An Interpretable Predictive Model for Early Detection of Hardware Failure

Artsiom Balakir, Alan Yang, and Elyse Rosenbaum
Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign
1308 W. Main Street, Urbana, IL 61801, USA
Email: artsiombalakir@gmail.com, asyang2@illinois.edu, elyse@illinois.edu

***Abstract*—This paper develops an accurate yet interpretable machine learning framework for predicting field failures from time-series diagnostic data with application to datacenter hard disk drive failure prediction. Interpretable models are accountable: model reasoning can be verified by a domain expert for critical reliability tasks. We develop an attention-augmented recurrent neural network that visualizes the temporal information used to generate predictions; visualizations correlate with physical expectations. Finally, we propose a clustering-based method for discovering failure modes.**

***Index Terms*—System Reliability, Hard Disk Drives, Failure Prediction, Machine Learning, Interpretable Prediction**

I. Introduction

This paper develops an accurate yet interpretable machine learning framework for predicting field failures from time-series diagnostic data, with application to datacenter hard disk drive (HDD) failure prediction. The methodology is equally applicable to solid-state devices but HDD offer the benefit of large publicly available datasets. Recurrent neural network models such as gated recurrent units (GRU) [1] have achieved state-of-the-art performance for this task [2], but lack interpretability. Unlike black-box models, human-interpretable models have the advantage of accountability: model reasoning can be verified by a domain expert for critical reliability tasks.

Inspired by the RETAIN model proposed by Choi et al. for healthcare data [3], we augment a GRU-based failure prediction model with an *attention mechanism* that provides a visual heatmap of the temporal information used to generate any particular prediction. We find evidence that heatmap behavior correlates with physical expectations; furthermore, we provide a clustering-based method for discovering failure modes. We also show that the generated heatmaps are consistent over different runs. The code and preprocessed dataset used in our experiments are publicly available[1].

II. Dataset

Our experiments use publicly available data collected from over 30,000 drives in Backblaze's datacenter [2]. The task is to predict the probability of imminent drive failure p_{fail} using the past $T = 365$ days' of data. In the dataset, each drive reports data collected from 22 sensors once a day. Those sensor channels are known as Self-Monitoring, Analysis and Reporting Technology (SMART) attributes. The dataset also includes entries that indicate if a drive failed on the last reported day; note that some drives have $T < 365$ days' of data. The SMART attributes are listed in Table I; we refer to them as the *features* used for failure prediction. The models in this work consider only a subset of 7 features. The 7 were obtained using Markov blanket feature selection [4]. The discarded features are statistically independent of drive failure when conditioned on the selected features.

TABLE I: HDD dataset features

Read error rate(1)	Spin up time(3)
Start/stop count(4)	Reallocated sector count(5)
*Seek count(7)	*Power on hours(9)
Power cycle count(12)	SATA downshift error count(183)
End-to-end error IOEDC(184)	Reported uncorrectable errors(187)
Command timeout count(188)	High fly writes(189)
Airflow temperature(190)	Power off retract count(192)
*Load cycle count(193)	*Temperature(194)
*Current pending sector count(197)	Offline uncorrectable sector count(198)
UltraDMA CRC error count(199)	Heat flying hours(240)
*Total logical blocks written(241)	*Total logical blocks read(242)

*Selected features.
$^{()}$SMART ID.

III. Models

In this work, we compare the performance of the proposed model for failure prediction ("augmented GRU") with that of a baseline GRU model ("classical GRU"). The augmented GRU is a classical GRU with the addition of an attention mechanism that facilitates interpretablilty. The two models are described in sections III-A and III-B, respectively, and model training is discussed in section III-C. Hyperparameters, which include the split of training, validation, and testing data, the class weight and the learning rate, are kept the same for the two models; details are provided in section III-C.

