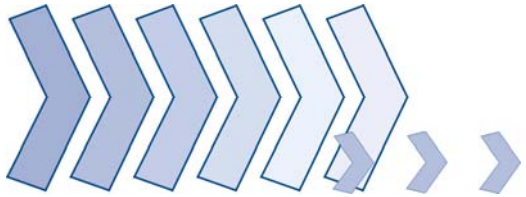
HUMBOLDT-UNIVERSITÄT ZU BERLIN



Proactive Fault Management

Felix Salfner

7.7.2010

www.rok.informatik.hu-berlin.de/Members/salfner

Contents

- **Introduction**
- Variable Selection
- Online Failure Prediction Overview
- Four Online Failure Prediction Techniques
- Assessing Failure Predictors
- Taking Action
- Summary

Our Credo

"Ordinary mortals know what's happening now, the gods know what the future holds because they alone are totally enlightened.

Wise men are aware of future things just about to happen"

C. P. Cavafy, (Greek poet, 1863-1933) "But the Wise Perceive Things about to Happen," a poem based on lines by Philostratos

Motivation

- Ever-increasing systems complexity
- Ever-growing number of attacks and threats, novice users and third-party or open-source software, COTS
- Growing connectivity and interoperability
- Dynamicity (frequent configurations, reconfigurations, updates, upgrades and patches, ad hoc extensions), and
- Natural and man-made disasters

A Different Mindset

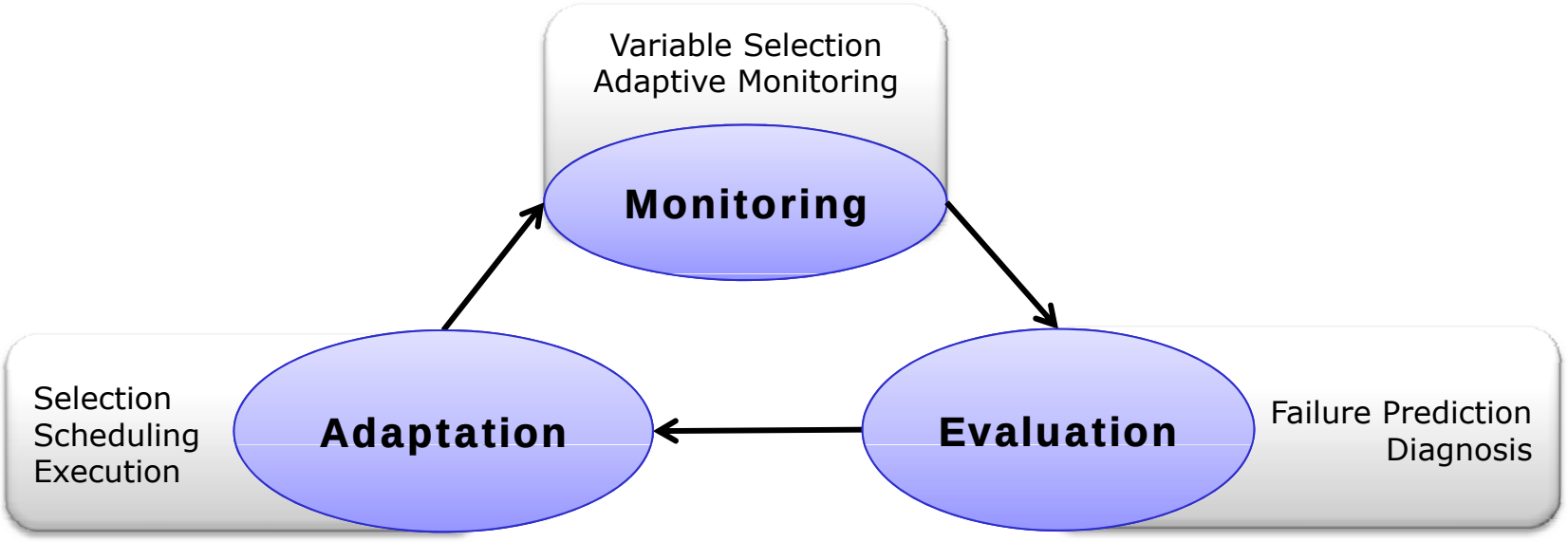
- Faults, errors and failures are common events so let us treat them as part of the system behavior and learn how to cope with them

- Attractive panacea:

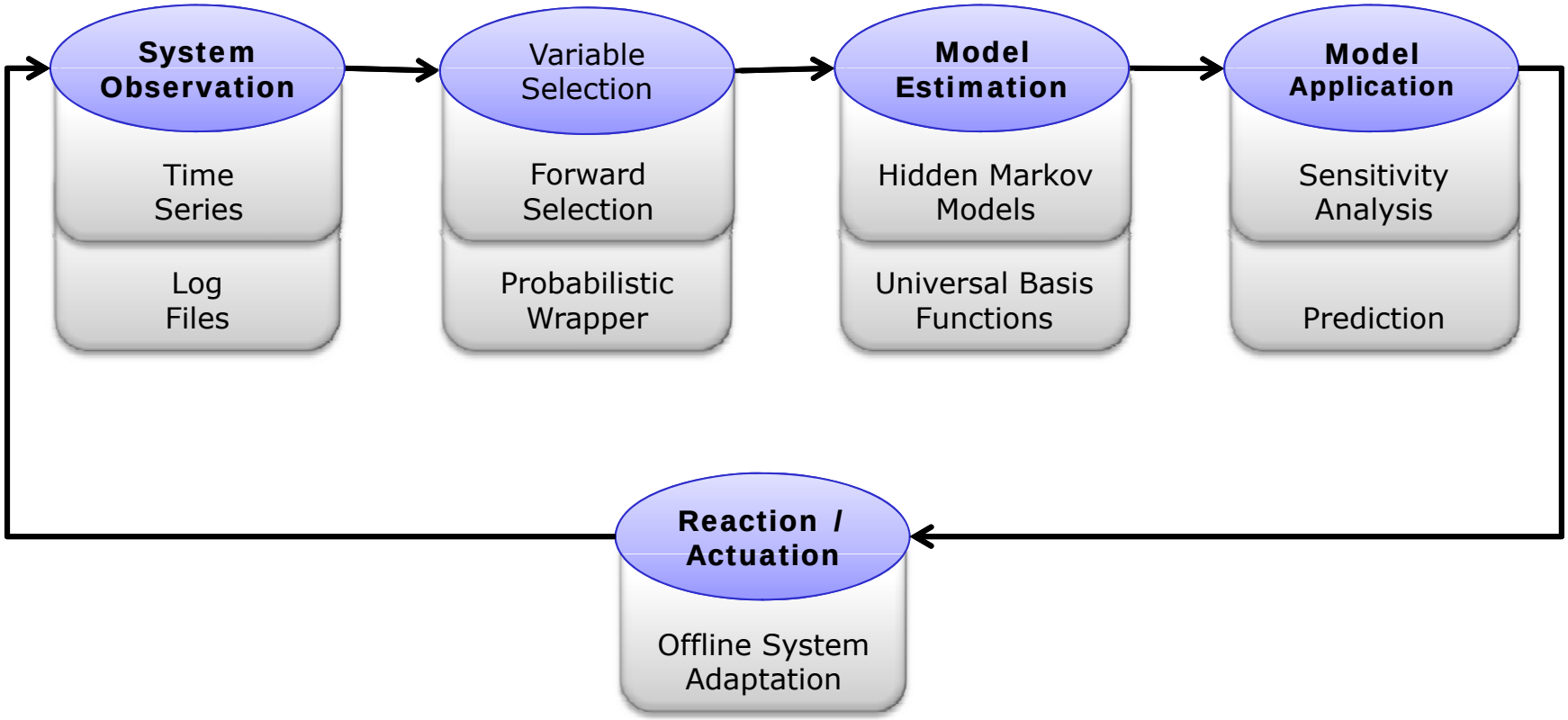
  (self) **Proactive Fault Management** (PFM)

Proactive Fault Management (PFM)

PFM is an umbrella term for techniques such as monitoring, diagnosis, prediction, recovery and preventive maintenance concerned with proactive handling of errors and failures: if the system knows about a critical situation in advance, it can try to apply countermeasures in order to prevent the occurrence of a failure, or it can prepare repair mechanisms for the upcoming failure in order to reduce time-to-repair.



How to Get There



[Hoffmann, Trivedi, Malek 2007]

Comparison to Classical Reliability Theory

- Classical reliability theory is typically useful for long term or average behavior predictions and comparative analysis

- Classical reliability theory may help but is not very good for short term prediction due to dynamics, mobility, systems/networks complexity, changing execution environments, upgrades, online repair, etc.

