Additive Explanations for Anomalies Detected from Multivariate Temporal Data

Ioana Giurgiu
IBM Research - Zurich
Ruschlikon, Switzerland
igi@zurich.ibm.com

Anika Schumann
IBM Research - Zurich
Ruschlikon, Switzerland
ikh@zurich.ibm.com

ABSTRACT

Detecting anomalies from high-dimensional multivariate temporal data is challenging, because of the non-linear, complex relationships between signals. Recently, deep learning methods based on autoencoders have been shown to capture these relationships and accurately discern between normal and abnormal patterns of behavior, even in fully unsupervised scenarios. However, validating the anomalies detected is difficult without additional explanations. In this paper, we extend SHAP – a unified framework for providing additive explanations, previously applied for supervised models – with influence weighting, in order to explain anomalies detected from multivariate time series with a GRU-based autoencoder. Namely, we extract the signals that contribute most to an anomaly and those that counteract it. We evaluate our approach on two use cases and show that we can generate insightful explanations for both single and multiple anomalies.

CCS CONCEPTS

• **Computing methodologies** → **Anomaly detection**; *Neural networks*; • **Mathematics of computing** → *Time series analysis*;

KEYWORDS

anomaly detection; time series; explainability

ACM Reference format:
Ioana Giurgiu and Anika Schumann. 2019. Additive Explanations for Anomalies Detected from Multivariate Temporal Data. In *Proceedings of The 28th ACM International Conference on Information and Knowledge Management, Beijing, China, November 3–7, 2019 (CIKM '19),* 4 pages.
https://doi.org/10.1145/3357384.3358121

1 INTRODUCTION

Automated anomaly detection is essential for managing complex environments and ensuring they maintain reliable operations with minimum burden on support teams. To detect varying and continually emerging anomalies as deviations from the baseline behavior of the system, deep learning approaches using autoencoders (AE) have been proposed in recent years [3, 4, 9]. The idea is to encode the data into a lower dimensional space and reconstruct it through the decoder. Given that most of the time a system behaves normally, an AE will learn to properly reconstruct the baseline, whereas anomalies will be reconstructed poorly (i.e., high reconstruction error), therefore allowing for a fully unsupersized approach, where the AE is trained on both normal and anomalous data simultaneously. While these models are effective, their outputs remain hard to explain, making it challenging to adopt them in the wild.

To enable trust in AE-based approaches, detected anomalies should be accompanied by explanations as to why each instance was deemed to be anomalous. One way to achieve this is by using an interpretable approximation of the original model. LIME [6] builds a linear model in the vicinity of the instance to be explained, while DeepLIFT[2] backpropagates the contributions of all neurons in the network to the input features. SHAP (SHapley Additive exPlanations) [12] is a unified framework for interpreting predictions via feature importance in supervised scenarios by using game theory.

We focus on explaining anomalies detected from unlabeled multivariate temporal data. More specifically, the type of anomalies we are interested in are not point anomalies (i.e., single peaks or dips), but entire time series, which is especially useful in various use cases – for example, detecting epileptic seizures from EEG recordings. Because no labels are available at training time, we use unsupervised methods. Our approach, based on a GRU-AE, identifies anomalies based on the reconstruction error. To explain the anomalies, we modify Kernel SHAP [12] to output both *contributing* (i.e., pushing the reconstructed value farther from the original) and *counteracting* signals (i.e., pushing the reconstructed value towards the original) and use influence weighting [11] to select the neighbourhood of time series required to compute the SHAP values for one time series sample. We evaluate our approach on two cases. The first uses EEG recordings to detect epileptic seizures, whereas the second detects performance anomalies for large-scale storage based on key performance indicators (KPIs). We show that intuitive explanations can be generated for both individual (*local*) and multiple samples, even the entire dataset (*global*). Such explanations can be used by domain experts in validating the anomalies, as well as gaining useful insights into how anomalous events are triggered or counteracted.

