

AER: Auto-Encoder with Regression for Time Series Anomaly Detection

Lawrence Wong, Dongyu Liu, Laure Berti-Equille, Sarah Alnegheimish, Kalyan Veeramachaneni

▶ **To cite this version:**

Lawrence Wong, Dongyu Liu, Laure Berti-Equille, Sarah Alnegheimish, Kalyan Veeramachaneni. AER: Auto-Encoder with Regression for Time Series Anomaly Detection. 2022 IEEE International Conference on Big Data, IEEE, pp.1152-1161, 2022, 978-1-6654-8045-1. 10.1109/BigData55660.2022.10020857 . ird-04382119

HAL Id: ird-04382119
https://ird.hal.science/ird-04382119

Submitted on 10 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

AER: Auto-Encoder with Regression for Time Series Anomaly Detection

Lawrence Wong
MIT
Cambridge, USA
lcwong@mit.edu

Dongyu Liu
MIT
Cambridge, USA
dongyu@mit.edu

Laure Berti-Equille
IRD ESPACE-DEV
Montpellier, France
laure.berti@ird.fr

Sarah Alnegheimish
MIT
Cambridge, USA
smish@mit.edu

Kalyan Veeramachaneni
MIT
Cambridge, USA
kalyanv@mit.edu

arXiv:2212.13558v1 [cs.LG] 27 Dec 2022

***Abstract*—Anomaly detection on time series data is increasingly common across various industrial domains that monitor metrics in order to prevent potential accidents and economic losses. However, a scarcity of labeled data and ambiguous definitions of anomalies can complicate these efforts. Recent unsupervised machine learning methods have made remarkable progress in tackling this problem using either single-timestamp predictions or time series reconstructions. While traditionally considered separately, these methods are not mutually exclusive and can offer complementary perspectives on anomaly detection. This paper first highlights the successes and limitations of prediction-based and reconstruction-based methods with visualized time series signals and anomaly scores. We then propose AER (Auto-encoder with Regression), a joint model that combines a vanilla auto-encoder and an LSTM regressor to incorporate the successes and address the limitations of each method. Our model can produce bi-directional predictions while simultaneously reconstructing the original time series by optimizing a joint objective function. Furthermore, we propose several ways of combining the prediction and reconstruction errors through a series of ablation studies. Finally, we compare the performance of the AER architecture against two prediction-based methods and three reconstruction-based methods on 12 well-known univariate time series datasets from NASA, Yahoo, Numenta, and UCR. The results show that AER has the highest averaged F1 score across all datasets (a 23.5% improvement compared to ARIMA) while retaining a runtime similar to its vanilla auto-encoder and regressor components. Our model is available in Orion[1], an open-source benchmarking tool for time series anomaly detection.**

***Index Terms*—anomaly detection, time series data, auto-encoder, regression, machine learning**

I. Introduction

Time series data is consistently generated and collected across various industries – examples include stock prices in finance, vital signs in healthcare, and retail sales in business. Effective monitoring and use of time series data are essential for increasing efficiency and productivity. In addition, analysis of time series data can extrapolate recurring patterns to predict future occurrences. Anomaly detection, an important task within time series analysis, explicitly aims to identify unexpected events. This research is increasingly relevant due to its broad applications in detecting crucial issues, such as financial fraud in trading networks [8], medical problems in electrocardiograms [5], [16], and ecosystem disturbances in satellite signals [21].

[1]The AER model is available in Orion: https://github.com/sintel-dev/Orion

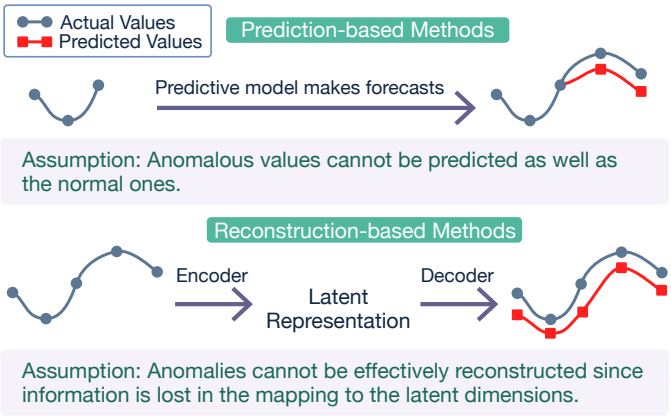


Fig. 1. A comparison between prediction-based and reconstruction-based methods for anomaly detection in time series data. Prediction-based methods learn a predictive model fitted to the given time series data. Reconstruction-based methods learn a model to capture the given time series data's latent structure and then reconstruct the data from their latent representations.

While the criteria differ across domains, anomalies in time series typically exhibit one of three identifiable patterns: point, contextual, or collective [4]. Point anomalies are singular data points that suddenly deviate from the normal range of the series. For example, a sensor malfunction is one common cause of point anomalies. Collective anomalies are a series of consecutive data points that are considered anomalous as a whole. Finally, contextual anomalies are groups of data points that fall within the series' normal range but do not follow expected temporal patterns.

Time series also exhibit unique properties that complicate anomaly detection. First, the temporality of time series implicates a correlation or dependence between each consecutive observation [11]. Second, the dimensionality of each observation influences the computational cost, imposing limitations on the modeling method. For example, modeling methods for multivariate datasets with more than one channel face the curse of dimensionality since they need to capture correlations between observations on top of temporal dependencies [3]. Third, noise due to minor sensor fluctuations during the process of capturing the signal can impact the performance [20]. The pre-processing stages must minimize noise to prevent models from confusing this noise with anomalies. Finally, time series are often non-stationary. They have statistical properties that change over time, like seasonality, concept drift, and change

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

points, that can easily be mistaken for anomalies.

Existing machine learning methods for anomaly detection on time series can be either prediction-based or reconstruction-based (Fig. 1). Prediction-based methods train a model to learn previous patterns in order to forecast future observations [6]. An observation is anomalous when the predicted value deviates significantly from the actual value. Prediction-based methods are good at revealing point anomalies but tend to produce more false detection [13]. On the other hand, reconstruction-based methods learn a latent low-dimensional representation to reconstruct the original input [6]. This method assumes that anomalies are rare events that are lost in the mapping to the latent space. Hence, regions that cannot be effectively reconstructed are considered anomalous. In our experiments, we observed that reconstruction-based methods tend to be more effective than prediction-based methods at identifying contextual and collective anomalies.

This paper proposes a new architecture – an auto-encoder with regression (AER) model – that leverages the successes and addresses the limitations faced by each method type. This architecture trains a reconstruction-based auto-encoder with a prediction-based regression component using a joint objective function. As a result, the model can produce both reconstruction-based and prediction-based anomaly scores (likelihood of an abnormal observation). This paper also explores several ways to calculate and combine scores to address several limitations of existing methods. Briefly, the contributions of this paper are as follows:

- We identified several successes and limitations of prediction-based and reconstruction-based methods using visualized examples.
- We propose a novel architecture – auto-encoder with regression (AER) – that leverages the successes of prediction-based and reconstruction-based methods for anomaly detection on time series data.
- We introduce the idea of masking anomaly scores created from the smoothing function to reduce start-of-sequence false-positive predictions. We applied masking to every baseline method and compared the method's performance to that of its unmasked counterpart.
- We present bi-directional anomaly scores, which combine prediction-based anomaly scores in the forward and reverse directions. This method addresses the limitation of missing forecasts faced by prediction-based methods.
- We demonstrated that AER outperformed five other baseline methods in anomaly detection on 12 time series datasets[2]. In addition, ablation studies show that the AER model achieved a 23.5% improvement in averaged F1 score compared to the baseline ARIMA model while retaining a runtime similar to its vanilla auto-encoder and LSTM regressor components.