A. Classical GRU

Fig. 1(a) illustrates the structure of the classical model – a time-reversed GRU model. We use a time-reversed model for which the sequence of data are input in the reversed order, since it performs better. The model consists of 2 GRU layers in series with a feed-forward neural network. For each day $1 \leq t \leq T$, x_t represents the length-7 vector of measurements

[1]github.com/artsiom-svm/interpretable_prediction_hdd
[2]backblaze.com/b2/hard-drive-test-data.html

on day t. For each time-point t, the network generates a low-dimensional vector b_t

$$b_t = f_{gru}(x_t \dots x_T), \tag{1}$$

where f_{gru} represents the trainable artificial neural network architecture shown in Figure 1(a). It consists of a linear embedding layer given by matrix W, a GRU, and a feed-forward neural network, connected in series. b_t represents, in essence, a summary of the reversed time series data $x_t, ..., x_T$. The probability p_{fail} that the drive fails on day T is given by

$$p_{fail} = \sigma(b_1), \tag{2}$$

where σ denotes the sigmoid function.

The sequence to vector mapping given by Eq. (1) obscures temporal information from the user; i.e., one cannot determine how far in advance a particular feature provides prognostically important information. Directly comparing the importance of different features is generally difficult for artificial neural network models.

B. Augmented GRU

An interpretable model may help drive reliability improvement or discover what sensors are most useful for failure diagnosis. We use the attention-augmented GRU model illustrated in Fig. 1(b), which resembles the RETAIN architecture proposed in [3]. The classical GRU is augmented with a reverse time attention mechanism in parallel. At time t, the attention mechanism generates a scalar importance weight a_t

$$a_t = g_{gru}(x_t \dots x_T), \tag{3}$$

where g_{gru} is implemented by a model structure similar to f_{gru}. Note that the linear embedding layer is shared between f_{gru} and g_{gru}. The vector $a = \left[a_1, ..., a_T\right]^{\mathsf{T}}$ represents the importance weighting; normalization is performed using the softmax function:

$$\tilde{a} = \left[\tilde{a}_1, \dots \tilde{a}_T\right]^{\mathsf{T}} = \text{softmax}(a). \tag{4}$$

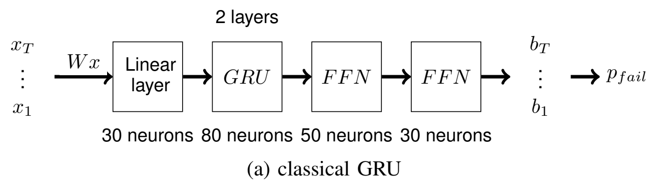
Equations (1) and (4) are combined to compute the *sensitivity* of feature j at time t, defined as

$$c_{t,j} = \tilde{a}_t w^{\mathsf{T}}(b_t \circ W^j), \tag{5}$$

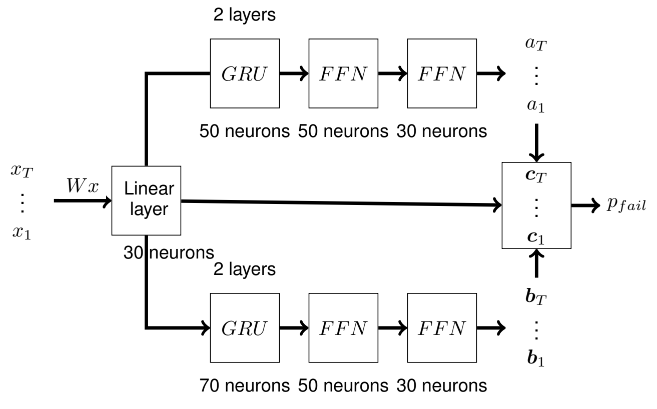
where $\circ$ denotes the element-wise product, w is a trainable vector of parameters, and W^j represents the j^{th} row of W. The sensitivity $c_{t,j}$ may be interpreted as a measure of the j^{th} feature's contribution to the predicted failure probability p_{fail} at time t, since p_{fail} has the form

$$p_{fail} = \sigma\left(\sum_{t=1}^{T}\sum_{j} c_{t,j} \cdot x_{t,j}\right). \tag{6}$$

The values of $c_{t,j}$ can be plotted as a two-dimensional heatmap in order to illustrate the influence of each feature across time.