2 RELATED WORK

Various approaches that quantify feature contributions have been proposed for supervised learning. SHAP [12] unifies attributions produced by LIME [6], DeepLIFT [2], layer-wise relevance propagation and variations on Shapley value estimation to compute feature contributions with game theory. [5] attributes the prediction of a deep network to its input features, by integrating gradients. AVA [8] combines SHAP [12] and Integrated gradients [5], with

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3358121

antecedent event influence to build post-hoc local explanations and global patterns in supervised classification tasks.

For anomaly detection, [10] proposes using variational autoencoders to detect and explain anomalies. The algorithm is based on an approximate probabilistic model that considers the existence of anomalies in the data, and by maximizing the log-likelihood, it estimates which features contribute to determining data as an anomaly. [1] learns appropriate mappings of the feature space to detect anomalies, by minimizing intraset variance and maximizing interset variance. Both works require training the detector with normal data, therefore are semi-supervised. Our goal is to explain anomalies in an unsupervised manner by using attribution methods. To the best of our knowledge, such approaches (e.g., SHAP) have not been applied on temporal data and in unsupervised tasks.

3 APPROACH

Consider a dataset $\mathcal{S}$ with n samples for which we want to detect the anomalous samples. Each sample is represented by a multivariate time series of length L, MTS_i, where MTS_i is a matrix $X = \{x_{\upsilon t_k}\} \in R^{\upsilon x T}$ (υ is the number of signals and T is the number of time steps in $\{t_1, ..., t_T\}$). We use a symmetric GRU-based AE to encode the high-dimensional input space into a lower-dimension embedding through a nonlinear mapping and to reconstruct the original input, $f : MTS_i \rightarrow z_i \rightarrow MTS_i^{'}$. The reconstruction error at time t_k is the sum of errors of each signal $L_r(MTS_{it_k}, MTS_{it_k}^{'}) = \sum_{i=1}^{\upsilon}(x_{it_k} - x_{it_k}^{'})^2$ and is used to detect anomalous samples MTS_i with the properties: 1) reconstruction error at each timestamp t_k exceeds d standard deviations from the mean (e.g., d = 3); 2) reconstruction error of MTS_i is the mean of reconstruction errors at each timestamp t_k in MTS_i. Our challenge is to explain for each anomalous MTS_i which input signals have contributed to it and which signals have counteracted it, by computing the SHAP values of the reconstructed signals. Not only that, but we also want to identify the most contributing signals for all or subsets of the anomalies.

3.1 Shapley Additive Explanations (SHAP)

SHAP [12] unifies methods like LIME [6] and DeepLIFT [2] under the class of additive feature attribution methods. These methods are explanation models in the form of a linear function of simplified binary variables, as in $f(x) = g(z) = \theta_0 + \sum_{i=1}^{m} \theta_i z_i$, where f is the original model (i.e., GRU-based AE in this paper), g is the explanation model, z is the simplified input, $x = h_x(z)$ is the mapping function to the original model, m is the number of simplified input features and θ_i is the effect attributed to each feature. Summing the effects of all feature attributions approximates f.

SHAP uses Shapley values from game theory to explain a prediction by assigning an importance value to each feature that meets the following criteria: (1) *local accuracy* - the explanation model matches the original model; (2) *missingness* - features missing in the original input have no impact; (3) *consistency* - if a model changes so that some simplified input's contribution increases or stays the same regardless of other inputs, that input's attribution should not decrease. Since the exact computation of SHAP values is challenging, we use Kernel SHAP, a model-agnostic approximation method which combines LIME with Shapley values to build a local explanation model.