The structure of the paper is as follows: Section II provides an overview of the existing pipeline and approaches for time series anomaly detection. Section III formally defines the problem, and Section IV documents the successes and limitations of existing methods. Section V introduces our solution, including the AER framework, smoothing function masking, and bi-directional scoring. Finally, Sections VI and VII evaluate the proposed framework, discuss the results, and summarize the key findings.

[2]Scripts to reproduce the results are available in https://github.com/sintel-dev/aer-paper

II. Related Work

A. Anomaly Detection Pipelines

Anomaly detection aims to find a set of anomalous intervals from either univariate or multivariate time series data. It is usually an unsupervised task due to the lack of labeled data. Recent work by Sintel [1] formalized this task as an end-to-end pipeline consisting of pre-processing, modeling, and post-processing stages. The pre-processing stage first transforms the raw data into suitable inputs for the models. The modeling stage then predicts or reconstructs the input to get the expected output. Finally, the post-processing stage finds discrepancies between the expected and real inputs. The methodology for finding these discrepancies significantly impacts the anomalies identified by this stage. Hence, our work focuses on the limitations in the post-processing stage for prediction-based and reconstruction-based methods. Understanding these limitations also enables us to make appropriate changes to the modeling stage.

B. Machine Learning-Based Approaches

Prediction-based approaches generally use the deviation between the predicted and actual values to identify anomalies. Autoregressive Integrated Moving Average (ARIMA) [15] and Long Short Term Memory Recurrent Neural Network with Non-parametric Dynamic Thresholding (LSTM-DT) [9] are well-known examples of prediction-based approaches. ARIMA uses lags and lagged forecast errors to predict future values. Statistical models like ARIMA require the user to have extensive domain knowledge about the time series data in order to adjust the parameters appropriately. Machine learning-based methods like LSTM-DT tend to require less domain knowledge. In the modeling stage, the method uses a separate LSTM neural network to model each channel in order to facilitate granular system control and mitigate errors from high-dimensionality outputs. In the post-processing stage, the method combines an exponentially-weighted average function with a non-parametric dynamic thresholding technique to detect anomalous intervals. Our work examines the limitations of the post-processing stage in LSTM-DT pipeline.

Reconstruction-based approaches learn a latent low-dimensional representation to reconstruct the original input. These methods assume that the latent space prioritizes capturing common patterns within the dataset. Rare events like anomalies are not captured in the latent representation and are less likely to be accurately reconstructed. Principal Component Analysis (PCA) [19], LSTM Auto-Encoders (LSTM-AE) [7], and LSTM Variational Auto-Encoders (LSTM-VAE) [14] are examples of reconstruction-based approaches. PCA

is a dimensionality-reduction technique limited to linear reconstructions and fails to leverage spatial-temporal correlation in multivariate settings. LSTM-AE is an auto-encoder built from LSTM layers that learns a latent space representation for the input. The size of the latent space needs to be calibrated to capture generalizable patterns while avoiding noise and anomalies. LSTM-VAE introduces regularization in the latent space using a probabilistic encoder and decoder. However, these methods tend to overfit to the training data, which results in decreased performances [6].

Generative Adversarial Network (GAN) is another reconstruction-based approach to address the overfitting issue. This form of adversarial learning offers regularization to the reconstruction errors. An early example is MAD-GAN [12], which uses spatial-temporal correlation and other dependencies among multiple variables to capture non-linear latent interactions. TadGAN [6] is another GAN-based approach trained with cycle consistency loss to address model instability issues and allow for better reconstruction of time series data. It also proposes several methods in the post-processing stage to calculate reconstruction-based anomaly scores. Similar to prediction-based methods, our work examines post-processing steps presented by TadGAN for reconstruction-based approaches.

Zhao et al. propose MTAD-GAT, a multivariate anomaly detection model that optimizes a joint loss of forecasting– and reconstruction–based models [23]. The architecture of MTAD-GAT differs from AER (our work) where MTAD-GAT is a graph attention network in comparison to AER that includes a bidirectional LSTM network. Moreover, Zhao et al. apply additional preprocessing steps to clean the data. Specifically, they apply Spectral Residual (SR) anomaly detection method [17] to filter out anomalous regions. In this work, we limit preprocessing to data scaling, imputing, and detrending. Furthermore, our approach still operates in an unsupervised setting where there is no prior knowledge about the anomalies in the dataset and no hyperparameter tuning, preventing information leakage. Lastly, we provide analysis to understand why the combination of prediction-based and reconstruction-based anomaly scores can be beneficial in predicting point and collective anomalies.

III. ML-Based Anomaly Detection Pipeline

Unsupervised time series anomaly detection aims to find a set of anomalous intervals given time series with one or more channels. Ideally, each interval captures an unexpected behavior that deviates from the expected patterns in the signal. This section first formulates the anomaly detection task into a sequence of steps (Fig. 2) similar to Alnegheimish et al.'s work [1] and then critically analyzes existing methods to learn their strengths and weaknesses.

A. Pre-processing Stage

The time series signal is pre-processed into inputs suitable for models similar to Geiger et al.'s work [6]. The time series t with d number of channels is divided into train and test splits. The train split is used to learn the parameters for subsequent

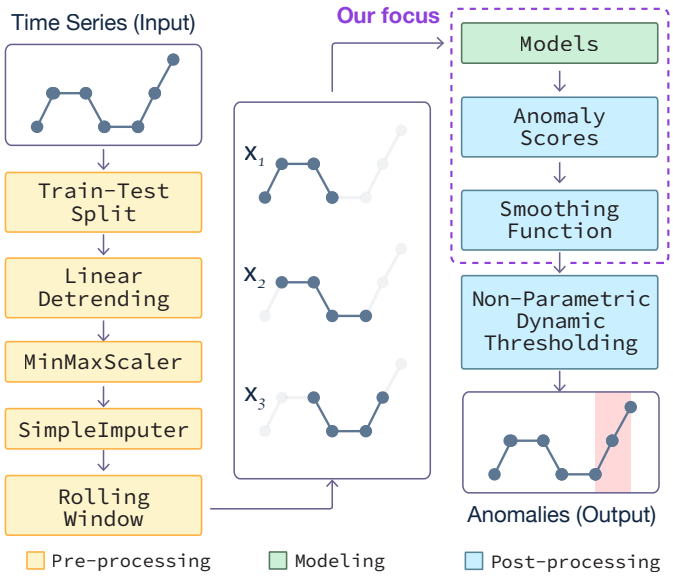


Fig. 2. The pipeline for anomaly detection on time series data consists of pre-processing, modeling, and post-processing stages. Our work focuses on models, anomaly scores, and the smoothing function steps of the pipeline.

transformations. Both splits are detrended, as necessary, by fitting and subtracting a least-square fit. Then, the values of each split are min-max normalized to the range `[-1, 1]`. Finally, any missing values are imputed with the mean. Let T be the total number of observations in the split without loss of generality. A rolling window with window size n and step size 1 creates $T - n$ number of inputs $\mathbf{x}_i = \{t_i, t_{i+1}, ..., t_{i+n-1}\}$ such that i represents the index of the first observation in the window. It is worth noting that pre-processing varies based on application scenarios, and the above summary only covers the most common steps.

