CAUSAL INFERENCE FOR EARLY DETECTION OF HARDWARE FAILURE

BY

ALAN YANG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Bachelor of Science in Electrical and Computer Engineering
in the College of Engineering of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Adviser:

Professor Elyse Rosenbaum

ABSTRACT

Many modern hardware systems are equipped with sensors which record time-series diagnostic data. These sensors enable data-driven failure prediction that can reduce the need for component redundancy and lengthen lifetime specifications, by allowing for identification and proactive replacement of a soon-to-fail component. In this work, we develop a causal inference framework for predicting data center hard disk drive failures using multivariate time series recordings of temperature, read error rate, and other attributes. Information-theoretic measures are developed to quantify relationships between sensor variables, select prognostic features, and train a predictor. Finally, a recurrent neural network demonstrating high predictive accuracy and a low false alarm rate is developed, using field data collected from an operating data center.

ACKNOWLEDGMENTS

I would like to thank Professor Elyse Rosenbaum for advising me and making this project possible. I would also like to thank Professor Maxim Raginsky, Professor Negar Kiyavash, and Amiremad Ghassami for their invaluable guidance and feedback.

Furthermore, I would like to thank the Center for Advanced Electronics through Machine Learning and its sponsors for supporting this research, and in particular Chris Cheng and Yongjin Choi from Hewlett Packard Enterprise for their advice and many useful discussions.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS v

LIST OF SYMBOLS vi

CHAPTER 1 INTRODUCTION 1

1.1 Motivation for Data-Driven Reliability 1

1.2 Why Causal Inference? 2

1.3 The Baidu Dataset 3

CHAPTER 2 METHODS 5

2.1 Notation and Overview 5

2.2 Causal Inference for Feature Selection 6

2.3 Markov Blanket Induction 8

2.4 Conditional Mutual Information Estimation 8

2.5 Recurrent Neural Networks for Time-Series Prediction 10

CHAPTER 3 RESULTS AND DISCUSSION 13

3.1 Feature Selection 13

3.2 Evaluation using RNNs 14

3.3 Discussion 14

CHAPTER 4 CONCLUSION 16

REFERENCES 17

LIST OF ABBREVIATIONS

CMI	Conditional Mutual Information
DAG	Directed Acyclic Graph
DR	Detection Rate
ECC	Error Correction Code
FAR	False Alarm Rate
GRU	Gated Recurrent Unit
HDD	Hard Disk Drive
i.i.d.	Independent and Identically Distributed
MB	Markov Blanket
MI	Mutual Information
RAID	Redundant Array of Independent Disks
RNN	Recurrent Neural Network
SMART	Self-Monitoring, Analysis, and Reporting Technology

LIST OF SYMBOLS

$X \perp Y$	X and Y statistically independent
$X \perp Y \mid Z$	X and Y independent when conditioned on Z
$X \perp Y \mid_{\mathcal{G}} Z$	X and Y are d-separated by Z
$X \setminus Y$	X set minus Y
$\mathbb{E}_Z[\cdot]$	Expected value with respect to Z
$X \odot Y$	Hadamard product of X and Y

CHAPTER 1

INTRODUCTION

1.1 Motivation for Data-Driven Reliability

Reliability is a critical concern in hardware design. In general, system designers attempt to mitigate the negative effects of component-level failures by introducing redundancy and conservatively limiting system lifetime specifications. For example, Redundant Array of Independent Disks (RAID) is a data storage paradigm used by commerical datacenters [1]. While many variants of RAID are used, all share the same goal of limiting unrecoverable data loss. Mirroring is one technique used by RAID; a single piece of data is copied to several drives, ensuring its immunity to any single disk failure. There is a clear trade-off between reliability and system cost.

At the same time, telemetry and data storage technologies have enabled diagnostic sensor arrays to be embedded within hardware systems. System designers collect large volumes of sensor data streamed from operational devices in the field; this data is typically used to drive design choices and assist in field repairs. An intelligent failure prediction system would allow for a new reliability paradigm that relies more heavily on data analytics rather than expensive redundancy or early retirement of hardware.

