Enhancing a Deep Learning Model for Anomaly Detection in Time-Series Data Using Attention Networks

Authors Names

Zahraa Modher Nabat[a]

Publication data: 14 /11 /2026

Keywords: anomaly detection, time series, attention mechanism, Transformer, deep learning, autoencoder, unsupervised learning.

ABSTRACT

Anomaly detection in time-series data is essential for monitoring industrial machinery and computing systems, yet labelled anomalies are scarce and real signals are often periodic and non-stationary. This paper presents the Attention-based Temporal Reconstruction Network (ATRN), an unsupervised Transformer encoder–decoder with a low-dimensional window bottleneck that prevents identity mapping, and an overlap-averaging scheme that converts window reconstruction errors into point-level anomaly scores. ATRN was evaluated on seven real-world series from the Numenta Anomaly Benchmark (realKnownCause) against Isolation Forest, an LSTM autoencoder and an attention-free ablation, under two label-free training protocols and three random seeds, using ROC-AUC, PR-AUC and best F1 without point adjustment. When trained on a clean initial segment, ATRN achieved the highest mean ROC-AUC among the neural models (0.695 vs. 0.660 for the LSTM autoencoder), and attention improved ROC-AUC on the most periodic series by 0.091. However, Isolation Forest obtained the best overall accuracy (mean ROC-AUC 0.750), training-set contamination removed the advantage of attention, and the learned attention maps were largely diffuse. The results indicate that attention is most useful when anomalies disrupt regular temporal patterns, and they underline the importance of strong baselines.

1. Introduction

The problem of time series anomalies is defined as identifying the points or sequences of points that diverge from the normal pattern of a particular system under consideration [4], [6]. It is crucial for many domains ranging from predictive maintenance of machines, monitoring of cloud systems, detecting fraud transactions in financial domain, and safety of telemetry data in spacecrafts [16], [28]. In this regard, overlooking of anomalies may lead to costly downtimes and dangerous scenarios, while the large amount of false positive alarms would rapidly decrease operators' trust. Thus, constructing effective detectors becomes a challenging task.

These three factors make the problem challenging. Firstly, labelled anomalies are rare and costly to acquire, meaning that any practical anomaly detector must be designed in an unsupervised way using potentially contaminated data [16], [20]. Secondly, the signals observed in practice are non-stationary, seasonal and subject to regime changes, meaning that an observation can be considered an anomaly only in the context of its temporal neighborhood. Thirdly, the anomalies themselves are heterogeneous, and can occur as spikes, level shifts, smooth drifts or collective anomalies of up to several hundred time steps [4].

The simplicity of classical statistical methods and distance-based methods like the Local Outlier Factor and Isolation Forest, on the other hand, is tempting; however, they do not consider temporal dependency because they see windows as unordered feature vectors [5], [13]. The Recurrent Neural Network (RNN), specifically the Long Short-Term Memory Network (LSTM network) [8], provided

Registration and Students Affairs, University of Babylon, Iraq, Email:zahraa.modher@uobabylon.edu.iq

an explicit model for temporal context and came to dominate the deep learning approach in both prediction-based and reconstruction-based detection [9], [14], [15]. However, due to the sequential processing of sequence data in recurrent networks, parallel training is difficult.

Both the attention mechanism [3] and the Transformer, developed based on it [24], overcome these issues by enabling each time step to interact directly with any other one in the window. Attention networks have proven to be successful in time series prediction and representation learning [25], [29], [31], and have also been applied to anomaly detection [23], [27]. However, several important gaps still exist. First, many of the works evaluate their methods using point adjustment procedure, which has been proved to lead to an overestimation of the scores, such that even random anomaly detectors score well [10]. Some popular datasets used for evaluation have also been accused of containing trivial or wrongly labelled anomalies [26]. Furthermore, most of the works lack a theoretically sound explanation of the use of attention in temporal anomaly detection and a reproducible experiment with appropriate baselines.

The current research seeks to fill in these gaps through design, implementation, and evaluation of an attention-based deep learning approach to anomaly detection in univariate time series, which is unsupervised. Main contributions include:

(1) A theoretical framework that treats anomaly detection as a reconstruction problem using a learnt model of normal behaviour and describes how multi-head self-attention captures short-term and long-term dependencies within windows.

(2) A compact autoencoder based on Transformer architecture, called Attention-based Temporal Reconstruction Network (ATRN), with a window-level bottleneck which prevents the identity mapping of inputs and overlap-averaged loss function that turns window-level errors into point-level anomalies.

