WINDOW PROJECTION FEATURES ARE ALL YOU NEED FOR TIME SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The challenge of time series anomaly detection has motivated the development of increasingly more complex deep representations and anomaly metrics. In this paper we demonstrate that a simple approach based on window projection features can achieve better results. Projection features are a common way to discretize multivariate data; they first multiply the data by a projection matrix followed by discretization of each output dimension. We first show that short temporal windows, encoded by projection features, are often already sufficiently expressive for linearly separating between normal and anomalous time series. However, we find that while the expressivity of projection features is sufficient, current one-class classification methods are unable to use them effectively to detect anomalies. We hypothesize this is due to the difficulty of density estimation. The difficulty can be overcome by estimating the probability density of the sample mean, which follows the Gaussian distribution when the conditions of the Central Limit Theorem are met. Simply put, we fit a multivariate Gaussian model to the average of the projection features of adjacent windows within a time series. Despite its simplicity, our method outperforms the state-of-the-art in diverse settings including: five UEA datasets, video trajectory anomaly detection and standard anomaly segmentation benchmarks. Code is provided.

1 INTRODUCTION

Time series anomaly detection (AD) methods aim to determine if either an entire time series or parts of it contain novel patterns. This has important applications in science (e.g. detecting unusual stellar orbits for discovering black holes), medicine (e.g. detecting unusual ECG patterns) and industry (e.g. detecting unusual network traffic patterns for intrusion detection). Despite the importance of the task and the significant research effort spent on solving it, it remains an open challenge. Anomaly detection is difficult as no anomalies are seen in training. Training samples of anomalies typically cannot be obtained as they are rare and unexpected. A successful anomaly detector must deal with all anomalies despite not having seen them previously.

Anomaly detection methods strive to model the distribution of normal time series sufficiently well so that novel time series lying outside the distribution can be detected. As time series are quite complex, modeling their distributions is not trivial and requires making some assumptions. Density-based anomaly detection methods typically represent the data using hand-crafted or learned representations and then use a density estimation technique for modeling this distribution. Other anomaly detection methods assume that machine learning models trained on the normal data using auxiliary objectives (e.g. orientation prediction) will fail to generalize on anomalous data. In recent years, deep learning methods have been adopted due to their advanced expressivity and representation learning ability. This resulted in significant improvements in time series anomaly detection benchmarks.

In this paper, we present a simple, hand-crafted approach for time series anomaly detection. We first explore if time series anomaly detection is truly limited by the expressivity of representations. We represent each window of the time series using a simple representation, discretized projected windows (DPW). This representation, which is commonly used for providing a non-linear representation for multivariate data: i) projects the raw features of the window using an unlearned linear projection matrix into d projection dimensions ii) discertizes each of the resulting dimensions into B bins. We then test if a simple (supervised) linear classifier can discriminate between normal and anomalous

windows. To our surprise, the linear classifier was indeed successful (although it failed on the raw data). We therefore conclude that the simple DPW representation is sufficiently expressive.

Unfortunately, expressive representations are necessary but not sufficient for anomaly detection; the existence of a discriminative hyper-plane does not guarantee that it can be found without supervision. In fact, we show that a broad range of one-class classification methods were unable to achieve good performance on DPW features. This demonstrates that the main challenge in time series anomaly detection is not the representation, but rather finding the correct classifier without supervision. More optimistically, we conduct a third experiment, where instead of using the representation of single windows, we average the representations of densely sampled windows. Here, simple anomaly detection methods achieve excellent performance, particularly simple multivariate Gaussian density estimation.

The above observations are explained using a simple theoretical reason based on the Central-Limit-Theorem (CLT). We then formalize a simple method based on the empirical evidence and theoretical explanation. A thorough experimental state-of-the-art evaluation is conducted. Our method typically performed better than all previous methods despite being easy-to-implement.

2 PREVIOUS WORK

Time series Anomaly detection. The task of anomaly detection in time series has been studied over several decades, see Blazquez-Garcia et al. (Blázquez-García et al., 2021) for a comprehensive survey. In this paper, we are concerned with collective anomaly detection i.e. abnormal patterns in a collection of points. Traditional approaches for this task include generic anomaly detection approaches such as: K nearest neighbors (kNN) based methods e.g. vanilla kNN (Eskin et al., 2002) and Local Outlier Factor (LOF) (Breunig et al., 2000). Tree-based methods e.g. Isolation Forest (Liu et al., 2008). One-class classification methods e.g. One-Class SVM (Tax & Duin, 2004) and SVDD (Schölkopf et al.). Some traditional methods are particular to time series anomaly detection, specifically auto-regressive methods (Rousseeuw & Leroy, 2005). With the advent of deep learning, the traditional approaches were augmented with deep-learned features. Deep one-class classification methods include DeepSVDD (Ruff et al., 2018) and DROCC (Goyal et al., 2020). Deep auto-regressive methods include RNN-based prediction and auto-encoding methods (Bontemps et al., 2016; Malhotra et al., 2016). In addition, some deep learning anomaly detection approaches were proposed that are conceptually different from traditional approaches. These methods are based on the premise that classifiers trained on the normal data will struggle to generalize to anomalous data. These approaches were originally developed for image anomaly detection (Golan & El-Yaniv, 2018) but have been extended to tabular and time series data (Bergman & Hoshen, 2020; Qiu et al., 2021).

