Technical Report

System Analysis and Modeling Group

Hasso-Plattner Institute

University of Potsdam

January 30, 2018

Christian M. Adriano

Sona Gharhemani

Holger Giese

Contents

1. Introduction

This report details the technical steps taken to train machine learning models and compare decisions make with these models. The report is divided in three parts: training, validating, ranking decisions, computing reward. The steps we followed are depicted in Figure 1.

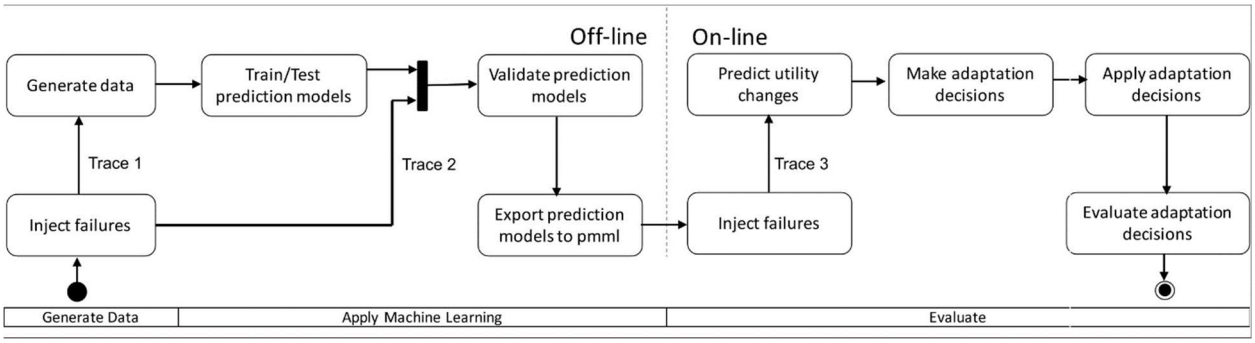

Figure 1 Methodology

The methodology that we followed divided between an off-line training process (performed in R) and an online execution of the trained models, which is performed in Java (utilizes an XML format exported by R). Both online and offline processes depend on injection of real failures. Below we explain each of these activities and provide data that was generated.

2. Inject Failures

We have two types of failure traces: sampled and full. The sampled failure traces consist of 5, 25, and 50 failures that are randomly injected to components. The full trace consists of multiple different set of failures.

3. Generate Data

4. Train and Test

The first step is to create Training comprises the machine learning models to predict utility functions

5. Validate Prediction Model

6. Export Prediction Model to pmml

The prediction model is developed in R and exported to an XML format that can later be instantiated and called from Java (see Utility Change Predictor in Figure 2).

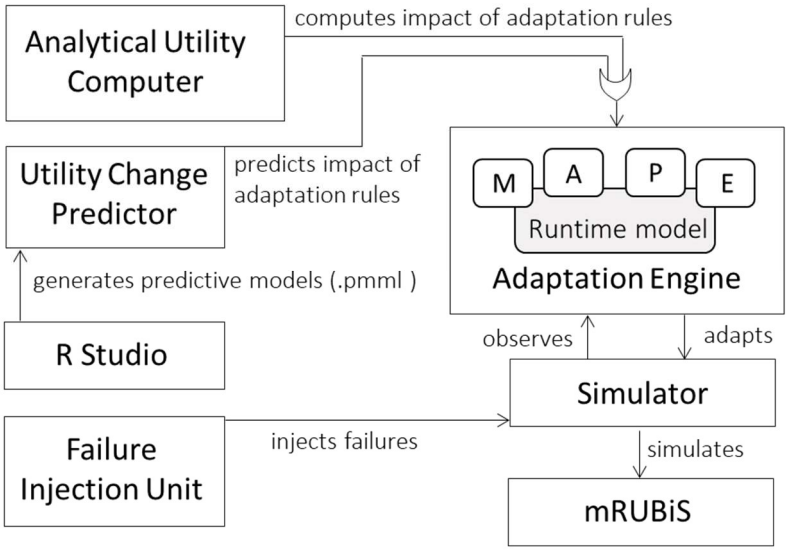

Figure 2 - On-line Prediction Architecture

7. Predict Utility Change

8. Make Adaptation Decision

A decision consists of an order list of rules to be applied. The order of these rules in the list follow a decreasing value of utility change. Since distinct prediction models have different prediction errors when computing the value of utility change, we obtain lists with different order. To compare these lists we used similarity metrics, which we investigated their behavior in depth.

