Sparse Neural Networks for Anomaly Detection in High-Dimensional Time Series

Narendhar Gugulothu, Pankaj Malhotra, Lovekesh Vig, Gautam Shroff
TCS Research, New Delhi, India
{narendhar.g, malhotra.pankaj, lovekesh.vig, gautam.shroff}@tcs.com,

Abstract

Online anomaly detection in time series is an important component for automated monitoring. In many applications, time series are high-dimensional with tens or even hundreds of variables being monitored simultaneously. We note that existing anomaly detection approaches based on recurrent autoencoders may not be very effective for high-dimensional time series. In this work, we propose a simple, yet effective, extension to such approaches for high-dimensional time series. Our approach combines the advantages of non-temporal dimensionality reduction techniques and recurrent autoencoders for time series modeling through an end-to-end learning framework. The recurrent encoder gets sparse access to the input dimensions via a feedforward layer while the recurrent decoder is forced to reconstruct all the input dimensions, thereby leading to better regularization and a robust temporal model. The autoencoder thus trained on normal time series is likely to give a high reconstruction error, and a corresponding high anomaly score, for any anomalous time series pattern. We prove the efficacy of the proposed approach through experiments on a public dataset and two real-world datasets with significant improvement in anomaly detection performance over several baselines. We observe that the proposed approach is able to perform well even without knowledge of relevant dimensions carrying the anomalous signature in a high-dimensional setting.

1 Introduction

In the current Digital Era, streaming data is ubiquitous and growing at a rapid pace, enabling automated monitoring of systems, e.g. using Industrial Internet of Things [Da Xu *et al.*, 2014] with large number of sensors capturing the operational behavior of an equipment. Complex industrial systems such as engines, turbines, aircrafts, etc., are typically instrumented with a large number (tens or even hundreds) of sensors resulting in high-dimensional streaming data. There is a growing interest among original equipment manufacturers (OEMs) to leverage this data to provide *remote health monitoring services* and help field engineers take informed decisions.

Anomaly detection from time series is one of the key components in building any health monitoring system. For example, detecting early symptoms of an impending fault in a machine in form of anomalies can help take corrective measures to avoid the fault or reduce maintenance cost and machine downtime. Recently, Recurrent Neural Networks (RNNs) have found extensive applications for anomaly detection in multivariate time series by building a model of normal behavior of complex systems from multi-sensor data, and then flagging deviations from the learned normal behavior as anomalies. Approaches such as LSTM-AD [Malhotra *et al.*, 2015], EncDec-AD [Malhotra *et al.*, 2016a] (described later in Section 4.1), and their extensions have been used in anomaly detection applications for real-world industrial equipment such as engines [Malhotra *et al.*, 2015; Malhotra *et al.*, 2016a], gasoil heating loop [Filonov *et al.*, 2016], turbomachinery [Vishnu *et al.*, 2017], and spacecrafts [Hundman *et al.*, 2018], etc. Further extensions of these models have been used for prognostics and health management applications for equipment such as turbofan engines, milling machines, and pumps, producing state-of-the-art results on benchmark datasets [Malhotra *et al.*, 2016b; Gugulothu *et al.*, 2017].

High-dimensional data is known to be sparse and almost any point can be considered to be an anomaly from the perspective of distance-based definitions. Consequently, the notion of finding meaningful anomalies becomes substantially more complex and non-obvious in high-dimensional data due to: i) exponential growth in possible subspaces, and ii) irrelevant dimensions acting as noise and masking the true anomalies [Zimek *et al.*, 2012; Erfani *et al.*, 2016]. Even though the above-stated RNN-based approaches have been successfully used for anomaly detection in various domains, they have not been tested for anomaly detection in high-dimensional time series (say, from more than 20 sensors). For instance, [Filonov *et al.*, 2016] explores an RNN-based approach for 19 sensors but then uses domain-knowledge to finally select only 6 sensors for anomaly detection. Similarly, [Hundman *et al.*, 2018] model each input sensor independently via an RNN-based approach for anomaly detection to deal with large number of sensors.

We note that in real-world industrial applications, any sub-

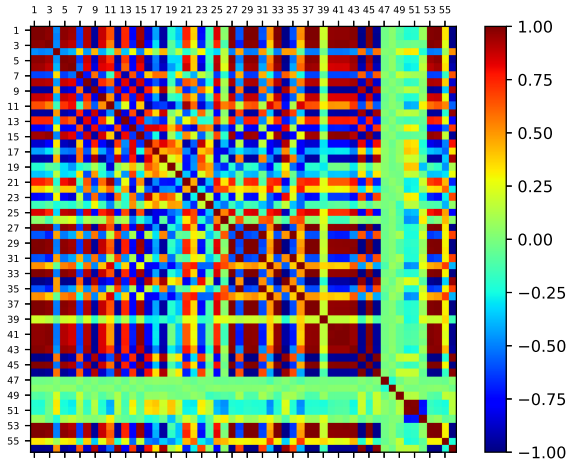


Figure 1: A real-world example of a correlation matrix of 56 parameters in a turbomachinery. Such complex systems often consist of multiple subsystems where sensors corresponding to a subsystem are highly correlated. In this example, sensors 40-46 are highly correlated among themselves but also with several other sensors, sensors 50-52 are highly correlated among themselves but very weakly correlated with any other sensor, sensors 47-49 are not correlated with any sensor.