This work develops a technique for predicting hard disk drive (HDD) failures. Using accurate, real-time failure prediction, RAID systems could reduce the need for disk redundancy while ensuring data reliability by copying only the contents of those drives with a high probability of failing in the near future. Such a system is enabled by the Self-Monitoring, Analysis, and Reporting Technology (SMART) standard widely adopted by HDD manufacturers. Modern HDDs are equipped with sensors that measure attributes such as internal temperature, read error rate, and counts of error correction code (ECC) invocations [1]. HDDs can be polled periodically for up to 81 diagnostic SMART attributes [2].

Prior works applied supervised machine learning techniques to predict HDD failure prediction from SMART data, using modeling techniques such as Naive Bayes [3], support vector machines [4], and recurrent neural networks (RNNs) [5]. The recurrent neural network developed in [5] reports a detection rate (DR) of 97.71% with a false alarm rate (FAR) of 0.06% on their data, the best results found in the open literature. At the same time, no known previous study has demonstrated an ability to transfer predictive models from one device to another (e.g. between HDD models).

This work builds on prior work by proposing a causal inference framework for discovering causal relationships between SMART attributes. Successful discovery of causal relationships between sensor variables allows for principled feature selection for failure prediction, analysis of HDD failure mechanisms, and ultimately causal transfer learning.

1.2 Why Causal Inference?

It is reasonable to question the need for feature selection in the failure prediction task. Indeed, state-of-the-art learning algorithms utilizing regularization techniques do not require feature selection to overcome the "curse of dimensionality" and achieve high prediction performance.

However, in data-driven system failure prediction, feature selection and causal modeling have a critical role to play. For datacenters managing and monitoring hundreds and thousands of drives, feature dimensionality reduction can reduce data storage requirements for analyzing SMART attributes.

Classical, supervised machine learning assumes identically and independently distributed data (i.i.d.). In other words, the conditions under which SMART measurements are taken are assumed to be universally identical. Since different datacenters expose drives to different operating conditions and workloads, the i.i.d. assumption does not hold. Furthermore, variability in failure conditions between different drive makes and models is expected, even if all drives record the same set of SMART attributes.

Changes in drive models and environments can be viewed as disturbances in the distribution of variables, and thus violations of the i.i.d. assumption. Therefore, a model trained using data collected from one drive model in one datacenter may not perform well for data collected from another. However, knowledge of causal relationships can be used to develop models robust to particular distribution changes, e.g. different drive models reporting the same SMART attributes. For this same reason, an understanding of causal relationships allows for improvements over the feature selection strategies used in [3],[4], and [5], which attempt to rank features based on correlation.

1.3 The Baidu Dataset

The data provided by the authors of [4] is used for this study. This "Baidu dataset" contains SMART attributes collected from a single model of HDD used in a Baidu-run datacenter. Of the 23,395 HDDs represented in the dataset, 433 failed within the data collection window. Table 1.1 lists the 10 SMART attributes reported for each drive.

HDD manufacturers report SMART values in two formats: raw and normalized. Raw values represent physical sensor measurements; normalized SMART values correspond to raw values quantized to 253 levels. Manufacturers provide both formats as an attempt to provide more human-interpretable SMART values. In the Baidu dataset, eight of the ten reported attributes are given by normalized SMART values. The authors of [4] who compiled the dataset likely included raw values specifically for the reallocated sector count and current pending sector count, since these quickly varying counter values likely sensitive to the lossy compression performed during vector quantization.

Drives in the dataset are polled for their SMART attributes hourly. For the 433 drives that fail within the measurement period, up to a week's worth of data prior to failure is collected. Twenty days' worth of data are collected for all other drives. Finally, all provided data are scaled to the range $[0, 1]$.