Contents

- Introduction
- **Variable Selection**
- Online Failure Prediction Overview
- Four Online Failure Prediction Techniques
- Assessing Failure Predictors
- Taking Action
- Summary

Variable Selection

- What are the right variables to use for modeling?
- There are up to 4200 variables (v) and up to hundreds of fault classes (f) per node
- For n nodes: m = v x f x n variables, the number of combinations c equals:

$$c = \sum_{r=1}^{m} \binom{m}{r}$$

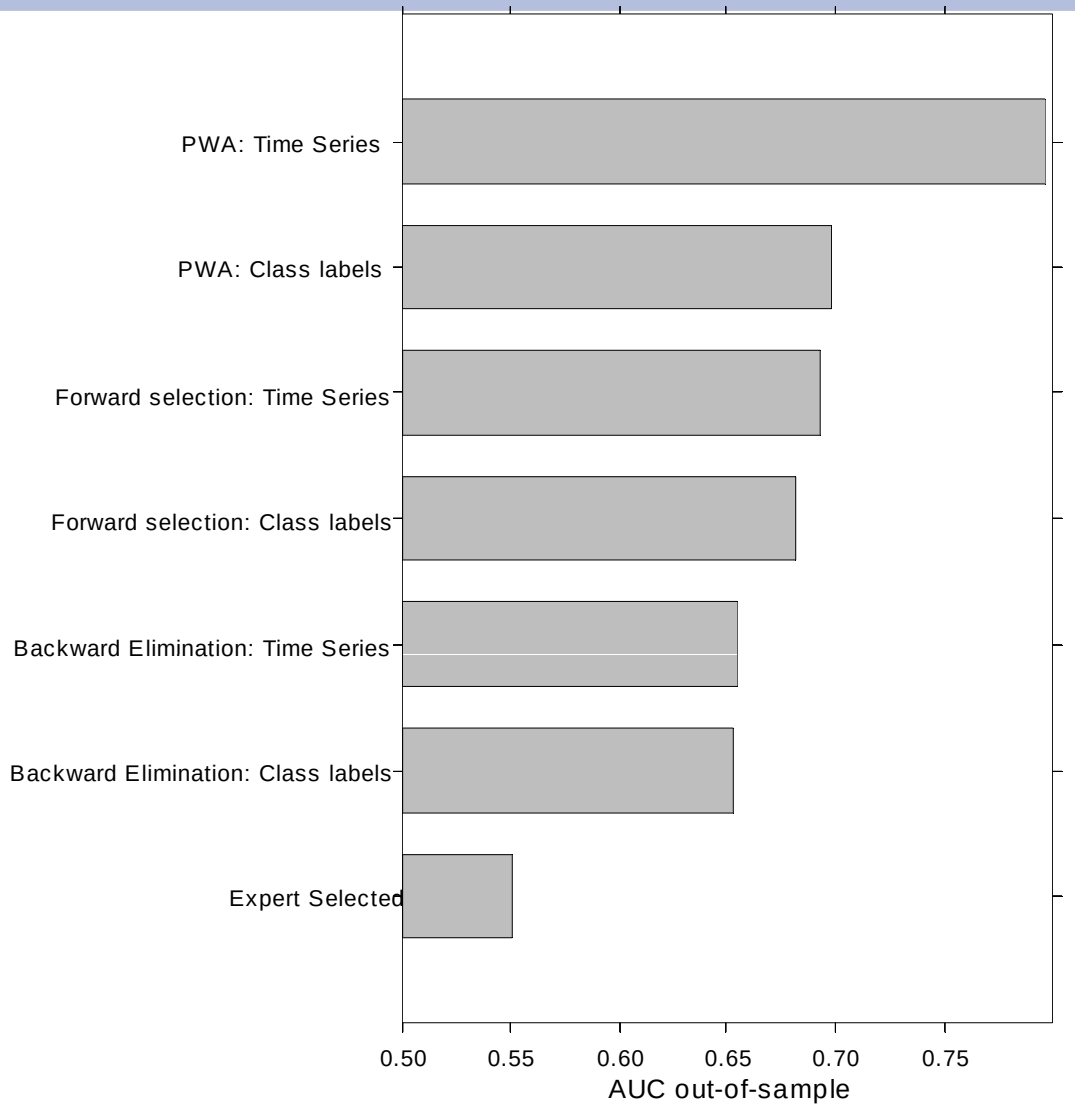
- Combinatorial explosion!

Variable Selection Methods

- Selection by experts
- Filter (e.g., mutual information criterion)
- Wrapper (making use of modeling procedure specifics)
  - feed forward selection, finding independent variables
  - backward elimination
  - probabilistic (only variables showing correlation and certain distribution)
- Forward Addition - a method of selecting random variables for inclusion in the regression model by starting with no variables and then gradually adding those that contribute most to prediction

Variable Selection

- Benchmarked four techniques
  - Forward selection
  - Backward elimination
  - Expert selected
  - PWA (Prob. Wrapper)
- Variables
  - *alloc*
  - *sema/s*
- PWA performs best on time series *and* class label data

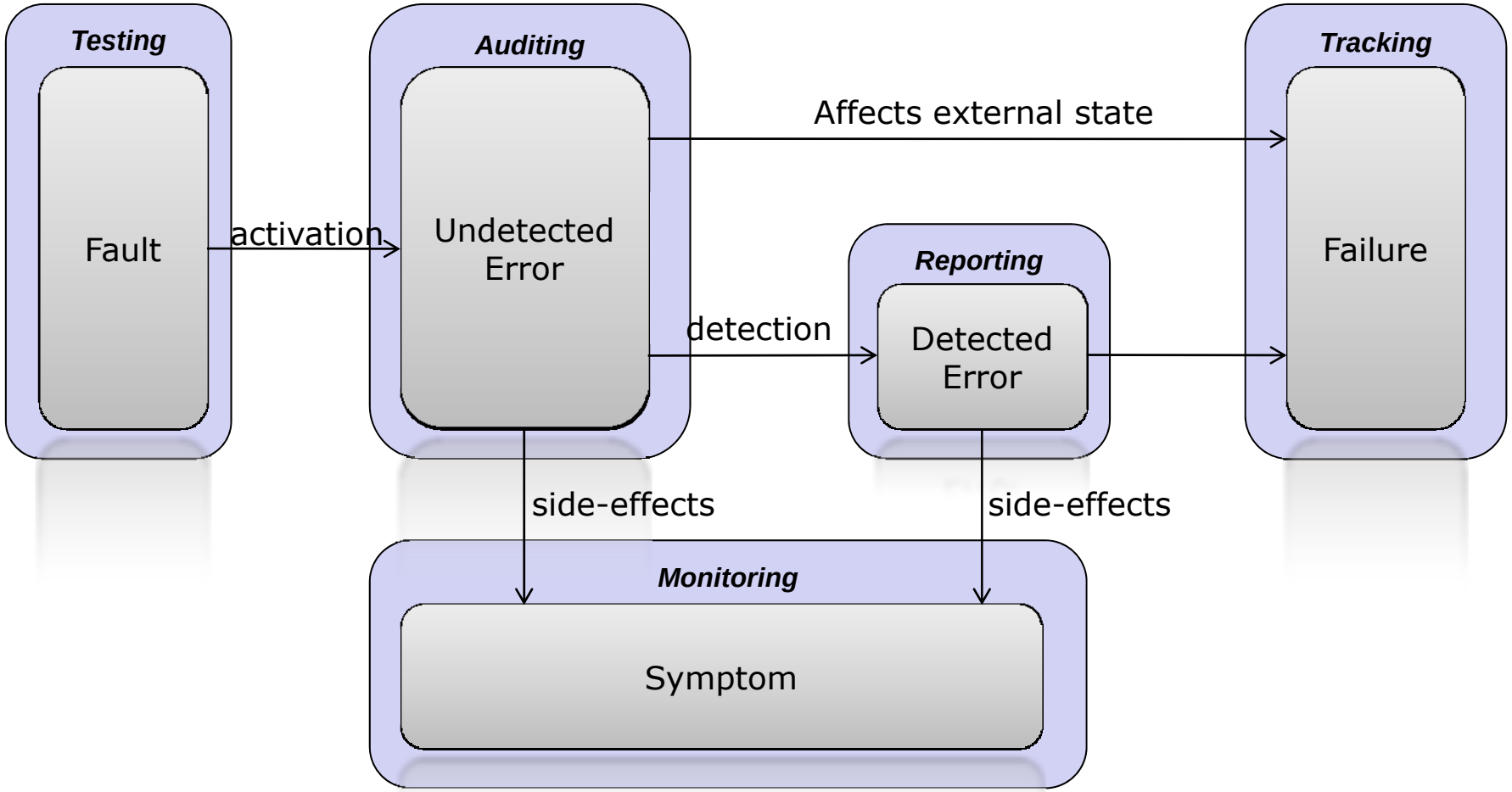


[Hoffmann, Malek 2006; Hoffmann 2005]

Contents

- Introduction
- Variable Selection
- **Online Failure Prediction Taxonomy**
- Online Failure Prediction Techniques
- Assessing Failure Predictors
- Taking Action
- Summary