Discretized Projections. Using discretized projections of multivariate data has been used in many previous works. Locally sensitive hashing uses random projection and subsequent binary quantization as a hash for high-dimensional data. This is used among other things to facilitate fast k nearest neighbor search. This representation was also used by , with the modification of discretization by the median (rather than by some random value). This transformation is also highly related to the Radon transform Radon (1917). Kolouri et al. (Kolouri et al., 2015) used this representation as a building block in their set representation. HBOS Goldstein & Dengel (2012) performs anomaly detection by representing each dimension of multivariate data using a histogram discretized variable and subsequent density estimation. LODA Pevnỳ (2016) extends this work, by first projecting the data using a random projection matrix (followed by discretization). We differ from LODA in the use of a better density estimator and in using averages over multiple windows rather than a single window. Rocket and mini-rocket Dempster et al. (2020; 2021) represent time series for classification using the average of their window projection representations. Our contribution differs from them in: i) proposing anomaly detection rather than a supervised classification approach ii) connecting projection features to density estimation and the central limit theorem iii) proposing a new approach for anomaly segmentation.

3 PRELIMINARY

Notation. Our training data consists of N time series, denoted as $S_1, S_2..S_N$. Each series S consist of a set of T multivariate temporal observations $S = x_1, x_2..x_T$ (where T is the duration of the time

Experiment	EPSY		NAT		SAD		CT		RS	
	w	s	w	s	w	s	w	s	w	s
Raw, s=1	71.5	74.0	81.6	94.4	67.6	92.7	58.9	83.0	68.5	84.4
Proj, s=1	98.0	99.8	86.8	98.8	71.2	98.6	82.7	99.7	83.0	95.9
Proj, s=10	95.1	98.9	95.4	98.7	88.3	99.2	92.0	99.9	84.4	94.2

Table 1: *Exp.1.* Supervised classification accuracy (%) for different features and context duration. w indicates window level accuracy, s indicates series level accuracy. Projection features are far more expressive than raw data. Pyramid features perform better at the level of a single window. The performance gap between single and multi-resolution features virtually disappears at the series level.

series). Each observation x is d dimensional (so that $x \in \mathbb{R}^d$). We can also parameterize each time series S as a set of T overlapping windows $W = w_1, w_2..w_T$ each of duration τ. Prior to window extraction, the series S is first right and left zero-padded by $\frac{\tau}{2}$ to form padded series S'. The first window w_1 is defined as the first τ observations in padded series S', i.e. $w_1 = x'_1, x'_2..x'_\tau$. We further define windows at higher scales W^s, which include observations sampled with stride s. At scale s, the original series S is right and left zero padded by $\frac{s \cdot \tau}{2}$ to form padded series S'^s. The first window w_1^s will be defined as $w_1^s = x'_1, x'_{1+s}, ..x'_{1+\tau s}$. Finally, for each series S, we define a pyramid feature R^s consisting of the concatenation of all windows $W^1, W^2..W^s$. The pyramid feature for time t will be given by the concatenation $w_t^1, w_t^2..w_t^s$.

Discertized projections. The discretized projection transformation operates on a multivariate input v of dimension $d \cdot T \cdot s$. It first projects the input v to another dimension r using projection matrix $P \in \mathbb{R}^{(d \cdot T \cdot s \times r)}$. The dimension r can be higher or lower than the original input dimension. The projection matrix P may be selected by different methods which are explored in the paper; the simplest being random independent sampling of each cell from the normal distribution. The resulting projected matrix is denoted as p:

$$p = P \cdot v \tag{1}$$

Each dimension j of p is then discretizes into B bins. The discretization is performed by dividing the region between maximal and minimum values of p_j in the training set into B equally spaced bins, and mapping p_j into the index b of the bin into which it falls.

4 Empirical Investigation: Representation and Supervision in Time Series Anomaly Detection

In this section, we perform an empirical investigation exploring the different dimensions of the time series anomaly detection task: i) representation ii) temporal window duration iii) supervision.

Experiment *1*: Are projection features of a short temporal window sufficiently expressive to distinguish normal from anomalous data? Our objective in the first experiment is to determine if the simple projection features advocated here a sufficiently expressive for anomaly detection. Another objective is determining what duration of temporal content is needed.

For each series S, we extract a set of temporal windows W each of duration $\tau = 9$. We also extracted pyramid features with 10 scales ($s = 10$). In this experiment only (but not in any other experiment in the paper), we utilize a training set consisting of both normal and anomalous time series, which are **labelled**. We trained a (linear) ridge regression model on the training set consisting of the DPW of each window of all training time series. The target was the label of the series (normal or anomalous).

Data. We run this experiment on the 5 datasets used in Qiu et al. (2021), which were adapted from the UEA repository. Each dataset was originally a multi-class time series classification dataset and was adapted to anomaly detection by Qiu et al. by setting a single class as normal and all other as anomalies. The reported results are obtained by averaging over classes.

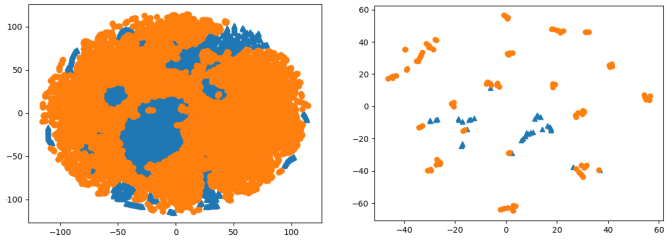

Figure 1: t-SNE plot of the normal (blue) and anomalous (orange) time series for window (left) and average (right) features. The average features are more compact / lower variance than the window features.