SMART Attribute Name	**SMART Atrribute Name**
Raw Read Error Rate	Spin Up Time
Reported Uncorrectable Errors	High Fly Writes
Reallocated Sector Count*	Temperature Celsius
Hardware ECC Recovered	Seek Error Rate
Current Pending Sector Count*	Power-on Hours

Table 1.1: SMART attributes provided in the Baidu dataset. * denotes attributes for which both raw and normalized values are provided; only normalized values are provided for the other attributes.

CHAPTER 2

METHODS

2.1 Notation and Overview

The Baidu data are denoted as:

$$\mathcal{D} = \left\{ \left\{ (X_k^{(i)})_{t=1}^{T} \right\}_{k=1}^{n}, (Y^{(i)})_{t=1}^{T} \right\}_{i=1}^{N} \tag{2.1}$$

Here, $X^{(i)}$ and $Y^{(i)}$ are random variables representing the SMART data and health status of the i-th drive, respectively. N is the number of drive examples, and n is the number of SMART attributes reported by each drive. Data are collected from time $t = 1$ to $t = T$. $X^{(i)}$ and $Y^{(i)}$ represent time series; the k-th SMART attribute reported by the i-th drive at time t is denoted as $(X_k^{(i)})_t$, and the health status of the i-th drive at time t is $(Y^{(i)})_t$.

There are many possible representations for Y. Following the scheme in [5], Y is taken to be a discrete random variable such that

$$(Y^{(i)})_t = \begin{cases} 1 & \text{Drive } i \text{ fails within time } [t, t+12] \\ 3/4 & \text{Drive } i \text{ fails within time } (t+12, t+24] \\ 1/2 & \text{Drive } i \text{ fails within time } (t+24, t+36] \\ 1/4 & \text{Drive } i \text{ fails within time } (t+36, t+168] \\ 0 & \text{else} \end{cases} \tag{2.2}$$

where time is measured in hours. Drives in $\mathcal{D}$ that do not fail have health status 0 at all times, while drives that fail have nonzero health status that increase in time as the failure event approaches.

Formally, we aim to predict Y from $\left\{X_k\right\}_{k\in\mathcal{F}}$, where the set $\mathcal{F} \subset \{1, .., n\}$ indexes the selected features used for prediction. Sections 2.2, 2.3, and 2.4

describe the causal discovery and feature selection methods used, and 2.5 describes the resulting sequence-to-sequence failure prediction model.

2.2 Causal Inference for Feature Selection

Bayesian networks provide an intuitive framework for modeling causal relationships. First, it is necessary to define causality for random variables given observational data. The following definitions follow those in [6], [7], and [8].

Definition 2.2.1. Feature X_k is a prima facie cause of Y with respect to observations $\mathcal{O} = \{X_k\}_{k=1,..,n}$ if

$$P(Y|\mathcal{O}) \neq P(Y|\mathcal{O} \setminus X_k)$$

In other words, we say that X_k is a cause of Y if knowledge of X_k changes the distribution of Y. Clearly, if $X_k \perp Y$ then X_k is not a cause of Y. Furthermore, knowledge of variable dependencies may be used to identify other non-causal features. For example, consider the case where $X_k \not\perp Y$, yet there is another feature X_j such that $X_k \perp Y|X_j$. Despite the fact that $X_k \not\perp Y$, X_k does not cause Y. Bayesian networks provides a concise framework for these concepts.

Definition 2.2.2. A Bayesian Network graphically represents a probability distribution over a set of n variables $V = \{X_1, X_2, ..., X_n\}$. Variables are represented as vertices in the directed acyclic graph (DAG) $\mathcal{G} = \{V, E\}$, where E is the set of edges, satisfying:

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P\big(X_i|\text{Parents}(X_i)\big)$$

where the set $\text{Parents}(X_i) = \{X_j \in V : \{X_j, X_i\} \in E\}$ represents the parents of X_i in $\mathcal{G}$. In this case, the DAG $\mathcal{G}$ is said to be faithful to the joint distribution over V.