Faults, Errors, Failures again …



Four Ways of Detecting Faults

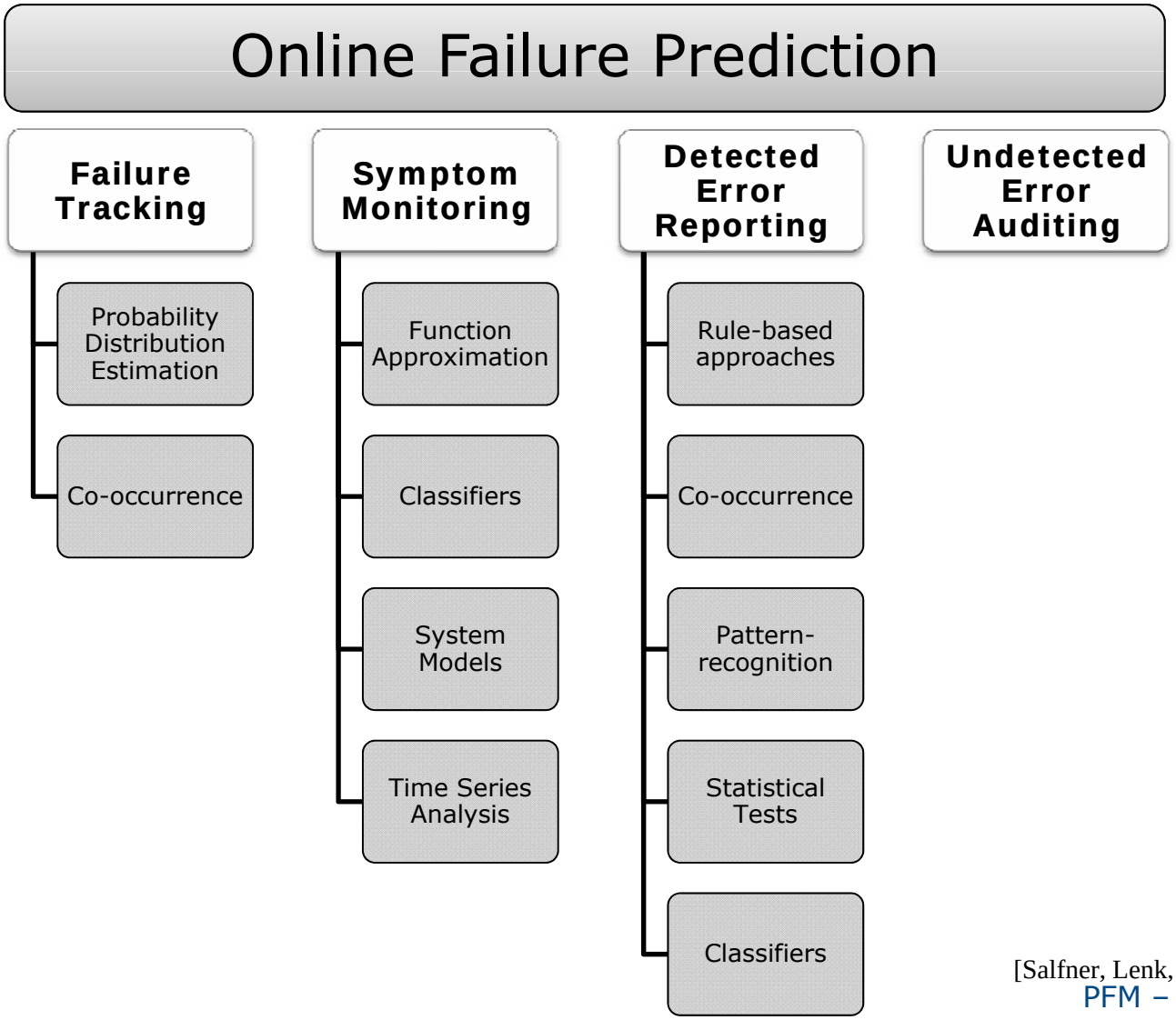
(1) The system can be *audited* in order to actively search for *faults*, e.g., by testing on checksums of data structures, etc.

(2) System parameters such as memory usage, number of processes, workload, etc., can be *monitored* in order to identify side-effects of the faults. These side-effects are called *symptoms*. For example, the side-effect of a memory leak is that the amount of free memory decreases over time.

(3) If a fault is activated and *detected* (observed), it turns into an *error*.

(4) If the fault is not detected by fault detection mechanisms, it might directly turn into a *failure* which can be observed from outside the system or component.

Taxonomy



[Salfner, Lenk, Malek, CSUR]

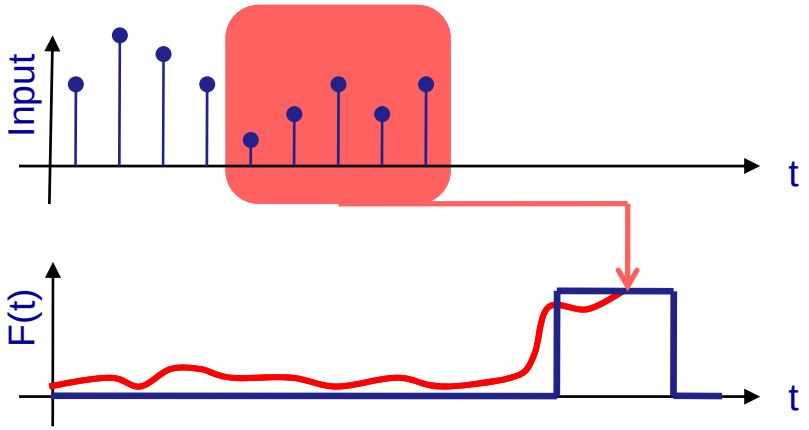
Online Failure Prediction - Definition

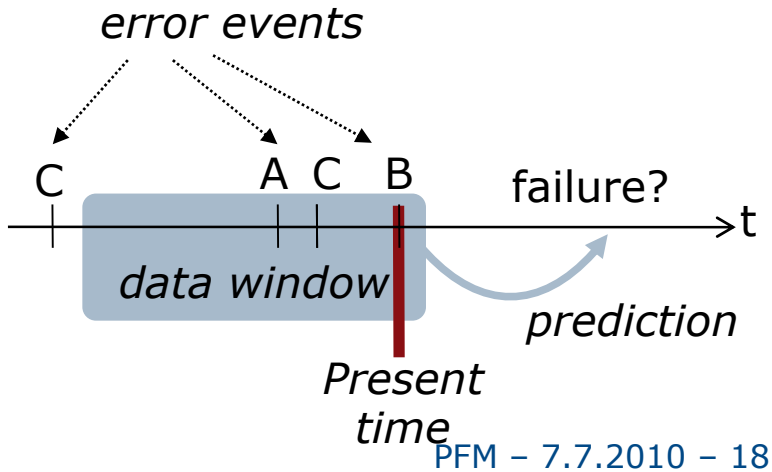
- The goal of online failure prediction is to ***identify failure-prone situations***, i.e. situations that will probably evolve into a failure. The evaluation is ***based on runtime monitoring data***.

- The output of online failure prediction can either be
  - a decision that a failure is imminent or not, or
  - some continuous measure evaluating how failure-prone the current situation is

Two Types of Input Data

- There are two types of system measurements
  - periodic, numerical
  - event-based, categorical
- Examples for periodic data
  - system- / CPU load
  - memory usage
- Examples for event-based data:
  - interrupts
  - threshold violations
  - error events





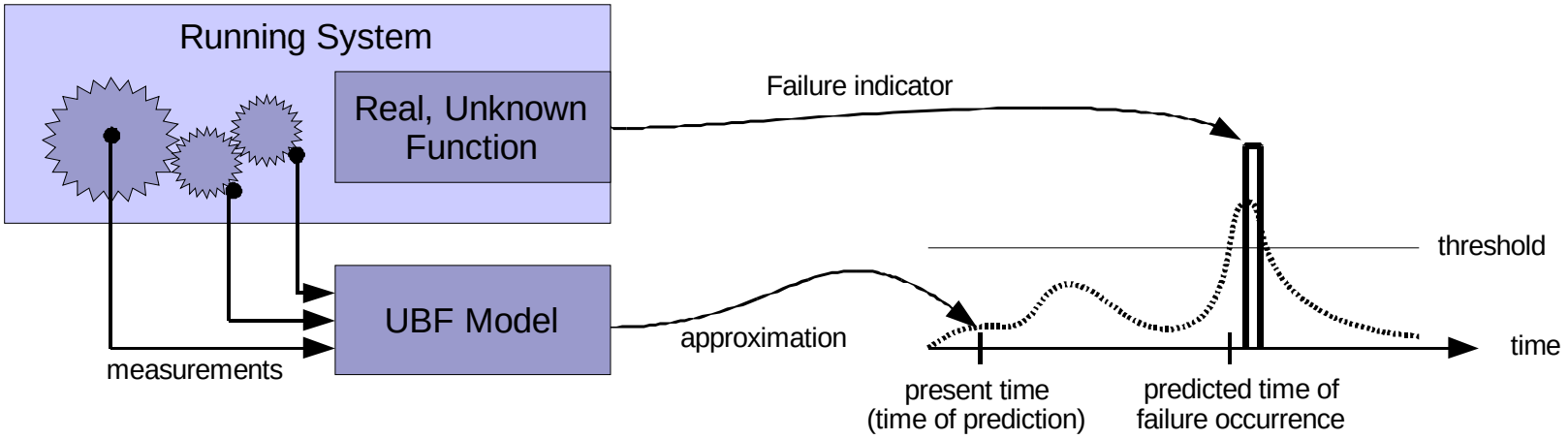
Contents

- Introduction
- Variable Selection
- Online Failure Prediction Overview
- **Four Online Failure Prediction Techniques**
- Assessing Failure Predictors
- Taking Action
- Summary

Prediction Techniques Examples

1. *Universal Basis Functions (UBF)*
2. *Hidden Semi-Markov Model (HSMM)*
3. Dispersion-Frame Technique (DFT)
4. Eventset method

Universal Basis Functions (UBF)



- Tailored to periodic measurements
- Function approximation approach: Express target value as function of input variables
- Examples for target values:
  - Availability
  - Memory consumption

UBF Background

- Starting from radial basis functions
  Linear combination of kernel functions G_i

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i * G_{\lambda_i}\left(\left\|\mathbf{x} - \mathbf{c}_i\right\|\right)$$