system within a complex system leads to a subspace of highly correlated dimensions that may, in turn, be either dependent on or correlated to dimensions of other subsystems. Refer Figure 1 for an example. This suggests that dimensionality reduction techniques such as Principal Components Analysis can be used to obtain low-dimensional time series which can then be used for anomaly detection. However, since the anomaly may be present in only a few dimensions (e.g. 1-3 dimensions in datasets we consider) out of tens of dimensions, dimensionality reduction may lead to loss of weak anomaly signature, and therefore, result in a sub-optimal anomaly detector. Also, point-wise dimensionality reduction techniques would not capture the temporal nature of correlations and dependencies.

In this work, we propose **Sp**arse **Re**current Neural Network based **A**nomaly **D**etection, or **SPREAD**: an approach that combines the point-wise (i.e. non-temporal) dimensionality reduction – via a sparsely-connected feedforward layer over the input layer – with a recurrent neural encoder in an end-to-end learning setting to model the normal behavior of a system. Once a model for normal behavior is learned, it can be used for detecting behavior deviating from normal by analyzing the reconstruction via a recurrent decoder that attempts to reconstruct the original time series back using output of the recurrent encoder. Having been trained only on normal data, the model is likely to fail in reconstructing an anomalous time series and result in high reconstruction error. This error in reconstruction is used to obtain an anomaly score (refer Section 4 for details).

SPREAD combines point-wise dimensionality reduction (via feedforward layer) and time series compression (via recurrent layers) in an *end-to-end learning approach* via an autoencoder neural network. Through empirical evaluation on two real-world datasets and a publicly available simulated dataset, we demonstrate the following key properties of SPREAD that are useful in practical high-dimensional time series anomaly detection scenarios:

- SPREAD *combines the advantages of dimensionality reduction as well as temporal encoding* to learn a robust temporal model of high-dimensional time series.
- The proposed feedforward dimensionality reduction layer with *sparse access to input dimensions acts as a strong regularizer* forcing the network to effectively capture dependencies across input dimensions.

The rest of the paper is organized as follows: We provide a review of related work in Section 2. We briefly introduce RNN-based Encoder Decoder (RNN-ED) architecture in Section 3. We provide details of SPREAD using RNN-ED in Section 4. We provide experimental details and observations in Section 5, and conclude in Section 6.

2 Related Work

Domain-driven sensor selection for anomaly detection using RNNs (e.g. in [Filonov *et al.*, 2016]) is restricted by the knowledge of important sensors to capture a given set of anomalies, and would therefore miss other types of anomalous signatures in any sensor not included in the set of relevant sensors. Similarly, approaches considering each sensor or a subset of sensors independently (e.g. in [Hundman *et al.*, 2018]) to handle such scenarios may not be appropriate given that: a) it leads to loss of useful sensor-dependency information, and b) when the number of sensors is large, building and deploying a separate RNN model for each sensor may be impractical and computationally infeasible.

Subspace outlier detection approaches (e.g. [Aggarwal and Yu, 2001; Kriegel *et al.*, 2009; Keller *et al.*, 2012]) find abnormal lower dimensional projection in which the density of the data is exceptionally lower than average. Multivariate online anomaly detection has been proposed in [Ahmed *et al.*, 2007] for non-temporal data. Also, various approaches for high-dimensional anomaly detection have been proposed for non-temporal data, e.g. [Tucker *et al.*, 2001; Zimek *et al.*, 2012; Scardapane *et al.*, 2017; Yi *et al.*, 2017]. However, all these approaches cannot directly capture the temporal aspect of data in case of application to time series anomaly detection. It is not clear as to how to easily extend such approaches to time series anomaly detection in an online setting. Dimensionality reduction using PCA [Ding and Kolaczyk, 2013] has been proposed for anomaly detection in high-dimensional data. On the other hand, our approach implicitly learns to map high-dimensional data to lower dimensional subspace, then uses recurrent autoencoder to deal with temporal aspect of data, and works in an online setting making it robust and more suitable for practical applications.