Experiment	EPSY		NAT		SAD		CT		RS	
	w	s	w	s	w	s	w	s	w	s
single win, kNN	77.7	77.8	86.0	95.1	62.9	95.0	76.2	99.0	65.7	81.2
single win, Gaussian	68.0	68.6	86.2	94.0	57.5	75.4	76.3	98.3	60.1	70.6
average, kNN	-	95.5	-	95.5	-	94.5	-	99.5	-	86.3
average, Gaussian	-	98.1	-	96.1	-	97.8	-	99.7	-	92.3

Table 2: *Exp.2 & 3*. Anomaly detection accuracy (%) by density estimation at the level of individual windows and series-level averages. w indicates window level accuracy, s indicates series level accuracy. kNN outperform Gaussian models when estimating at the window level. Gaussian models outperform kNN when estimating at the series-level. A theoretical explanation is presented in Sec. 5

Results. The results are presented in Tab. 4. we can see that even a single window provides significant discrimination between normal and anomalous data. We also performed the same as above, but instead of DPW, we used the raw data. We can see the raw features were insufficient for the classification task. Finally, we conducted the same experiment as above with pyramid features and $s = 10$. We can see that only SAD benefited from the increased number of scales. The SAD dataset requires identification of Arabic digits which typically require the context of the entire digit.

Key findings. We showed that even a short temporal window can be sufficient for identifying anomalies in many cases. In cases where longer scales are needed, a temporal pyramid can be used. Most significantly, while the raw data is not linearly separable, DPW features are able to linearly separate between normal and anomalous data and therefore are sufficiently expressive.

Experiment *2*: Can the DPW features of a single window be used for detecting anomalous windows? As we showed in Exp *1* that DPW features are sufficiently expressive, one could hypothesize that they can be used for window anomaly detection in the one-class classification setting. The difference between the setting here and the one in Exp. *1* is that no examples of anomalies are presented during training. The challenge is that a separating hyper-plane (or manifold in general) needs to be found between one class without seeing the second class at all. In this experiment, we extract DPW features from all normal training windows (no anomalous data are used for training anywhere in the paper, except for Exp. *1*). We then test two classical anomaly detection methods on the DPW features including: kNN and a Gaussian estimator. We do not use deep learning methods, as the representation is already sufficiently expressive without deep features.

Data. We used the smaller datasets from Exp. *1* (as the larger dataset required too much memory). No anomalous time series were used in training.

Results. Tab. 4 presents the detailed results. We can observe that the accuracy is much lower than in the supervised case, as the correct manifold is not found without supervision. kNN achieved the best results, presumably as it does not require learning a parametric separating manifold, however the results are still much weaker than in the supervised case.

Key findings. One-class classification methods perform poorly with DPW features. The challenge is not expressivity but finding the correct separating manifold without anomaly supervision.

Experiment *3*: Averaging DPW features across adjacent windows. While OCC methods did not perform well when applied on top of DPW features of a single window, we investigate the utilization of the continuous nature of time series. Specifically, anomalies in time series tend to occupy a duration of time, often spanning multiple windows. Here, we test if averaging the DPW features of multiple adjacent windows leads to improved OCC performance. The hypothesis is that while individual windows may be quite noisy, making density estimation tricky, the average of multiple windows may be more stable and more distinct. A theoretical explanation is provided in Sec. 5.

Data. We use exactly the same data as Exp. *2*.

Results. A t-SNE plot of the features of normal and anomalous data on the EPSY dataset are presented in Fig. 1. In the left plot, the features are DPW of single windows. In the right plot, the features are the average DPW of all windows across the series. We can see that distributions of normal and anomalous data are much easier to distinguish for average DPW features. We present the results of OCC method on the average window features in Tab. 4. It is clear that the Gaussian estimator performs very well for anomaly detection, while kNN performs well but underperforms it. Averaged DPW features perform much better than single window features.

Key findings. Averaging DPW features across multiple adjacent windows significantly improves OCC accuracy.

5 Using the Central Limit Theorem for Time Series AD

The previous section presented an important empirical insight; window projection features are sufficiently informative for separating normal from anomalous time series series. While OCC methods cannot use these features directly for density estimation, averaging the features of all windows is more powerful. In fact, aa Gaussian estimator performs poorly at the window level, it is highly effective when used on the averaged feature.

Theoretical explanation. We present a simple explanation for the above observations. We model the features of each window f from normal time series as IID observations from a probability distribution function $p(f)$. The distribution function is *not* assumed to be Gaussian. Using a Gaussian density estimator trained on the features of windows observed in training is unlikely to be effective for anomaly detection (due to the non-Gaussian $p(f)$). A kNN estimator will often be be more effective for estimating $p(f)$ due to its non-parametric nature.

We propose to estimate the likelihood of the sample mean i.e. the mean of a set of sampled windows rather than the features of a single window. The sample mean has superior statistical properties, in particular, the Central Limit Theorem states that under some conditions the sample mean follows the Gaussian distribution regardless of the distribution of windows $p(f)$. While typically in anomaly detection only a single sample is presented at a time, the situation is different for time series. We consider a time series as a set of windows. While a the windows are often not IID, given a time series spanning multiple periods, an IID approximation is justifiable (as windows that are well separated in time are roughly independent).

Given the above analysis, we conclude that the average of the features of all window in a time series approximately follows the Gaussian distribution. This explains the observations in Exp *2* and *3*.