This definition suggests that the joint distribution over all variables can be factored into "local" distributions represented by edges between parent and child variables. Specifically, for variables X_i, X_j, and Y, we can make a few observations.

1. If X_i and Y do not share an edge, $X_i \perp Y$.
2. The connection patterns $X_i \rightarrow X_j \rightarrow Y$ and $X_i \leftarrow X_j \rightarrow Y$ both imply that $X_i \perp Y|X_j$. In this case, X_i may be eliminated as a feature provided that X_j is measured.
3. Since we may assume that disk failure Y cannot be a cause of SMART attributes, we rule out the possibility of the structure $X_i \rightarrow X_j \leftarrow Y$.

It is clear that conditional independence tests can be used to identify features to eliminate, as in case 2. The relationship between conditional independencies and DAGs is formalized by the notion of d-separation.

Definition 2.2.3. Let A, B, and C be disjoint subsets of vertices in the DAG $\mathcal{G} = \{V, E\}$, and p be an undirected path in $\mathcal{G}$. C blocks p if there is a vertex $z \in V$ on p satisfying one of:

- z has converging arrows along p, and neither z nor any of its descendants are in C.
- z does not have converging arrows along p, and is in C.

Then, C d-separates A and B, written as $A \perp_{\mathcal{G}} B|C$, if and only if every path from a vertex in A to a vertex in B. $\mathcal{G}$ is said to be a faithful Bayesian network if d-separation implies conditional independence between variables and vice-versa, i.e.

$$A \perp_{\mathcal{G}} B|C \leftrightarrow A \perp B|C$$

An example of d-separation for three variables is illustrated in Figure 2.1.

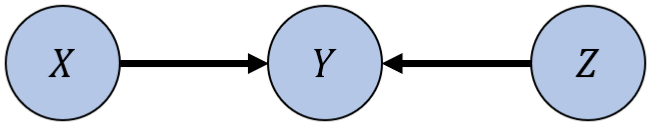

Figure 2.1: An example illustrating d-separation. $X \perp_{\mathcal{G}} Z$, but $X \not\perp_{\mathcal{G}} Z|Y$.

The above definitions imply that feature selection is equivalent to finding and eliminating the variables that are d-separated from Y. Note that since Y is not a cause of any X_i, it has no children. Therefore, we attempt to find and use the set Parents(Y) as prognostic features for predicting Y. The desired set of features is known as the Markov Blanket (MB) of Y, and an algorithm for finding such sets is proposed in Section 2.3.

2.3 Markov Blanket Induction

The Markov Blanket of Y, MB(Y), is a set containing the parents of Y, children of Y, and other parents of children of Y. Specifically, Y is d-separated from all other variables when conditioned on a minimal set MB(Y). Since we assume that Y is childless (see Section 2.2), we have MB(Y) = Parents(Y).

Algorithms exist for finding MB(Y), many of which use conditional independence testing to rule out possible connections to Y. By definition 2.2.3, in the true, faithful DAG $\mathcal{G} = \{V, E\}$,

$$X_i \in \text{MB}(Y) \text{ iff } \nexists S \subset V \text{ such that } Y \perp X_i | S \tag{2.3}$$

For each hypothetical edge of interest $X_i \rightarrow Y$, a series of binary hypothesis tests given by the form of Equation 2.4 may be performed. If H_0 is declared for any edge of interest between Y and X_i, conditioning on any set of variables S, it is eliminated. The following section describes a procedure for binary hypothesis testing using mutual information measures. Due to the limited size of our dataset, we limit the size of S to be at most one for statistical considerations.

$$\begin{cases} H_0: & Y \perp X_i | S \\ H_1: & Y \not\perp X_i | S \end{cases} \tag{2.4}$$

2.4 Conditional Mutual Information Estimation

Taking a probabilistic view of SMART attributes, conditional mutual information (CMI) is used as a metric for performing the conditional independence tests at the heart of causal inference. Formally, the CMI between random variables X and Y conditioned on Z is defined as follows (e.g. [9]).

