a practical implementation perspective, our use of a convolutional encoder and decoder potentially limits its modelling of long-range dependencies. This is not a limitation of the method itself, but rather the chosen architecture. This could be resolved by using a transformer so that relationships between signal or image patches at distant spatio/temporal locations could be captured. Nevertheless, the fact that our implementation is still competitive with transformer-based architectures (see Table 2) shows the benefit of our method regardless of architectural choices. Second, finding the optimal latent parameter via optimisation requires in-network iterative optimisation which is more expensive than a simple forward pass. While it is possible to use only the encoded mean latent variable without optimisation of the additional noise parameter (results in third rows of ablation study in Table 3) this does reduce performance. It amounts to assuming that the continuation is the most likely without considering the contents of the actual continuation. In the context of anomaly detection, this is not optimal. Finally, while our model learns the distribution of temporal continuation, its application to anomaly detection requires selection of a similarity function in order to distinguish normal from abnormal. It is likely that performance would be improved if this function could itself be learnt. However, this would require supervision in the form of example anomalies which is not always available depending on the problem.

In future, we would like to explore using the trained encoder as a pre-trained backbone for downstream tasks. The encoder has learned to embed sufficient information about a given time series segment to predict the following segment. We believe that this means it would likely perform well when fine tuned for other tasks such as classification or object detection. Secondly, we would also like to explore the use of other architectures such as transformers which naturally handle sequential data and so may perform well for time series data. Finally, we would like to test whether our method generalises to other temporal data modalities such as video and audio.

References

[1] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, E. Keogh, The ucr time series archive, IEEE/CAA Journal of Automatica Sinica 6 (6) (2019) 1293–1305.

[2] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, H. E. Stanley, Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals, circulation 101 (23) (2000) e215–e220.

[3] G. B. Moody, R. G. Mark, The impact of the mit-bih arrhythmia database, IEEE engineering in medicine and biology magazine 20 (3) (2001) 45–50.

[4] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, O. Winther, Autoencoding beyond pixels using a learned similarity metric, in: International conference on machine learning, PMLR, 2016, pp. 1558–1566.

[5] L. Jing, Y. Tian, Self-supervised visual feature learning with deep neural networks: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (11) (2021) 4037–4058. doi:10.1109/TPAMI.2020.2992393.

[6] C. Doersch, A. Gupta, A. A. Efros, Unsupervised visual representation learning by context prediction, in: Proc. ICCV, 2015, pp. 1422–1430.

[7] R. Zhang, P. Isola, A. A. Efros, Colorful image colorization, in: Proc. ECCV, 2016, pp. 649–666.

[8] S. Gidaris, P. Singh, N. Komodakis, Unsupervised representation learning by predicting image rotations, in: Proc. ICLR, 2018.

[9] I. Misra, C. L. Zitnick, M. Hebert, Shuffle and learn: unsupervised learning using temporal order verification, in: Proc. ECCV, 2016, pp. 527–544.

[10] D. Wei, J. J. Lim, A. Zisserman, W. T. Freeman, Learning and using the arrow of time, in: Proc. CVPR, 2018, pp. 8052–8060.

[11] X. Wang, A. Jabri, A. A. Efros, Learning correspondence from the cycle-consistency of time, in: Proc. CVPR, 2019.

[12] S. Hitawala, Comparative study on generative adversarial networks, arXiv preprint arXiv:1801.04271 (2018).

[13] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).

[14] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[15] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 16000–16009.

[16] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, S. Levine, Stochastic adversarial video prediction, arXiv preprint arXiv:1804.01523 (2018).

[17] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, N. Ballas, Self-supervised learning from images with a joint-embedding predictive architecture, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15619–15629.

[18] H. Fan, F. Zhang, Y. Gao, Self-supervised time series representation learning by inter-intra relational reasoning, arXiv preprint arXiv:2011.13548 (2020).

[19] S. Liu, A. Mallol-Ragolta, E. Parada-Cabeleiro, K. Qian, X. Jing, A. Kathan, B. Hu, B. W. Schuller, Audio self-supervised learning: A survey, arXiv preprint arXiv:2203.01205 (2022).

[20] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, A. Krishnaswamy, Self-supervised classification for detecting anomalous sounds., in: DCASE, 2020, pp. 46–50.

[21] X. Chang, T. Maekaku, P. Guo, J. Shi, Y.-J. Lu, A. S. Subramanian, T. Wang, S.-w. Yang, Y. Tsao, H.-y. Lee, et al., An exploration of self-supervised pretrained representations for end-to-end speech recognition, in: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE, 2021, pp. 228–235.