8.1. Uncertainty Analysis

8.1.1. Variance of metrics and variance of reward

Below are variance comparisons for models trained with all three dataset sizes (1k, 3k, and 9k), respectively in Table 1, Table 2, and Table 3.

These tables show that system reward does not vary as much the similarity values between predicted and optimal decisions (order lists of lists to be applied). This is positive results because it suggests that the propagation of prediction errors is followed by a reduction in error impact after each step.

Table 1 - Reward versus Similarity Metric Variance (1K dataset trained models)

Model	Reward	Jaccard	Kendall	DCG
Linear	0.0E+00	3.0E-02	1.5E-04	9.1E-03
Saturating	3.1E-05	3.6E-02	4.6E-05	1.7E-02
Discontinuous	3.0E-05	5.2E-02	1.4E-04	1.9E-02
Combined	4.0E-03	6.6E-02	9.1E-06	2.9E-02
Var(Reward) > Var(Metric)	-	**0**	**1**	**0**

Table 2 - Reward versus Similarity Metric Variance (3K dataset trained models)

Model	Reward	Jaccard	Kendall	DCG
Linear	0.0E+00	1.8E-02	1.8E-04	1.4E-02
Saturating	3.1E-05	2.1E-02	1.1E-04	6.7E-03
Discontinuous	3.0E-05	4.2E-02	2.7E-04	1.5E-02
Combined	4.0E-03	4.2E-02	4.1E-04	2.2E-02
Var(Reward) > Var(Metric)	-	**0**	**1**	**0**

Table 3 - Reward versus Similarity Metric Variance (9K dataset trained models)

Model	Reward	Jaccard	Kendall	DCG
Linear	0.0E+00	1.1E-02	5.9E-06	3.3E-03
Saturating	3.3E-11	2.0E-02	1.2E-04	7.0E-03
Discontinuous	2.9E-08	1.5E-02	1.5E-05	8.4E-03
Combined	4.1E-05	3.3E-02	2.3E-04	1.7E-02
Var(Reward) > Var(Metric)	-	**0**	**0**	**0**

8.2. Correlation between reward and similarity metrics

Although we noted that system reward does not vary more than the similarity metrics, variance comparison does not tell us about the direction and intensity of a potential co-variance. We studied that by analyzing the correlation between reward and similarity values.