Method. We formalize our method here. Let us denote the DPW features for each window w as f. The average of DPW features for all windows W in series S is denoted as a. We compute the mean and full covariance matrix of the average features $a_1, a_2..a_N$ of all training time series and denote them as μ_{tr}, Σ_{tr}. As the full covariance matrix is very high-dimensional, we use a constant shrinkage factor (we used 0.03 but other values are fine). We model the distribution of the averaged DPW of the normal data using a Gaussian distribution:

$$p_{norm}(a) = \mathcal{N}(a|\mu_{tr}, \Sigma_{tr}) \tag{2}$$

By taking the logarithm and neglecting the constant term, the series-level anomaly score becomes:

$$score(a) = \frac{1}{2}(a - \mu_{tr})^T \Sigma_{tr}^{-1}(a - \mu_{tr}) \tag{3}$$

Window-level Anomaly Scoring. As an additional contribution, we derive a window-level anomaly score. Let us rewrite the averaged DPW feature a as the sum of the window DPW features f_w weighted by per-window factor α_w (in fact we used $\alpha_w = \frac{1}{n}$ for all windows):

$$a = \sum_{w \in W} \alpha_w f_w \tag{4}$$

We measure the per-window anomaly score by the influence the window has on the anomaly score. More precisely, the per-window anomaly score $score_w(f_w)$ is given by the derivative of $score(a)$ for the entire time series with respect to the weight of the window α_w:

$$score_w(f_w) = \frac{\partial score(a)}{\partial \alpha_w} = (f_w^T - \mu_{tr})^T \Sigma_{tr}^{-1} (a - \mu_{tr}) \tag{5}$$

One interpretation is that the linear hyperplane between normal and anomalous windows has vector $\Sigma_{tr}^{-1}(a - \mu_{tr})$. This "classifier" varies for different test time series (each has a different averaged DPW value a). One can interpret this mechanism as test-time training, see App. G for details.

Limitation. Our method assumes that the average of window features is an unbiased estimate of the sample mean. When the observed time series is very short, this estimate of the sample mean might be biased. In that case, averaging the window features is still helpful in reducing the variance of observations. Conversely, the CLT result will not hold in that case and there is no guarantee that the averaged features will follow the Gaussian distribution. We investigate this phenomenon in Sec. 6.3.

Relation to previous methods. Our method is related to several previous methods. HBOS (Goldstein & Dengel, 2012) and LODA (Pevnỳ, 2016) also used similar projection features for anomaly detection but performed histogram-based density estimation by ignoring the dependency across projections. As they can only be applied to a single window (similarly to Exp. 2), they do not achieve competitive performance for time series AD. Rocket/mini-rocket (Dempster et al., 2020; 2021) also average projection features across windows but do not tackle anomaly detection. The connection to density estimation and the central limit theorem is novel. Finally, there is a subtle connection to Radon transform (Kolouri et al., 2015) and sliced Wasserstein-distance-based methods (Bonneel et al., 2015) which also use similar projection and histogram features. However, the sliced Wasserstein distance does not admit a simple Gaussian interpretation as presented here and therefore was not used for time series anomaly detection.

6 Experiments

We extensively evaluate our approach against a large range of time series anomaly detection approaches. Our evaluation spans both long standing, time-tested methods as well as more recent deep learning methods that currently achieve the state-of-the-art on different benchmark datasets. In Sec. 6.1, we evaluate our method against a large number of methods on a set of benchmark anomaly detection datasets. In Sec. 6.2, our method is evaluated on activity anomaly detection on landmark trajectory data against specialized deep learning approaches. In Sec. 6.3, our method is evaluated on anomaly segmentation (sub-sequence). We perform extensive ablations in Sec. 6.4.

6.1 Evaluation on Whole Series Anomaly Detection

Dataset. We compare our results from Exp. *3* on the UEA datasets (obt) against the state-of-the-art methods presented in the paper by Qiu et al. (2021). The datasets were described in App. A.

Metric. Following Qiu et al. (2021), we the series level ROCAUC metric (also used in Sec. 4).

Baselines. We copy the results of several baselines methods reported by Qiu et al. (2021). The method cover the following paradigms: One-class classification (OCC) - One-class SVM (OCSVM) and its deep versions DeepSVDD (commonly used) and DROCC (recently published) are one-class classifiers. Tree-based - Isolation Forest (IF). Nearest-neighbors - LOF as a specialized version of nearest neighbor anomaly detection. Auto-regressive - RNN and ED (LSTM encoder-decoder) are deep neural network-based version of auto-regressive prediction models and are probably the most commonly used methods in time series anomaly detection. Transformation prediction

Table 3: UEA datasets, average ROCAUC (%) over all classes. (σ presented in Tab. 8)

	OCSVM	IF	LOF	RNN	ED	DeepSVDD	GOAD	DROCC	NeuTraL	Ours
EPSY	61.1	67.7	56.1	80.4	82.6	57.6	76.7	85.8	92.6	**98.1**
NAT	86.0	85.4	89.2	89.5	91.5	88.6	87.1	87.2	94.5	**96.1**
SAD	95.3	88.2	98.3	81.5	93.1	86.0	94.7	85.8	**98.9**	97.8
CT	97.4	94.3	97.8	96.3	79.0	95.7	97.7	95.3	99.3	**99.7**
RS	70.0	69.3	57.4	84.7	65.4	77.4	79.9	80.0	86.5	**92.3**
Avg.	82.0	81.0	79.8	86.5	82.3	81.1	87.2	86.8	94.4	**96.8**

Table 4: Trajectory AD accuracy on the ShanghaiTech Campus Dataset (ROCAUC %).