Sparse architectures via L_1 regularization have been proposed for deep neural network (e.g. [Feng and Simon, 2017; Wen *et al.*, 2016; Scardapane *et al.*, 2017; Yoon and Hwang, 2017]). To the best of our knowledge, our proposed approach is the first instance that shows an effective way to leverage sparse networks via L_1 regularization for anomaly detection in high-dimensional time series.

3 Background: RNN Encoder-Decoder

Let $\mathbf{x}_{1...T}$ denote a multivariate real-valued time series $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T$ of length T where each $\mathbf{x}_t \in \mathbb{R}^d$, ($d$ being the input dimension, e.g. number of sensors in our case). An RNN Encoder-Decoder (RNN-ED), illustrated in Figure 2(a), is a sequence-to-sequence learning model [Sutskever *et al.*, 2014] consisting of a pair of (multilayered) RNNs trained simultaneously to learn a mapping from input sequences of the form $\mathbf{x}_{1...T}$ to output sequences of the form $\mathbf{x}'_{1...T'}$. The encoder ingests $\mathbf{x}_{1...T}$, and maps it to a fixed dimensional representation $\mathbf{z}_T \in \mathbb{R}^n$, where $n = h \times L$, h is the number of LSTM units in one layer and L is the number of LSTM layers (refer Appendix A.1 for more details). Then, the decoder uses $\mathbf{z}_T$ to generate an estimate $\hat{\mathbf{x}}'_{1...T'}$ for $\mathbf{x}'_{1...T'}$.

The encoder iterates through the points in $\mathbf{x}_{1...T}$ as follows: At time t, the current input $\mathbf{x}_t$ and the previous hidden state $\mathbf{z}_{t-1}$ are used to compute the hidden state $\mathbf{z}_t$ (through a sequence of operations as described in Appendix A.1). Once the encoder has effectively represented the useful information from $\mathbf{x}_{1...T}$ in $\mathbf{z}_T$, the decoder goes through a similar set of transformations as the encoder, using $\mathbf{z}_T$ as its initial hidden state. The decoder iteratively uses its hidden state $\mathbf{z}'_t$ to generate an estimate $\hat{\mathbf{x}}'_t$ for $\mathbf{x}'_t$ via a linear transform for T' steps (or till some stopping criterion is met). Figure 2(c) illustrates RNN-ED unrolled over time for standard and sparse scenarios.

4 RNN-ED based Anomaly Detection

We first describe EncDec-AD as proposed in [Malhotra *et al.*, 2016a], and then provide details of our proposed approach SPREAD that extends EncDec-AD to better deal with high-dimensional time series.

4.1 EncDec-AD

EncDec-AD first trains an RNN-ED as a temporal autoencoder using reconstruction error as the loss function. The autoencoder is trained only on normal time series[1] such that the network learns to reconstruct a normal time series well but is likely not to reconstruct an anomalous time series - having not seen such a pattern during training. The reconstruction error is then used to obtain an anomaly score. It is worth noting that in real-world applications, getting access to large amount of normal data is relatively easy compared to getting access to similar amount of anomalous data, thereby making EncDec-AD useful in practice.

More specifically, RNN-ED is trained in such a manner that the target time series $\mathbf{x}^{(i)}_{T...1}$ is reverse of the input time series $\mathbf{x}^{(i)} = \mathbf{x}^{(i)}_{1...T}$, for ith time series instance. The overall process can be thought of as a non-linear mapping of the input multivariate time series to a fixed-dimensional vector $\mathbf{z}^{(i)}_T$ via an encoder function f_E, followed by another non-linear mapping of the fixed-dimensional vector to a multivariate time series via a decoder function f_D. RNN-ED is trained to minimize the loss function $\mathcal{L}$ given by the average of squared reconstruction error:

$$\begin{aligned}
\mathbf{z}^{(i)}_T &= f_E(\mathbf{x}^{(i)}; \mathbf{W}_E) \\
\hat{\mathbf{x}}^{(i)} &= f_D(\mathbf{z}^{(i)}_T; \mathbf{W}_D) \\
\mathbf{e}^{(i)}_t &= \mathbf{x}^{(i)}_t - \hat{\mathbf{x}}^{(i)}_t,\ t = 1 \dots T \\
C_1(\hat{\mathbf{x}}^{(i)}, \mathbf{x}^{(i)}) &= \frac{1}{T} \sum_{t=1}^{T} \parallel \mathbf{e}^{(i)}_t \parallel_2^2 \\
\mathcal{L} &= \frac{1}{N} \sum_{i=1}^{N} C_1(\hat{\mathbf{x}}^{(i)}, \mathbf{x}^{(i)})
\end{aligned} \tag{1}$$

where, N is the number of train instances, $||.||_2$ denotes L_2-norm, and $\mathbf{W}_E$ and $\mathbf{W}_D$ represent the parameters of the encoder and decoder RNNs, respectively.

