Monte Carlo EM for Deep Time Series Anomaly Detection

François-Xavier Aubet [1] **Daniel Zügner** [2] **Jan Gasthaus** [1]

Abstract

Time series data are often corrupted by outliers or other kinds of anomalies. Identifying the anomalous points can be a goal on its own (anomaly detection), or a means to improving performance of other time series tasks (e.g. forecasting). Recent deep-learning-based approaches to anomaly detection and forecasting commonly assume that the proportion of anomalies in the training data is small enough to ignore, and treat the unlabeled data as coming from the nominal data distribution. We present a simple yet effective technique for augmenting existing time series models so that they explicitly account for anomalies in the training data. By augmenting the training data with a latent anomaly indicator variable whose distribution is inferred while training the underlying model using Monte Carlo EM, our method simultaneously infers anomalous points while improving model performance on nominal data. We demonstrate the effectiveness of the approach by combining it with a simple feed-forward forecasting model. We investigate how anomalies in the train set affect the training of forecasting models, which are commonly used for time series anomaly detection, and show that our method improves the training of the model.

1. Introduction

In many time series anomaly detection applications one only has access to unlabeled data. This data is usually mostly nominal but may contain some (unlabeled) anomalies. Examples of this setting are e.g. the widely used anomaly detection benchmarks SMAP, MSL (Hundman et al., 2018), and SMD (Su et al., 2019).

This "true" unsupervised setting with *mixed* data can be contrasted with the "nominal-only" setting, where one assumes access to "clean" nominal data. In practice, techniques that (explicitly or implicitly) assume access to nominal data can often also successfully be applied to mixed data by assuming it is nominal, as long as the proportion of anomalies is sufficiently small., they are however biased by training on some anomalous data.

While some time series anomaly detection model rely on the one class classification paradigm which does not suffer from this assumption (Shen et al., 2020; Carmona et al., 2021), the vast majority of the current time series anomaly detection methods are either forecasting methods (Shipmon et al., 2017; Zhao et al., 2020) or reconstruction methods (Su et al., 2019; Xu et al., 2018; Park et al., 2018; Zhang et al., 2019). Forecasting methods detect anomalies as deviations of observations from predictions, while reconstruction methods declare observations that deviate from the reconstruction as anomalous. In both cases, a probabilistic model of the observed data is assumed and its parameters are learned. However, by training the model on the observed data which contains both normal and anomalous data points, the model ultimately learns the wrong data distribution. We propose to address this issue using a simple technique based on latent indicator variables that can readily be combined with existing probabilistic anomaly detection appoaches. By using latent indicator variables to explicitly infer which observations in the training set are anomalous, we can subsequently suitably account for the anomalous observations while training the probabilistic model.

Probabilistic models that use latent (unobserved) indicator variables to explicitly distinguish between nominal and anomalous data points are well-established in the context of robust mixture models (e.g. Fraley & Raftery, 1998) and classical time series models (e.g. Wang et al., 2018). However, these techniques have not yet been utilized in the context of recent advances in *deep* anomaly detection and time series modeling, presumably due to the (perceived) increased complexity of the required probabilstic inference and training procedure. We show that combining latent anomaly indicators with a Monte Carlo Expectation-Maximization (EM) (Wei & Tanner, 1990) training procedure, results in a simple yet effective technique that can be combined with (almost) all existing deep anomaly detection and time series forecasting techniques.

We demonstrate the effectiveness of our approach with a

[1] AWS AI Labs [2]Technical University of Munich. Correspondence to: François-Xavier Aubet <aubetf@amazon.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

simple model for anomaly detection on the Yahoo anomaly detection dataset and on the electricity dataset for forecasting from a noisy training set.

2. Background

For non-time series data, one common approach of formalizing the notion of anomalies is to assume that the observed data is generated by a mixture model (Ruff et al., 2020): each observation $\mathbf{x}$ is drawn from the mixture distibution $p(\mathbf{x}) = \alpha p^+(\mathbf{x}) + (1-\alpha)p^-(\mathbf{x})$, where $p^+(\mathbf{x})$ is the distribution of the nominal data and $p^-(\mathbf{x})$ the anomalous data distribution. Typically one assumes a flexible parametrized distribution for p^+ and a broad, unspecific distribution for p^- (e.g. a uniform distribution over the extent of the data).