(a) classical GRU



(b) augmented GRU

Fig. 1: Model structures, labeled with the size of each layer. The input and output are sensor data and predicted failure probability, respectively. Notation is consistent with Eq. (5). FFN stands for a single layer feed-forward network with tanh activation function.

C. Optimization

Both models are trained by minimizing a weighted binary cross-entropy loss over the model parameters. The loss function is given by

$$J(\phi) = -\frac{1}{N}\sum_{i=1}^{N}\Big[\lambda y^i \log(p^i_{fail}) + (1-y^i)\log(1-p^i_{fail})\Big], \tag{7}$$

where N is the number of drives and λ is a class weight hyperparameter. y^i represents the true label of the i^{th} drive; y^i = 0 corresponds to a healthy drive which does not fail on day T, whereas y^i = 1 corresponds to a drive which fails on day T. p^i_{fail} is the model's output for the i^{th} drive; it is the predicted probability of failure on day T and is used to predict the class label. If $p^i_{fail} \geq p_{threshold}$, then the predicted label is 1, otherwise it is 0. $p_{threshold}$ is a constant that is computed to satisfy a specific constraint, e.g. a maximum allowable false alarm rate. In our experiments, a value of $\lambda = 2.5$ was used; this was found to compensate for the fact that the dataset contains a 1:15 ratio of failing to healthy drives.

When training the augmented GRU model, we maximize the model's predictive performance Eq. (7) while simultaneously favoring zero-valued $c_{t,j}$. This is done to encourage the nonzero values of $c_{t,j}$ to carry meaning; a large positive $c_{t,j}$ implies that the corresponding $x_{t,j}$ provides evidence of failure, whereas a large negative $c_{t,j}$ provides evidence to the contrary. Specifically, we minimize the objective function

$$\mathcal{L} = J(\phi) + \rho R(\tilde{a}), \tag{8}$$

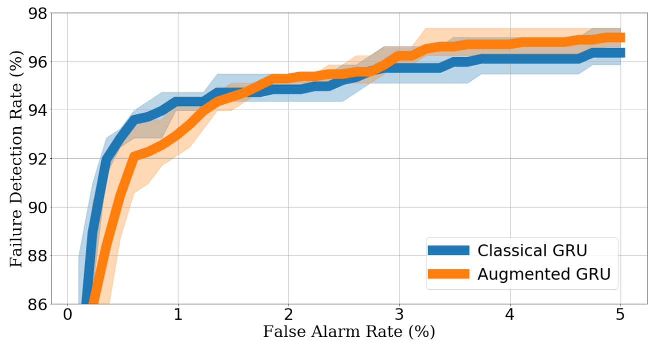


Fig. 2: Model performance of classical and augmented GRU. Note that small differences in model performance are not significant. Neural network training is a non-convex problem and slightly different solutions are produced on each training run. Shaded area shows range of performance over 6 training runs with different initial state. Solid line is mean curve.

TABLE II: Confusion Matrix

(a) Classical GRU

$N = 3,566$	Predicted Healthy	Predicted Failure
Actual Healthy	3265	36
Actual Failure	12	253