Anomaly score from error vectors

Given the error vector $\mathbf{e}^{(i)}_t$, Mahalanobis distance [De Maesschalck *et al.*, 2000] is used to compute the anomaly score $a^{(i)}_t$ as follows:

$$a^{(i)}_t = \sqrt{(\mathbf{e}^{(i)}_t - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{e}^{(i)}_t - \boldsymbol{\mu})} \tag{2}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance matrix of the error vectors corresponding to the normal training time series instances. This anomaly score can be obtained in an online setting by using a window of length T ending at current time t as the input, making it possible to generate timely alarms related to anomalous behavior. A point $\mathbf{x}^{(i)}_t$ is classified as anomalous if $a^{(i)}_t > \tau$; the threshold τ can be learned using a hold-out validation set while optimizing for F-score.

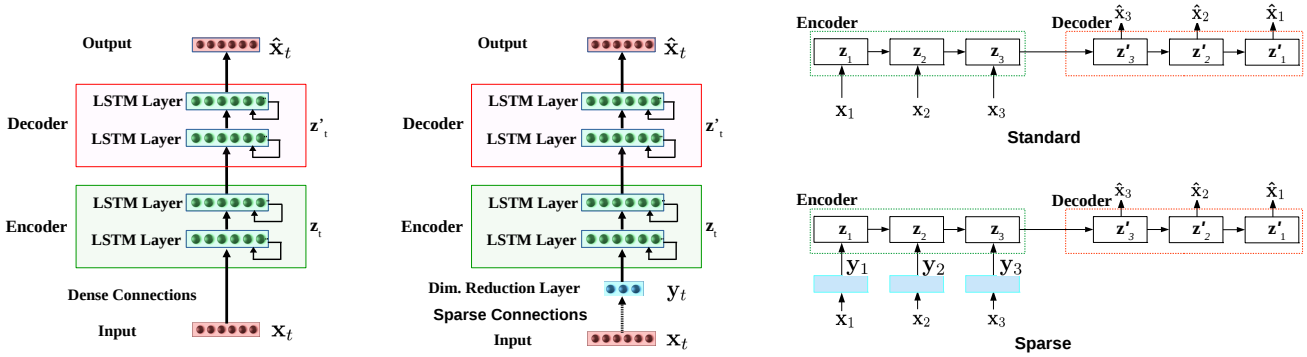
4.2 SPREAD Algorithm

We extend EncDec-AD to SPREAD as follows: We explicitly provision for mapping each high-dimensional point in the input time series to a reduced-dimensional point via a feedforward dimensionality reduction layer, and then use the time series in reduced-dimensional space to reconstruct the original high-dimensional time series via RNN-ED, as in EncDec-AD. We further add a sparsity constraint on the weights of this feedforward layer such that each unit in the feedforward layer has access to a subset of the input dimensions. Figure 2 illustrates the difference in RNN-ED and the proposed Sparse RNN-ED.

A feedforward layer with sparse connections $\mathbf{W}_R$ from the input layer is used to map $\mathbf{x}^{(i)}_t \in \mathbb{R}^d$ to $\mathbf{y}^{(i)}_t \in \mathbb{R}^r$, such that $r < d$, through a non-linear transformation via Rectified Linear Units (ReLU). The transformed lower-dimensional input $\mathbf{y}^{(i)}_t$ is then used as input to the RNN-ED network instead of $\mathbf{x}^{(i)}_t$ modifying the steps in Equation 1 as follows:

$$\begin{aligned}
\mathbf{y}^{(i)}_t &= ReLU(\mathbf{W}_R \cdot \mathbf{x}^{(i)}_t),\ t = 1 \dots T \\
\mathbf{z}^{(i)}_T &= f_E(\mathbf{y}^{(i)}; \mathbf{W}_E) \\
\hat{\mathbf{x}}^{(i)} &= f_D(\mathbf{z}^{(i)}_T; \mathbf{W}_D) \\
\mathbf{W}^* &= \arg\min_{\boldsymbol{W}} \mathcal{L} + \frac{\lambda}{d \times r} ||\boldsymbol{W}_R||_1
\end{aligned} \tag{3}$$

[1] We assume that a machine/system exhibits normal behavior during its initial life. We use data from this period to learn the models.



(a) Standard RNN Encoder-Decoder.

(b) Sparse RNN Encoder-Decoder.

(c) RNN Encoder-Decoder unrolled over time.