(3) Reproducible empirical study of the proposed approach using seven real-world time series from the Numenta Anomaly Benchmark (NAB) realKnownCause dataset [1], [12] in comparison with Isolation Forest, LSTM-based autoencoder, and ATRN with disabled attention layer under two different training scenarios using threshold-free metrics and without point-wise adjustments.

(4) Discussion of when attention works and when it does not work and insights about learnt attention maps.

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 presents the theoretical framework and the proposed model. Section 4 describes the experimental setup, Section 5 reports and discusses the results, and Section 6 concludes the paper.

2. Related Work

2.1 Classical and machine-learning approaches

The early taxonomy classifies anomaly detection techniques into statistical, distance-based, density-based, and model-based categories [4], [6]. Density-based approaches like LOF measure the anomaly score of an example based on the difference between the local density of the example and that of its neighbors [5]. The Isolation Forest creates a forest of isolation trees and utilizes the insight that anomalies are isolated with fewer splits compared to normal examples [13]. These approaches can be

extended for use with time series through sliding windows in which the signal is embedded, yet they do not take into account the order of values within a window. Industrial approaches have leveraged spectral residual transformations in combination with convolutional neural networks to monitor massive numbers of metrics in production settings [19]. It is evident from extensive comparison studies that there is no single winning approach among classical and deep methods. [20].

2.2 Recurrent and generative deep models

Deep learning detectors are classified as either forecasting-based and reconstruction-based approaches [16], [28]. Under the forecasting approach, a model predicts future values, while the prediction error becomes the anomaly score. For instance, Malhotra et al. [14] applied stacked LSTM predictors in this regard, and Hundman et al. [9] combined LSTM forecasting and non-parametric dynamic thresholding for spacecraft telemetry monitoring. With reconstruction approaches, an encoder transforms a window into its latent representation, which is then reconstructed by the decoder, and the windows unable to reconstruct are anomalies. A good example in this case is an LSTM encoder-decoder presented by Malhotra et al. [15]. Probabilistic deep learning detectors learn the distribution of normal behavior, where OmniAnomaly combines recurrent cells with a stochastic latent space [22], DAGMM learns an autoencoder and Gaussian mixture model together [32], and USAD employs adversarial training of autoencoders for reconstruction error augmentation [2]. Graph-based approaches like MTAD-GAT [30] and GDN [7] apply the mechanism of attention to variable relationships in multivariate time series.

2.3 Attention and Transformer models

Attention mechanism was designed to help a decoder concentrate on the most informative components of an input sequence [3]; later, the Transformer model abandoned all recurrences and adopted solely stacks of self-attention modules [24]. In time series domain, Transformers were utilized for long-horizon forecasting via sparse attention [31], for unsupervised representation learning via masked-value pretraining [29] and surveyed in the recent paper [25]. Concerning anomaly detection, the Anomaly Transformer evaluates the difference between the learned prior association and the observed series association [27], while TranAD integrates the Transformer encoder and adversarial, self-conditioned learning [23]. Although both methods exhibit good performance, a large part of their reported advantage is gained using point-adjusted evaluation [10]; some of the utilized datasets suffer from quality problems [26]. Thus, this work adopts a compact architecture, performs ablation of the attention mechanism and follows an evaluation protocol that is not affected by those flaws.

3. Theoretical Framework and Proposed Model

3.1 Problem formulation

Let X=(x_1, x_2, …, x_T) be a univariate time series where $x_t \in R$. Our objective is to score each time step with an anomaly score s_t such that the anomaly time steps score higher than non-anomaly time steps. There are no labeled data available for training; however, some anomalies might be present in the training data as well. The time series is standardized to have a zero mean and unit variance and is divided into overlapping windows of length w:

$$\boldsymbol{x}^{(i)} = (x_i, x_{i+1}, \dots, x_{i+w-1}), \qquad i = 1, \dots, T - w + 1. \qquad (1)$$

Reconstruction-based detection works on the premise that the model trained for window compression and reconstruction will learn the prevalent and normal patterns of the system. Since anomalies are rare and varied, the model will not have enough representation for these, and therefore they will be poorly reconstructed [15], [16].