B. Modeling Stage

The input and output depend on the type of anomaly detection model. Each input $\mathbf{x}_i \in \mathbb{R}^{n \times d}$ has n observations based on the window size (default to $n = 100$ for reconstruction-based models and $n = 250$ for prediction-based models) with d channels. In the case of multivariate inputs, separate models are trained for each channel to ensure traceability [9]. Usually, one channel is selected as the model's target channel. For example, many-to-one prediction-based models will produce single timestep predictions $f_i \in \mathbb{R}$ for index i of the target channel. On the other hand, many-to-one reconstruction-based models reconstruct the entire target channel and produce a sequence $y_{i:i+n-1} \in \mathbb{R}^n$ with the same starting index i as the input $\mathbf{x}_i$.

C. Post-processing Stage – Computing Anomaly Scores

The computation of anomaly scores differs between prediction-based and reconstruction-based models since they produce different outputs.

Prediction-based models produce a one-step forecast in the forward direction f_{i+n} at index $i + n$ given the input $\mathbf{x}_i$ starting at index i. Only forecasts f_i for indices $i \in [n+1, T]$ can be computed since prediction-based models require at least

Anomaly Scores	Limitations (L)		Successes (S)	
Prediction-based (P)	**PL1**	High anomaly scores at the early indices often result in false-positive predictions.	**PS1**	Prediction-based anomaly scores are better at capturing point anomalies than reconstruction-based anomaly scores.
	PL2	Low prediction-based anomaly scores for contextual anomalies with simple patterns result in false-negative predictions.		
	PL3	Missing prediction-based anomaly scores at the early indices result in false-negative predictions.		
Reconstruction-based (R)	**RL1**	Reconstruction-based anomaly scores reducing peaks for point anomalies result in false-negative predictions.	**RS1**	Reconstruction-based anomaly scores are better at capturing contextual and collective anomalies.
			RS2	Reconstruction-based DTW anomaly scores are better at capturing anomalies than AD and PD anomaly scores.

TABLE I
OVERVIEW OF SUCCESSES (S) AND LIMITATIONS (L) FOR PREDICTION-BASED (P) AND RECONSTRUCTION-BASED (R) METHODS.

n observations to forecast the first value at the index $n+1$. The absolute error between the sequence of forecasts in the forward direction f and the time series t creates the prediction-based anomaly score α_p as defined in Eq. (1).

$$\alpha_p(t, f) = \begin{cases} 0 & i \in [1, n+1) \\ |t_i - f_i| & i \in [n+1, T] \end{cases} \tag{1}$$

Reconstruction-based models reconstruct a sequence of values $y_{i:i+n-1}$ of one channel given the input $\mathbf{x}_i$ starting at index i. Each index i in the time series signal has multiple reconstructed values since that index occurs in multiple sequences of $y_{i:i+n-1}$. The median of the collection of reconstructed values is used as the final value for index i since using the median achieves better performance than using the mean [6]. Unlike prediction-based anomaly scores, reconstruction-based anomaly scores can be calculated for every index. The reconstruction-based anomaly scores can be calculated in three ways given sequences t and y: point-wise differencing, area differencing, or dynamic time warping.

Point-wise differencing (PD). The reconstruction-based PD anomaly score $\alpha_{r,p}$ defined in Eq. (2) takes the absolute error between the time series t and the reconstructed value y at every index i.

$$\alpha_{r,p}(t, y) = |t_i - y_i| \qquad i \in [1, T] \tag{2}$$

Area differencing (AD). The reconstruction-based AD anomaly score $\alpha_{r,a}$ defined in Eq. (3) is created using a fixed length window size that measures the similarity between local regions.

$$\alpha_{r,a}(t, y) = \frac{1}{2l} \left| \int_{i-l}^{i+l} t_i - y_i \, dt \right| \qquad i \in [1, T] \tag{3}$$

The similarity is measured as the average difference between areas beneath two curves of length $2l$ calculated using the trapezoidal rule ($l = 10$ by default).

Dynamic Time Warping (DTW). The reconstruction-based DTW anomaly score $\alpha_{r,d}$ defined in Eq. (4) created with dynamic time warping allows for many-to-many mapping between two sequences that are locally out of phase [2]. DTW creates a cost matrix $C \in \mathbb{R}^{2l \times 2l}$ such that each (i, j) coordinate represents the distance c_q between t_i and y_j.

$$\alpha_{r,d}(t, y) = \min_{C} \left[\frac{1}{Q} \sqrt{\sum_{q=1}^{Q} c_q} \right] \tag{4}$$

Dynamic programming solves for the optimal warp path C^* with the minimum warp distances between t and y.

Exponentially weighted moving average (EWMA) [10] with a smoothing window of $0.1T$ is applied to both prediction-based and reconstruction-based anomaly scores to reduce noise.

D. Post-processing Stage – Identifying Anomalous Sequences

Hundman et al. [9] used the locally adaptive thresholding function to identify anomalous intervals from the anomaly scores. This function uses a sliding window to compute local thresholds, merges continuous observations to create anomalous sequences, and mitigates false-positives by pruning anomalies.

Let α be the sequence of anomaly scores with a maximum size of length T (one score for each observation). The window size defaults to $\frac{T}{3}$ with a step size of $\frac{T}{3*10}$ to optimally identify anomalies. The adaptive threshold for each sliding window is four standard deviations from the window's mean. Observations with scores that exceed that threshold are identified as anomalous. Consecutive anomalous time steps are joined together to create K anomalous sequences. Hundman et al. [9] additionally employed a pruning method to reduce the number of false positives. Let $K^{(i)}_{max}$ represent the maximum anomaly score in each anomalous sequence $K^{(i)}$. The maxima are sorted in descending order, and the decrease percentage change $p^{(i)}$ is calculated between $K^{(i)}_{max}$ and $K^{(i+1)}_{max}$. At the sequence $K^{(j)}$ whose percentage change $p^{(j)}$ does not exceed an empirically defined threshold θ (defaults to 0.13), that sequence and all subsequent sequences are reclassified as normal, i.e., all sequences between `[j, K]`.