Replace fixed Gaussian by flexible domain specific kernel

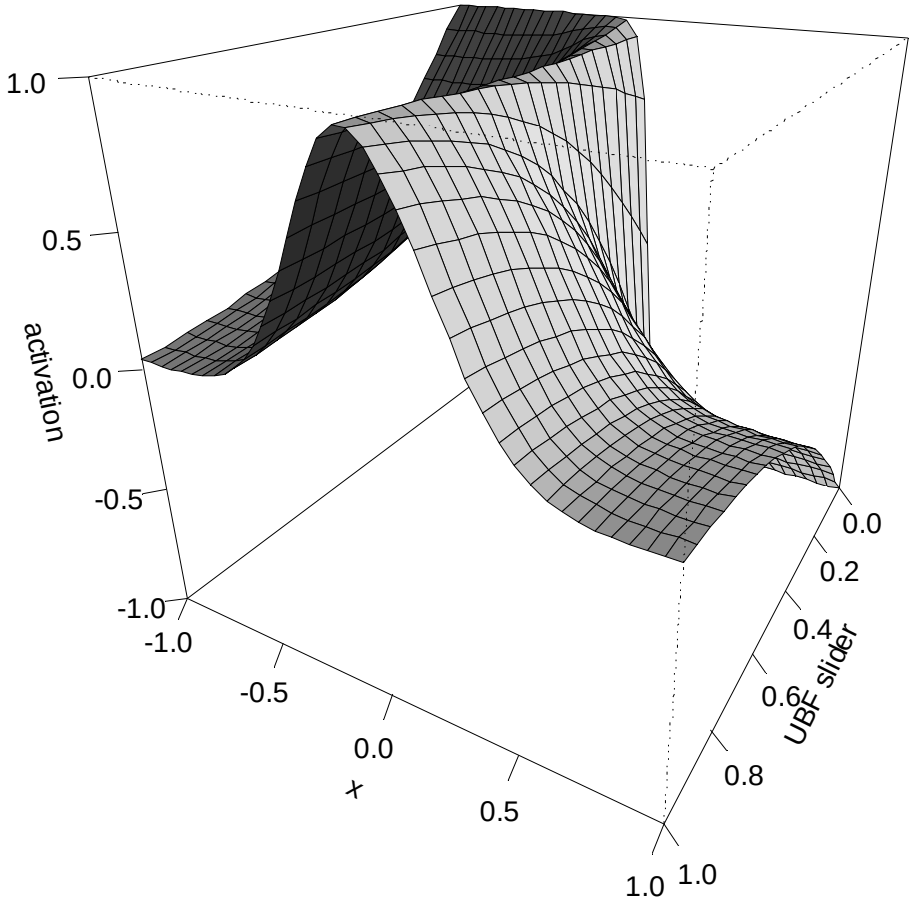
$$G_{\lambda_i}(\mathbf{d}) = \boldsymbol{\omega}\Phi_1(\mathbf{d}) + (1-\boldsymbol{\omega})\Phi_2(\mathbf{d})$$

- Determine parameters by minimizing, e.g., mean square error

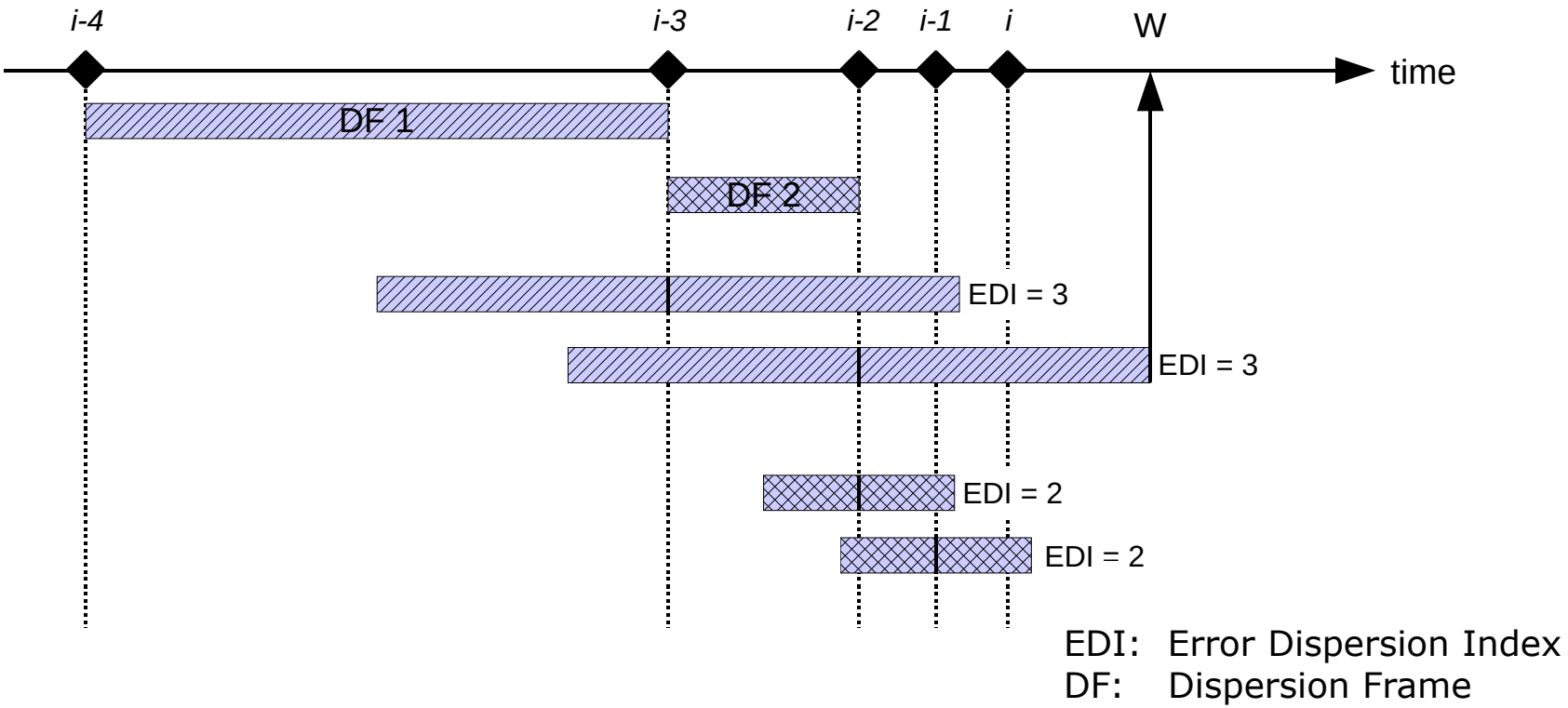
$$\min H[f] = \sum_{i=1}^{N} \left(f(x_i) - y_i\right)^2$$

Effect of ω (UBF slider)

- Linear combination of nonlinear kernel functions
- Examples:
  - Gaussian
  - Sigmoide functions, …
- RBF is special case
- Improve efficacy by introducing data specific flexible kernels
- Universal approximator
- Large number of kernels to cover heavy tailed distributions



Dispersion Frame Technique



EDI: Error Dispersion Index
DF: Dispersion Frame

- Classic technique for error-log analysis
- Evaluates the time of error occurrence
- Applies a set of heuristic rules evaluating the number of errors within successive dispersion frames

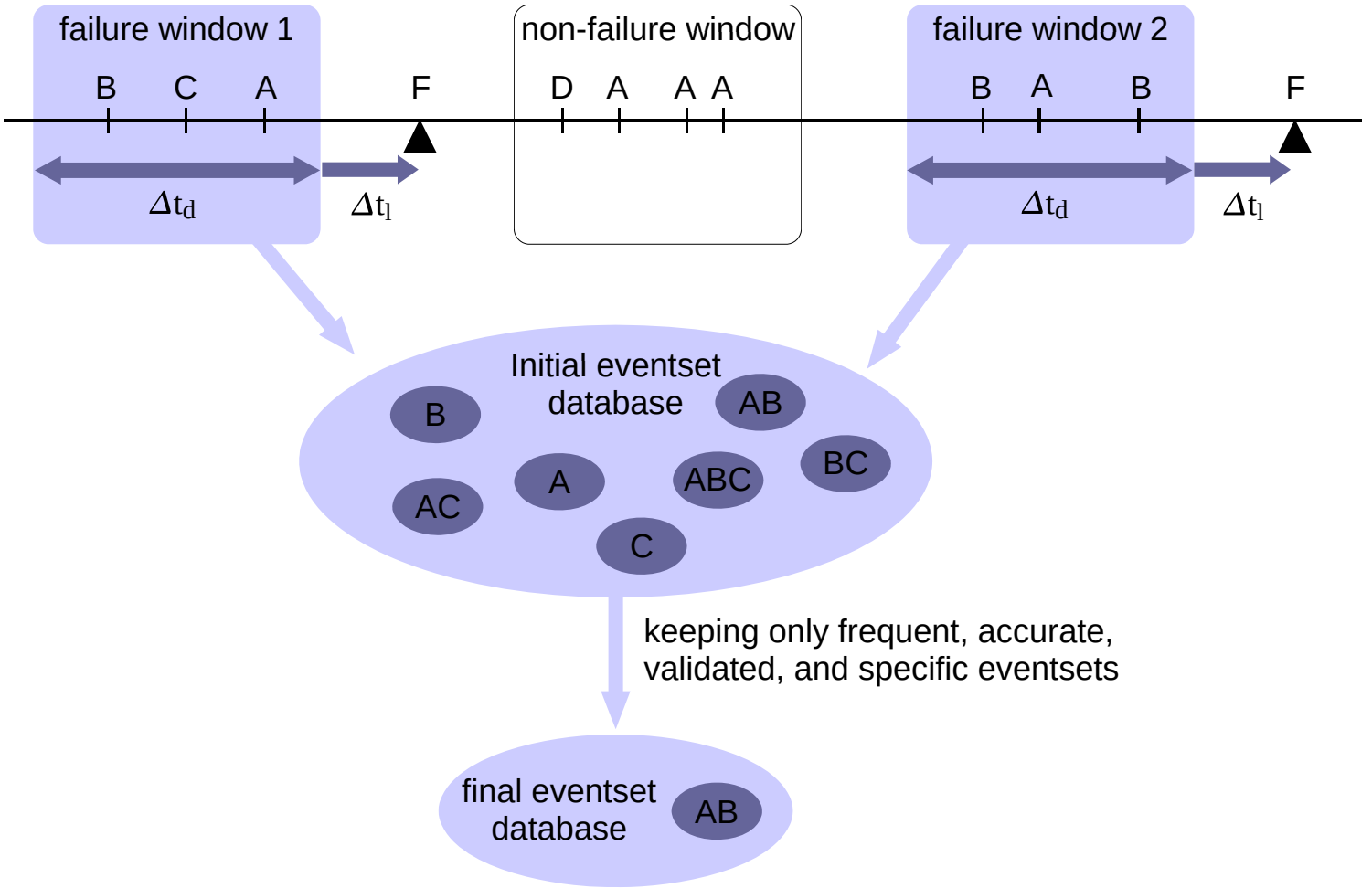
[Lin, Siewiorek 1990]