3.2 Self-attention as a model of temporal context

The mapping from each window to a series of d dimensional tokens $Z \in R^{w \times d}$ is done using a linear embedding. Sinusoidal positional encoding P is used to enable the model to determine the temporal ordering of each time step [24]. The scaled dot product attention mechanism then determines the weighted sum of all time steps within the window:

$$Attention(Q,K,V) = softmax\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V, \qquad Q = ZW_Q,\ K = ZW_K,\ V = ZW_V. \tag{2}$$

Each row of the matrix A = softmax($QK^{\top}/\sqrt{d_k}$) corresponds to a different time step, and each row is a probability distribution over the window. This means that the matrix is a learned and data-specific model of temporal context, where a step on a normal cycle will be able to attend to the same phase in the window while a step after a level shift will attend to its neighbouring steps. Multi-head attention works by running:

$$MHA(Z) = Concat(head_1, \dots, head_h)\, W_O, \qquad head_j = Attention\left(ZW_Q^j, ZW_K^j, ZW_V^j\right). \tag{3}$$

This architecture provides three theoretical benefits over recurrence. Firstly, the distance between two steps is one, as compared to being linear in proportion to the distance in an LSTM [8], [24]. This makes it simpler to detect long-term regularities such as those associated with daily periods. Secondly, all steps are computed simultaneously, thus reducing the training time. Lastly, the attention matrix is readily interpretable and this helps in making decisions about taking action in response to an alarm.

3.3 The ATRN architecture

The introduced Attention-based Temporal Reconstruction Network (ATRN) depicted in **Fig. 1** is an autoencoder, comprising of three modules:

- **Attention encoder.** Each scalar is mapped to d=64 dimensions using a linear layer and positional encoding is applied to it. Next, there are two pre-normalized Transformer blocks, wherein the multi-head self-attention mechanism is used with h=4 heads and a two-layer feedforward network with GELU activation, both enclosed by residual connections with layer normalization and dropout of 0.1 [21], [24].
- **Window bottleneck.** The encoder output (w×d) is flattened and projected to a latent vector of just k=16 dimensions. The bottleneck is crucial since without it, residual connections will enable the network to copy the input and reconstruct anomalies as well as normal sequences.
- **Attention decoder.** Latent vectors are projected into a w × d sequence, the position encoding is once again added, and a single Transformer block along with a linear output layer generates the reconstructed window. $\hat{\mathbf{x}}^{(i)}$.

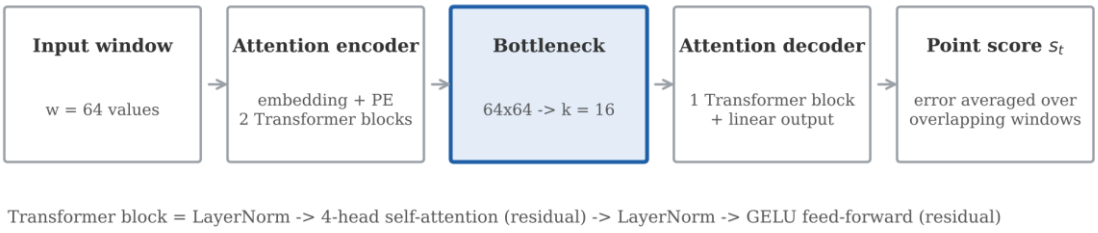


Fig. 1 - Architecture of the proposed Attention-based Temporal Reconstruction Network (ATRN).

The network parameters θ are learned by minimising the mean squared reconstruction error over the N training windows with the Adam optimiser [11]:

$$\mathcal{L}(\theta) = \frac{1}{Nw}\sum_{i=1}^{N}\|\boldsymbol{x}^{(i)} - \hat{\boldsymbol{x}}^{(i)}\|_2^2. \qquad (4)$$

3.4 Point-level anomaly scoring

During the inference process, the windows are taken with a stride of one, and thus each step t is included in at most w windows. The average of the squared errors for step t from all covering windows is called the point score:

$$s_t = \frac{1}{|\mathcal{W}_t|}\sum_{i\in\mathcal{W}_t}\left(x_t - \hat{x}_{t-i+1}^{(i)}\right)^2, \qquad (5)$$

where W_t is the set of windows that include step t. By averaging over the overlapping windows, we smoothen the score and make it independent from the starting point of the window. An alert is triggered whenever the value of s_t becomes higher than some threshold τ. As τ needs to be tuned depending on the cost of false alarms, our evaluation uses threshold-independent measures.

4. Experimental Setup

4.1 Dataset

The experiments utilize the realKnownCause subset of the Numenta Anomaly Benchmark (NAB) [1], [12]. This subset is comprised of seven real-world univariate series in which all anomalies have their cause stated explicitly. Examples of causes include technical problems in the system, wrong configuration, or a national holiday. In NAB, all anomalies are annotated as windows centered around the event. The length of windows has been selected to make sure that labeled points occupy approximately 10% of the time series. These are shown in **Table 1**. There is variation in series' lengths, rates and behaviors which makes this set of series particularly challenging for generality. As a reaction to the benchmark quality issues described in [26], this subset was selected.

Table 1 - Characteristics of the NAB real Known Cause series used in this study.