(b) Augmented GRU

$N = 3,566$	Predicted Healthy	Predicted Failure
Actual Healthy	3265	36
Actual Failure	15	250

where ρ is a regularization hyperparameter and R is given by

$$R(\tilde{a}) = \sum_{t=1}^{T} |\tilde{a}_t|. \tag{9}$$

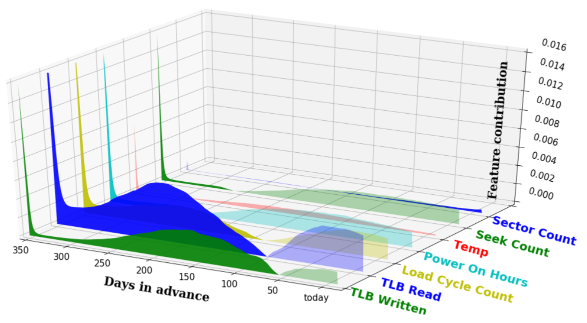
The regularization term R is the ℓ_1-norm of vector $\tilde{a}$. In our experiments, we chose $\rho = 10^{-5}$. We observed that the addition of the ℓ_1 regularization reduced the variance of model performance over different runs.

We used the Adam optimizer [5] with AMSGrad [6] to minimize Eq. (8) with a learning rate 5×10^{-4} and Adam parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We randomly split the dataset into training/validation/test sets with a 80/10/10 ratio. The split was fixed for both the classical GRU and augmented GRU models to ensure a fair comparison.

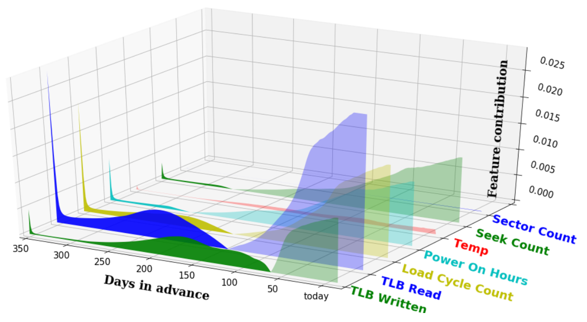
IV. Results

A. Comparison of Model Architectures

All models were trained using NVIDIA Tesla V100 GPUs. When comparing the two model architectures, we ensured that the number of trainable parameters was the same between the classical and augmented GRU models. Fig. 2 shows a comparison of the two models' receiver operating curves (ROC) across 6 runs. The ROC illustrates the trade-off between a model's false alarm rate and its failure detection rate as the threshold on p_{fail} is varied. The augmented and classical models perform comparably across different threshold values. Table II summarizes the performance of the two models when the threshold $p_{threshold}$ is chosen to meet a 1% false alarm rate constraint; these tables are known as confusion matrices. The values shown in Fig. 2 and Table II are computed on the test data.



(a) Average heatmap of failing drives



(b) Average heatmap of healthy drives

Fig. 3: Average sensitivity maps for (a) drives which fail at the end of the sequence and (b) drives which do not fail. The absolute value of sensitivity is shown on the z-axis, the features are shown on the y-axis, and reversed time in days is shown on the x-axis. Negative-valued $c_{t,j}$ are plotted with higher transparency.

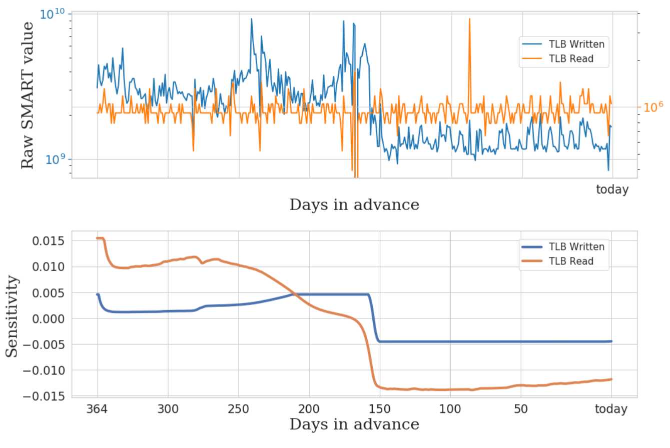
B. Learned Sensitivity Heatmaps

Eq. (6) indicates that negative sensitivity values push p_{fail} towards 0, since the values of $x_{t,j}$ are constrained to be non-negative. Large magnitude sensitivity implies high importance of the corresponding feature, and the sign of the sensitivity indicates whether the feature provides evidence that the device will be healthy at the end of the sequence or if it will fail.