IV. CRITICAL ANALYSIS OF EXISTING METHODS

Despite some minor differences, most prediction-based (P) and reconstruction-based (R) methods follow the same course presented in Fig. 2. Both method types generally have their successes (S) and limitations (L), which are summarized in Table I. Fig. 3 and 4 illustrate the time series signal and the anomaly scores produced by each method in real-world datasets to understand each of their behavior better. In each figure, graph (a) shows the time series signal (blue) with the ground truth anomalous intervals (red). Graph (b) shows the prediction-based anomaly score α_p produced by ARIMA (orange) and LSTM-DT (sky blue) models. Graphs (c,d) correspond to the reconstruction-based PD $\alpha_{r,p}$ and DTW

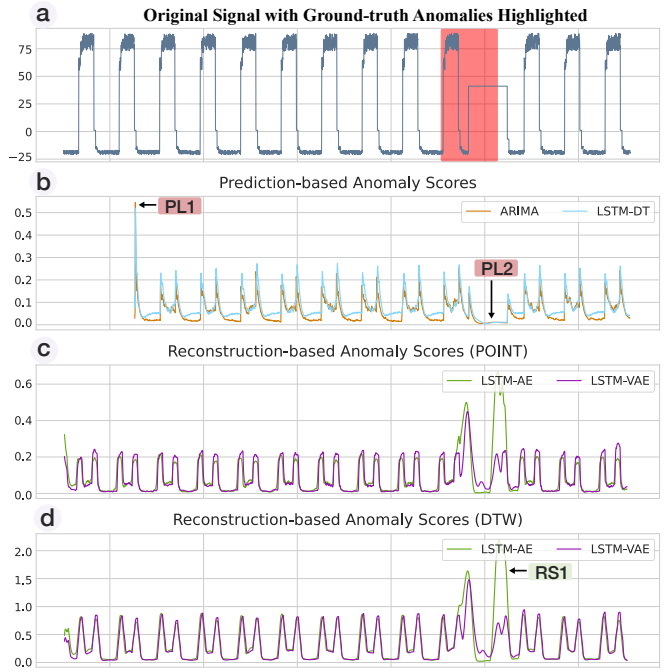


Fig. 3. Anomaly scores for the `art_daily_flatmiddle` signal from the `artificialWithAnomaly` dataset with one contextual anomaly. Prediction-based anomaly scores are high near the beginning (PL1) and low for contextual anomalies with simple patterns (PL2). On the other hand, all variations of reconstruction-based anomaly scores could capture the simple contextual anomaly (RS1).

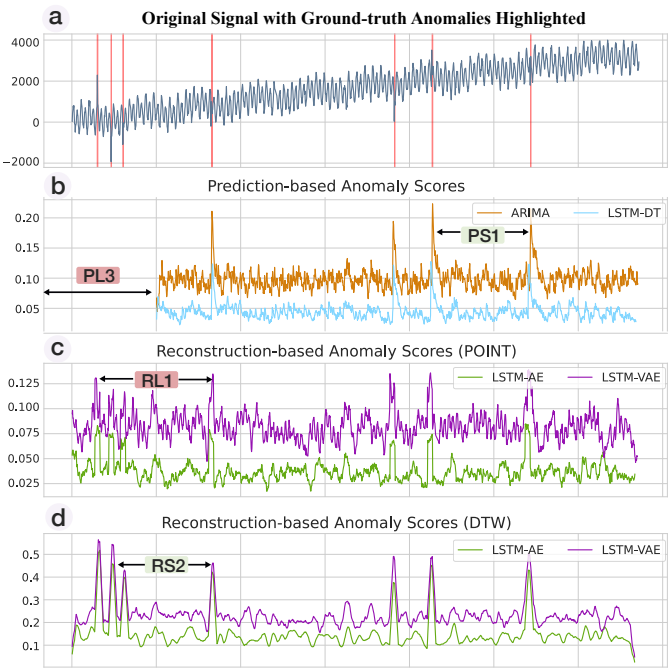


Fig. 4. Anomaly scores for the `A3Benchmark-TS11` signal from the `YAHOOA3` dataset with multiple point anomalies. Prediction-based anomaly scores are better at identifying point anomalies than reconstruction-based anomaly scores (PS1) but fail to find anomalies at the start of the signal (PL3). Of the variations in calculating reconstruction-based anomaly scores, DTW was the best at capturing the point anomalies (RS2).

$\alpha_{r,d}$ anomaly scores for LSTM-AE (green) and LSTM-VAE (purple) models.

PL1: High anomaly scores at the early indices often result in false-positive predictions. This error is likely the byproduct of using the exponential weighted moving average function to smooth the anomaly scores. The function requires at least the same number of observations as the size of the smoothing window before it can produce stable anomaly scores. While this limitation occurs in many signals, an example is seen in the prediction-based anomaly scores from the `art_daily_flatmiddle` signal (see PL1 in Fig. 3(b)).

PL2: Low prediction-based anomaly scores for contextual anomalies with simple patterns result in false-negative predictions. The cyclic pattern in prediction-based anomaly scores suggests that the models could not fully capture the structure, especially at the change point in the time series. However, in this case, the contextual anomaly is a simple pattern. Therefore, the models can easily forecast the pattern, resulting in nearly zero anomaly scores at the interval. Hence, the adaptive threshold failed to find the contextual anomaly (see PL2 in Fig. 3(b)).

PL3: Missing prediction-based anomaly scores at the early indices result in false-negative predictions (see PL3 in Fig. 4(b)). This limitation occurs only in prediction-based models since they require at least n observations to forecast the first value at index $n+1$. This behavior usually results in false-negative predictions for signals with anomalies occurring at the beginning, mainly from datasets like `YAHOOA3` with a decent number of point anomalies at the start of the time series.

PS1: Prediction-based anomaly scores are better at capturing point anomalies than reconstruction-based anomaly scores. For example, prediction-based anomaly scores showed more prominent peaks at anomalies than reconstruction-based anomaly scores for the `A3Benchmark-TS11` signal from the `YAHOOA3` dataset. As a result, the locally adaptive thresholding function can quickly identify anomalies using prediction-based anomalies, resulting in higher F1 scores for datasets like `YAHOOA3` with more point anomalies (see PS1 in Fig. 4(b)).

RL1: Reconstruction-based anomaly scores reducing peaks for point anomalies result in false-negative predictions. The reconstruction-based anomaly scores are calculated from the median of all predicted values for index i. Since some reconstructed outputs are better at capturing the point anomalies than others, the median value is closer to the true value at index i. This calculation lowers the anomaly scores such that the window-based threshold no longer captures those point anomalies, since the scores are now closer to the window's mean (see RL1 in Fig. 4(c)).

RS1: Reconstruction-based anomaly scores are better at capturing contextual and collective anomalies. For example, reconstruction-based anomaly scores from the `art_daily_flatmiddle` signal spiked while prediction-based anomaly scores remained close to zero at the contextual anomaly (see RS1 in Fig. 3(d)). This behavior occurs for prediction-based anomaly scores since the contextual anomaly pattern was easy to model. On the other hand, reconstruction-

based models struggled to recreate the entire interval, since the model tries to reconstruct values from simple anomalous intervals and complex non-anomalous intervals. The sudden shift from an intricate cyclic pattern to a simple pattern results in high reconstruction-based anomaly scores.

RS2: Reconstruction-based DTW anomaly scores are better at capturing anomalies than AD and PD anomaly scores. Reconstruction-based anomaly scores for the `A3Benchmark-TS11` signal show that reconstruction-based DTW anomaly scores are less noisy than reconstruction-based PD anomaly scores (see RS2 in Fig. 4(d)). The success of DTW scores is attributed to the method's ability to handle shifts in the alignment of two series. The ablation study by Geiger et al. [6] also reports that DTW slightly outperforms the other two reconstruction error types.