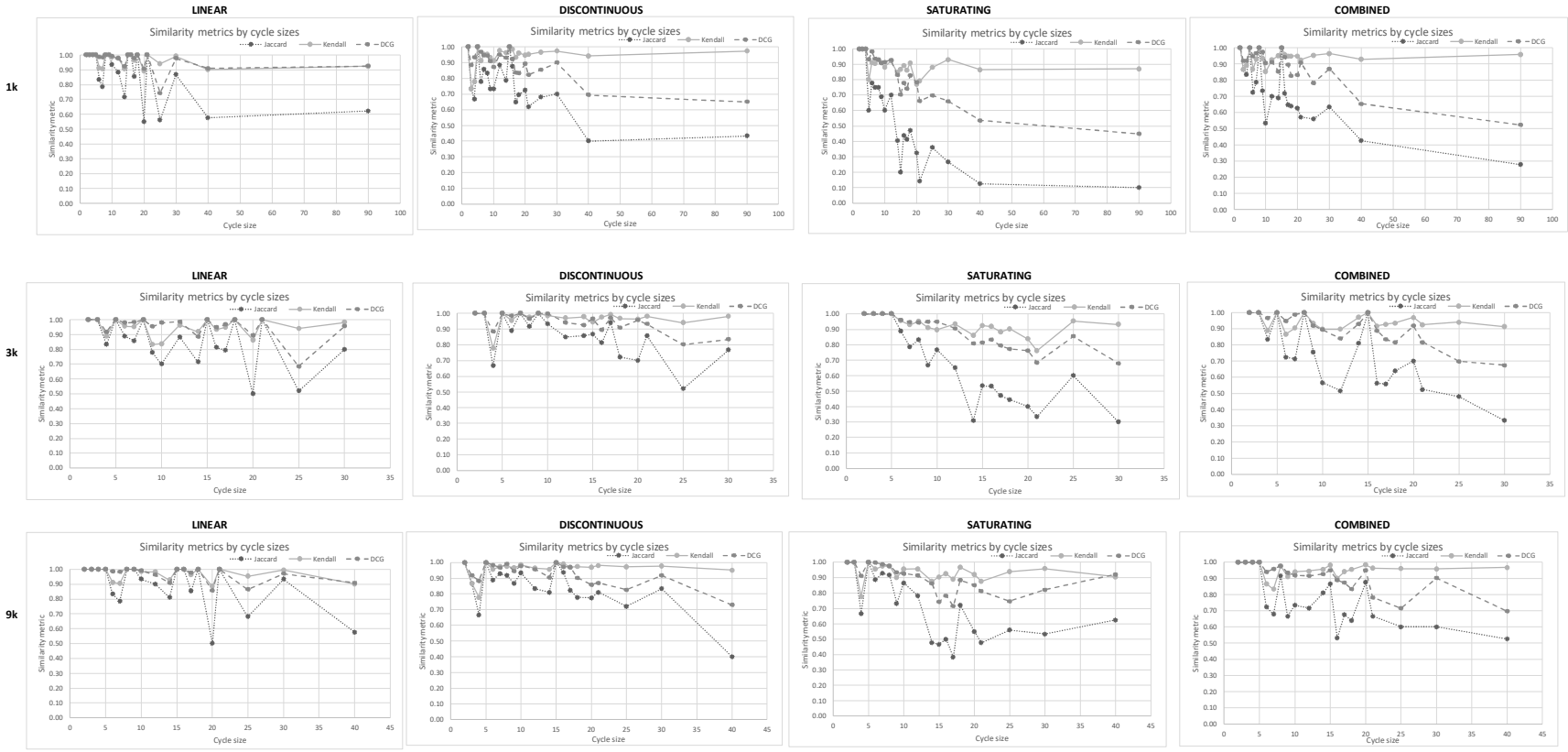

Figure 3 Similarity metrics by Failure Trace Cycles Sizes (up to 90 failures)