Figure 2: RNN-ED versus the proposed Sparse RNN-ED

where, $\boldsymbol{W} = \{\boldsymbol{W}_R, \boldsymbol{W}_E, \boldsymbol{W}_D\}$, $ReLU(x) = max(x, 0)$. L_1-norm $||\mathbf{W}_R||_1 = \sum_j |w_j|$ (where w_j is an element of matrix $\mathbf{W}_R$) is the LASSO penalty [Tibshirani, 1996] employed to induce sparsity in the dimensionality reduction layer, i.e., constrain a fraction of the elements of $\mathbf{W}_R$ to be close to 0 (controlled via the parameter λ). This converts a dense, fully-connected feedforward layer to a sparse layer. The sparse feedforward layer and the RNN-ED are trained in an end-to-end manner via stochastic gradient descent. $\frac{\partial||\boldsymbol{W}_R||}{\partial w_i} = sign(w_i)$, $w_i \neq 0$, w_i is an element of matrix $\mathbf{W}_R$. As L_1-norm is not differentiable at 0, the subgradient 0 is used in practice. Once trained, the anomaly score is computed as in Equation 2.

The resulting sparse weight matrix $\mathbf{W}_R$ ensures that the connections between the input layer and the feedforward layer are sparse such that each unit in the feedforward layer potentially has access to only a few of the input dimensions. Therefore, each dimension of $\mathbf{y}_t^{(i)}$ is a linear combination of a relatively small number of input dimensions, effectively resulting in unsupervised feature selection.

It is to be noted that even though the ReLU layer implies dimensionality reduction, the autoencoder is trained to reconstruct the original time series itself. This ensures that the anomaly scores are still interpretable as contribution of each original dimension to the anomaly score can be estimated. Further, the sparse feedforward layer acts as a strong regularizer such that the reduced dimensions in the ReLU layer are forced to capture the information relevant to reconstruct all the original input dimensions. In a nutshell, RNN-ED ensures that the temporal dependencies are well captured in the network while the sparse feedforward layer ensures that the dependencies between various dimensions at any given time are well captured.

5 Experimental Evaluation

5.1 Approaches considered for comparison

We compare SPREAD with standard EncDec-AD – hereafter referred to as **AD**. We also consider following variants of AD for comparison:

- A simple non-temporal anomaly detection model, namely **MD**, based on Mahalanobis Distance in the original input space using $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of the original point-wise inputs from the train instances (similar to Equation 2 where $\mathbf{x}_t$ is used instead of $\mathbf{e}_t$ to get the anomaly score).
- **Relevant-AD** where AD model is trained only on the most relevant sensors sufficient to determine the anomalous behavior or fault (as suggested by domain experts). This is used to evaluate the efficacy of SPREAD in being able to detect weak anomaly signatures present in only a small subset of the large number of input sensors.
- To compare implicit dimensionality reduction in SPREAD via end-to-end learning with standard dimensionality reduction techniques, we consider **PCA-AD** where Principal Components Analysis (PCA) is first used to reduce the dimension of input being fed to AD (we take top principal components capturing 95% of the variance in data).
- To evaluate the effect of sparse connections in the feedforward layer with LASSO sparsity constraint, we consider **FF-AD** (feedforward EncDec-AD) model which is effectively SPREAD without the L_1 regularization (i.e. $\lambda = 0$).

For performance evaluation, each point in a time series is provided ground truth as 0 (normal) or 1 (anomalous). Anomaly score is obtained for each point in an online manner, and Area under ROC curve (AUROC) (obtained by varying the threshold τ) is used as a performance metric.

5.2 Datasets Considered

We use three multi-sensor time series datasets as summarized in Table 4 for our experiments: i) *GHL*: a publicly avail-

Table 1: Performance Comparison of Anomaly Detection Models in terms of AUROC. AD refers to EncDec-AD.

Dataset	Relevant-AD	MD	PCA-AD	AD	FF-AD	SPREAD
GHL	0.944	0.692	0.903	0.974	0.962	**0.977**
Turbomachinery	0.981	0.903	0.688	0.878	0.879	**0.945**
Pulverizer	0.882	0.812	0.757	0.953	**0.966**	0.964

Table 2: Sparsity Factors

Approach	GHL	Turbo.	Pulverizer
FF-AD($\lambda = 0$)	0.041	0.045	0.074
SPREAD($\lambda = 0.01$)	0.491	0.310	0.581

Table 3: Turbomachinery: Effect of treating sensors independently

Sensor	R_1	R_2	R_1&R_2
AUROC	0.888	0.922	0.981

able Gasoil Heating Loop dataset [Filonov *et al.*, 2016], ii) *Turbomachinery*: a real-world turbomachinery dataset, and iii) *Pulverizer*: a real-world pulverizer dataset. Anomalies in GHL dataset correspond to cyber-attacks on the system, while anomalies in Turbomachinery and Pulverizer dataset correspond to faulty behavior of system. Each dataset is divided into train, validation and test sets - whereas the train and validation sets contain only normal time series, the test set contains normal as well as anomalous time series. Refer Appendix A.2 for more details.