This mixture distribution can equivalently be written using a binary *indicator latent variable* z taking value 0 with probability $p(z=0) = \alpha$ and value 1 with probability $p(z=1) = 1-\alpha$, and specifying the conditional distribution

$$p(\mathbf{x}|z) = \begin{cases} p^+(\mathbf{x}) & \text{if } z = 0 \\ p^-(\mathbf{x}) & \text{if } z = 1, \end{cases} \tag{1}$$

so that $p(\mathbf{x}) = \sum_z p(\mathbf{x}|z)p(z) = \alpha p^+(\mathbf{x}) + (1-\alpha)p^-(\mathbf{x})$. In this setup, anomaly detection can be performed by inferring the posterior distribution $p(z|\mathbf{x})$ (and thresholding it if a hard choice is desired). Yet another way of representing the same model is generatively: first, draw $\mathbf{y}^+ \sim p^+(\cdot)$, $\mathbf{y}^- \sim p^-(\cdot)$, and $z \sim \text{Bernoulli}(1-\alpha)$, and then set $\mathbf{x} = \mathbf{I}[z=0]\,\mathbf{y}^+ + \mathbf{I}[z=1]\,\mathbf{y}^-$, i.e. the observation $\mathbf{x}$ is equal to $\mathbf{y}^+$ if it is nominal ($z=1$) and equal to $\mathbf{y}^-$ otherwise. Introducing the additional latent variables $\mathbf{y}^+$ and $\mathbf{y}^-$ is unnecessary in the IID setting, but becomes useful in the time series setting described next.

In time series setting, where the the observations are time series $\mathbf{x}_{1:T} = \mathbf{x}_1, \ldots, \mathbf{x}_T$ that exhibit temporal dependencies, and anomalies are time points or regions within these time series, we have one anomaly indicator variable z_t corresponding to each time point $\mathbf{x}_t$. Like before, the nominal data is drawn from a parametrized probabilistic model $p^+_\theta(\mathbf{y}_{1:T})$, and the anomalies are generated from a fixed model $p^-(\mathbf{y}_{1:T})$. For time series data, the mixture data model then amounts to drawing $\mathbf{y}^+_{1:T} \sim p^+(\cdot)$, $\mathbf{y}^-_{1:T} \sim p^-(\cdot)$, and $z_{1:T} \sim p^z(z_{1:T})$, and seting $\mathbf{x}_t = \mathbf{I}[z_t = 0]\,\mathbf{y}^+_t + \mathbf{I}[z_t = 1]\,\mathbf{y}^-_t$.

3. Method

Forecasting or reconstruction models are designed to learn a model of $p^+(\cdot)$ but are typically trained directly on the observed time series $\mathbf{x}_{1:T}$. We propose to learn the model of $p^+(\cdot)$ only from $\mathbf{y}^+_{1:T}$ by inferring $z_{1:T} \sim p^z(z_{1:T})$ on the training set. This way we can train the model only on the observed points that are normal, the ones that are equal to $\mathbf{y}^+_{1:T}$. Depending on the model, the anomalous points can be treated as missing or the normal point can be inferred.

3.1. Models

Each of the three latent time series is modeled with a probabilistic model: a parametrized model p^+_θ of the nominal data $\mathbf{y}^+_{1:T}$, a fixed model p^- to model the anomalous data $\mathbf{y}^-_{1:T}$, and a model p^z of the indicator time series $z_{1:T}$.

Nominal Data Model Many existing deep anomaly detection methods aim to model the nominal data (e.g. (Shipmon et al., 2017; Zhao et al., 2020; Su et al., 2019; Xu et al., 2018; Park et al., 2018; Zhang et al., 2019)), and any of them can be used to model $\mathbf{y}^+$, the latent nominal time series. Our method is agnostic to the type of model used, so that it can be combined with any probabilistic time series model, be it a deep or shallow probabilistic forecasting method, a reconstruction method, or any other type of model. We call the model of the latent normal time series p^+_θ, which is parametrised by a set of parameters θ.