Algorithm 1 Compute SHAP values for $\{MTS_i\}_{i=1}^{b \leq n}$

1: **procedure** GETSHAPVALS($\{MTS_i\}_{i=1}^{b \leq n}, top_m^{signals}, f, \mathcal{N}_p^i$)
2: weigths $\leftarrow f$.weights
3: **for each** $sgnl \in top_m^{signals}$ **do**
4: $explainer = shap.KernelExplainer(g)$
5: $shap_{sgnl} = explainer.shapvalues(\{MTS_i\}_{i=1}^{b \leq n}, \mathcal{N}_p^i)$
6: $shaptop_m^{signals}.add(shap_{sgnl})$
7: **return** $shap_{sgnl}$

3.2 Generating Explanations for Anomalies

Since the reconstruction error of an anomaly MTS_i is the mean of reconstruction errors at each of its timestamps t_k, $L_r(MTS_i, MTS_i^{'}) = \frac{\sum_{l=1}^{k}\sum_{i=1}^{\upsilon}(x_{it_l} - x_{it_l}^{'})^2}{k}$. Henceforth, we simplify the notation by removing the timestamp t_k. Let $x_{(1)}$, ..., $x_{(\upsilon)}$ be a reordering of the signals such that $(x_{(1)} - x_{(1)}^{'})^2 \geq ... \geq (x_{(\upsilon)} - x_{(\upsilon)}^{'})^2$ for MTS_i, and $top_m^{signals} = \{x_{(1)}, ..., x_{(m)}\}$ contains the minimal set of reconstructed signals that account for at least 85% of $L_r(MTS_i, MTS_i^{'})$.

For each sample MTS_i or set of samples $\{MTS_i\}_{i=1}^{b \leq n}$, we want to detect the signals that had an impact on the reconstruction error by using Kernel SHAP (Alg. 1). We compute the SHAP values, namely the importance of each signal x_1, ..., x_υ in predicting whether a sample (or subset) is anomalous. Kernel SHAP receives the model g and a background set of s samples, $smpl_i^s$, for building the local explanation model and calculating the SHAP values (line 4). As more samples lead to lower variance estimates of the SHAP values (i.e., triggers as many re-evaluations of the model as the number of instances), [12] recommends using at least s=2*υ instances. In lines 5-6, we build a two-dimensional vector $shaptop_m^{signals}$, in which each row holds the SHAP values for one signal from $top_m^{signals}$. Since Kernel SHAP expects one-dimensional vectors, we average over the timestamps per each signal and collapse them into single values, namely $MTS_i^c \in R^{\upsilon x 1}$.

Selecting background samples – By default, Kernel SHAP requires a fixed number of background samples $smpl_i^s$ randomly chosen from $\mathcal{S}$ to compute the SHAP values for a sample MTS_i. Instead, we use influence weighting [11] to generate a neighbourhood around MTS_i. The influence weight ρ_j of sample MTS_j on MTS_i is $\rho_j = \mathcal{I}_{up,loss}(MTS_j, MTS_i) = \frac{d}{d\epsilon}\mathcal{L}(f_{\epsilon,MTS_j}, MTS_i)|_{\epsilon=0}$.

$\{\rho_j\}_{MTS_j} \in \mathcal{S}$ is the set of influence weights for all samples in $\mathcal{S}$ apart from MTS_i, where $\rho_j > 0$. These weights induce a probability distribution over the signal space centered at MTS_i. We select a local neighbourhood $\mathcal{N}_p^i$ of the most p most influential samples on MTS_i, defined as $\mathcal{N}_p^i(MTS_i, \mathcal{S}) = argmax \sum_{MTS_j \in \mathcal{N}} \rho_j$, and use it to compute the SHAP values (line 6 in Alg. 1).

Finally, we divide the signals into *contributing* to the anomaly MTS_i and *counteracting* the anomaly. Depending on how the reconstructed value of the signal relates to the original, we divide the signals as follows: if $x_i > x_i^{'}$, *contributing* signals are those with negative SHAP values, while *counteracting* signals have positive SHAP values. The opposite applies when $x_i < x_i^{'}$. We maintain two lists, $shap_{contributing}$ and $shap_{counteracting}$ for each signal

in $top_m^{signals}$. In both lists, the signals with higher SHAP values are of most interest to explain the detected anomaly.