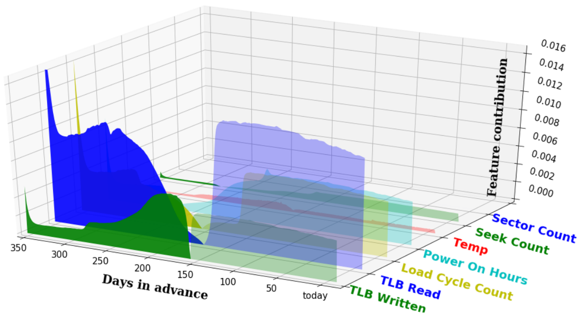
We generated heatmaps for each drive in the test set. Fig. 3 shows the average value of sensitivity for each feature for (a) drives that fail at the end of the sequence and (b) healthy drives. Only drives with data spanning $T = 365$ or longer were used to generate those plots, and negative-valued $c_{t,j}$ are plotted with higher transparency for contrast.

We observe that the features representing total logical blocks (TLB) written and read have high sensitivity for both healthy and failing drives. An examination of the heatmaps for individual drives reveals that, in a large number cases, the model assigns the highest sensitivity to TLB written and/or read, suggesting that those features are crucial in predicting the class label.

Data for a particular drive are shown in Fig. 4. Assuming that TLB written is a good measure of workload, Fig. 4(a) shows that this drive had a marked reduction in its workload around 150 days in advance of time T. The sensitivity measures for two features, TLB written and TLB read, show a step-down at the same time, changing from high and positive to high and



(a)



(b)

Fig. 4: (a) Top: Daily TLB written and TLB read SMART attributes for a single drive. Bottom: Corresponding plot of the sensitivity values. (b) Complete heatmap for this drive.

negative (Fig. 4(b)). It is interesting that a reduction in TLB written causes the sign of the sensitivity measures for both TLB written and TLB read to change, even though this drive had a near-constant value of TLB read. One might conclude that read operations are better tolerated under a low workload.

We further notice that temperature seems to be not as important as other features. This may be an artifact of the experimental conditions, since all the drives were located in data centers where the environment is carefully controlled.

C. Clustering

It is worthwhile to investigate whether or not useful information can be extracted by grouping (clustering) similar heatmaps. In this section, we demonstrate how the sensitivity heatmaps can be clustered. Dynamic Time Warping (DTW) is used as a similarity measure between sensitivity maps for different drives. DTW was chosen because of its success for many time-series problems [7], [8]. Subsequently, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [9] algorithm is used to perform unsupervised clustering of pairwise similarities. It is was observed that each drive was assigned into one of two distinct clusters, labeled a and b. Cluster a contains only failing drives, whereas cluster b is a mixture of failing and healthy drives; see Fig. 5. We refer