Event-set Method

- Approach inspired by data-mining
- Focus on type of events
- Based on sets of events
  - Each set contains decisive events that occur prior to a target event
  - Events correspond to errors in our taxonomy
  - Target events correspond to failures
  - Event sets do not keep timing information
- Result: rule-based failure prediction system containing a database of indicative eventsets

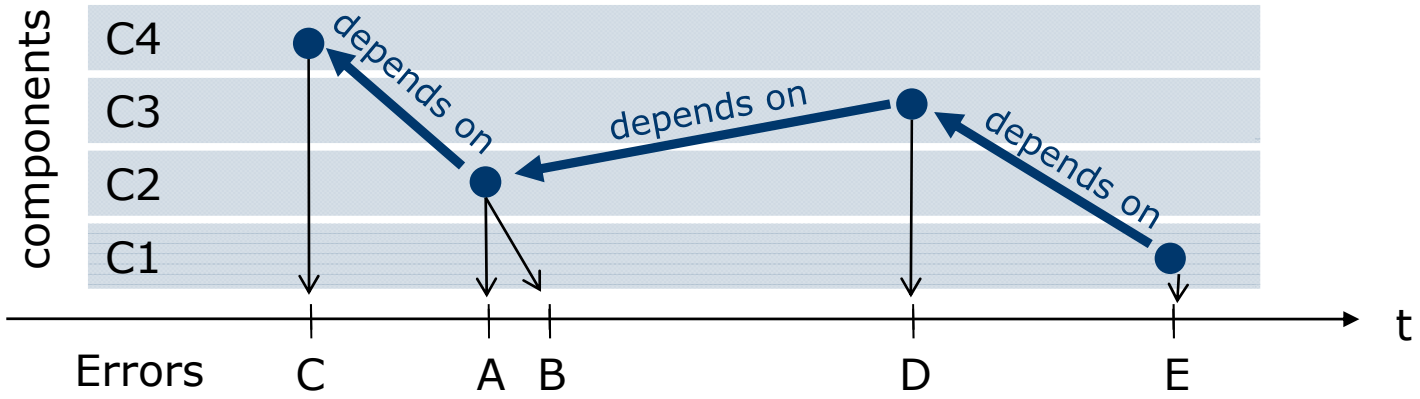
[Vilalta, Ma 2002]

Event-set Method



Hidden Semi-Markov Model Prediction

- Components of complex systems depend on each other
- Dependencies lead to error patterns



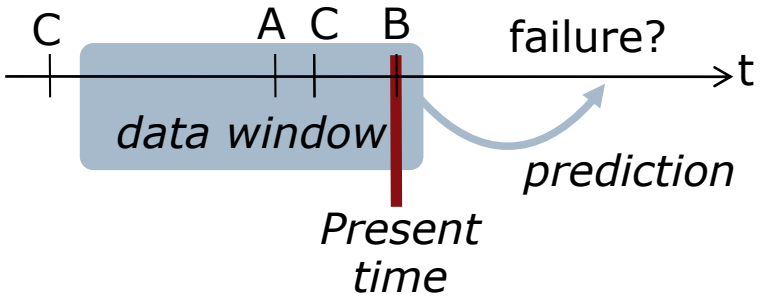
- Fault-tolerant systems:
  - Failures occur only under certain conditions
  - Failure-prone conditions can be identified by specific error patterns

➢ Use pattern recognition to identify symptomatic situations

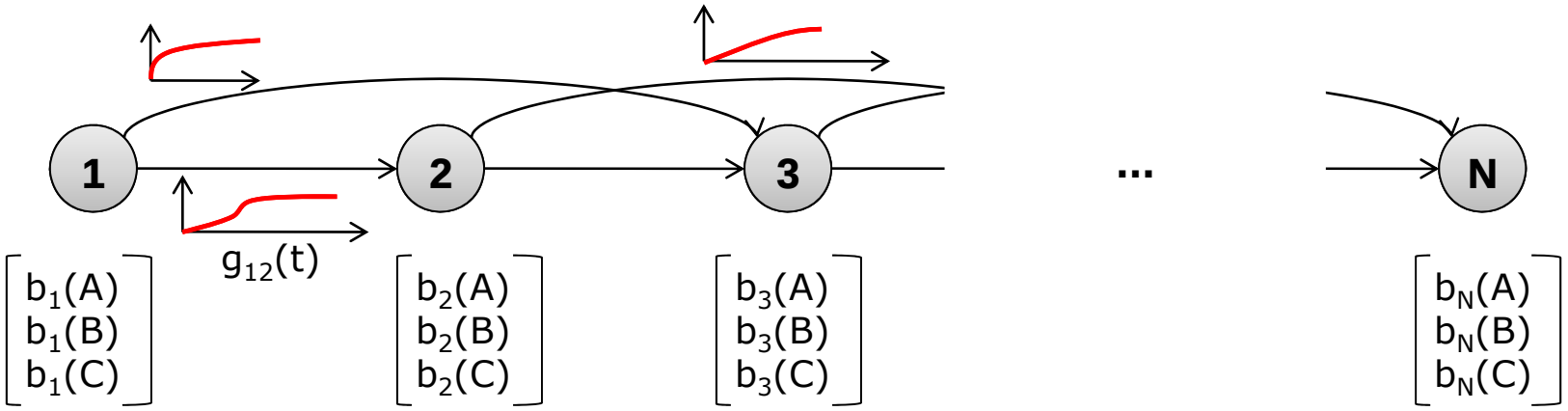
[Salfner, Malek 2007; Salfner 2008]

Approach

- Standard tool for pattern recognition: Hidden Markov Models
- Identify symptomatic patterns
  - Algorithmically
  - From recorded training data
  - ➢ **Machine learning**
- Additional assumption:
  - Time between events is decisive (temporal sequence analysis)
  - Standard Hidden Markov Models need to be extended
  - ➢ **Development of a Hidden Semi-Markov Model (HSMM)**
- The approach incorporates both type and time-of-occurrence of error events