Our observations show that prediction-based and reconstruction-based anomaly scores have successes and limitations that complement one another. For example, we observe from our experiments that prediction-based anomaly scores have an easier time identifying point anomalies but produce relatively more false positives. On the other hand, reconstruction-based anomaly scores have an easier time identifying contextual and collective anomalies but produce relatively more false negatives. Therefore, our method strives to address these limitations and leverage strengths from both types of models as an alternative solution for anomaly detection in time series.

V. AER: Auto-Encoder with Regression

Our solution has three components targeting the models, anomaly scores, and smoothing function steps in the anomaly detection pipeline, as summarized in Fig. 5.

A. Modeling Stage

The AER model borrows ideas from LSTM-AE and LSTM-DT to produce prediction-based and reconstruction-based anomaly scores simultaneously. The goal is to combine the strengths of both types of methods while overcoming some of their limitations.

The input to the model is $\mathbf{x}_i \in \mathbb{R}^{n \times d}$ with n observations and d channels. Like other auto-encoder architectures, AER consists of an encoder and a decoder. While AER uses a regular encoder, the decoder reconstructs $n+2$ instead of n observations by increasing the number of units of the repeated vector layer by two. This minor change allows the model to create an output consisting of three components: the one-step reverse prediction $r_{i-1} \in \mathbb{R}$, the reconstructed sequence $y_{i:i+n-1} \in \mathbb{R}^n$, and the one-step-ahead prediction $f_{i+n} \in \mathbb{R}$.

The loss function (Eq. 5) is divided into prediction and reconstruction portions. The prediction loss V_{pred} is the average of the mean squared error between the pairs of true and prediction values in the reverse (t_{i-1}, r_{i-1}) and forward direction (t_{i+n}, f_{i+n}). Likewise, the reconstruction loss V_{rec} is the mean squared error between the time series $t_{i:i+n-1}$ and the reconstructed sequence $y_{i:i+n-1}$. The contribution of the prediction and reconstruction loss is determined by $\gamma \in [0, 1]$. The full objective function is defined as follows:

$$Loss = \frac{\gamma}{2} V_{pred}(t_{i-1}, r_{i-1}) + \frac{\gamma}{2} V_{pred}(t_{i+n}, f_{i+n}) + (1-\gamma) V_{rec}(t_{i:i+n-1}, y_{i:i+n-1}) \quad (5)$$

By default, the hyperparameters are $n = 100$ observations per input and $\gamma = 0.5$ to give equal importance to the prediction and reconstruction losses. One biLSTM layer with $b = 30$ units is used for both the encoder and decoder. The latent space is the same dimension as the last hidden state of the bidirectional LSTM layer, which is $2b$.

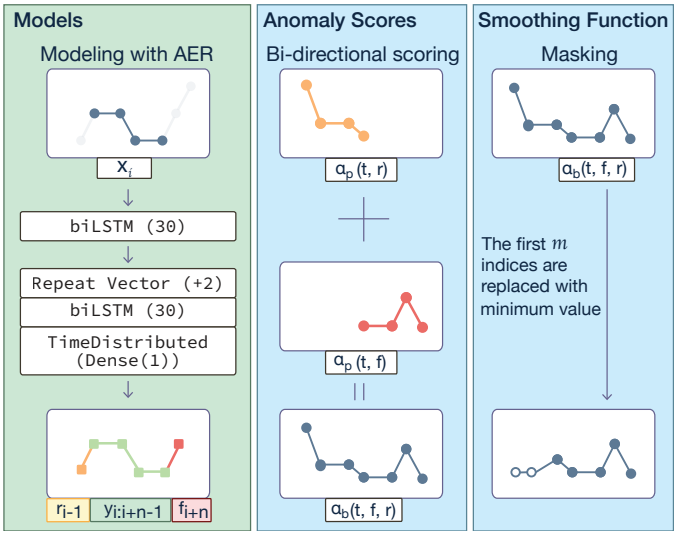


Fig. 5. Our solution targets three steps in the anomaly detection pipeline: models, anomaly scores, and smoothing function. AER is a joint model consisting of an LSTM auto-encoder and regressor capable of producing forecasts and reconstructing the input sequence. Second, bi-directional scoring combines prediction-based anomaly scores in the forward and reverse directions to address the issue of missing anomaly scores (PL3). Finally, masking in the smoothing step replaces the values of the anomaly scores at the first few indices to address start-of-sequence false-positive predictions (PL1).

B. Post-processing Stage: Masking

To overcome the false-positive predictions created from the exponential weighted moving average smoothing function (PL1), we introduce masking. The proposed solution is to mask m indices from the start of the sequence with some value. Our observations show that using the minimum anomaly scores as the masking value produced the best results. By default, m is equal to $0.01T$ (size of smoothing window) where T is the time series length.

C. Post-processing Stage: Bi-Directional Scoring

Bi-directional anomaly scores target the missing start of sequence anomaly scores since prediction-based methods require at least n observations to make the first forecast (PL3). A solution is to produce anomaly scores using the sequence of predictions in the forward direction f and in the reverse direction r. The anomaly scores created using r can fill in the missing prediction-based anomaly scores produced by f.

Datasets	Source	NASA		YAHOO				NAB					UCR
	Name	MSL	SMAP	A1	A2	A3	A4	Art	AdEx	AWS	Traffic	Tweets	UCR
Properties	Synthetic	No	No	No	Yes	Yes	Yes	Yes	No	No	No	No	Mix
	# Signals	27	53	67	100	100	100	6	5	17	7	10	250
# Anomalies	Point (*len=1*)	0	0	68	33	935	833	0	0	0	0	0	3
	Collective (*len>1*)	36	67	110	167	4	2	6	11	30	14	33	247
# Data Points	Anomalous Points	7766	54696	1669	466	943	837	2418	795	6312	1560	15651	49363
	Total Points	132046	562800	94866	142100	168000	168000	24192	7965	67644	15662	158511	19353766

TABLE II
HIGH-LEVEL OVERVIEW OF ALL 12 BENCHMARK DATASETS.

Again, let α_p denote the function to calculate prediction-based anomaly scores. Prediction-based anomaly scores are calculated in the forward direction $\alpha_p(t, f)$ for indices $i \in [n+1, T]$ and in the reverse direction $\alpha_p(t, r)$ for indices $i \in [1, T-n]$. If masking is used, then the first m values of $\alpha_p(t, f)$ are replaced with zeros, and the first m values of $\alpha_p(t, r)$ are replaced with $min(\alpha_p(t, r))$. Then, the scores $\alpha_p(t, f)$ are padded with n zeros in the beginning while $\alpha_p(t, r)$ are padded with n zeros at the end to align the anomaly scores.

$$\alpha_b(t, f, r) = \begin{cases} \alpha_p(t, r) & i \in [1, n+m+1) \\ \frac{1}{2}\alpha_p(t, r) + \frac{1}{2}\alpha_p(t, f) & i \in [n+m+1, T-n+1) \\ \alpha_p(t, f) & i \in [T-n+1, T] \end{cases} \quad (6)$$

The bi-directional anomaly scores α_b defined in Eq. (6) consist of averages of both scores in overlapping intervals and the max between both scores in non-overlapping intervals.

D. Post-processing Stage: Combination Scores

The bi-directional prediction-based anomaly scores α_b and reconstruction-based anomaly errors α_r can be used to create the combined anomaly scores α_c.