4 EVALUATION

GRU-based AE – The AE dimensions are v-50-5-50-v. We set batch size to 32, learning rate to 0.01, use Adam and stop training if the loss does not improve over 10 consecutive epochs. The activation function for each layer is ReLU. Our implementation is in Keras.

Datasets – We generate additive explanations for two datasets. In the first, we use EEG recordings [7] to detect abnormal brain activity that leads to epileptic seizures. The dataset contains one signal collected in 178 consecutive windows of 23 seconds each, for 500 patients and is labeled into *seizure* (20%) and *noseizure* (80%). We treat each of the consecutive windows as individual features. Thus, our objective is to provide **temporal** explanations, namely to identify the 23s windows that mostly *contribute* to and *counteract* a *seizure*. We use the labels to compute the precision and recall of the anomaly detection, as the model is fully unsupervised. In the second dataset, we use KPIs collected with 5-minute granularity for large-scale storage. The dataset contains 798 signals for 100+ environments over 24h windows. While no labels are available, a fraction of the anomalies detected has been validated by domain experts. We will refer to some of them as illustrating examples for generating **feature** explanations.

4.1 Epileptic seizure detection use case

First, we report on the precision and recall of the anomaly detector when L_r exceeds d standard deviations from the mean (Table 1). Since detecting *seizure* samples is our objective, we are specifically interested in the precision and recall for this class ($Prec_s$ and Rec_s). With $d = 1$, the false positive rate is 1.6%, and reduces to 0 when $d \geq 2$. At the same time, as the detector filters more anomalies when increasing d, recall drops as expected. For the *noseizure* samples, precision and recall ($Prec_{ns}$ and Rec_{ns}) are 1 independent of d.

	d=1	d=2	d=3		d=1	d=2	d=3
$Prec_s$	0.984	1	1	$Prec_{ns}$	1	1	1
Rec_s	0.75	0.42	0.34	Rec_{ns}	1	1	1

Table 1: Precision and recall when reconstruction error exceeds d standard deviations from the mean.

For every detected anomaly, the AE reconstructs poorly on an average of 26 out of 178 windows. An example of ranked windows based on their reconstruction error for a correctly detected *seizure* sample when $d = 2$ is shown in Fig. 1. While such a representation is easy to understand, we identify the following severe limitations: (1) contributing windows can only be identified for each individual sample, which inherently induces a strong per-sample locality factor; across the 75 (out of 100) *seizure* anomalies correctly detected, 96 out of 178 windows (54%) contribute to the anomalies, therefore making it impossible for a domain expert to narrow down the most contributing windows over all or a subset of the *seizure* anomalies; (2) counteracting windows cannot be identified, as their reconstruction error would be lower than the set threshold.

By applying our approach, we can alleviate these limitations. First, we compute the SHAP values for the *seizure* anomaly shown in Fig. 1. In this example, the 26 reconstructed windows have a cummulated L_r = 0.565. Since they explain 85% of the error, the total L_r = 0.67 (Note: input signal is normalized prior to training the AE). As shown in Fig. 2, the smallest overall reconstruction loss for a sample is 0.09, while the largest is 1.42. The base value represents the average AE reconstruction error over the training samples passed (i.e., in our case, the entire dataset). Given that the set is imbalanced (only 20% of samples are anomalies), the neighbourhood of samples built around any *seizure* sample still contains many *noseizure* samples, which explains the low base value. The output value is the total reconstruction error for the example anomaly. By computing the SHAP values, we are able to generate the lists of *contributing* and *counteracting* 23s windows.

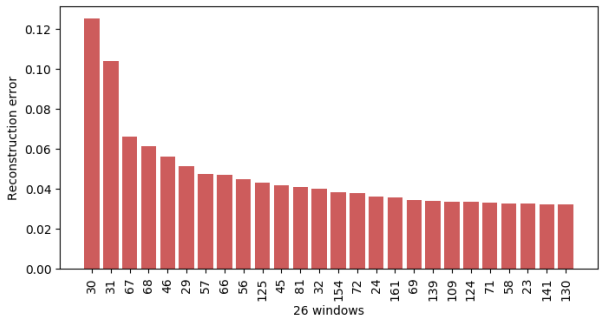

Figure 1: Reconstruction errors for 26 windows that explain 85% of a *seizure* sample's L_r.