Series	Domain	Length	Anomaly windows	Labelled points (%)
ambient_temperature_system_failure	Office temperature	7,267	2	10.0
cpu_utilization_asg_misconfiguration	Cloud CPU usage	18,050	1	8.3
ec2_request_latency_system_failure	Cloud request latency	4,032	3	8.6
machine_temperature_system_failure	Industrial machine sensor	22,695	4	10.0
nyc_taxi	Taxi passenger counts	10,320	5	10.0
rogue_agent_key_hold	Keyboard event timing	1,882	2	10.1
rogue_agent_key_updown	Keyboard event timing	5,315	2	10.0

4.2 Compared models

The four detectors have been used; their windows are the same windows of length w=64, and all were transformed into the point scores by Eq. (5).

- Isolation Forest (IForest) [13] – a detector using 200 trees trained on the raw window vectors. It ignores temporal order within a window and is a good classical baseline.
- LSTM autoencoder (LSTM-AE) after [15] – it uses two-layer LSTM encoder with 64 hidden units, bottleneck with dimensionality of 16 and LSTM decoder. It belongs to the family of recurrent reconstruction-based models.
- ATRN without attention - an ablation case in which every self-attention sub-layer has been omitted and all other components remain unchanged. Thus, information mixing is possible only via the bottleneck projection.
- ATRN – the proposed model from Section 3.3 (235,793 parameters).

4.3 Training protocols

Two label-free methodologies were used. Methodology A (full-series) involves fitting each model to the whole series and then scoring the same series. This is typical of a retrospective analysis case scenario, where the training set is polluted with the anomalies to detect. Methodology B (clean-prefix) involves fitting each model only to the first 15% of each series, which corresponds to the probationary period, as specified by NAB [12], and scoring the whole series. This is typical of an online deployment scenario, where a detector is trained on a set of data supposed to be primarily anomaly-free. No labels were used to train, tune, or select any model in either methodology.

4.4 Implementation and evaluation

All the models have been implemented in PyTorch [17] and Isolation Forest in scikit-learn [18]. Neural network models were trained using the Adam optimiser [11], learning rate 10^{-3} and batch size of 128, 10 epochs on windows with stride of four for Protocol A and 15 epochs for all prefix windows for Protocol B. Hyperparameters were fixed beforehand and not tuned on the evaluation labels. Each configuration was evaluated using three random seeds (0, 1 and 2), and results are presented as mean value with standard deviation across seeds. All experiments are performed on a two-core CPU. Multithreaded CPU kernels are not bit deterministic, and, thus, multiple runs with the same seed can be different; three seeds and reporting the standard deviations are the way to account for that difference.

Performance of the detection method is evaluated by comparison with the NAB label points using three metrics: the area under the ROC curve (ROC-AUC), the area under the precision-recall curve calculated as average precision (PR-AUC), and the best F1 score across all possible thresholds (F1*). Point adjustment protocol was not used deliberately as it can make even random scores look highly precise [10]. ROC-AUC equals 0.5 for a random detector, whereas the random baseline for PR-AUC equals the anomaly ratio, about 0.10 here, which makes PR-AUC the more demanding metric for imbalanced data.

5. Results and Discussion

5.1 Overall performance

The performances of all detectors for the seven series are reported in **Table 2**. There are two notable results. Firstly, when using Protocol B, the proposed ATRN yielded the best ROC-AUC of all three neural models, 0.695, better than the one without the attention mechanism (0.679) and LSTM autoencoder (0.660). Secondly, the Isolation Forest was consistently the best-performing detector across both protocols, having a mean ROC-AUC of approximately 0.75, and mean PR-AUC which was between 1.7 to 2.8 times higher than those of the neural models. The standard deviations between all seeds were less than 0.015, so these differences are not due to random initialisation.

Table 2 - Mean performance over the seven series (mean ± standard deviation over three seeds). Best value per protocol and metric in bold; best neural model underlined.

Protocol	Model	ROC-AUC	PR-AUC	F1*
A (full-series)	IForest	**0.745 ± 0.004**	**0.412 ± 0.006**	**0.432 ± 0.005**
	LSTM-AE	<u>0.604 ± 0.008</u>	<u>0.151 ± 0.004</u>	<u>0.236 ± 0.006</u>
	ATRN w/o attention	0.601 ± 0.003	0.145 ± 0.002	0.230 ± 0.003
	ATRN (proposed)	<u>0.604 ± 0.004</u>	0.146 ± 0.003	0.231 ± 0.005
B (clean-prefix)	IForest	**0.750 ± 0.003**	**0.375 ± 0.010**	**0.422 ± 0.001**
	LSTM-AE	0.660 ± 0.010	<u>0.225 ± 0.006</u>	<u>0.304 ± 0.013</u>
	ATRN w/o attention	0.679 ± 0.013	0.207 ± 0.009	0.295 ± 0.006
	ATRN (proposed)	<u>0.695 ± 0.007</u>	0.212 ± 0.005	0.296 ± 0.005