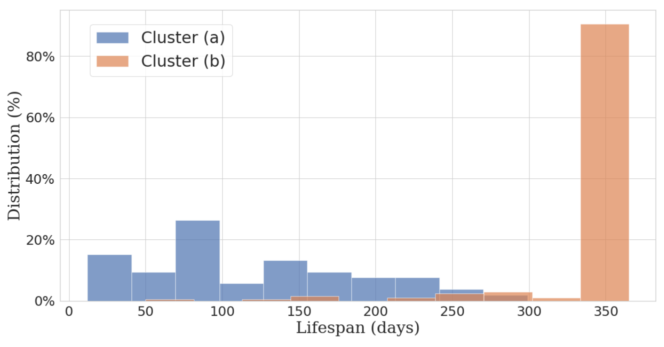
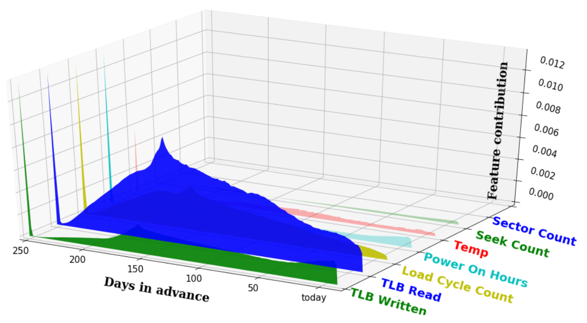
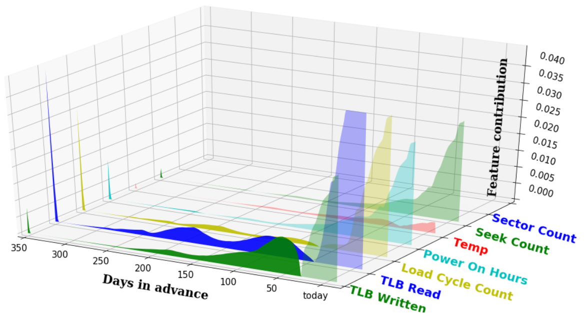


Fig. 5: Histograms of failed drives in clusters a and b

to cluster a as the failure cluster and cluster b as the regular drive cluster. We define the *cluster center* of each cluster to be the heatmap most similar to that of every other member, as measured by DTW. As indicated in Table III, the variance of the DTW-based distance within each cluster is small relative to the distance between cluster centers. This confirms that heatmaps can be easily clustered and that cluster centers may be used for a visual representation of the cluster.



(a)



(b)

Fig. 6: Sensitivity maps of the cluster centers. (a) Failed drive cluster. (b) Regular drive cluster. The absolute value of sensitivity is shown on the z-axis, the features are shown on the y-axis, and reversed time in days is shown on the x-axis. Negative-valued $c_{t,j}$ are plotted with higher transparency.

Fig. 6 shows the cluster centers. Plausibly, cluster a – the failure cluster – represents drives with manufacturing defects and the other cluster represents all other drives. We observe that for features with high sensitivity, the heatmap for cluster a exhibits nearly positive sensitivity over the lifespan, which

TABLE III: clusters statistics

	Cluster radius	distance to cluster		class labels		Mean lifespan
	$\mu \pm 3\sigma$	(a)	(b)	failure	healthy	in days (failure)
(a)	4.9± 2.3	0	8.5	53	1	119
(b)	0.2 ± 1.5	8.5	0	212	3300	352

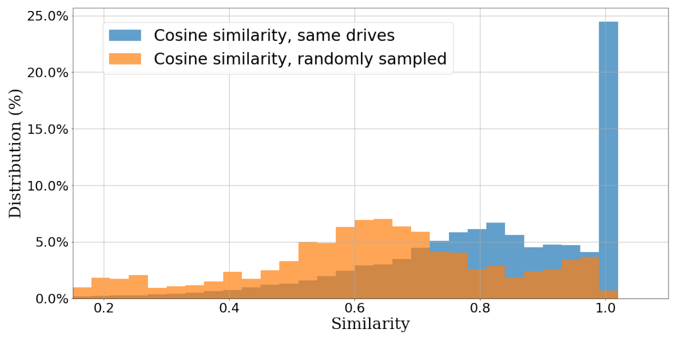


Fig. 7: Histogram of computed similarities between drives. Note that the distribution of similarities for the same drives over different runs is significantly shifted to the right compared to the distribution of similarities between randomly sampled pairs of drives.

is only a few months in most cases (Fig. 5).

D. Consistency of Sensitivity Heatmaps

Neural network models are generally trained using stochastic gradient descent and, as a result there is some randomness in the final learned model. To verify that the model learning approach generates consistent results, we will show that learned augmented GRU models with comparable prediction capabilities generate similar heatmaps for a fixed drive, across different training runs.