5.3 Training details

Table 4: Details of datasets. Here $\mathbf{T}$: window length, $\mathbf{d}$: no. of sensors, $\mathbf{d_r}$: no. of relevant sensors for anomaly, $\mathbf{p}$: no. of principal components, $\mathbf{n_f}$: no. of faults, $\mathbf{n}_a$: no. of anomalous points, $\mathbf{n}$: no. of windows.

Dataset	$\mathbf{T}$	$\mathbf{d}$	$\mathbf{d_r}$	$\mathbf{p}$	$\mathbf{n_f}$[2]	$\mathbf{n}_a$	$\mathbf{n}$
GHL	100	14	1	9	24	8,564	32,204
Turbo.	20	56	2	10	2	57	4353
Pulverizer	60	35	3	13	1	443	16,344

We use Adam optimizer [Kingma and Ba, 2014] for optimizing the weights of the networks with initial learning rate of 0.0005 for all our experiments. We chose the best architecture as the one with least reconstruction error on the hold-out validation set containing only normal time series via grid search on following hyper-parameters: number of recurrent layers in RNN encoder and decoder $L = \{1, 2, 3\}$, number of hidden units per layer in the range of $50 - 250$ in steps of 50, and number of units $r = \{\frac{d}{4}, \frac{d}{2}\}$ in the feedforward layer. We used $\lambda = 0.01$ for SPREAD, and dropout rate of 0.25 in feed-forward connections in encoder and decoder (as described in Appendix A.1) for regularization.

5.4 Results and Observations

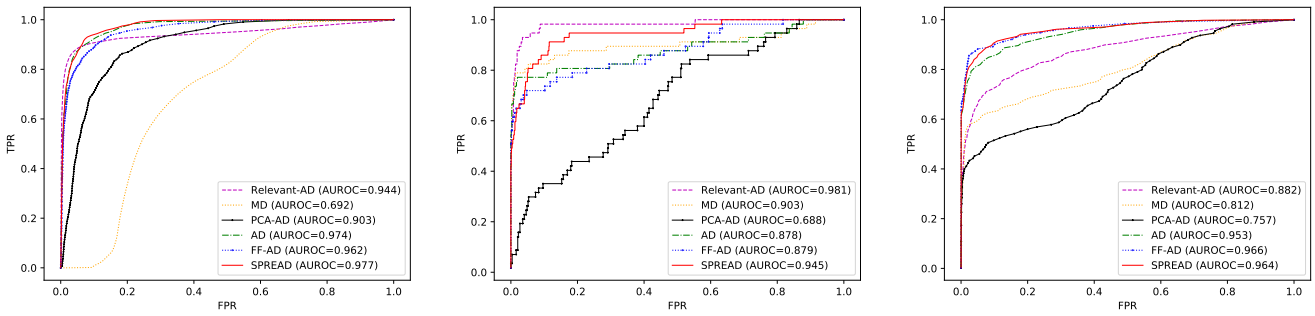
We make the following ***key observations*** from the results in Table 1 and Figure 3:

[2]Each fault or anomaly is spread over a period of time and has multiple anomalous points.

1. The non-temporal MD approach performs poorly across datasets highlighting the *temporal nature of anomalies*, and therefore, the applicability of temporal models including AD and SPREAD. It also suggests that Mahalanobis distance as applied in the error space (as in Equation 2) instead of original input space amplifies the effect of weak temporal anomalies.
2. PCA-AD does not perform well compared to FF-AD and SPREAD suggesting that *explicit dimensionality reduction via PCA leads to loss of information related to anomalous signatures*, whereas FF-AD and SPREAD are able to leverage the benefits of internal dimensionality reduction via the feedforward dimensionality reduction layer.
3. As expected, Relevant-AD – leveraging the knowledge of relevant sensors – is a strong baseline. This highlights the fact that *EncDec-AD performs well in low-dimensional cases* such as the Relevant-AD scenario. In other words, poor performance of AD compared to Relevant-AD highlights that *detecting anomalous signature is difficult when prior knowledge of relevant dimensions is not available* - which is often the case in practice. However, for Pulverizer and GHL datasets, we observe that AD performs better than Relevant-AD because in these cases the effect of anomaly originating in a sensor is also visible in other correlated sensors making it easier to detect anomalies due to amplification of anomalous signature when considering more sensors together.
4. SPREAD performs significantly better compared to other methods on most datasets (except Relevant-AD as discussed above). SPREAD performs better than or comparable to FF-AD highlighting the *regularizing effect of sparse connections*. Sparsity factors (Table 2) indicate sparse nature of connections in SPREAD compared to FF-AD. We measure sparsity factor as the fraction of weights with absolute value < 0.1 times the average of absolute weights.
5. We also tried Relevant-AD on Turbomachinery dataset with the two relevant sensors R_1 and R_2 considered independently, and observed a significant drop in performance compared to model using both the relevant sensors together as shown in Table 3. This suggests that *capturing correlation (or dependence) between sensors is important for detecting anomalies*.