1) Prediction-based Only (PRED): The combined anomaly scores α_c are calculated using only the bi-directional prediction-based anomaly scores.

$$\alpha_c(t, r, y, f) = \alpha_b(t, f, r). \quad (7)$$

2) Reconstruction-based Only (REC): The combined anomaly scores α_c are calculated using only the reconstruction-based anomaly scores. The calculation of reconstruction-based anomaly scores defaults to using DTW since it outperforms reconstruction-based PD and AD (RS2).

$$\alpha_c(t, r, y, f) = \alpha_{r,d}(t, y). \quad (8)$$

3) Convex (SUM): The combined anomaly scores α_c are calculated using a convex combination with parameter weight β that controls the two errors' relative importance (by default $\beta = 0.5$). Both prediction-based and reconstruction-based anomaly scores are min-max scaled to between `[0, 1]` before the combination.

$$\alpha_c(t, r, y, f) = (1-\beta)\alpha_{r,d}(t, y) + \beta\alpha_b(t, f, r). \quad (9)$$

4) Product (MULT): The combined anomaly scores α_c are calculated using a point-wise product between the two scores to emphasize both scores' high values. β controls the relative importance of the two errors (by default $\beta = 1$). Both prediction-based and reconstruction-based anomaly scores are min-max scaled to between `[1, 2]` before the combination.

$$\alpha_c(t, r, y, f) = \beta\alpha_{r,d}(t, y) \odot \alpha_b(t, f, r). \quad (10)$$

VI. EXPERIMENTAL RESULTS

The three main points we seek to validate in our experimental study are as follows:

- **RQ1**: Does the AER framework enable us to discover anomalies more efficiently than we can through other approaches?
- **RQ2**: What is the impact of smoothing function masking and bi-directional scoring on anomaly detection?
- **RQ3**: Do mixture anomaly scores offer additional information compared to using either a prediction-based or reconstruction-based anomaly score on its own?

A. Data Sources

We use 12 datasets (742 signals) spanning various domains to evaluate the models' generalizability and adaptability. The National Aeronautics and Space Administration (NASA) provided two spacecraft telemetry datasets[3]: Soil Moisture Active Passage (SMAP) and Mars Science Laboratory (MSL) acquired from a satellite and a rover, respectively [9]. Each numeric measurement in the target channel is accompanied by one-hot encoded information about commands sent or received by specific spacecraft modules in a given time window. The Yahoo Webscope Program provided the S5 datasets[4] consisting of one set of real production traffic to Yahoo properties (A1) and three synthetic datasets (A2, A3, A4) with varying trends, noise, and pre-specified or random seasonality. The A2 and A3 datasets only contain outliers inserted at random positions, while A4 has outliers and change points. The Numenta Anomaly Benchmark (NAB) provided several datasets[5] from various domains: artificialWithAnomaly (Art), realAdExchange (AdEx), realAWSCloudwatch (AWS), realTraffic (Traffic), realTweets (Tweets). The UCR Time Series Anomaly Archive[6] is a dataset created to address flaws like triviality, unrealistic anomaly density, mislabeled ground truth, and run-to-failure bias faced by popular datasets [22].

[3]NASA data: https://github.com/khundman/telemanom/

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

[5]NAB data: https://github.com/numenta/NAB

[6]UCR data: https://www.cs.ucr.edu/~eamonn/time_series_data_2018/UCR_TimeSeriesAnomalyDatasets2021.zip

Models	NASA		YAHOO				NAB					UCR	Avg. F1 ($\mu \pm \sigma$)
	MSL	SMAP	A1	A2	A3	A4	Art	AdEx	AWS	Traffic	Tweets	UCR	
ARIMA	0.442	0.333	0.733	0.807	0.818	0.700	0.353	0.518	**0.741**	0.500	0.567	0.124	0.553 ± 0.21
LSTM-DT	0.515	0.707	0.721	**0.980**	0.744	0.638	0.400	0.513	**0.741**	**0.667**	0.580	0.391	0.633 ± 0.16
LSTM-AE	0.500	0.705	0.610	0.866	0.420	0.253	0.545	**0.750**	0.692	0.457	0.483	0.314	0.550 ± 0.17
LSTM-VAE	0.526	0.653	0.575	0.823	0.432	0.240	**0.667**	0.700	0.643	0.483	**0.590**	0.317	0.554 ± 0.16
TadGAN	**0.584**	0.617	0.533	0.842	0.391	0.297	0.571	0.677	0.720	0.581	0.588	0.162	0.547 ± 0.18
AER*	0.541	**0.772**	**0.772**	0.959	**0.896**	**0.722**	0.615	0.635	0.621	0.606	0.585	**0.470**	**0.683 ± 0.14**

TABLE III
F1 SCORES FOR AER COMPARED TO PREDICTION-BASED AND RECONSTRUCTION-BASED BASELINE MODELS. THE HIGHEST SCORES ARE HIGHLIGHTED IN DARK GREEN, WHILE THE LOWEST SCORES ARE HIGHLIGHTED IN DARK RED PER DATASET.

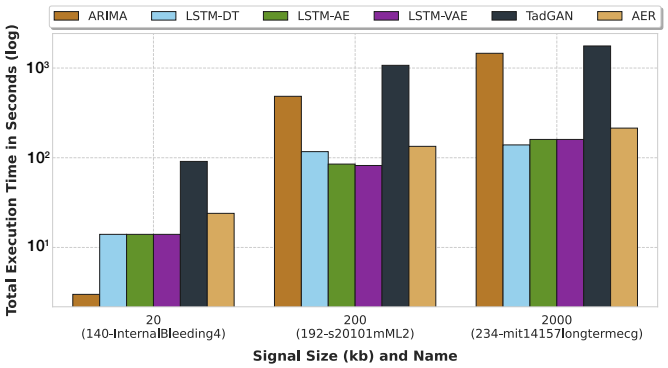


Fig. 6. A comparison between the total execution time in seconds on signals from the UCR dataset with increasing sizes. The total execution time consists of the time to train the model pipeline (training time) and the time to convert an input into an output (pipeline latency).

Similar to Geiger et al. [6], Table II summarizes basic information about each dataset. It differentiates between real and synthetic datasets and provides the number of signals and anomalies for each dataset. Each anomaly is classified as either point or collective, depending on the length of the anomaly. Lastly, the total number of anomalous and overall data points are provided for each dataset.

B. Evaluation Metrics

Like Hundman et al. [9] and Geiger et al. [6], the metric used in this study is unweighted contextual F1 scores for each dataset. The motivation is that anomalies are rare and window-based in many real-world application scenarios. The end user's goal is to detect timely true alarms without receiving many false positives. Hence, this evaluation metric is preferable since it prioritizes finding any part of the anomalies. Anomaly scoring is based on overlapping segments: a true positive (TP) if a known anomalous window overlaps any detected windows, a false negative (FN) if a known anomalous window does not overlap any detected windows, and a false positive (FP) if a detected window does not overlap any known anomalous region.

We performed all experiments in an instance of MIT Supercloud [18] with an Intel Xeon Gold 6249 processor, 10 CPU cores, 9 GB RAM per core, and 1 Nvidia Volta V100 GPU. The environment is created using the *anaconda/2022a* module, which includes TensorFlow 2.0. All models are implemented as primitives and benchmarked using Orion [6].