[22] Z. Chen, Y. Zhang, A. Rosenberg, B. Ramabhadran, G. Wang, P. Moreno, Injecting text in self-supervised speech pretraining, in: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE, 2021, pp. 251–258.

[23] L. C. Pickup, Z. Pan, D. Wei, Y. Shih, C. Zhang, A. Zisserman, B. Scholkopf, W. T. Freeman, Seeing the arrow of time, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2035–2042.

[24] D. Kim, D. Cho, I. S. Kweon, Self-supervised video representation learning with space-time cubic puzzles, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 33, 2019, pp. 8545–8552.

[25] S. Saha, F. Bovolo, L. Bruzzone, Change detection in image time-series using unsupervised lstm, IEEE Geoscience and Remote Sensing Letters (2020).

[26] L. Tao, X. Wang, T. Yamasaki, Self supervised video representation using pretext-contrastive learning, arXiv preprint arXiv:2010.15464 2 (2020) 2.

[27] J. Wang, Y. Lin, A. J. Ma, P. C. Yuen, Self-supervised temporal discriminative learning for video representation learning, arXiv preprint arXiv:2008.02129 (2020).

[28] J. Kaur, S. Das, Future frame prediction of a video sequence, arXiv preprint arXiv:2009.01689 (2020).

[29] Y. Wang, L. Jiang, M.-H. Yang, L.-J. Li, M. Long, L. Fei-Fei, Eidetic 3d lstm: A model for video prediction and beyond, in: International conference on learning representations, 2018.

[30] S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J. A. Castro-Vargas, S. Orts-Escolano, J. Garcia-Rodriguez, A. Argyros, A review on deep learning techniques for video prediction, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).

[31] L. Jiang, M. Xu, Z. Wang, Predicting video saliency with object-to-motion cnn and two-layer convolutional lstm, arXiv preprint arXiv:1709.06316 (2017).

[32] W. Lotter, G. Kreiman, D. Cox, Deep predictive coding networks for video prediction and unsupervised learning, arXiv preprint arXiv:1605.08104 (2016).

[33] C. Finn, I. Goodfellow, S. Levine, Unsupervised learning for physical interaction through video prediction, Advances in neural information processing systems 29 (2016).

[34] R. Villegas, J. Yang, S. Hong, X. Lin, H. Lee, Decomposing motion and content for natural video sequence prediction, arXiv preprint arXiv:1706.08033 (2017).

[35] M. Giannoulis, A. Harris, V. Barra, Ditan: A deep-learning domain agnostic framework for detection and interpretation of temporally-based multivariate anomalies, Pattern Recognition 143 (2023) 109814.

[36] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M. A. Zuluaga, Do deep neural networks contribute to multivariate time series anomaly detection?, Pattern Recognition 132 (2022) 108945.

[37] T. Mokoena, T. Celik, V. Marivate, Why is this an anomaly? explaining anomalies using sequential explanations, Pattern Recognition 121 (2022) 108227.

[38] J. Pereira, M. Silveira, Learning representations from healthcare time series data for unsupervised anomaly detection, in: 2019 IEEE international conference on big data and smart computing (BigComp), IEEE, 2019, pp. 1–7.

[39] J. Yang, Y. Shi, Z. Qi, Learning deep feature correspondence for unsupervised anomaly detection and segmentation, Pattern Recognition 132 (2022) 108874.

[40] X. Zhang, J. Mu, X. Zhang, H. Liu, L. Zong, Y. Li, Deep anomaly detection with self-supervised learning and adversarial training, Pattern Recognition 121 (2022) 108234.

[41] V. Zavrtanik, M. Kristan, D. Skočaj, Reconstruction by inpainting for visual anomaly detection, Pattern Recognition 112 (2021) 107706.

[42] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, L. Liu, Feature encoding with autoencoders for weakly supervised anomaly detection, IEEE Transactions on Neural Networks and Learning Systems 33 (6) (2021) 2454–2465.

[43] S. Akcay, A. Atapour-Abarghouei, T. P. Breckon, Ganomaly: Semi-supervised anomaly detection via adversarial training, in: Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14, Springer, 2019, pp. 622–637.

[44] B. Zhou, S. Liu, B. Hooi, X. Cheng, J. Ye, Beatgan: Anomalous rhythm detection using adversarially generated time series., in: IJCAI, Vol. 2019, 2019, pp. 4433–4439.

[45] T.-W. Tang, H. Hsu, W.-R. Huang, K.-M. Li, Industrial anomaly detection with skip autoencoder and deep feature extractor, Sensors 22 (23) (2022) 9327.