We remark the following. First, the windows contributing to the anomaly (shown in red) are far fewer than the 26 identified through the reconstruction loss (42%). Second, there is agreement between SHAP and L_r for the top-4 windows (i.e., 30, 31, 67, 68) in terms of ranking. Since L_r is specific to the example anomaly, but SHAP values are computed based on neighbourhood samples, this agreement suggests that these 4 windows vary wildly between *seizure* and *noseizure* samples, indicating different dynamic properties of brain electrical activity. Indeed this is the case and we show the EEG signal measured from windows 12 to 31 for the example anomaly in comparison to a *noseizure* sample in Fig. 3. Third, 15 of the 26 poorly reconstructed windows are neither *contributing* nor *counteracting* the anomaly, because they have positive reconstruction errors for *noseizure* samples as well, a fact captured due to the neighbourhoods built with influence weighting. Fourth, our model identifies 4 *counteracting* windows (shown in blue), which are pushing the AE's output towards zero, but by a small margin. Fig. 3 captures 3 of these 4 windows (i.e., 12, 14, 17), whose signals deviate both from the baseline (i.e., 13, 16, 17 and 18-29) and from the *contributing* windows. Such *counteracting* windows could not be identified by using the reconstruction loss and are extremely useful for a domain expert to understand whether the patient's activity during those windows could potentially reduce the risk of epileptic seizures. Based on Fig. 3, it is clear that our approach indeed focuses on the windows that show abnormal brain activity (*contributing*) and those that exhibit an increased, but normal activity (*counteracting*) to derive explanations.

Finally, we compute the SHAP values for all 178 windows across all detected anomalies and identify the overall *contributing* and *counteracting* windows (Table 2). There are 24 *contributing* and 9

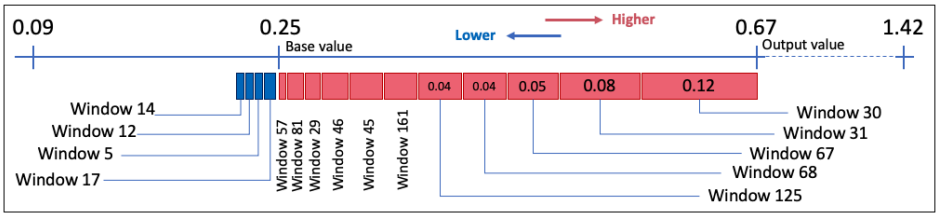

Figure 2: *Contributing* **and** *counteracting* **signals identified via computed SHAP values for the** *seizure* **anomaly in Fig. 1.**

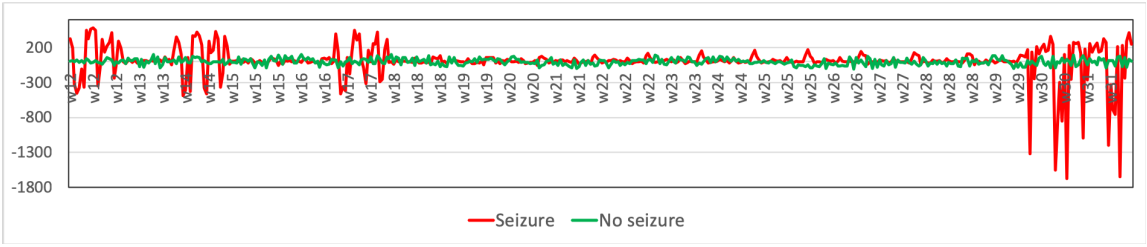

Figure 3: EEG signal in windows 12 to 31 compared for the *seizure* **anomaly in Fig. 1 and a** *noseizure* **sample.**

counteracting signals. Their contributions to either push the model output to higher (max. is 1.42) or lower values (min. is 0.09) are shown in gradient reds and blues, respectively.