Morais et al. (2019)	Markovitz et al. (2020)	Ours
73.4	75.2	**76.1** $\pm$ 0.3

- GOAD and NeuTraL-AD are based on transformation prediction, and are adaptations of RotNet-based approaches (such as GEOM (Golan & El-Yaniv, 2018)).

Results. Our results are presented in Tab. 3. We can observe that within the baselines, different approaches are effective for different datasets. kNN-based LOF is highly effective for SAD which is a large dataset but achieves worse results for EPSY. Auto-regressive approaches achieve strong results on CT. Transformation-prediction approaches, GOAD and NeuTraL achieve the best performance of all the baselines. The learned transformations of NeuTraL achieved better results than the random transformations of GOAD. Our method achieves the best overall results both on average and individually on all datasets apart from SAD (where it is comparable but a little lower than NeuTraL). Note that differently from NeuTraL, our method is far simpler, does not use deep neural networks and is very fast to train and evaluate. It also has few hyper-parameters and is well-grounded. Note that DAGMM and LSTM-AE were not presented in the table due to lack of space, both significantly under-performed the top methods presented in the table,

6.2 Detecting Anomalous Activity from Landmark Trajectory Data

We compare our method on trajectory anomaly detection against two state-of-the-art methods.

Benchmark. We use exactly the same setup as Markovitz et al. Markovitz et al. (2020). We briefly describe it here for completeness. The benchmark first extracts 12 frame sequences from all training and test videos in the Shanghai Tech video anomaly detection dataset. The 27 $2D$ landmark positions of all people are extracted for each frame. The trajectory of each person is associated across time, creating time series of duration $T = 12$ time steps each. Each observation consists of the x and y pixel positions of each of 27 landmarks (so dimension $r = 54$ in total). The training set consist of time series extracted from videos containing only normal trajectories, while the test set consists of both normal and anomalous time series.

Metric. We follow Markovitz et al. (2020) in measuring performance using frame-level ROCAUC. Methods calculate an anomaly score for each trajectory. The anomaly score for each video frame is the maximum of the anomaly score of all trajectories that intersect it. The ROCAUC is calculated between the frame anomaly score and the groundtruth value provided by the dataset.

Baselines. We presents a comparison against the other trajectory-based methods presented in Markovitz et al. (2020). MPED-RNN by Morais et al. (2019) uses RNN-based prediction and reconstruction losses. The best performing approach, ST-GCAE, is quite complex, it uses a temporal graph neural network with multiple training stages.

Results. The results of the evaluation are presented in Tab. 4. Our method achieves better results than the state-of-the-art baselines which are much more complex and compute intensive.

Table 5: Anomaly Segmentation (F1 %).

	Norm	kNN	Omni	THOC	AT	GDN	GANF	Ours
MSL	23.5	19.1	19.2	24.5	19.1	20.5	21.9	**32.5**
SMAP	22.8	22.7	23.4	22.7	22.7	25.4	24.2	**27.6**
PSM	48.3	55.4	43.5	43.8	43.4	54.7	49.7	**58.6**
SWaT	77.1	78.1	23.8	21.7	21.7	76.2	76.2	77.8

Table 6: An ablation of the effect of covariance estimation (ROCAUC).

	EPSY	RS	NA	CT	SAD
Identity	62.1	70.9	93.6	98.5	78.8
Full	98.1	92.3	96.1	99.7	97.8

6.3 Anomaly Segmentation

The theoretical justification of our method assumes that input time series are long enough so that they approximate the distribution of windows, but this is not always the case. The task of detecting temporally localized anomalies in a long online time series is often tackled by splitting the time series into a set of short sequences. The task is to detect if each sequence is normal or anomalous. Sequences are unable to provide an unbiased estimate of the sample mean of window features, as they contain only a few correlated time steps. To use our method on such datasets, we treat each sequence as a time series, and use windows of duration 1. While our method holds no guarantees of Gaussianity in this case, we find that it is effective in practice.

Benchmark Datasets. We apply our method on 4 widely used benchmark time series anomaly segmentation datasets: SMAP Hundman et al. (2018), MSL Hundman et al. (2018), PSM Abdulaal et al. (2021) and SWaT Mathur & Tippenhauer (2016). The datasets are described in App. C.

Metric. We use the F1 score, as is typical in this field. We do not use point-adjust due to its documented limitations (see App. E). For details of the protocol and implementation see App. F.

Baselines. We compare our method and five deep learning baselines. Current state-of-the-art methods: AnomalyTransformer (AT) Xu et al. (2021) and GANF Dai & Chen (2022) both highlighted as Spotlight papers in ICLR'22. Prominent methods: GDN Deng & Hooi (2021) from AAAI'21, THOC Shen et al. (2020) from NIPS'20 and OmniAnomaly Su et al. (2019b) from KDD'19. We additionally compute two simple but powerful baselines: kNN and simply using the euclidean norm of the window (Norm). We did not include the SMDSu et al. (2019a) dataset as it consists of many simple point anomalies, which are easily detected by the Norm or kNN baselines, and using longer contexts was not particularly useful there.

Results. We present the numerical evaluation in Tab. 5. We can observe that deep learning baselines did not consistently outperform traditional methods. GDN was the best performing deep learning method; it also outperformed the classical baselines on SMAP. Most methods perform comparably on SWaT. Our method achieves the best performance of all methods on SMAP, SML and PSM.