In our experiments, the selection of training protocol turned out to be more critical than the selection of architecture for the neural models. The transition from Protocol A to Protocol B increased the mean ROC-AUC of ATRN by 0.091 and PR-AUC by 0.066; on the contrary, the ROC-AUC of Isolation Forest changed less than 0.01. This is in accordance with the reconstruction assumption of Section 3.1, namely when anomalies are present in the training set, then, in addition to the normal data, an adequately flexible autoencoder learns to reconstruct the anomalies and their reconstruction error decreases. In the case of Protocol A, all three neural models converged to almost the same values of ROC-AUC (0.601-0.604).

5.2 Per-series analysis

Table 3 shows ROC-AUC decomposition by series. The order of detectors differed substantially from one series to another, which confirms the finding reported in [20] that there is no uniformly best method. In terms of Protocol B, both attention-based models significantly outperformed Isolation Forest on cpu_asg (0.914 – 0.918 versus 0.848), as the abnormality on this series consists of shape

changes in the utilisation curve instead of shifts in its level. Both attention-based models slightly outperformed all other detectors on ec2_latency and key_updown series, but the performance of all models was near chance on these two series. Isolation Forest clearly dominated on ambient_temp and machine_temp series, for which labelled anomalies correspond to large fluctuations in signal levels.

Table 3 - ROC-AUC per series (mean over three seeds) under Protocol A (full-series) and Protocol B (clean-prefix). Best value per series and protocol in bold.

Series	IForest A	LSTM-AE A	w/o attn A	ATRN A	IForest B	LSTM-AE B	w/o attn B	ATRN B
ambient_temp	**0.815**	0.596	0.558	0.554	**0.858**	0.724	0.681	0.690
cpu_asg	**0.972**	0.593	0.724	0.721	0.848	0.849	**0.918**	0.914
ec2_latency	**0.743**	0.541	0.705	0.706	0.655	0.503	**0.660**	0.654
machine_temp	**0.870**	0.702	0.633	0.632	**0.863**	0.744	0.663	0.672
nyc_taxi	**0.708**	0.522	0.491	0.510	**0.720**	0.529	0.538	0.629
key_hold	0.543	**0.719**	0.495	0.504	**0.741**	0.709	0.694	0.711
key_updown	0.563	0.555	0.603	**0.604**	0.565	0.560	**0.596**	0.592

Fig. 2 illustrates the PR-AUC for Protocol B. This is in line with the findings shown in **Table 3**, but it serves to emphasize just how far most detectors have yet to go before reaching perfection. For the two critical scenarios of key_updown and ec2_latency, all models fall within around 0.08 of the random baseline of 0.10, which implies that these labeled anomalies are difficult to detect in [26].

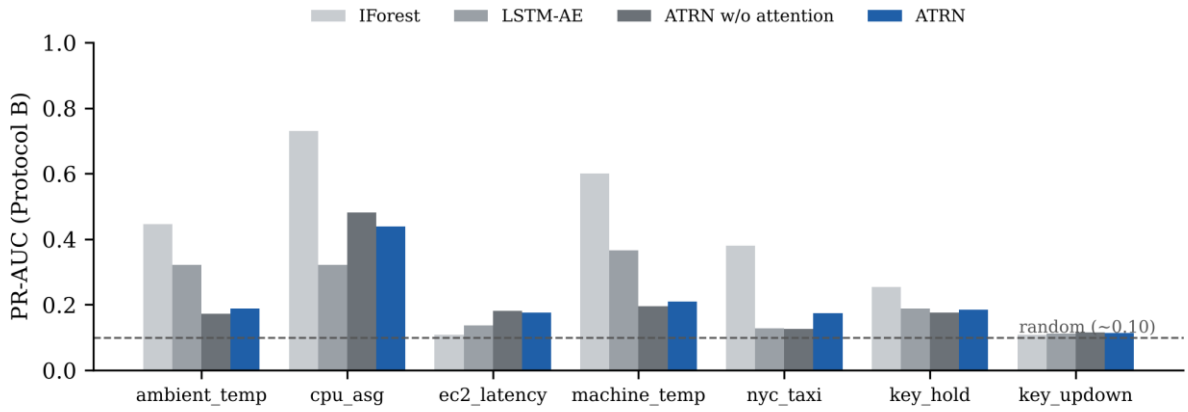


Fig. 2 - PR-AUC per series under Protocol B (mean over three seeds). The dashed line marks the random baseline, equal to the anomaly ratio.