In our experiments we demonstrate the general setup by modeling $p^+(\mathbf{y}^+_{1:T})$ with a simple deep probabilistic forecasting model. We decompose $p(\mathbf{y}^+_{1:T})$ into the telescoping product $p(\mathbf{y}^+_0)\prod_{t=0}^{T} p(\mathbf{y}^+_{t+1}|\mathbf{y}^+_{t:0})$ and, making an l-th order Markov assumption, approximate it with a network $p(\mathbf{y}^+_{t+1}|\mathbf{y}^+_{t:t-l}) = \mathcal{N}(f_\theta(\boldsymbol{y}_{t:t-l}), g_\theta(\boldsymbol{y}_{t:t-l}))$ taking as input the last l time points.

Anomalous Data Model A simple model can be used to model p^-, it does not need to take into account the time component as there are typically few anomalous points. It can be modeled with a mixture of Gaussians for example, with the risk of overfitting to the few anomalies of the train set. We simply model p^- with a uniform distribution over the domain of the training data, not assuming any prior on the kind of anomalies that we may expect.

Anomaly Indicator Model We model the latent anomaly indicator with a Hidden Markov Model (HMM) with two states, state $z_t = 0$ corresponds to the point being normal and state $z_t = 1$ corresponds to the point being anomalous. Any kind of time series model parameterizing a Bernoulli distribution can be used to model the latent anomaly indicators, we pick an HMM as it encodes basic time dependencies while staying a simple model.

If it is available, prior knowledge about the dataset can be used to initialise the transition matrix. The expected length of anomalous windows can be used to initialise the transition probability $p(z_{t+1} = 1|z_t = 1)$. The expected percentage of anomalous points in the dataset can be used to initialise the transition probability $p(z_{t+1} = 1|z_t = 0)$.

3.2. Training

Our training procedure follows Monte Carlo EM (Wei & Tanner, 1990). In the E-step we infer $p^z(z_{1:T})$. In the M-step we sample from $p^z(z_{1:T})$, using these samples to update p_θ^+ and the transition matrix of the HMM. Algorithm 1 sketches this procedure.

Algorithm 1 Monte Carlo EM for Latent Anomaly Indicator

Input: Observed time series $\mathbf{x}_{1:T}$, model to be trained p_θ^+
for $e \in \{1, \dots, \textit{numb\_epochs}\}$ **do**
  `// E-step:`
  $\rightarrow$ infer $p^z(z_{1:T})$
  `// M-step:`
  for $s \in \{1, \dots, \textit{numb\_samples}\}$ **do**
    $\rightarrow$ sample indicator time series z_s from $p^z(z_{1:T})$
    $\rightarrow$ perform one epoch of p_θ^+ on $\mathbf{x}_{\neg z_s}$ where the points at sampled anomalous indices are replaced
  end
  $\rightarrow$ update the transition matrix of the HMM
end

3.2.1. E-STEP

We infer $p^z(z_{1:T})$ by using the standard forward-backward algorithm for HMMs, using the following distributions:

$$p(\mathbf{x}_t|z_t = 0) = p_\theta^+(\mathbf{x}_t) \tag{2}$$
$$p(\mathbf{x}_t|z_t = 1) = p^-(\mathbf{x}_t) \tag{3}$$

and $p(z_{t+1}|z_t)$ is given by the HMM transition matrix.

3.2.2. M-STEP

We want to train $p^+(\cdot)$ only from $\mathbf{y}_{1:T}^+$. As most models may not allow for an analytical update using $\boldsymbol{x}_{1:T}$ and $z_{1:T}$, we propose to a Monte Carlo approximation of the expectation under $p^z(z_{1:T})$. We draw multiple samples from $p^z(z_{1:T})$ giving us possible normal points on which p_θ^+ can be trained. Each path sampled gives us a set of observed points that can be considered as coming from the normal data distribution p^+. We maximise the probability of these points under p_θ^+, treating the points coming from p^- points as missing.

Depending on the choice of model for p_θ^+, one may not be able to simply ignore anomalous points and they would have to be imputed. For deep forecasting or reconstruction models for example the model has to be given an input for each time point. In these cases, we propose to impute the point with the forecast or reconstruction obtained from p_θ^+ at the last M-step. This way, we use p_θ^+ to infer the time points of $\mathbf{y}_{1:T}^+$ that were not observed. With this method we can recover the full $\mathbf{y}_{1:T}^+$ time series and train p_θ^+ on it.