6.4 Ablations

Number of projections. Using a high output dimension for projection matrix P increases the expressively but increase the computation cost. We investigate the effect of the number of projections on the final accuracy of our method. The results are provided in Fig. 2. We can observe that although a small number of projections hurts performance, even a moderate number of projections is sufficient. We found 100 projections to be a good tradeoff between peformance and runtime.

Number of bins. We compute the accuracy of our method as a function of the number of bins per-projection. Our results (Fig. 2) show that beyond a very small number of bins - larger numbers are not critical. We found 20 bins to be sufficient in all our experiments.

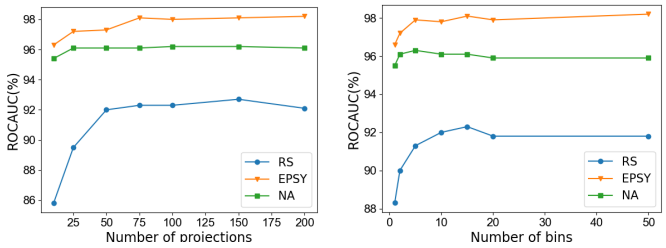

Figure 2: Ablation of accuracy vs. number of projections (left) and number of bins (right).

Table 7: An ablation of projection sampling methods (ROCAUC).

	EPSY	RS	NA	CT	SAD
Id.	97.1	90.2	91.8	98.2	78.3
PCA	98.2	91.6	95.8	99.7	96.7
Rand	98.1	92.3	96.1	99.7	97.8

Effect of Gaussian density estimation. Standard projection methods such as HBOS Goldstein & Dengel (2012) and LODA Pevnỳ (2016) do not use a multivariate density estimator but instead estimate the of each dimension independently. We compare using a full and per-variable density estimation in Tab. 6. We can see that our approach achieves far better results, attesting to the importance of modeling the correlation between projections. .

Comparing projection sampling methods. We compare three different projection selection procedures: i) Gaussian: sampling the weights cell in P from a random Normal Gaussian distribution ii) Using a identity projection matrix: $P = I$. iii) PCA: selecting P from the eigenvectors of the matrix containing all (raw) features of all training windows. PCA selects the projections with maximum variation but is computationally expensive. The results are presented in Tab. 7. We find that the identity projection matrix under-performed the other approaches (as it provides no variable mixing). Surprisingly, we do not see a large difference between PCA and randomly projections.

7 Discussion

Incorporating deep features. It was shown that our method was able to outperform the state-of-the-art in time series anomaly detection without using deep neural networks. Although an interesting and surprising result, we believe that deep features will be incorporated into our approach in the future. One direction is replacing the window projection features by suitable deep representations, while keeping the averaging and Gaussian modeling steps unchanged.

Anomaly detection beyond time series. The CLT ideas presented here can be applied to any anomaly detection task where the input is an IID set e.g. set or video anomaly detection.

8 Conclusion

This paper presented a method for time series anomaly detection using window projection features. The features were shown to be sufficiently expressive for discriminating between normal and anomalous data, but finding the correct classifier required supervision. On the other hand, we found that averaging features across adjacent windows removed unwanted degrees of variation, making finding an accurate classifier even without supervision. This approach was then extended to allow window level scores. We presented theoretical motivation for our approach, making the connection to the central limit theorem. Our approach can be extended to other anomaly detection tasks, where a set of similar samples from the same class are presented at test time.

REFERENCES

Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. Practical approach to asynchronous multivariate time series anomaly detection and localization. pp. 2485–2494, 08 2021. doi: 10.1145/3447548.3467174.

Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020.

Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Loïc Bontemps, Van Loi Cao, James McDermott, and Nhien-An Le-Khac. Collective anomaly detection based on long short-term memory recurrent neural networks. In *International conference on future data and security engineering*, pp. 141–152. Springer, 2016.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Enyan Dai and Jie Chen. Graph-augmented normalizing flows for anomaly detection of multiple time series. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=45L_dgP48Vd.

Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.

Angus Dempster, Daniel F Schmidt, and Geoffrey I Webb. Minirocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 248–257, 2021.

Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4027–4035, 2021.

Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pp. 77–101. Springer, 2002.

Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pp. 9758–9769, 2018.

Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, 2012.

Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. Drocc: Deep robust one-class classification. In *International Conference on Machine Learning*, pp. 3711–3721. PMLR, 2020.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. pp. 387–395, 07 2018. doi: 10.1145/3219819.3219845.

Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. *AAAI*, 2022.

Soheil Kolouri, Se Rim Park, and Gustavo K Rohde. The radon cumulative distribution transform and its application to image classification. *IEEE transactions on image processing*, 25(2):920–934, 2015.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. IEEE, 2008.

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

Amir Markovitz, Gilad Sharir, Itamar Friedman, Lihi Zelnik-Manor, and Shai Avidan. Graph embedded pose clustering for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10539–10547, 2020.

Aditya Mathur and Nils Ole Tippenhauer. Swat: a water treatment testbed for research and training on ics security. pp. 31–36, 04 2016. doi: 10.1109/CySWater.2016.7469060.

Romero Morais, Vuong Le, Truyen Tran, Budhaditya Saha, Moussa Mansour, and Svetha Venkatesh. Learning regularity in skeleton trajectories for anomaly detection in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11996–12004, 2019.

Tomáš Pevnỳ. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, 2016.

Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. Neural transformation learning for deep anomaly detection beyond images. *ICML*, 2021.

Johann Radon. 1.1 über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Proceedings of the Royal Saxonian Academy of Sciences at Leipzig*, 1917.

Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402, 2018.

Bernhard Schölkopf, John C Platt, et al. Support vector method for novelty detection. Citeseer.

Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. In *Advances in Neural Information Processing Systems*, 2020.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. KDD '19, New York, NY, USA, 2019a. Association for Computing Machinery. doi: 10.1145/3292500.3330672. URL https://doi.org/10.1145/3292500.3330672.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 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, 2019b.

David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1): 45–66, 2004.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy, 2021.

A UEA DATASETS

In this section, we provide a brief description of the UEA datasets used in the experiments in Sec. 6:

RacketSports (RS). Accelerometer and gyroscope recording of players playing four different racket sports. Each sport is designated as a different class.

Table 8: UEA datasets, average ROCAUC over all classes inlcuding error bounds

	OCSVM	IF	LOF	RNN	LSTM-ED
EPSY	61.1	67.7	56.1	80.4 ± 1.8	82.6 ± 1.7
NAT	86	85.4	89.2	89.5 ± 0.4	91.5 ± 0.3
SAD	95.3	88.2	98.3	81.5 ± 0.4	93.1 ± 0.5
CT	97.4	94.3	97.8	96.3 ± 0.2	79 ± 1.1
RS	70	69.3	57.4	84.7 ± 0.7	65.4 ± 2.1
Avg.	82	81	79.8	86.5	82.3
	DeepSVDD	GOAD	DROCC	NeuTraL	Ours
EPSY	57.6 ± 0.7	76.7 ± 0.4	85.8 ± 2.1	92.6 ± 1.7	**98.1** ± 0.3
NAT	88.6 ± 0.8	87.1 ± 1.1	87.2 ± 1.4	94.5 ± 0.8	**96.1** ± 0.1
SAD	86 ± 0.1	94.7 ± 0.1	85.8 ± 0.8	**98.9** ± 0.1	97.8 ± 0.1
CT	95.7 ± 0.5	97.7 ± 0.1	95.3 ± 0.3	99.3 ± 0.1	**99.7** ± 0
RS	77.4 ± 0.7	79.9 ± 0.6	80 ± 1	86.5 ± 0.6	**92.3** ± 0.3
Avg.	81.1	87.2	86.8	94.4	**96.8**

Epilepsy (EPSY). Accelerometer recording of healthy actors simulating four different activity classes, one of them being am epileptic shock.

Naval air training and operating procedures standardization (NAT). Positions of sensors mounted on different body parts of a person performing activities. There are six different activity classes in the dataset.

Character trajectories (CT). Velocity trajectories of a pen on a WACOM tablet. There are 20 different characters in this dataset.

Spoken Arabic Digits (SAD). MFCC features of ten arabic digits spoken by 88 different speakers. We follow the processing of the dataset as done by Qiu et al. Qiu et al. (2021). In private communications the authors explained that only sequences of lengths between 20 and 50 time steps were selected. The other time series were dropped.

The datasets are freely available from `https://www.timeseriesclassification.com/`. We provided 4 of the datasets in our repo. SAD is too large for GitHub and should be downloaded from the original website.

B UEA Results with Standard Errors

We present an extended version of the UEA results including error bounds for our method and baselines that reported them. The difference between the methods is significantly larger than the standard error.

C Anomaly Segmentation Benchmark Datasets.

In this section we provide a brief description of the benchmark datasets used in the experiments in Sec. 6.3.

Soil Moisture Active Passive (SMAP) and *Mars Science Laboratory (MSL)* Hundman et al. (2018). Both are public NASA datasets collected by a spacecraft, which contain telemetry anomaly data. The anomalous data is derived from the Incident Surprise Anomaly (ISA) reports of the spacecraft monitoring system. Anomalies are present in both training and testing data, with labels only for the latter.

Pooled Server Metrics (PSM) Abdulaal et al. (2021). collected from multiple application server nodes at eBay. The training set consists of 26 dimensions of server machine metrics such as CPU utilization and memory. The data was collected over 21 weeks. The first 13 weeks data is the training data, followed by eight weeks for testing. Anomalies are present in both training and testing

Table 9: Anomaly segmentation detailed results, with Precision (P), Recall (R) and F1 scores

Method	MSL			SMAP			PSM			SWaT		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Norm	13.6	84.4	23.5	12.9	98.9	22.8	32.4	95.2	48.3	98.2	63.4	77.1
kNN	10.5	100.0	19.1	16.3	37.2	22.7	39.7	91.6	55.4	97.5	65.1	78.1
Omni	10.7	91.4	19.2	13.4	92.1	23.4	27.8	100.0	43.5	13.6	98.2	23.8
THOC	16.4	48.5	24.5	12.8	100.0	22.7	28.5	94.5	43.8	12.1	100.0	21.7
GDN	11.7	85.5	20.5	14.9	85.1	25.4	50.4	59.9	54.7	98.3	62.4	76.2
GANF	13.3	61.6	21.9	13.9	92.8	24.2	47.6	52.0	49.7	98.9	62.0	76.2
AT	10.5	100.0	19.1	12.8	100.0	22.7	27.7	100.0	43.4	12.1	100.0	21.7
Ours	24.8	47.3	**32.5**	17.3	67.5	**27.6**	43.7	89.3	**58.6**	99.4	63.9	77.8

Table 10: Adjusted F1 Metric - Random (p=0.01) and Anomaly Transformer over 4 datasets

	Random		**AT**	
	F1	Adjusted	F1	Adjusted
MSL	0.0202	0.9533	0.1907	0.9420
SMAP	0.0347	0.9311	0.2268	0.9650
PSM	0.0191	0.9797	0.4342	0.9790
SWaT	0.0344	0.9333	0.2165	0.9335

data, with labels only for the latter. The labels were manually created by engineers and application experts.