5.3 Contribution of attention

The ablation allows isolation of the effect of attention layers. On average across all series, attention had an impact of only 0.016 reduction of ROC-AUC score under Protocol B and 0.003 reduction under Protocol A. There is, however, one series where attention had much more impact – on nyc_taxi, attention increased ROC-AUC score from 0.538 to 0.629 (difference 0.091) and PR-AUC from 0.126 to 0.174; each ATRN run on this series (ROC-AUC 0.590-0.652) achieved a higher score than each non-attention run (0.519-0.561). It is the most periodic of all series in the dataset, having a well-defined daily and weekly pattern of passenger demand. Anomalies of this series are calendar events like a city marathon, holidays and snowstorm, which disrupt the expected pattern without necessarily having outliers. It is exactly the situation where a model of temporal context described in Section 3.2 is expected to work.

This behavior can be seen in **Fig. 3**. In the ATRN score, there is little variation, and it reaches its highest peaks within the windows, namely, the marathon peak and during the New Year season, and another peak within the snowstorm window. There is no response from the ATRN score for the Thanksgiving and Christmas windows, where there is a gradual decrease in demand, and this shows a flaw in reconstruction-based scores with slow collective anomalies.

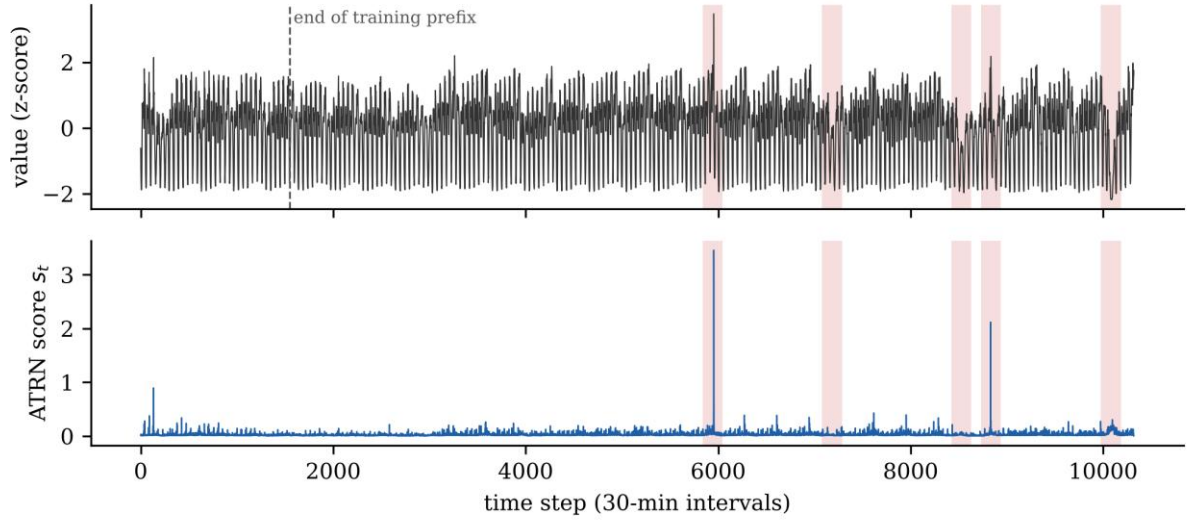


Fig. 3 - Top: nyc_taxi series with labelled anomaly windows shaded; the dashed line marks the end of the 15% training prefix (Protocol B). Bottom: ATRN anomaly score s_t from a separate training run with seed 0.

In order to explain the low average benefit of the attention, the attention weights of the trained model on nyc_taxi dataset were examined. The average entropy of the attention distributions was in the range of 3.56 to 4.09 nats per head/layer, close to the upper bound of 4.16 nats for the uniform distribution over 64 positions, while the average attended distance was around 20-22 steps, which is close to the number expected from the uniform attention pattern. There was only one attention head in the second encoder layer, which had an apparently more selective attention distribution. In other words, with short univariate time windows, most heads learned relatively broad and smoothing attention patterns, similar to temporal averaging, and the bottleneck projection has an equally good global view of the window. Thus, the interpretability of attention in this setup is constrained, which is typically not considered when attention maps are used to explain the model's behavior. It can be seen from **Fig. 4** that the structure that exists is based on the vertical stripes: all query positions have higher weights on certain few important positions, which is around two times more than the uniform weights of 1/64. This means that attention serves more as a shared, learned representation of the window, than a context model at certain positions, and the maps are quite similar for both windows.