6 Conclusion and Discussion

We observe that RNN based autoencoders for anomaly detection may yield sub-optimal performance in practice for high-dimensional time series. To address this, we have proposed SPREAD which explicitly provisions for dimensionality reduction layer that is trainable in an end-to-end manner along with the autoencoder and acts as a strong regularizer for high-dimensional time series modeling. SPREAD works in an online manner which is desirable for streaming applications. Experiments on a public dataset and two real-world datasets



(a) GHL (b) Turbomachinery (c) Pulverizer

Figure 3: ROC curves. TPR: True Positive Rate, FPR: False Positive Rate. Image best viewed on zooming and in color.

prove the efficacy of the proposed approach. Further, even though SPREAD uses dimensionality reduction internally, anomaly detection happens in the input feature space such that reconstruction error for each input dimension is accessible making the anomaly scores interpretable in practice. Our approach is generic and applicable to any high-dimensional time series anomaly detection. In future, it would be interesting to test SPREAD in an online learning setting for non-stationary time series, e.g. as in [Saurav *et al.*, 2018].

A Appendix

A.1 Long Short Term Memory (LSTM) Unit

We use a variant of LSTMs [Hochreiter and Schmidhuber, 1997] as described in [Zaremba *et al.*, 2014] in the recurrent hidden layers of RNN-ED. Consider $A_{n_1,n_2} : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_2}$ is an affine transform of the form $\mathbf{z} \mapsto \mathbf{W}\mathbf{z} + \mathbf{b}$ for matrix $\mathbf{W}$ and vector $\mathbf{b}$ of appropriate dimensions. The values for input gate $\mathbf{i}$, forget gate $\mathbf{f}$, output gate $\mathbf{o}$, hidden state $\mathbf{z}$, and cell activation $\mathbf{c}$ at time t are computed using the current input $\mathbf{x}_t$, the previous hidden state $\mathbf{z}_{t-1}$, and memory cell value $\mathbf{c}_{t-1}$ as given by Equations 4.

The time series goes through the following transformations iteratively at l-th hidden layer for $t = 1$ through T, where T is length of the time series:

$$\begin{pmatrix} \mathbf{i}_t^l \\ \mathbf{f}_t^l \\ \mathbf{o}_t^l \\ \mathbf{g}_t^l \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} A_{m+n,4n} \begin{pmatrix} \mathbf{D}(\mathbf{z}_t^{l-1}) \\ \mathbf{z}_{t-1}^l \end{pmatrix} \tag{4}$$

where cell state $\mathbf{c}_t^l = \mathbf{f}_t^l \cdot \mathbf{c}_{t-1}^l + \mathbf{i}_t^l \cdot \mathbf{g}_t^l$, $\mathbf{z}_t^l = \mathbf{o}_t^l \cdot tanh(\mathbf{c}_t^l)$, m is input dimension, and n is number of units present in the hidden layer. For a multilayered RNN with L hidden layers, the hidden state $\mathbf{z}_t^l$ at time t for l-th hidden layer is obtained from $\mathbf{z}_{t-1}^l$ and $\mathbf{z}_t^{l-1}$. For example, for $l = 1$, $\mathbf{z}_t^{l-1} \in \mathbb{R}^r$ s.t. $m = r$ in case of SPREAD encoder, and $\mathbf{z}_{t-1}^l \in \mathbb{R}^h$ s.t. $n = h$, and for $l > 1$, $m = n = h$. Dropout is used for regularization [Pham *et al.*, 2014] and is applied only to the non-recurrent connections, ensuring information flow across time-steps. $\mathbf{D}(\cdot)$ is dropout operator that randomly sets the dimensions of its argument to zero with probability equal to dropout rate, $\mathbf{z}_t^0$ equals the input at time t. The sigmoid (σ) and $tanh$ activation functions are applied element-wise.

A.2 Datasets Details

GHL: GHL dataset [Filonov *et al.*, 2016] contains data for normal operations of a gasoil plant heating loop, and faulty behavior (due to cyber-attacks) in a plant induced by changing the control logic of the loop. There are 14 main variables and 5 auxiliary variables: in our experiments, we consider 14 main variables, use fault IDs 25-48, and use *Danger* sensor as ground truth (1:Anomalous, 0:Normal). We downsample the original time-series by 4 for computational efficiency using 4-point average, and then take a window of 100 points to generate time-series instances.