Depending on the choice of model for p^-, one can update it using the points that are sampled as coming from $\mathbf{y}_{1:T}^-$.

We can update the transition matrix of the HMM like in the classical M-step. The average number of transitions from one state to the next in the samples from $p^z(z_{1:T})$ become the new transition probabilities.

3.3. Inference

At inference time, we propose to use the HMM to perform filtering on z and infer if incoming points are more likely to be drawn from p^+ or p^-. If an incoming point $\mathbf{x}_t$ is more likely to be coming from p^- it can be treated as missing or replaced with a sample from $p_{\theta t}^+$ or by its mode. This way we ensure that the trained model is only used on points coming from $\mathbf{y}_{1:T}^+$.

4. Experiments

Model We evaluate our approach with a simple forecasting model on both anomaly detection and forecasting tasks. We show the performance of the model when trained in a standard way and when trained with our procedure, which we call our procedure Latent Anomaly Indicator (LAI). We use a simple Multi-Layer Perceptron (MLP) model to parametrise the mean and the variance of a predictive Gaussian distribution. It takes as input the last 25 points.

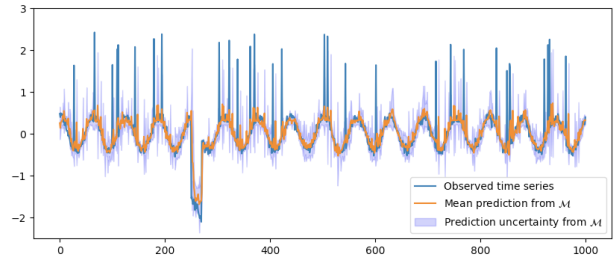
Datasets For the anomaly detection evaluation, we use the **Yahoo** dataset, published by Yahoo labs.[1] It consists of 367 real and synthetic time series, divided into four subsets (A1-A4) with varying level of difficulty. The length of the series vary from 700 to 1700 observations. Labels are available for all the series. We use the last 50% of the time points of each of the time series as test set, like (Ren et al., 2019) did, and split the rest in 40% training and 10% validation set. We evaluate the performance of the model using the adjusted F1 score proposed by Xu et al. (2018) and subsequently used in other work.

In addition, we evaluate the method on forecasting tasks using the commonly used **electricity** dataset (Dheeru & Taniskidou, 2017), composed of 370 time series of 133k points each. Given the length of the dataset, we sub-sample it by a factor 10. We select the last 50% of the points of each time series for testing. We scale each time series using the median and inter-quartile range on the train set.

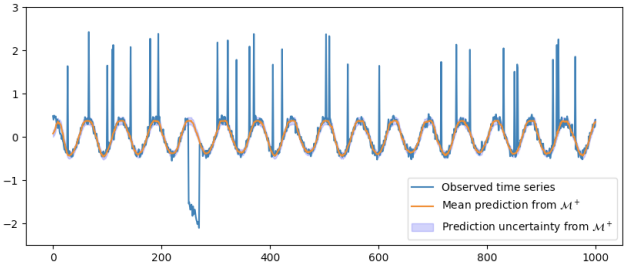
4.1. Visualization on synthetic data

Figure 1 visualises the advantage of the method on a simple sinusoidal time series with the simple MLP for p_θ^+. We generate a synthetic time series and inject outliers in it. We observe that our approach allows to train the model p_θ^+ while ignoring the outliers in the data, whereas the outliers heavily

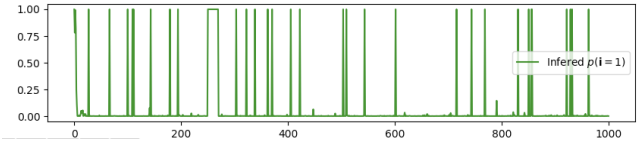
[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70



(a) The fit of a simple MLP without LAI.



(b) The fit of a simple MLP with LAI.



(c) The time series of latent anomaly indicator $p(z_t = 1)$.

Figure 1. We fit a MLP on this simple synthetic time series with anomalies. (a) shows the fit of the model trained in a conventional way, (b) shows the fit of the model trained as we propose to, (c) show the inferred $p(z_{1:T})$ distribution at the end of the training.

influence the model trained conventionally. We observe from figure 1c that the model is able to infer accurately which of the training points are likely to be anomalous.