C. Baseline Models

We compare our solution against the following five state-of-the-art methods:

ARIMA *(Prediction-based)*: Autoregressive Integrated Moving Average [15] is implemented with the `StatsModels` library. The hyperparameters are empirically set to `p=1, d=0, q=0`.

LSTM-DT *(Prediction-based)*: LSTM non-parametric Dynamic Threshold [9] uses two LSTM layers with 80 units and a dropout rate of 0.3. The training hyperparameters were: 35 epochs, batch size of 64, and Adam optimizer.

LSTM-AE *(Reconstruction-based)*: LSTM auto-encoders [7] use one LSTM layer with 60 units for the encoder and generator. A time-distributed layer with a dense one-unit layer is used to create the output.

LSTM-VAE (Reconstruction-based): LSTM variational auto-encoders [14] consist of an encoder and a decoder. The encoder uses one shared LSTM layer with 60 units and separate dense layers, each with 60 units, to create the mean and standard deviation vector. The decoder uses a repeat vector layer, an LSTM layer with 60 units, and a time-distributed layer with a dense one-unit layer.

TadGAN *(Reconstruction-based)*: TadGAN [6] consists of an encoder and generator that use bi-directional LSTM layers, and critics that use 1D convolution layers. The reconstruction-based anomaly scores can be used in combination with the critic scores to create the final anomaly scores. Geiger et al. [6] reported an ablation study merging these scores using summation, product, critic-only, and reconstruction-only combinations.

D. Benchmarking Results

AER outperforms baseline models based on averaged F1 scores (RQ1). Table III shows that AER has an averaged F1 score of 0.683, which is 23.5% higher than the score of the standard ARIMA model. The flexibility of combining prediction-based and reconstruction-based anomaly scores leads to an improvement in F1 scores across the datasets. The graph in Fig. 6 shows the runtime of AER scales in the same order as LSTM-DT, LSTM-AE, and LSTM-VAE. While the runtime is slighter higher for AER than for those models, this is a very reasonable computation cost considering the performance increase.

AER v.s. TadGAN (RQ1). Similarly, Table III shows that AER outperforms TadGAN by 24.9% in terms of averaged

A: Masking and Bi-Directional Scoring Comparison

Models	NASA		YAHOO				NAB					UCR	Avg. F1 ($\mu \pm \sigma$)
	MSL	SMAP	A1	A2	A3	A4	Art	AdEx	AWS	Traffic	Tweets	UCR	
ARIMA	0.442	0.333	0.733	0.807	**0.818**	**0.700**	0.353	0.518	0.741	0.500	0.567	0.124	0.553 ± 0.21
ARIMA (M)*	**0.457**	**0.359**	**0.752**	**0.809**	0.807	0.690	**0.500**	0.518	**0.769**	0.500	**0.576**	**0.148**	**0.574 ± 0.19**
LSTM-DT	0.515	0.707	0.721	0.980	0.744	0.638	0.400	0.513	0.741	0.667	0.580	0.391	0.633 ± 0.16
LSTM-DT (M)*	**0.521**	**0.754**	0.729	**0.987**	0.734	0.638	**0.600**	0.513	0.769	**0.686**	**0.588**	**0.446**	0.664 ± 0.14
LSTM-DT (M, Bi)*	0.505	0.662	**0.755**	0.949	**0.895**	**0.792**	0.545	0.488	**0.786**	0.684	0.587	0.432	**0.673 ± 0.16**
LSTM-AE	0.500	**0.705**	0.610	0.866	0.420	**0.253**	0.545	0.750	**0.692**	0.457	0.483	0.314	0.550 ± 0.17
LSTM-AE (M)*	**0.522**	0.701	**0.644**	**0.882**	**0.442**	0.236	**0.667**	0.750	0.609	**0.533**	**0.542**	**0.334**	**0.572 ± 0.17**
LSTM-VAE	**0.526**	0.653	0.575	0.823	0.432	0.240	**0.667**	0.700	**0.643**	0.483	0.590	0.317	0.554 ± 0.16
LSTM-VAE (M)*	0.521	**0.710**	**0.628**	**0.901**	**0.460**	**0.246**	0.545	**0.764**	0.615	**0.519**	0.590	**0.333**	**0.569 ± 0.17**
TadGAN	0.584	0.617	0.533	0.842	0.391	**0.297**	0.571	0.677	0.720	0.581	0.588	0.162	0.547 ± 0.18
TadGAN (M)*	0.584	**0.630**	**0.534**	**0.846**	**0.395**	0.291	**0.615**	0.677	0.720	0.581	0.588	**0.164**	**0.552 ± 0.18**

B: Ablation Study

Models	NASA		YAHOO				NAB					UCR	Avg. F1 ($\mu \pm \sigma$)
	MSL	SMAP	A1	A2	A3	A4	Art	AdEx	AWS	Traffic	Tweets	UCR	
AER (PRED)*	0.494	0.685	0.705	0.923	**0.896**	**0.722**	0.500	0.541	0.688	**0.615**	0.556	0.461	0.649 ± 0.14
AER (SUM)*	0.488	0.680	0.714	0.936	0.719	0.553	0.500	**0.702**	**0.733**	0.606	0.559	0.428	0.635 ± 0.13
AER (REC)*	0.500	0.683	0.707	**0.985**	0.620	0.416	0.444	0.644	0.692	0.571	0.519	0.363	0.595 ± 0.16
AER (MULT)*	**0.541**	**0.772**	**0.772**	0.959	0.752	0.572	**0.615**	0.635	0.621	0.606	**0.585**	**0.470**	**0.658 ± 0.13**

TABLE IV
F1 SCORES FOR MASKING, BI-DIRECTIONAL SCORING, AND ABLATION STUDY. ALL NEW METHODS ARE MARKED WITH *. THE VARIATION WITH THE HIGHEST SCORE IS IN **BOLD** FOR EACH MODEL TYPE. THE FINAL AER MODEL USES THE GREEN FILL SCORES.

F1 scores while requiring less execution time (see Fig. 6). This result suggests that combining prediction-based with reconstruction-based anomaly scores could lead to better F1 scores than combining critic-based with reconstruction-based anomaly scores.

Masking improves averaged F1 scores slightly (RQ2). Table IV-A shows that masking scores improved averaged F1 scores by 4.3%, on average, for prediction-based methods and 2.6%, on average, for reconstruction-based methods. Masking anomaly scores benefited prediction-based methods more than reconstruction-based methods since those methods tend to make more false-positive predictions. However, masking may remove anomalies at the start of the signal and hurt model performance on datasets like `YAHOOA3` and `YAHOOA4`.

Bi-directional scoring greatly improves F1 scores on some datasets (RQ2). LSTM-DT (M, Bi) consists of two separate LSTM-DT models trained on the sequence in the forward and reversed direction respectively. Table III-A shows that using bi-directional scoring with LSTM-DT (M, Bi) improved F1 scores by 20.3% for the `YAHOOA3` dataset and 24.1% for the `YAHOOA4` dataset compared to LSTM-DT. These datasets have signals with point anomalies at the beginning that uni-directional prediction-based models cannot predict. However, bi-directional scoring may negatively impact the performance of models on other datasets. Since prediction-based methods tend to produce false-positive predictions, filling in anomaly scores missed by prediction-based anomaly scores allows for more opportunities to produce false positives.