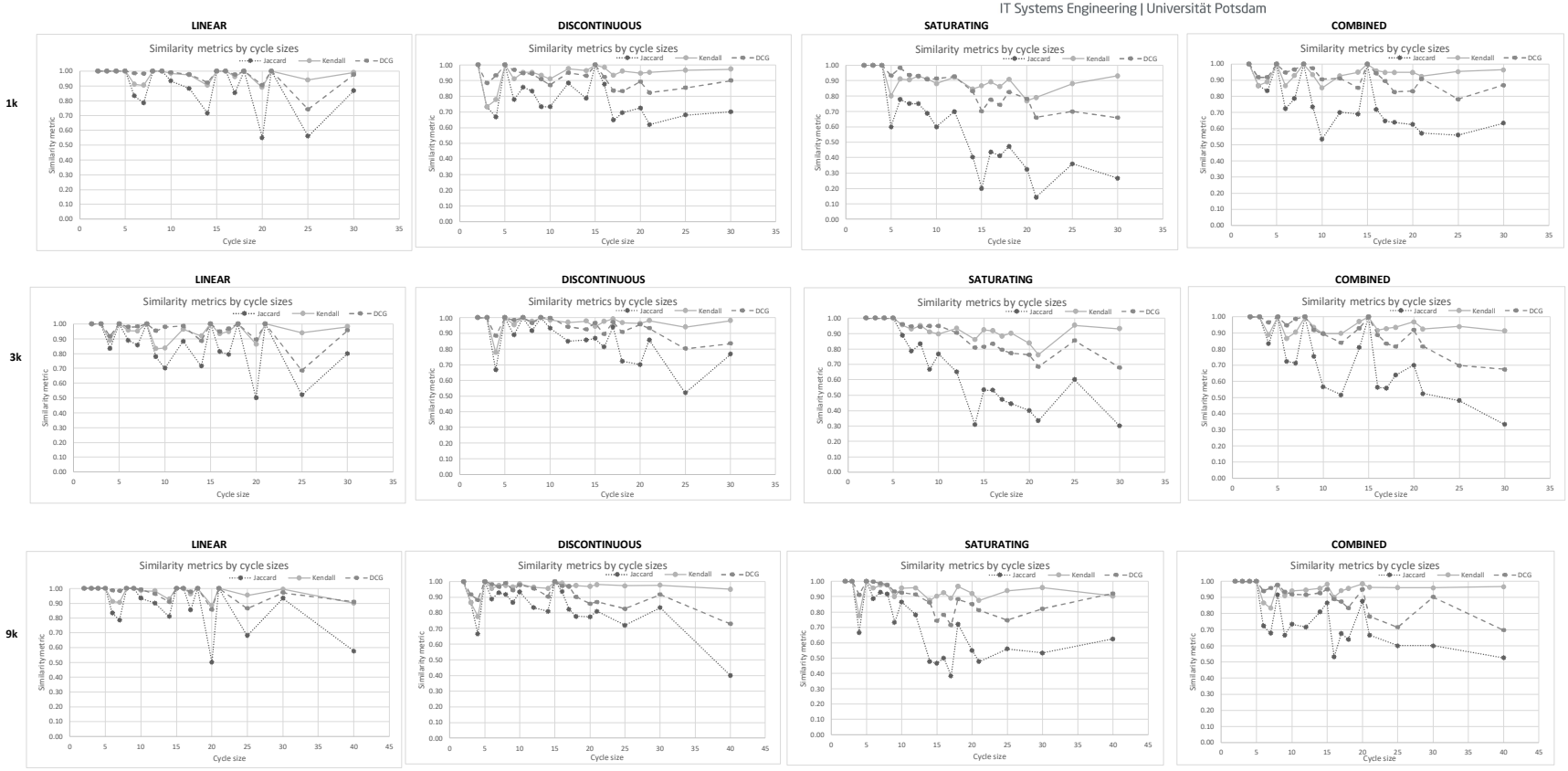

Figure 4 Similarity metrics by Failure Cycle Sizes (showing up to cycles with 40 failures)

Table 4 - Correlations between reward and similarity metric values

Cycle range	Linear	Discontinuous	Saturating	Combined	Metric	Trainind dataset size
All	-0.39	-0.86	-0.82	-0.90	DCG	1
All	-0.36	-0.28	0.25	0.15	Kendall	1
All	-0.57	-0.69	-0.68	-0.76	Jaccard	1
All	-0.30	-0.91	-0.91	-0.88	DCG	3
All	-0.53	-0.79	-0.75	-0.71	Jaccard	3
All	-0.19	-0.02	-0.58	-0.03	Kendall	3
All	-0.46	-0.88	-0.80	-0.86	DCG	9
All	-0.57	-0.80	-0.71	-0.74	Jaccard	9
All	-0.37	0.16	-0.20	0.08	Kendall	9
Up to 30	-0.14	-0.46	0.40	0.21	Kendall	1
Up to 30	-0.45	-0.86	-0.45	-0.65	Jaccard	1
Up to 30	-0.49	-0.93	-0.56	-0.68	DCG	1
Up to 30	-0.44	-0.72	-0.91	-0.86	DCG	3
Up to 30	-0.45	-0.62	-0.87	-0.75	Jaccard	3
Up to 30	0.01	0.07	-0.58	-0.17	Kendall	3
Up to 30	-0.50	-0.54	-0.78	-0.75	DCG	9
Up to 30	-0.36	-0.40	-0.76	-0.67	Jaccard	9
Up to 30	-0.10	0.34	-0.17	-0.01	Kendall	9

Analysis

- We can look from Figure 3 and Figure 4 that larger cycle sizes imply lower Jaccard an DCG similarity values. This is confirmed by negative correlations between cycle size and the Jaccard and DCG similarity metrics (Table 4). This pattern is present in the results from prediction models trained by all three dataset sizes (1k, 3K, 9K).
- Two variables showed small or no correlation (Table 4) with cycle sizes: the Kendall-tau similarity metric and the Linear model.
- Kendall-tau similarity showed either small correlations or non-significant ones. Looking at the chart, the Kendall-tau correlation remains mostly constant across cycle sizes. This is confirmed by either non-significant correlation values (p-value>0.05) or low correlation values.
- Looking at the correlations for different complexities , the Linear function showed consistently lower correlations than the other functions. The reason is the low prediction error of the linear model, which causes very few ranking mismatches, even across different cycle sizes.

- Even when controlling for higher complexity models (DCG and Saturating) and varying dataset sizes (1k, 3k, 9k), we could not detect any pattern in the correlation between cycle size and the similarity metrics. i.e., increasing dataset sizes not necessarily showed an increase in the correlation between similarity metric and cycle size. This means that, having models with larger prediction error (i.e., trained by smaller datasets) does not necessarily imply in a proportionate larger number of mismatches.
- One possible explanation is that the prediction error of the cheapest model (trained with 1k dataset) is already too small to cause major variations in the ranking that would reflect in major mismatches for larger cycles. i.e., smaller prediction error (produced by training with the 3k and 9K datasets) does not imply in fewer proportionate mismatches for larger cycles. Although this is a benefit in terms of cost by training with smaller datasets, we do not know if we could reduce the current level of mismatches by reducing the prediction error.