$$I(X;Y|Z) = \mathbb{E}_Z\big[I(X;Y)|Z\big] \tag{2.5}$$

where the mutual information (MI) between X and Y is defined as

$$I(X;Y) = \int_{\mathcal{X}\times\mathcal{Y}} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right)dxdy \tag{2.6}$$

$\mathcal{X}$ and $\mathcal{Y}$ denote the respective domains of X and Y, and $p(\cdot,\cdot)$ and $p(\cdot)$ represent respective joint and marginal densities. Intuitively, the CMI is an information-theoretic quantity that quantifies the relative entropy of the joint distribution of $X, Y|Z$ from the product of the marginal distributions of $X|Z$ and $Y|Z$. Clearly, $I(X;Y|Z)$ is non-negative, and is equal to zero when X and Y are independent when conditioned on Z.

In light of these definitions, the conditional independence binary hypothesis test in Equation 2.4 can be carried out as follows:

$$\begin{cases} H_0 : & \hat{I}(X_i;Y|X_j) \leq \eta \\ H_1 : & \hat{I}(X_i;Y|X_j) > \eta \end{cases} \tag{2.7}$$

where $\hat{I}$ is the estimated CMI and η is a threshold, selected ad-hoc.

MI estimation from data can be a challenging task. It has been observed that "plug-in" estimators, which attempt to substitute empirically estimated distributions into the definition 2.6, are generally suboptimal [10]. For continuous variables, k-nearest-neighbor methods, first proposed in [11], are the state-of-the-art in the literature, and generalization to CMI estimation has been studied in [12], [13], and [14]. These results are combined with the MI estimator proposed by [9] to estimate CMI in the case that data are drawn from a combination of discrete and continuous domains. Since SMART data consists of a combination of continuous measurements and discrete status indicators, such an estimator is needed.

The proposed estimator is shown in Algorithm 1. In essence, it returns an average of local CMI estimations around each data point; care is taken to locally distinguish discrete distributions from continuous ones. In this work, the 1-norm is used as the distance metric $\|\cdot\|$; this is chosen to accommodate the multivariate nature of the SMART data. $\psi(\cdot)$ denotes the digamma function, which obeys the recurrence relation

$$\psi(z+1) = \psi(z) + 1/z \tag{2.8}$$

and is related to the harmonic numbers as follows:

$$\psi(z) = \sum_{k=1}^{z-1} \frac{1}{k} - \gamma \tag{2.9}$$

where γ is the Euler-Mascheroni constant. The digamma function is used to estimate the log in Equation 2.6, because $\psi(x) \approx \log x - \frac{1}{2x}$ for large x [11].

Algorithm 1 Proposed Conditional Mutual Information Estimator

```
 1: procedure MIXEDCMI($\{X_i, Y_i, Z_i\}_{i=1}^N$, $k$)
 2:     for $i \in \{1, ..., N\}$ do
 3:         $\epsilon_i =$ The k-th smallest distance among
 4:                 $\{d_{i,j} = \max\{\|X_i - X_j\|, \|Y_i - Y_j\|, \|Z_i - Z_j\|\} : j \neq i\}$
 5:         if $\epsilon_i = 0$ then
 6:             $\tilde{k}_i = \#$ samples with $d_{i,j} = 0$
 7:         else
 8:             $\tilde{k}_i = k$
 9:         end if
10:         $n_z = \#$ samples with $\|Z_i - Z_j\| \leq \epsilon_i$
11:         $n_{xz} = \#$ samples with $\max\{\|X_i - X_j\|, \|Z_i - Z_j\|\} \leq \epsilon_i$
12:         $n_{yz} = \#$ samples with $\max\{\|Y_i - Y_j\|, \|Z_i - Z_j\|\} \leq \epsilon_i$
13:         $\xi_i = \psi(\tilde{k}_i) - \psi(n_{xz}) - \psi(n_{yz}) + \psi(n_z)$
14:     end for
15:     return $\hat{I} = \frac{1}{N}\sum_{i=1}^N \xi_i$
16: end procedure
```

2.5 Recurrent Neural Networks for Time-Series Prediction

Feature selection helps a predictive model learn by reducing dimensionality. However, a model trained using only selected features should perform comparably to one trained using all available data. In order to evaluate our

CMI-based feature selection feature selection procedure, we train two recurrent neural network (RNN) models. The first considers the full set of 12 features, while the second uses the selected set of features.