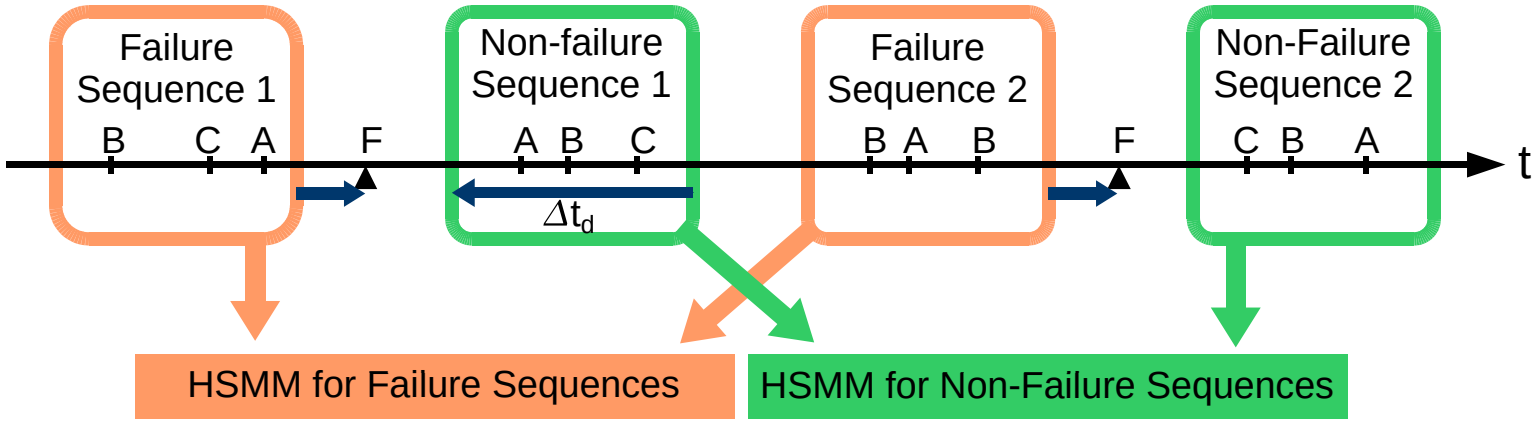
Hidden Semi-Markov Models



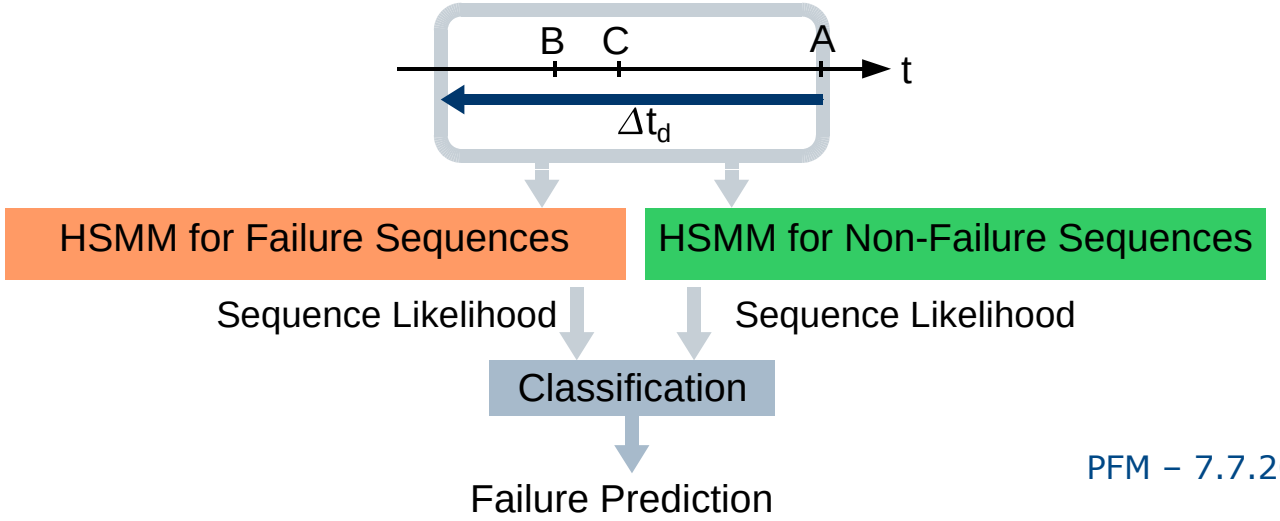
- Discrete Time Markov Chains (DTMC) consist of states (1…N) and transition probabilities p_{ij} between states
- In Hidden Markov Models (HMM) each state can generate a symbol A,B,C according to probability distribution $b_i(o_k)$
- Hidden semi-Markov models (HSMMs) replace transition probabilities p_{ij} by time-continuous cumulative probability distributions $g_{ij}(t)$

Machine Learning: Two Steps

1. Training: Fit model parameters to training data



2. Prediction: Processing of runtime measurements



Contents

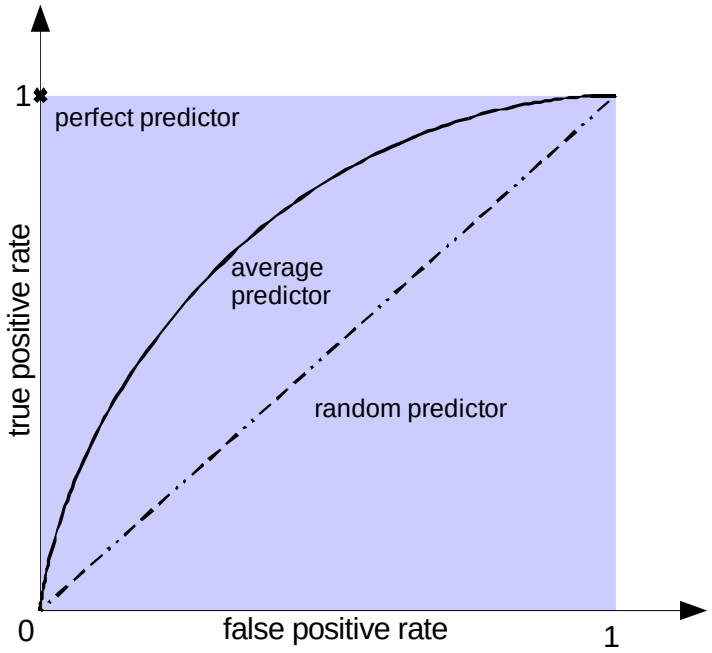
- Introduction
- Variable Selection
- Online Failure Prediction Overview
- Four Online Failure Prediction Techniques
- **Assessing Failure Predictors**
- Taking Action
- Summary

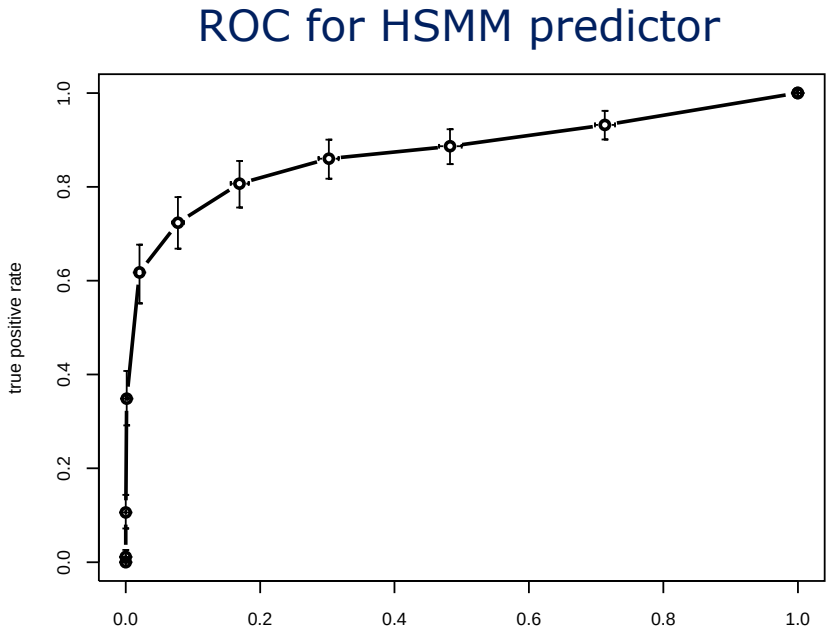
Precision, Recall and other Metrics

contingency table	**True failure**	**True success**	Sum
Failure alarm	Correct alarm (TP)	False alarm (FP)	# Alarms
No warning	Missing alarm (FN)	Correct no-alarm (TN)	# No-Alarms
Sum	# Failures	# Successes	# Total

- Precision: fraction of correct alarms: $$\text{precision} = \frac{\text{correct alarms}}{\text{total } \sharp \text{ of alarms}}$$
- Recall: fraction of predicted failures: $$\text{recall} = \frac{\text{correct alarms}}{\text{total } \sharp \text{ of failures}}$$
- False positive rate (fpr): $$\text{false positive rate} = \frac{\text{false positives}}{\sharp \text{ of successes}}$$
- True positive rate is equal to recall

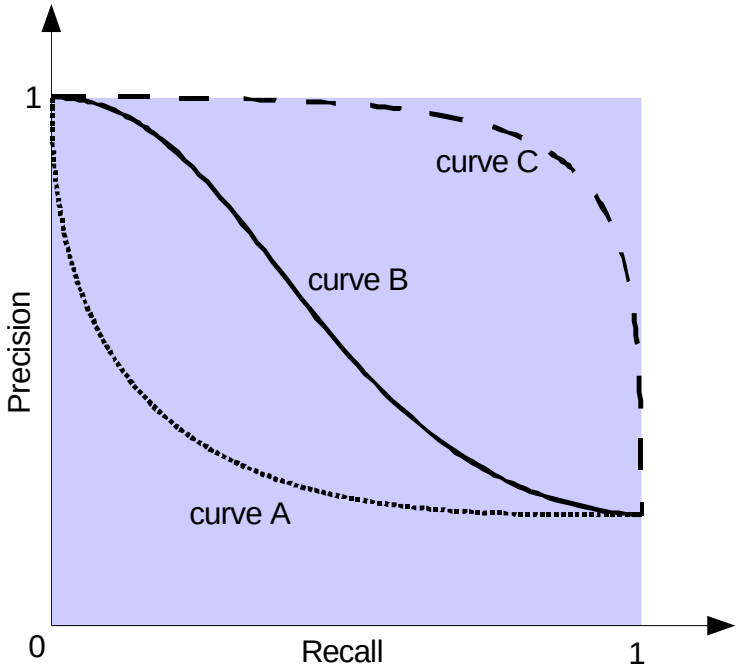
Receiver Operating Characteristics (ROC)