We performed 54 training runs with various initial state, optimization hyperparameters, and number of trainable parameters, and selected the models that have better than a 93% detection rate at a 1% false alarm rate. The subsequent analysis is carried out using the held-out test data. We compute average pairwise cosine similarities across $57,000$ heatmaps sampled uniformly from different drives and different training runs.

Since the goal is to compare the heatmaps' shapes rather than magnitudes, we scale each heatmap by its maximum value over time. Furthermore, any sensitivity value whose magnitude exceeds the mean by more than 3 standard deviations is clipped; this is done to lessen the effect of outliers. The average and standard deviation of $c_{t,j}$ computed for a given drive are denoted by μ and σ, respectively. The clipped and normalized sensitivity values are denoted as $\hat{c}_{t,j}$, which is given by

$$\hat{c}_{t,j} = \operatorname{sign}(c_{t,j}) \frac{\min\{|c_{t,j}|, |\mu| + 3\sigma\}}{\min\{\max_t |c_{t,j}|, |\mu| + 3\sigma\}}. \qquad (10)$$

Note that the set of all $\hat{c}_{t,j}$ for a given drive is a $T \times 7$ matrix. In order to measure similarity, we compute the cosine similarity between maps after flattening those matrices to vectors of length $7 \cdot T$.

A histogram of the computed cosine similarity measures is provided in Fig. 7; from that data, one obtains an average and standard deviation of 0.80 ± 0.07 for the same drive and 0.58 ± 0.26 for different drives. The learned models for a single drive have a similarity measure (0.80) that is significantly larger than the baseline (0.58). This indicates that the heatmaps generated following different training runs are similar.

V. Conclusion

An attention-based recurrent neural network model is used as an interpretable model for hard drive failure prediction. The model produces a heatmap illustration of the usefulness of different features across time for predicting impending drive failure, and we find that the heatmaps visually correlate with physical expectations. We believe that interpretable machine learning has the potential to provide insights to engineers working on other data-driven reliability problems.

Acknowledgment

This material is based on work supported by a Research Experience for Undergraduates supplement to National Science Foundation Grant No. CNS 16-24822 (Illinois) and the industry members of the CAEML I/UCRC.

This work utilizes resources supported by the National Science Foundation's Major Research Instrumentation program, grant #1725729, as well as the University of Illinois at Urbana-Champaign.

References

[1] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *Proc. Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, 2014.

[2] F. D. S. Lima, G. M. R. Amaral, L. G. M. Leite, J. P. P. Gomes, and J. C. Machado, "Predicting failures in hard drives with lstm networks," *razilian Conference on Intelligent Systems*, 2017.

[3] E. Choi, M. T. Bahadori, J. A. Kulas, A. Schuetz, W. F. Stewart, and J. Sun, "Retain: an interpretable predictive model for healthcare using reverse time attention mechanism," *Proc. of Neural Information Processing Systems (NIPS)*, pp. 3504–3512, 2016.

[4] A. Yang, A. Ghassami, M. Raginsky, N. Kiyavash, and E. Rosenbaum, "Model-augmented nearest-neighbor estimation of conditional mutual information for feature selection," *arXiv:1911.04628 [cs.LG]*, 2019.

[5] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2015.

[6] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *International Conference on Learning Representations*, 2018.

[7] T. Giorgino, "Computing and visualizing dynamic time warping alignments in r: The dtw package," *Journal of Statistical Software*, vol. 31, no. 7, pp. 1–24, 2009. [Online]. Available: http://www.jstatsoft.org/v31/i07/

[8] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli, "Matching incomplete time series with dynamic time warping: An algorithm and an application to post-stroke rehabilitation," *Artificial Intelligence in Medicine*, vol. 45, no. 1, pp. 11–34, 2008.

[9] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Proceedings of the 2nd international conference on knowledge discovery*, 1996.