Contr.	30	161	162	125	107	156	124	31	68
	157	168	106	151	29	44	67	45	126
	155	177	178	108					
Countr.	12	20	27	128	135	64	40	146	

Table 2: Overall *contributing* **and** *counteracting* **windows across all** *seizure* **anomalies.**

4.2 Performance anomaly detection use case

Due to the high-dimensionality in the KPI space (798 metrics), the complexity of the storage environment and the value of d, the number of poorly reconstructed KPIs explaining 85% of L_r for a sample varies from 20 to 95. For a domain expert, inspecting 95 different KPIs is extremely tedious. By applying our approach, we are able to reduce the *contributing* KPIs by 30-50%. For instance, in the case of a storage environment with 4 ports and 32 nodes, computing the SHAP values indicates that the 12 *contributing* KPIs to the detected performance anomaly are read cache hits, write cache hits, write-cache delay, write I/O rate, read I/O rate, cache-to-disk transfer rate, peak read response time, read response time, read transfer size, write transfer size, write data rate and read data rate, in that order. At the same time, disk and CPU utilization, as well as port to local node send queue time are identified as *counteracting* KPIs. This suggests that the example anomaly points to an I/O performance problem in the environment, rather than intensive loads on the disks or node CPUs. Finally, being able to reduce the volume of *contributing* KPIs allows domain experts to validate anomalies faster and reduces their inspection time into the anomalies themselves. Even more so, with the help of additive explanations, such as the ones generated by our approach, it is possible to further narrow down the profile of a performance anomaly, which should speed up problem resolution times.

5 CONCLUSIONS

In this paper, we have shown how to extend Kernel SHAP to provide additive explanations for anomalies detected via an unsupervised GRU-based AE from high-dimensional multivariate temporal data. Specifically, we use influence weighting to generate informative neighbourhoods of samples used to compute SHAP values per each signal of a time series sample. Then, we generate lists of *contributing* and *counteracting* signals for individual or multiple (even all) anomalies. We evaluate our approach on two use cases and show that we can provide both local and global explanations in space or time, that can be used by domain experts to validate anomalies and gain useful insights into how anomalous events are triggered or counteracted.

REFERENCES

[1] A. Del Giorno et al. 2016. Informative Features for Anomaly Detection. In *ICML Anomaly Detection Workshop*.
[2] A. Shrikumar et al. 2017. Learning Important Features Through Propagating Activation Differences. In *ICML*.
[3] C. Zhang et al. 2018. A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. In *arXiv:1811.08055*.
[4] E. Gutflaish et al. 2019. Temporal Anomaly Detection: Calibrating the Surprise. In *AAAI*.
[5] M. Sundararajan et al. 2017. Axiomatic Attribution for Deep Networks. In *ICML*.
[6] M. T. Ribeiro et al. 2016. Why Should I Trust You?: Explaining the Predictions of Any Classifier. In *KDD*.
[7] R. Andrzejak et al. 2001. Indications of Nonlinear Deterministic and Finite Dimensional Structures in Time Series of Brain Electrical Activity: Dependence on Recording Region and Brain State. In *Phys. Rev. E*.
[8] U. Bhatt et al. 2019. Towards Aggregating Weighted Feature Attributions. In *AAAI Workshop on Network Interpretability for Deep Learning*.
[9] Y. Guo et al. 2018. Multidimensional Time Series Anomaly Detection: A GRU-based Gaussian Mixture Variational Autoencoder Approach. In *ACML*.
[10] Y. Ikeda et al. 2019. Estimations for Dimensions Contributing to Detected Anomalies with Variational Autoencoders. In *AAAI*.
[11] P. W. Koh and P. Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *ICML*.
[12] S. M. Lundberg and S. Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *NIPS*.