[46] J. Liu, Y. Huang, S. Wang, X. Zhao, Q. Zou, X. Zhang, Rail fastener defect inspection method for multi railways based on machine vision, Railway Sciences 1 (2) (2022) 210–223.

[47] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. Aggarwal, M. Salehi, Deep learning for time series anomaly detection: A survey, ACM Computing Surveys 57 (1) (2024) 1–42.

[48] Z. Wang, C. Pei, M. Ma, X. Wang, Z. Li, D. Pei, S. Rajmohan, D. Zhang, Q. Lin, H. Zhang, et al., Revisiting VAE for unsupervised time series anomaly detection: A frequency perspective, in: Proceedings of the ACM Web Conference 2024, 2024, pp. 3096–3105.

[49] H. Kang, P. Kang, Transformer-based multivariate time series anomaly detection using inter-variable attention mechanism, Knowledge-Based Systems 290 (2024) 111507.

[50] J. Miao, H. Tao, H. Xie, J. Sun, J. Cao, Reconstruction-based anomaly detection for multivariate time series using contrastive generative adversarial networks, Information Processing & Management 61 (1) (2024) 103569.

[51] Z. Z. Darban, G. I. Webb, S. Pan, C. C. Aggarwal, M. Salehi, CARLA: Self-supervised contrastive representation learning for time series anomaly detection, Pattern Recognition 157 (2025) 110874.

[52] D. Kim, S. Park, J. Choo, When model meets new normals: test-time adaptation for unsupervised time-series anomaly detection, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 38, 2024, pp. 13113–13121.

[53] Y. Zhou, X. Xu, J. Song, F. Shen, H. T. Shen, Msflow: Multiscale flow-based framework for unsupervised anomaly detection, IEEE Transactions on Neural Networks and Learning Systems (2024).

[54] H. Yao, M. Liu, Z. Yin, Z. Yan, X. Hong, W. Zuo, GLAD: towards better reconstruction with global and local adaptive diffusion models for unsupervised anomaly detection, in: European Conference on Computer Vision, Springer, 2024, pp. 1–17.

[55] S. Dai, Y. Wu, X. Li, X. Xue, Generating and reweighting dense contrastive patterns for unsupervised anomaly detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 1454–1462.

[56] J. Liu, K. Wu, Q. Nie, Y. Chen, B.-B. Gao, Y. Liu, J. Wang, C. Wang, F. Zheng, Unsupervised continual anomaly detection with contrastively-learned prompt, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 38, 2024, pp. 3639–3647.

[57] H. Liu, Z. Wan, W. Huang, Y. Song, X. Han, J. Liao, PD-GAN: Probabilistic diverse gan for image inpainting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9371–9381.

[58] D. Brunet, E. R. Vrscay, Z. Wang, On the mathematical properties of the structural similarity index, IEEE Transactions on Image Processing 21 (4) (2011) 1488–1499.

[59] Q. Lei, J. Yi, R. Vaculin, L. Wu, I. S. Dhillon, Similarity preserving representation learning for time series clustering, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 2845–2851.

[60] F. Karim, S. Majumdar, H. Darabi, S. Chen, Lstm fully convolutional networks for time series classification, IEEE access 6 (2017) 1662–1669.

[61] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, Lstm-based encoder-decoder for multi-sensor anomaly detection, arXiv preprint arXiv:1607.00148 (2016).

[62] Y. Liu, J. Chen, S. Wu, Z. Liu, H. Chao, Incremental fuzzy c medoids clustering of time series data using dynamic time warping distance, Plos one 13 (5) (2018) e0197499.

[63] M. Thill, S. Däubener, W. Konen, T. Bäck, P. Barancikova, M. Holena, T. Horvat, M. Pleva, R. Rosa, Anomaly detection in electrocardiogram readings with stacked lstm networks., in: ITAT, 2019, pp. 17–25.

[64] G. Sivapalan, K. K. Nundy, S. Dev, B. Cardiff, D. John, Annet: A lightweight neural network for ECG anomaly detection in iot edge sensors, IEEE Transactions on Biomedical Circuits and Systems 16 (1) (2022) 24–35.

[65] P. Matias, D. Folgado, H. Gamboa, A. V. Carreiro, Robust anomaly detection in time series through variational autoencoders and a local similarity score., in: Biosignals, 2021, pp. 91–102.

[66] A. Alamr, A. Artoli, Unsupervised transformer-based anomaly detection in ECG signals, Algorithms 16 (3) (2023) 152.