Secure Water Treatment (SWaT) Mathur & Tippenhauer (2016). The data is of a water treatment testbed compromising 51 sensors, that was collected over 11 days. The first 7 days data is the normal training data. Then, in the last 4 days, 41 attacks were launched with different intents and diverse durations.

D Anomaly Segmentation Detailed Results

The detailed Precision (P), Recall (R) and F1 scores on the anomaly segmentation experiments are in Tab. 9.

E F1-Adjusted Score

The adjusted-F1 score have been used by many recent papers to evaluate performance on subsequence anomaly detection. The metric adjust the score for each ground truth anomalous sequence with the following rule: if the model correctly predicted at least one time point from an entire (potentially long) anomaly duration, all the time points of the anomaly are marked as true positives. It has been shown very recently (e.g. Kim et al. Kim et al. (2022)) that this metric does not correctly rank different methods. For example, a random binomial baseline predicting each time point as anomalous with probability 0.01 outperforms all existing methods on this metric. In Tab. 10 we evaluate the performance of the random binomial prediction vs. Anomaly Transformer prediction (which is the top performing method in terms of adjusted F1). The scores of Anomaly Transformer on 4 datasets are comparable with the random guess.

F IMPLEMENTATION DETAILS

UEA Experiments. We used each time series as an individual training sample. We chose a kernel size of 9, 100 projection, 20 quantiles max number of resolutions of 10. The results varied only slightly within a reasonable range of the hyperparameters e.g. using 5, 10, 15 resolutions yielded an average ROCAUC of 97, 96.8, 96.8 across the five UEA datasets.

For all experiments we used the full train/test datasets without any sub-sampling nor strides. We allowed different sub-sequence length and other processing within the models. However, f1-best score evaluation was obtained in relation to the original labels. We padded few missing scores where needed with last score (caused by different sub-sequence length used by different models).

Landmark Experiment. We used the same hyperparameter values as the UAE datasets. We used the code of Markovitz et al. Markovitz et al. (2020) for preparing the data and for evaluation metric. The only change was replacing the authors' predictive model by our own.

Sub-sequence Experiments. We used a common data loader and scoring metrics for all methods, and reported scores by F1-best. kNN used one nearest neighbor from all train set, and is implemented with Faiss. Norm simply computes the Euclidean norm computed over a sub-sequences of length 100 centered at each time point in the test set. For the deep learning methods, we used the authors' code for training each model and followed the suggested parameters given in the authors' paper and their code instructions. GANF is tested with sub-sequences of length 60, and without subsampling. GDN was evaluated with their suggested sub-sequences of length 5. To evaluate their default sub-sampling method, we also evaluated it over sub-sequences of length 50. The presented scores using sub-sequences of length 5, which performs better. AnomalyTransformer is with default parameters for each of the datasets, and default sub-sequences length of 100. OmniAnomaly was evaluated with default hyper-parameters suggested by the authors. THOC was evaluated with default parameters given by the authors - sub-sequences length is 100 with stride 100 for MSL and SMAP, and 1 for the others (PSM was experimented for both of them). Our method is evaluated with 200 projections and 10 bins, sub-sequences length of 100 and window size of length 1.

Analysis in Sec. **??**. We used the same hyperparameters as the above experiments. We split the time series into overlapping windows of length 9. Each window was used to predict the label for its central time point. We found that using projections from the range $[-1, 2]$ worked better than random weights. For the point anomaly detection experiments, we used an autoregressive model to predict the next time point after the window. Poor prediction for a specific time point is used to it as an anomaly.

Computational resources. The experiments were run on a modest number of CPUs on a computing cluster. The baseline methods were run on a single RTX2080-GT GPU

G A TEST-TIME TRAINING INTERPRETATION.

We provide an intuitive interpretation of this result, as test time training. One of the most standard methods for calculating the separating hyper-plane between two classes in linear discriminant analysis (LDA). In this method, the separating hyper-plane $score(f) = f \cdot l + u$ has its slope l given by:

$$l = \Sigma^{-1}(\mu_2 - \mu_1) \tag{6}$$

Where Σ is the covariance of the data. It is assumed to be shared across classes, which is a strong assumption often not satisfied, but the formulation has been found to not hurt performance in the out-of-distribution literature (e.g. Lee et al.). μ_1, μ_2 are the means of the first and second classes respectively. In anomaly detection, we can estimate the average of the normal class features, but do not know the average of the features of the anomalous data (as none are provided). Time series AD present a unique opportunity, as in test time, an entire series with multiple windows is presented. This allows us to estimate the mean of the (suspected) anomalous class. This motivates the per-window classifier of $\Sigma_{tr}^{-1}(a - \mu_{tr})$. Note, there is a gap between the theory and practice as we estimate Σ_{tr} using the average DPW features a instead of the window features f as we found this estimate to work much better.

H SOCIETAL IMPACT

Detecting anomalies is of high practical value for science and industry. E.g.,the theory of quantum mechanics originate from Plank's empirical discovery of an anomalous photoelectric effect. Conversely, anomaly detection can be for suppressing phenomena that are unusual or unique by bad actors. By making our research open, we are letting the community have full knowledge of the current technological capabilities.