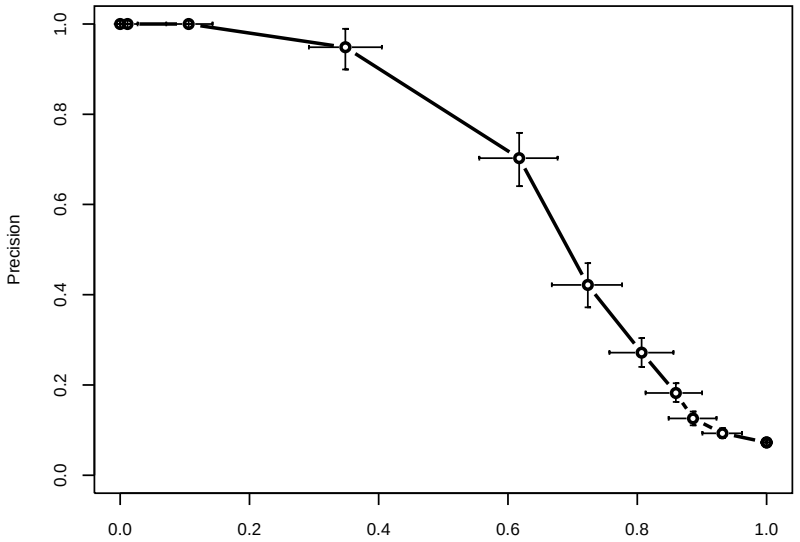


- Plot true positive rate (recall) over false positive rate for various thresholds
- Threshold ∞ : tpr and fpr equal to zero
- Threshold -∞ : tpr and fpr equal to one

Precision-Recall-Plots



Precision-Recall plot for HSMM predictor



- Plot precision over recall for various thresholds
- Threshold ∞ : precision equal to one, recall equal to zero
- Threshold -∞ : precision equal to ratio of positive and negative examples, recall equal to one

Skalar Metrics

- ROC, Precision/Recall diagrams etc. are graphs
- Good for visual inspection, bad for algorithmic decisions
- Goal: obtain one real number to evaluate „quality“ of predictor
- Examples:
  - Precision-Recall-Breakeven: Value at which precision and recall cross
  - Area-under-curve (AUC): Area under the ROC curve
  - F-Measure: Harmonic mean of precision and recall:

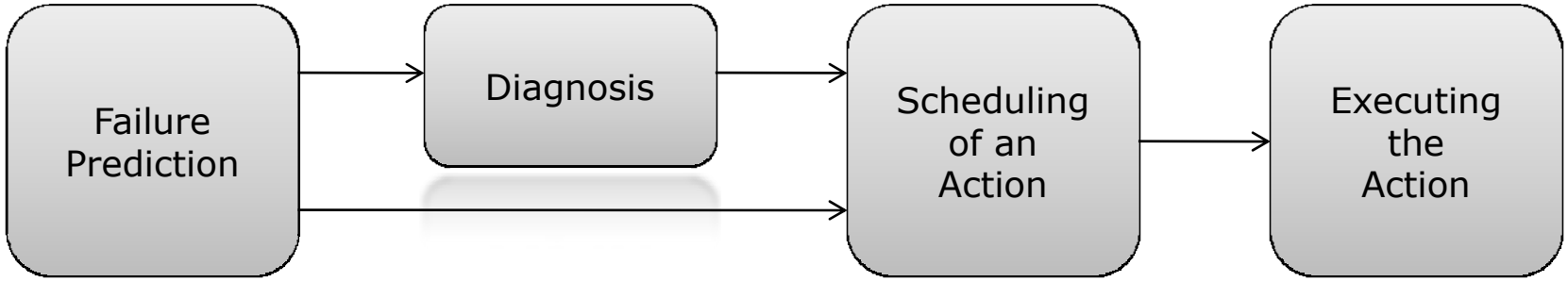
$$\text{F-Measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Contents

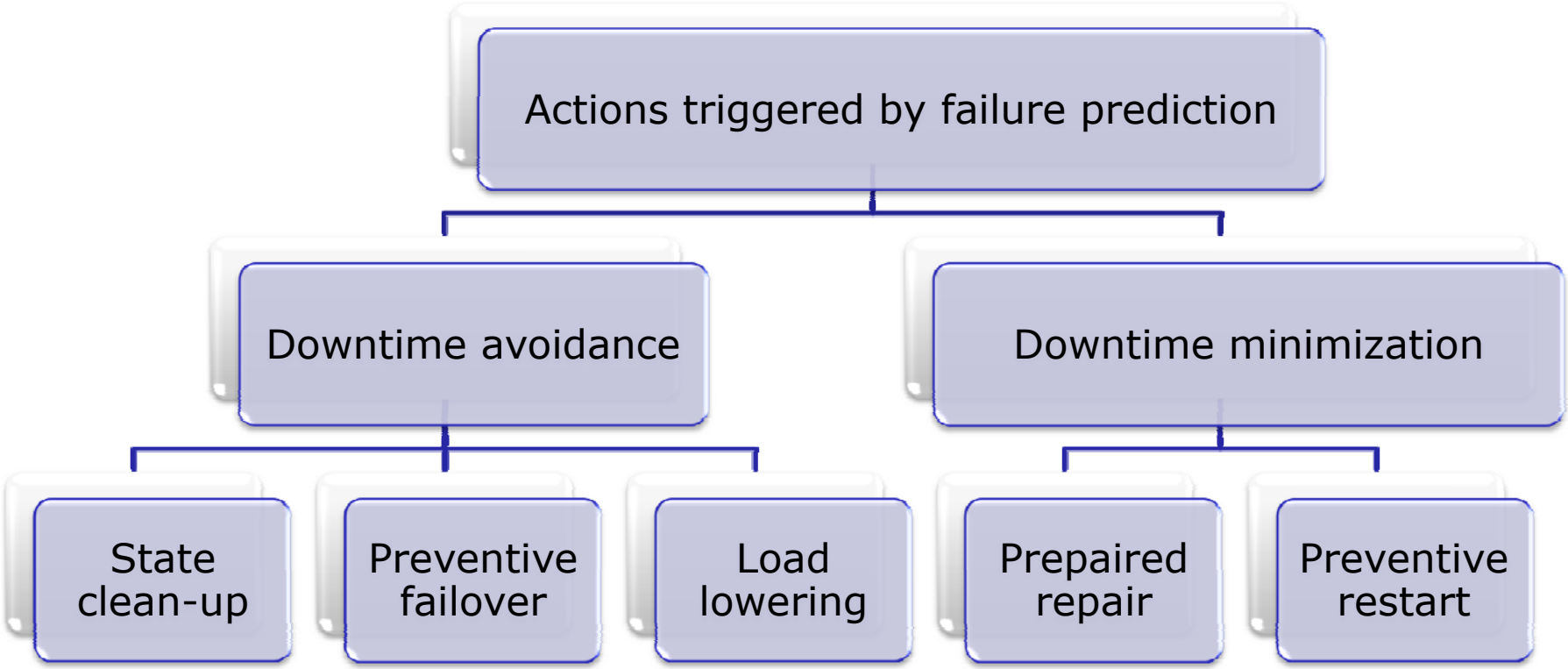
- Introduction
- Variable Selection
- Online Failure Prediction Overview
- Four Online Failure Prediction Techniques
- Assessing Failure Predictors
- **Taking Action**
- Summary

Taking Action

- Failure prediction is only the first step in managing faults proactively
- After a potential failure has been predicted, an action must be taken in order to
  - Avoid the failure (prevent it from occurring)
  - Prepare the system such that TTR can be reduced
  - (See lecture seven)
- In general, the following steps have to be performed:

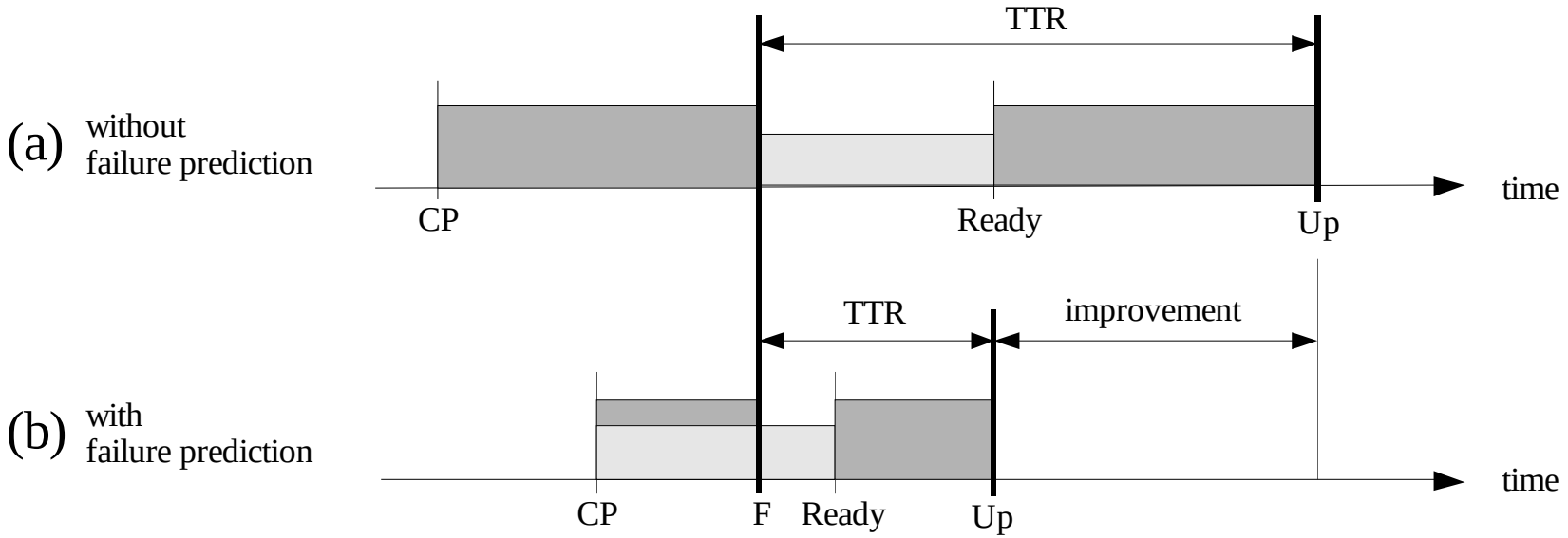


Taxonomy of Reaction Methods



Effects of Proactive Methods

- Downtime avoidance improves MTTF
- Downtime minimization reduces MTTR:



- However, In case of frequent false positive and false negative predictions, proactive fault management can also reduce availability!

