BASELINE

The BASELINE_PREFACE corresponds to instantiating PREFACE by removing the Deep Autoencoder neural network and adding a threshold-based approach for detecting the anomalous rKPIs. (We recall that PREFACE uses the Deep Autoencoder to predict anomalous rKPIs, out of the rKPIs computed with the PREFACE's Rectifier.) By comparing PREFACE with BASELINE_PREFACE, we evaluate the contribution of the Deep Autoencoder neural network, the core component of PREFACE.

BASELINE_PREFACE calculates a threshold for each rKPI computed by the PREFACE's Rectifier, by referring to the training data. It detects the anomalous rKPIs at runtime, as the KPIs with a value greater than the corresponding threshold. BASELINE_PREFACE computes the threshold T for each rKPI as T=M+3V, where M is the mean and V the variance of the rKPI in the training data. This is consistent with PREFACE that detects anomalies at runtime when the reconstruction error exceeds the threshold of mean+3*variance, with respect to the mean and the variance of the reconstruction errors observed during training.

RESULTS

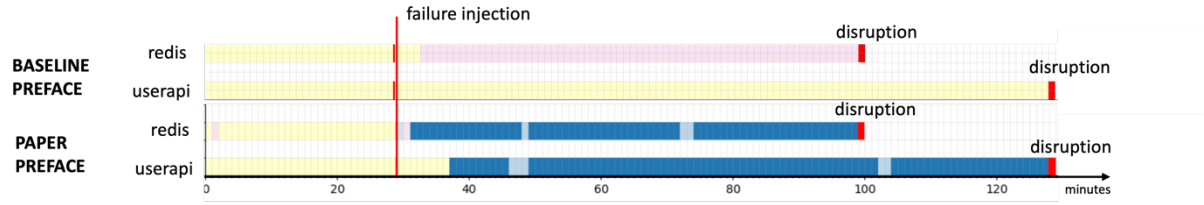
Below we compare the results of PREFACE with BASELINE_PREFACE, in the same format that we use in the paper (Section 4.3, Fig.5).

We observe that PREFACE performs significantly better than the threshold-based BASELINE_PREFACE. In particular, BASELINE_PREFACE cannot predict any of the failures that we injected in the microservice userapi, and cannot predict the Network_Delay failures that we injected in the microservice redis. BASELINE_PREFACE does not produce any valid localization (pink squares only) for Cpu_Stress and Memory_Stress in redis, and it reacts much later than PREFACE for Memory_Stress in redis.

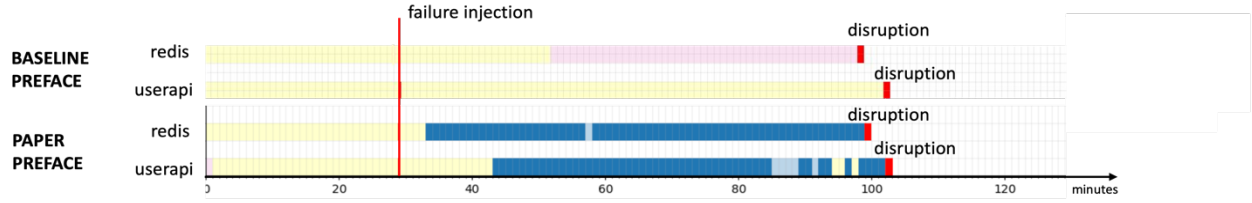
These data confirm the merit of the design of PREFACE based on the Deep Autoeconder.

PLOTS

CPU_Stress



Memory_Stress



Network_Delay