E. Ablation Study

Product (MULT) combination of anomaly scores have the highest averaged F1 score across all combination methods (RQ3). The product (MULT) combination of prediction-based and reconstruction-based anomaly scores produced the highest F1 scores on 6 of 12 datasets (see Table IV-B). Most of these datasets were non-synthetic, including `MSL`, `SMAP`, `YAHOOA1`, and `Tweets`. This combination method outperformed the convex (SUM) combination by 3.7%, the reconstruction-based only (REC) combination by 10.6%, and the prediction-based only (PRED) combination by 1.5% in terms of averaged F1 scores. Additionally, excluding `YAHOOA3` and `YAHOOA4` synthetic datasets with many point anomalies result in an averaged F1 score of 0.658 for the product (MULT) combination, a 6.6% increase compared to 0.617 for prediction-based only (PRED) combination. These results support the idea that mixture anomaly scores offer more information than reconstruction-based anomaly scores in general and prediction-based anomaly scores in cases other than identifying point anomalies.

Prediction-based only (PRED) anomaly scores perform better on datasets with mostly point anomalies. Bi-directional scoring produced the highest F1 scores on datasets like `YAHOOA3` and `YAHOOA4` with mostly point anomalies (see Table IV-B). This finding is consistent with our findings in the LSTM-DT (M, Bi) model.

The selection of the combination method for each dataset is based on the use case. We recommend that users default to using product (MULT) anomaly scores and using prediction-based only (PRED) scores when users primarily want to

identify point anomalies. The AER model reports the F1 scores of AER (PRED) for the YAHOOA3 and YAHOOA4 datasets with mostly point anomalies and AER (MULT) for the other datasets, even though they might not be the best combination method according to the ablation study. In practice, datasets come without labels since anomaly detection is an unsupervised problem. Hence, it is impossible to retroactively tune the best method to calculate anomaly scores for each dataset.

F. Limitations and Discussion

While product mixture scores offer unique insights for anomaly detection, several ways exist to improve the AER framework. For example, the model architecture could be better since our study uses a vanilla auto-encoder architecture with one biLSTM layer for both the encoder and decoder. Our framework is designed to easily extend to any reconstruction-based method with minimum changes to the objective function. Another improvement involves experimenting with the γ (defaults to 0.5), which controls the contribution of prediction and reconstruction loss to the objective function. An optimal γ could lead to more accurate prediction-based and reconstruction-based anomaly scores that ultimately improve F1 scores. Lastly, the findings in our analysis of existing methods in section IV are for datasets we are currently investigating. The identified constraints may not always hold in other datasets.

Although researchers pay increasing attention to building more powerful models to improve the accuracy of prediction-based and reconstruction-based methods, we would like to call for more attention to the post-processing stage. Our study demonstrated that changes in the post-processing stage could significantly improve performances in addition to our proposed model. Future exploration directions could include additional methods to create mixture scores and better heuristics for the selection of such methods (e.g., between PRED and MULT) for each signal.

VII. Conclusion

This study analyzed the successes and limitations of existing reconstruction-based and prediction-based methods. We proposed a threefold solution to address existing limitations: (1) the AER framework that leverages the successes of prediction-based and reconstruction-based methods, (2) masking anomaly scores to reduce start-of-sequence false-positive predictions, and (3) bi-directional scoring to address missing forecast issues. In addition, we conducted an ablation study to test several ways of combining prediction-based and reconstruction-based anomaly scores. Our results showed that (1) AER has the highest F1 score averaged across 12 datasets, (2) masking and bi-directional scoring improve F1 scores given the right conditions, (3) the product combination (MULT) of bi-directional and reconstruction-based anomaly scores produces better results, on average, for datasets with mostly collective anomalies. Finally, the code is available at https://github.com/sintel-dev/Orion.

References

[1] S. Alnegheimish, D. Liu, C. Sala, L. Berti-Equille, and K. Veeramachaneni. Sintel: A machine learning framework to extract insights from signals. In *SIGMOD'22*. ACM, jun 2022.

[2] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, page 359–370, 1994.

[3] R. Chalapathy and S. Chawla. Deep learning for anomaly detection: A survey. *CoRR*, abs/1901.03407, 2019.

[4] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009.

[5] F. Cheng, D. Liu, F. Du, Y. Lin, A. Zytek, H. Li, H. Qu, and K. Veeramachaneni. Vbridge: Connecting the dots between features and data to explain healthcare models. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):378–388, 2021.

[6] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni. Tadgan: Time series anomaly detection using generative adversarial networks. *CoRR*, abs/2009.07769, 2020.

[7] R.-J. Hsieh, J. Chou, and C.-H. Ho. Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing. In *IEEE SOCA'19*, pages 90–97, Nov 2019.

[8] D. Huang, D. Mu, L. Yang, and X. Cai. Codetect: Financial fraud detection with anomaly feature detection. *IEEE Access*, 6:19161–19174, 2018.

[9] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. *ACM SIGKDD'18*, Jul 2018.

[10] J. S. Hunter. The exponentially weighted moving average. *Journal of Quality Technology*, 18(4):203–210, 1986.

[11] J. Kusuma, L. Doherty, and K. Ramchandran. Distributed compression for sensor networks. In *Proceedings 2001 International Conference on Image Processing*, volume 1, pages 82–85 vol.1, 2001.

[12] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S. Ng. MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks. *CoRR*, abs/1901.04997, 2019.

[13] T. Li, M. L. Comer, E. J. Delp, S. R. Desai, J. L. Mathieson, R. H. Foster, and M. W. Chan. Anomaly scoring for prediction-based anomaly detection in time series. In *2020 IEEE Aerospace Conference*, pages 1–7, 2020.

[14] D. Park, Y. Hoshi, and C. C. Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *CoRR*, abs/1711.00614, 2017.

[15] E. H. M. Pena, M. V. O. de Assis, and M. L. Proença. Anomaly detection using forecasting methods arima and hwds. In *Proceedings 2013 SCCC*, pages 63–66, 2013.

[16] J. Pereira and M. Silveira. Learning representations from healthcare time series data for unsupervised anomaly detection. In *IEEE BigComp'19*, pages 1–7, 2019.

[17] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang. Time-series anomaly detection service at microsoft. In *ACM SIGKDD'19*, pages 3009–3017, 2019.

[18] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Houle, M. Hubbell, M. Jones, A. Klein, L. Milechin, J. Mullen, A. Prout, A. Rosa, C. Yee, and P. Michaleas. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *IEEE HPEC'18*, pages 1–6. IEEE, 2018.

[19] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of pca for traffic anomaly detection. *SIGMETRICS Perform. Eval. Rev.*, 35(1):109–120, jun 2007.

[20] V. P. Tuzlukov and Cheng. *Signal Processing Noise*. CRC Press, Inc., USA, 2002.

[21] J. Verbesselt, A. Zeileis, and M. Herold. Near real-time disturbance detection using satellite image time series. *Remote Sensing of Environment*, 123:98–108, 2012.

[22] R. Wu and E. Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021.

[23] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang. Multivariate time-series anomaly detection via graph attention network. In *ICDM'20*, pages 841–850. IEEE, 2020.