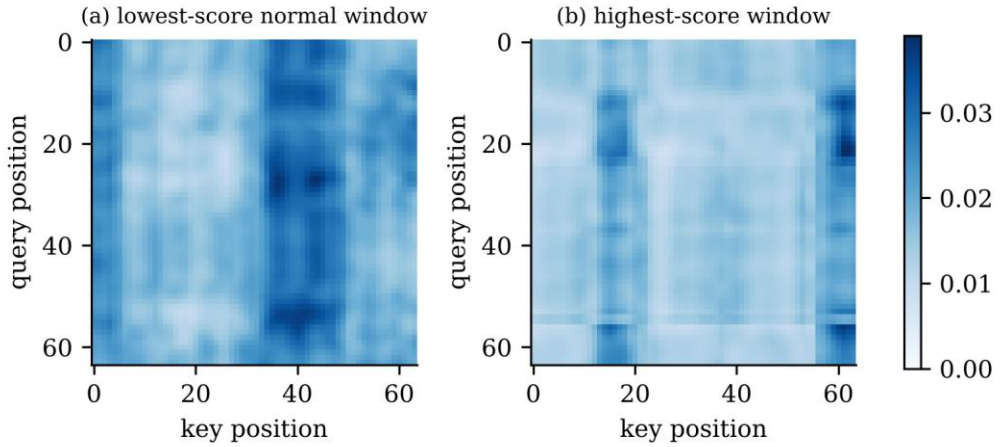


Fig. 4 - Head-averaged attention weights of the first ATRN encoder layer on nyc_taxi (Protocol B, seed 0) for (a) the normal window with the lowest score and (b) the window with the highest score.

5.4 Computational cost

ATRN needed 50 seconds on average for one training and scoring session on the two-core CPU in Protocol A, whereas both the LSTM autoencoder and the ablated version needed 16 seconds each and Isolation Forest needed less than 1 second. The self-attention process is quadratic to the window length, and hence this difference will only be larger with longer windows but will become smaller on a GPU because of high parallelism.

5.5 Discussion and practical implications

The outcomes suggest four implications for practice. First, the benefits of attention arise solely in the case where anomalies are characterized by an interruption of the pattern of temporal regularity, such as in the periodic series of nyc_taxi and shape-based cpu_asg; when anomalies are large shifts in level, the classical Isolation Forest model is not only more accurate but also substantially more efficient. Second, training procedure plays no less role than architecture of a detector: a brief calibration period consisting mostly of normal observations was helpful for every deep network, while contamination of the training set rendered the benefit of attention irrelevant. Third, good performance of a classical benchmark confirms the importance of the recommendation of recent evaluation studies [10], [20], [26]. Fourth, one cannot assume that attention maps are explanations, since in this paper most heads paid attention to features almost evenly.

There are multiple limitations to this study. There was a reliance on univariate series of just one benchmark, windows from NAB are relatively large and overlap with normal behavior, and there was a lack of hyperparameter tuning, which may put all neural networks at a disadvantage. The decision to rely on those choices was made on purpose to prevent the use of evaluation labels for the design of the model.

6. Conclusions

In this paper, an attention-based deep learning architecture called ATRN has been proposed to detect anomalies in unsupervised manner. This model is based on the combination of Transformer Encoder/Decoder together with a low-dimensional window and overlap-averaging point scoring

approach, and its architecture was guided by a theoretical analysis of the self-attention mechanism for modeling temporal context. Experiments were carried out on seven real-life series from the NAB realKnownCause dataset under two unsupervised learning scenarios without point adjustment and using the clean prefix as input, and results indicated that ATRN was the most precise of the three neural approaches (mean ROC-AUC=0.695) for the clean prefix scenario, and its performance was also significantly better on strongly periodic datasets. However, Isolation Forest reached the highest level of precision while having lower costs, and the learned attention mechanism was rather diffuse.

These results imply that attention is a good feature to include in time series anomaly detection under temporal contextual anomalies, but is not always an improvement. In future work, we plan to evaluate on more challenging multivariate benchmarks, as well as more stringent metrics. Additionally, we will look into using a hybrid approach with attention for contextual scoring in addition to isolation methods for level scores. Finally, we will study sparse and locality-biased attention.

References

[1] S. Ahmad, A. Lavin, S. Purdy and Z. Agha, Unsupervised real-time anomaly detection for streaming data, Neurocomputing, 262(2017), 134-147.

[2] J. Audibert, P. Michiardi, F. Guyard, S. Marti and M. A. Zuluaga, USAD: Unsupervised anomaly detection on multivariate time series, Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining, (2020), 3395-3404.