4.2. Time series anomaly detection

Table 1 shows the F1 score of the model with and without LAI on the different subsets of the Yahoo dataset. We train one MLP on each of the time series and average the F1 scores obtained on the different time series of the subset. We observe that using our approach greatly improves the performance of the model.

Table 1. F1 score on the different subsets of the Yahoo dataset.

Model	A1	A2	A3	A4
MLP	33.64	53.28	63.25	47.30
MLP + LAI	41.84	87.26	87.91	61.62

In addition to the improved F1 score, we compare the inferred anomalous points on the training set with the actual labeled anomalous points. Table 2 shows the F1 score on the training set when using the anomaly indicator as anomaly score. We observe that our method allows to find accurately the anomalies present in the training set. While the training and test sets are different, we propose that the higher F1 on the train set is due to the fact that the model can use the whole training set to infer if a point is anomalous, and not only the past points.

Table 2. F1 score on the training set the different subsets of the Yahoo dataset using the inferred $p(z_t = 1)$ as anomaly score.

Model	A1	A2	A3	A4
MLP + LAI	59.48	94.02	81.89	73.77

4.3. Forecasting using a corrupted train set

Our method can be used more generally to train a forecasting model on a forecasting dataset containing anomalies. We take the electricity forecasting dataset and inject point outliers in the training set so that about 0.4% of the training point have an added or subtracted spike. Table 3 shows the mean absolute error (MAE) on the test set in the setting where the original train set is used and in the setting where the noisy train set is used. We see that using our method allows to reduce significantly the increase in error from the outliers in the training set, only 0.0146 increase in the mean absolute error versus 0.0542 when training the model normally.

Table 3. MAE on electricity with and without injecting point outliers in the train set

Model	electricity	electricity + outliers
MLP	0.1551	0.2092
MLP + LAI	0.1558	0.1704

5. Conclusion

We present LAI, a method that canbe used to wrap any probabilistic time series model to perform anomaly detection without being impacted by unlabeled anomalies in the training set. We present the details of the approach and propose preliminary empirical results on commonly used public benchmark datasets. The approach seems to greatly help both for anomaly detection tasks and for training a forecasting model on a contaminated training set.

We aim to extend this work by wrapping other bigger models such as OmniAnomaly (Su et al., 2019) or state-of-the-art forecasting models. Finally, with our current method at inference time, one has to decide at each incoming point if it is to be replaced or not, one could use particles which would mimic the Monte Carlo approach of the training time.

References

Carmona, C., Aubet, F.-X., Flunkert, V., and Gasthaus, J. Neural contextual anomaly detection for time series. 2021.

Dheeru, D. and Taniskidou, E. K. electricity: hourly time series of the electricity consumption of 370 customers, 2017. URL http://archive.ics.uci.edu/ml.

Fraley, C. and Raftery, A. E. How many clusters? Which clustering method? Answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018.

Park, D., Hoshi, Y., and Kemp, C. C. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

Ren, H., Xu, B., Wang, Y., Yi, C., Huang, C., Kou, X., Xing, T., Yang, M., Tong, J., and Zhang, Q. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3009–3017, 2019.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. A unifying review of deep and shallow anomaly detection. *arXiv preprint arXiv:2009.11732*, 2020.

Shen, L., Li, Z., and Kwok, J. Timeseries anomaly detection using temporal hierarchical one-class network. *Advances in Neural Information Processing Systems*, 33, 2020.

Shipmon, D. T., Gurevitch, J. M., Piselli, P. M., and Edwards, S. T. Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data. *arXiv preprint arXiv:1708.03665*, 2017.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. 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*, pp. 2828–2837, 2019.

Wang, H., Li, H., Fang, J., and Wang, H. Robust Gaussian Kalman filter with outlier detection. *IEEE Signal Processing Letters*, 25(8):1236–1240, 2018.

Wei, G. C. and Tanner, M. A. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704, 1990.

Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pp. 187–196, 2018.

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., and Chawla, N. V. A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:1409–1416, jul 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33011409. URL www.aaai.orghttps://aaai.org/ojs/index.php/AAAI/article/view/3942.

Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., and Zhang, Q. Multivariate time-series anomaly detection via graph attention network. *arXiv preprint arXiv:2009.02040*, 2020.