Inspired by the RNN structure used in [5], we developed a model using the gated recurrent unit (GRU)[15]. Since we found the GRU architecture to outperform the simple RNN structure used in [5], we used it as our evaluator. In order to train deep neural networks using gradient-based methods, errors at the network's output need to be repeatedly differentiated backwards through the network in order to update the model parameters. Unfortunately, high-order derivatives numerically tend to zero. The GRU attempts to overcome this problem by storing a history of values seen in the past.

The GRU is visualized in Figure 2.2 and defined using the following recurrence relation:

$$y_t = \sigma_o(V_o \cdot h_t + b_o)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \sigma_h(W_h \cdot x_t + U_h \cdot (r_t \odot h_{t-1}) + b_h)$$

where the internal signals z_t and r_t are defined as:

$$z_t = \sigma_g(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z)$$

$$r_t = \sigma_g(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r)$$

The parameters are given by the W, U, and V matrices, as well as the bias vectors b. The latent variable at time t is represented by h_t, and the predicted output is y_t. Following standard practices [16], we use the sigmoid function for σ_g, the hyperbolic tangent for σ_h, and softmax for σ_o. The symbol $\odot$ represents the Hadamard (element-wise) product. At each time step, the model predicts a vector y_t as a one-hot encoding of the predicted drive health defined in Equation 2.2. Training is done using standard gradient-based backpropagation.

The models were trained and evaluated using 5 days' worth of data, sampled once every 6 hours. The data are preprocessed so that drives which fail do so at the very end of the sampling period. After training, the models are evaluated based on failure detection rate (DR) and false alarm rate (FAR), following the conventions used in [4] [5]. For each drive's data sequence, the value of the last predicted health status is used to assign a binary healthy/failed status. A drive is marked as failed if its last predicted health

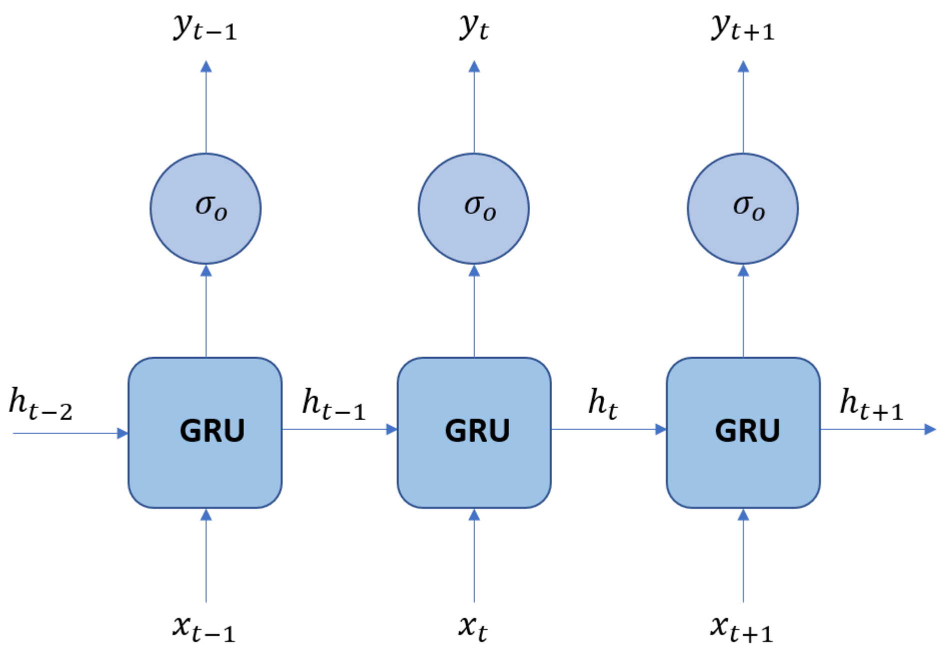

Figure 2.2: Gated Recurrent Unit (GRU) Structure.