[3] D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate, Proc. 3rd Int. Conf. Learning Representations (ICLR), San Diego, USA, (2015).

[4] A. Blázquez-García, A. Conde, U. Mori and J. A. Lozano, A review on outlier/anomaly detection in time series data, ACM Comput. Surv., 54(3)(2021), 1-33.

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng and J. Sander, LOF: Identifying density-based local outliers, Proc. ACM SIGMOD Int. Conf. Management of Data, Dallas, USA, (2000), 93-104.

[6] V. Chandola, A. Banerjee and V. Kumar, Anomaly detection: A survey, ACM Comput. Surv., 41(3)(2009), 1-58.

[7] A. Deng and B. Hooi, Graph neural network-based anomaly detection in multivariate time series, Proc. AAAI Conf. Artificial Intelligence, 35(5)(2021), 4027-4035.

[8] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Comput., 9(8)(1997), 1735-1780.

[9] K. Hundman, V. Constantinou, C. Laporte, I. Colwell and T. Soderstrom, Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding, Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining, (2018), 387-395.

[10] S. Kim, K. Choi, H.-S. Choi, B. Lee and S. Yoon, Towards a rigorous evaluation of time-series anomaly detection, Proc. AAAI Conf. Artificial Intelligence, 36(7)(2022), 7194-7201.

[11] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, Proc. 3rd Int. Conf. Learning Representations (ICLR), San Diego, USA, (2015).

[12] A. Lavin and S. Ahmad, Evaluating real-time anomaly detection algorithms – The Numenta Anomaly Benchmark, Proc. 14th IEEE Int. Conf. Machine Learning and Applications (ICMLA), Miami, USA, (2015), 38-44.

[13] F. T. Liu, K. M. Ting and Z.-H. Zhou, Isolation forest, Proc. 8th IEEE Int. Conf. Data Mining (ICDM), Pisa, Italy, (2008), 413-422.

[14] P. Malhotra, L. Vig, G. Shroff and P. Agarwal, Long short term memory networks for anomaly detection in time series, Proc. 23rd European Symp. Artificial Neural Networks (ESANN), Bruges, Belgium, (2015), 89-94.

[15] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal and G. Shroff, LSTM-based encoder-decoder for multi-sensor anomaly detection, ICML 2016 Workshop on Anomaly Detection, New York, USA, (2016), arXiv:1607.00148.

[16] G. Pang, C. Shen, L. Cao and A. van den Hengel, Deep learning for anomaly detection: A review, ACM Comput. Surv., 54(2)(2021), 1-38.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al., PyTorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst., 32(2019), 8024-8035.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res., 12(2011), 2825-2830.

[19] 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, Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining, (2019), 3009-3017.

[20] S. Schmidl, P. Wenig and T. Papenbrock, Anomaly detection in time series: A comprehensive evaluation, Proc. VLDB Endow., 15(9)(2022), 1779-1797.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res., 15(2014), 1929-1958.

[22] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun and D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining, (2019), 2828-2837.

[23] S. Tuli, G. Casale and N. R. Jennings, TranAD: Deep transformer networks for anomaly detection in multivariate time series data, Proc. VLDB Endow., 15(6)(2022), 1201-1214.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst., 30(2017), 5998-6008.

[25] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan and L. Sun, Transformers in time series: A survey, Proc. 32nd Int. Joint Conf. Artificial Intelligence (IJCAI), (2023), 6778-6786.

[26] R. Wu and E. J. Keogh, Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress, IEEE Trans. Knowl. Data Eng., 35(3)(2023), 2421-2429.

[27] J. Xu, H. Wu, J. Wang and M. Long, Anomaly Transformer: Time series anomaly detection with association discrepancy, Proc. 10th Int. Conf. Learning Representations (ICLR), (2022).

[28] Z. Zamanzadeh Darban, G. I. Webb, S. Pan, C. C. Aggarwal and M. Salehi, Deep learning for time series anomaly detection: A survey, ACM Comput. Surv., 57(1)(2024), 1-42.

[29] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty and C. Eickhoff, A transformer-based framework for multivariate time series representation learning, Proc. 27th ACM SIGKDD Conf. Knowledge Discovery & Data Mining, (2021), 2114-2124.

[30] 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, Proc. 20th IEEE Int. Conf. Data Mining (ICDM), (2020), 841-850.

[31] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong and W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, Proc. AAAI Conf. Artificial Intelligence, 35(12)(2021), 11106-11115.

[32] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho and H. Chen, Deep autoencoding Gaussian mixture model for unsupervised anomaly detection, Proc. 6th Int. Conf. Learning Representations (ICLR), Vancouver, Canada, (2018).