Turbomachinery: This is a real-world dataset with per-minute sensor readings from 56 sensors, recorded for 4 days of operation with faulty signature being present for 1 hour before a forced shutdown. The sensors considered include temperature, pressure, control sensors, etc. belonging to different components of the machine. Out of these 56 sensors, the fault first appears in only 2 sensors. Eventually, few other sensors also start showing anomalous behavior.

Pulverizer: Pulverizer is a real-world dataset obtained from a pulverizer mill with per-minute sensor readings from 35 sensors. This dataset has sensor readings of 45 days of operation, and symptoms of fault start appearing intermittently for 12 hours before forced shutdown. The sensors considered include temperature, differential pressure, load, etc. belonging to different components of the machine. This dataset has 3 relevant sensors sufficient to identify the anomalous behavior.

References

[Aggarwal and Yu, 2001] Charu C Aggarwal and Philip S Yu. Outlier detection for high dimensional data. In *ACM Sigmod Record*, volume 30, pages 37–46. ACM, 2001.

[Ahmed *et al.*, 2007] Tarem Ahmed, Mark Coates, and Anukool Lakhina. Multivariate online anomaly detection using kernel re-

cursive least squares. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 625–633. IEEE, 2007.

[Da Xu *et al.*, 2014] Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.

[De Maesschalck *et al.*, 2000] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[Ding and Kolaczyk, 2013] Qi Ding and Eric D Kolaczyk. A compressed pca subspace method for anomaly detection in high-dimensional data. *IEEE Transactions on Information Theory*, 59(11):7419–7433, 2013.

[Erfani *et al.*, 2016] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

[Feng and Simon, 2017] Jean Feng and Noah Simon. Sparse-input neural networks for high-dimensional nonparametric regression and classification. *arXiv preprint arXiv:1711.07592*, 2017.

[Filonov *et al.*, 2016] Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model. *NIPS Time Series Workshop 2016, arXiv preprint arXiv:1612.06676*, 2016.

[Gugulothu *et al.*, 2017] Narendhar Gugulothu, Vishnu TV, Pankaj Malhotra, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Predicting remaining useful life using time series embeddings based on recurrent neural networks. *International Journal on Prognostics and Health Management, IJPHM, arXiv preprint arXiv:1709.01073*, 2017.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hundman *et al.*, 2018] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. *arXiv preprint arXiv:1802.04431*, 2018.

[Keller *et al.*, 2012] Fabian Keller, Emmanuel Muller, and Klemens Bohm. Hics: high contrast subspaces for density-based outlier ranking. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 1037–1048. IEEE, 2012.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kriegel *et al.*, 2009] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 831–838. Springer, 2009.

[Malhotra *et al.*, 2015] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long Short Term Memory Networks for Anomaly Detection in Time Series. In *ESANN, 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 89–94, 2015.

[Malhotra *et al.*, 2016a] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. In *Anomaly Detection Workshop at 33rd International Conference on Machine Learning (ICML 2016)*. CoRR,https://arxiv.org/abs/1607.00148, 2016.

[Malhotra *et al.*, 2016b] Pankaj Malhotra, Vishnu TV, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Multi-Sensor Prognostics using an Unsupervised Health Index based on LSTM Encoder-Decoder. *1st ACM SIGKDD Workshop on ML for PHM. arXiv preprint arXiv:1608.06154*, 2016.

[Pham *et al.*, 2014] Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. Dropout improves recurrent neural networks for handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR)*, pages 285–290. IEEE, 2014.

[Saurav *et al.*, 2018] Sakti Saurav, Pankaj Malhotra, Vishnu TV, Narendhar Gugulothu, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Online anomaly detection with concept drift adaptation using recurrent neural networks. 2018.

[Scardapane *et al.*, 2017] Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

[Tibshirani, 1996] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[Tucker *et al.*, 2001] Allan Tucker, Stephen Swift, and Xiaohui Liu. Variable grouping in multivariate time series via correlation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(2):235–245, 2001.

[Vishnu *et al.*, 2017] TV Vishnu, Narendhar Gugulothu, Pankaj Malhotra, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Bayesian networks for interpretable health monitoring of complex systems. *AI4IOT Workshop at International Joint Conference on Artificial Intelligence IJCAI*, 2017.

[Wen *et al.*, 2016] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2074–2082, 2016.

[Yi *et al.*, 2017] Subin Yi, Janghoon Ju, Man-Ki Yoon, and Jaesik Choi. Grouped convolutional neural networks for multivariate time series. *arXiv preprint arXiv:1703.09938*, 2017.

[Yoon and Hwang, 2017] Jaehong Yoon and Sung Ju Hwang. Combined group and exclusive sparsity for deep neural networks. In *International Conference on Machine Learning*, pages 3958–3966, 2017.

[Zaremba *et al.*, 2014] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[Zimek *et al.*, 2012] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012.