status is greater than or equal to 1/2 on the scale given in Equation 2.2. The following equations may then be applied:

$$\text{Detection Rate (DR)} = \frac{\#\text{ predicted failed drives}}{\#\text{ true failed drives}} \tag{2.10}$$

$$\text{False Alarm Rate (FAR)} = \frac{\#\text{ false predicted drive failures}}{\#\text{ total predicted drive failures}} \tag{2.11}$$

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Feature Selection

The results of the conditional independence tests performed on the Baidu dataset are shown in Table 3.1. The following attributes have been eliminated:

- Spin Up Time
- Reallocated Sector Count
- Reported Uncorrectable Errors
- Current Pending Sector Count
- Raw Current Pending Sector Count

First conditioning on nothing, drive failure was found to be statistically independent of Reported Uncorrectable Errors and Current Pending Sector Count. When conditioning on single variables, Spin Up Time, Reallocated Sector Count, and Raw Current Pending Sector Count were also found to be conditionally independent. While the true DAG structure relating these variables is yet unknown, these tests rule out direct edges between the eliminated features and drive failure.

Discovered Conditional Independencies
Drive Failure $\perp$ Reported Uncorrectable Errors
Drive Failure $\perp$ Current Pending Sector Count
Drive Failure $\perp$ Spin Up Time | Raw Current Pending Sector Count
Drive Failure $\perp$ Reallocated Sector Count | Raw Current Pending Sector Count
Drive Failure $\perp$ Raw Current Pending Sector Count | Power On Hours

Table 3.1: Results of conditional independence testing.

3.2 Evaluation using RNNs

A summary of the GRU RNN performance metrics is given in Table 3.2. The model was developed using Tensorflow and trained using the RMSprop algorithm [17]. The RNN using selected features performs comparably to the model using all data, and both models have the same, low false alarm rate. The similarity in performance between the two RNNs supports the hypothesis that the features listed in Table 3.1 are not prognostic for failure prediction.

	RNN(all 12/12 features)	RNN(selected 7/12 features)
Detection Rate	98.85%	97.70%
False Alarm Rate	0.109%	0.109%

Table 3.2: RNN performance comparison using all 12 features (full) and 7 chosen features (selected).

3.3 Discussion

Several of the features found to be conditionally independent of failure had not been anticipated. The Reallocated Sector Count attribute is a measure of the number of bad sectors in an HDD. Since these sectors are no longer readable or writable, the HDD controller maps their contents to spare sectors. While the normalized Reallocated Sector Count was eliminated as a feature, the raw representation of the value was not. As explained in section 3.1, it is possible that important information in the raw Reallocated Sector Count was lost in the quantization process.

Notably, both raw and normalized Current Pending Sector Count values were eliminated. These attributes count the number of sectors currently "on probation" for being reallocated, due to failed reads and writes. However, a pending sector is not necessarily reallocated; temporary read and write failures can occur randomly due to the sensitive nature of the HDD's storage media.

Similarly, Reported Uncorrectable Errors keeps track of the number of read errors that could not be corrected by ECC. Since this information is entirely captured by counts of reallocated sectors and counts of ECC invocations, it is unlikely to be a useful feature for prediction.

Finally, Spin Up Time was removed as a feature. This attribute is a measure of the amount of time required for an HDD to accelerate its disk from rest to its operating speed. While irregular Spin Up Times could indicate mechanical problems, it is not an attribute that is often updated in datacenter HDDs. Drives are operated around the clock, and are not often power-cycled. As a result, the Spin Up Time of a drive is not measured regularly.

The RNN results given in Table 3.2 indicate that the removed features were not important to begin with, since very little performance was lost. While the model trained using the full dataset has a slightly higher detection rate, the inclusion of eliminated features is not immediately justified. For example, the model may be learning patterns within this specific dataset that do not generalize well. As a direction of future work, it is necessary to select features between different drive models operating in different datacenters, in order to find feature subsets that can be used to build general yet accurate predictors.

CHAPTER 4

CONCLUSION

In summary, a causal feature-selection method for choosing SMART attributes for HDD failure prediction is presented. The technique has been demonstrated on a dataset provided by one of Baidu's datacenters, and we were able to demonstrate strong failure detection rates while maintaining low false alarm rates. Our models' performance confirms the feasibility of failure detection from SMART data, supporting the results in [18] and [4]. Furthermore, we demonstrated that a GRU using the full dataset performs comparably to one using only our selected features. This suggests that certain SMART attributes may be eliminated from consideration without significant loss in performance, which is an important contribution for successful failure detection systems.

There are several steps for future work. Currently, our feature selection and modeling aims to predict failures as they occur in real-time. This is unfortunately not useful in practice, since datacenters need to be warned of impending failures in advance, so that at-risk data can be transferred in time. Once larger sets of SMART data are available, a thorough analysis of prediction time-in-advance can be performed.

REFERENCES

[1] J. Elerath, "Hard-disk drives: The good, the bad, and the ugly," *Commun. ACM*, vol. 52, no. 6, pp. 38–45, June 2009. [Online]. Available: http://doi.acm.org/10.1145/1516046.1516059

[2] BackBlaze, "Hard drive data and stats," 2013. [Online]. Available: https://www.backblaze.com/b2/hard-drive-test-data.html

[3] G. Hamerly and C. Elkan, "Bayesian approaches to failure prediction for disk drives," in *Proceedings of the Eighteenth International Conference on Machine Learning*, 12 2001, pp. 202–209.

[4] B. Zhu, G. Wang, X. Liu, D. Hu, S. Lin, and J. Ma, "Proactive drive failure prediction for large scale storage systems," in *2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST)*, May 2013, pp. 1–5.

[5] C. Xu, G. Wang, X. Liu, D. Guo, and T. Y. Liu, "Health status assessment and failure prediction for hard drives with recurrent neural networks," *IEEE Transactions on Computers*, vol. 65, no. 11, pp. 3502–3508, Nov 2016.

[6] A. E. Isabelle Guyon, Constantin Aliferis, "Causal feature selection," Clopinet, Tech. Rep., 2007.

[7] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 171–234, 2010.

[8] J. Pearl, *Causality.* Cambridge university press, 2009.

[9] W. Gao, S. Kannan, S. Oh, and P. Viswanath, "Estimating mutual information for discrete-continuous mixtures," in *Advances in Neural Information Processing Systems*, 2017, pp. 5988–5999.

[10] L. Paninski, "Estimation of entropy and mutual information," *Neural computation*, vol. 15, no. 6, pp. 1191–1253, 2003.

[11] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical review E*, vol. 69, no. 6, p. 066138, 2004.

[12] S. Frenzel and B. Pompe, "Partial mutual information for coupling analysis of multivariate time series," *Physical review letters*, vol. 99, no. 20, p. 204101, 2007.

[13] M. Vejmelka and M. Paluš, "Inferring the directionality of coupling with conditional mutual information," *Physical Review E*, vol. 77, no. 2, p. 026214, 2008.

[14] I. Vlachos and D. Kugiumtzis, "Nonuniform state-space reconstruction and coupling detection," *Physical Review E*, vol. 82, no. 1, p. 016207, 2010.

[15] K. Cho, B. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.

[17] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[18] C. Xu, G. Wang, X. Liu, D. Guo, and T.-Y. Liu, "Health status assessment and failure prediction for hard drives with recurrent neural networks," *IEEE Transactions on Computers*, vol. 65, no. 11, pp. 3502–3508, 2016.