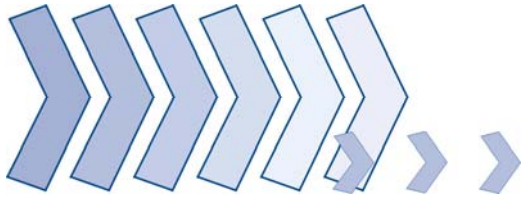
Contents

- Introduction
- Variable Selection
- Online Failure Prediction Overview
- Four Online Failure Prediction Techniques
- Assessing Failure Predictors
- Taking Action
- **Summary**

Summary

- Dynamics and complexity of today‘s systems require adaptive and proactive mechanisms to handle faults
- Proactive Fault Management (PFM) builds on
  - Continuously observing the system
  - Predict whether a failure is coming up
  - In case of an upcoming failure:
    - Analyze the fault that causes the upcoming failure
    - Decide what to do: Either try to avoid the failure or prepare repair mechanisms for the upcoming failure
- Since online failure predictors build on monitoring data: The best set of variables need to be identified: Variable selection
- Analyses of PFM suggest that PFM has the potential to enhance system availability by up to an order of magnitude.

HUMBOLDT-UNIVERSITÄT ZU BERLIN



Thanks!
Questions?

Prediction in Computer Science

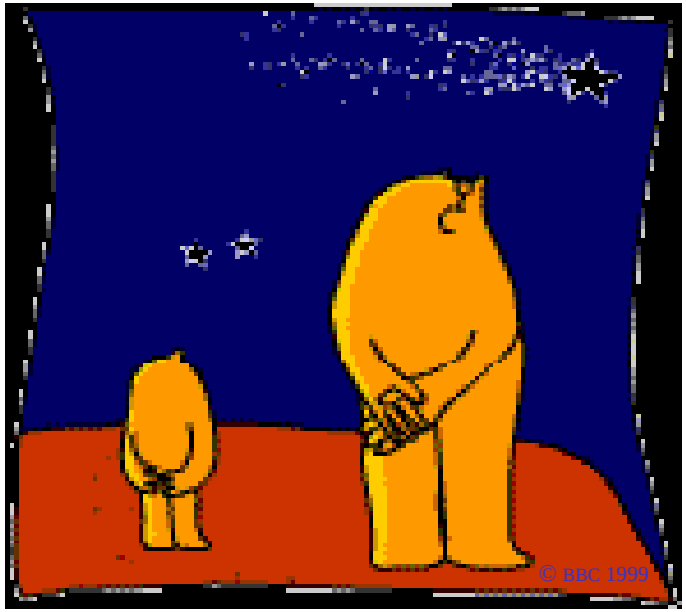
- Scheduling
- Branch prediction
- Memory management
- Performance evaluation
- Reliability prediction
- …
- Failure prediction

A Long Way to Go

- Daimler predicted in 1900 that the Europe's production of cars will not exceed 1,000 a year because it will not be possible to train enough chauffeurs
- In 1900 General Post Office of United Kingdom has predicted that there will be a demand for about ONE phone for each city with a population of more than one hundred thousand
- "There is not the slightest indication that nuclear energy will ever be obtainable." — Albert Einstein, 1932
- "There is no reason for any individual to have a computer in their home", Ken Olsen, 1977

Predicting the Future

- Predicting the future has fascinated people from the beginning of times
- Several millions of people work on prediction daily
- Astrologists, meteorologists, politicians, pollsters, stock analysts, doctors,…, and many computer scientists/engineers



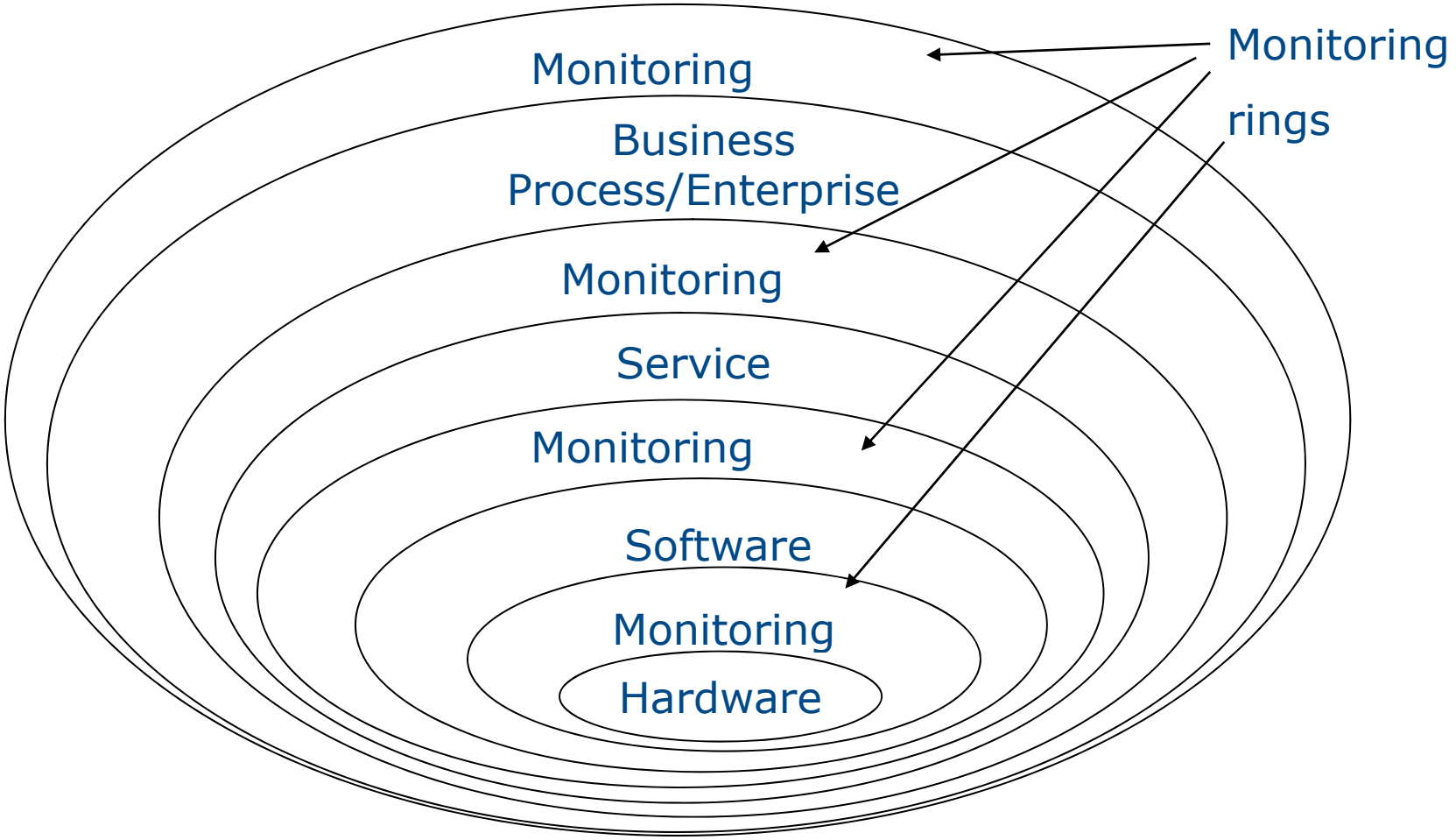
Contents

- **Runtime Monitoring**
- Error Logs
- Variable Selection
- Online Failure Prediction Taxonomy
- Online Failure Prediction Techniques
- A Case Study
- Taking Action
- Summary

Runtime (Online) Monitoring

- Runtime monitoring is a continuous observation of system variables for a given purpose such as diagnosis or failure prediction
- Fundamental questions:
  - Which variables to monitor?
  - How frequently (sampling rate)?
  - At what level should we monitor?
  - How will performance be affected?
  - What storage will be needed?
  - When and how to process the monitoring data?
- **Remember:** monitoring is not free and never complete

Monitoring – At What Level?



Types of Data Sources

- Error Logs (Logfiles)
  - Lack of uniformity
  - Standards are just emerging
  - Redundancy (in some cases a problem is reported 60,000 times)

- System Activity Reporter (SAR) data
  - Up to 4200 parameters can be monitored in real systems
  - Up to five can usually be measured and processed in real time for 1-5 minutes prediction

Contents

- Runtime Monitoring
- **Error Logs**
- Variable Selection
- Online Failure Prediction Taxonomy
- Online Failure Prediction Techniques
- A Case Study
- Taking Action
- Summary

What Is the Use of Logfiles Today?

- Development
  - Programming Errors
- Operation and maintenance
  - Root cause diagnosis of system failures
  - Configuration errors
- Research
  - Source data for analysis

Requirements Evolution

- Target
  - Today: Human readable
  - Tomorrow: Both human and machine readable
- Domain knowledge
  - Today: Domain specific, implicit domain knowledge, e.g., selected thresholds
  - Tomorrow: Comprehensive description
- Standardization
  - Today: Proprietary formats, homogeneous environment
  - Tomorrow: Standards for heterogeneous environments, universal tools, analysis server

Typical Problems with Error Logs

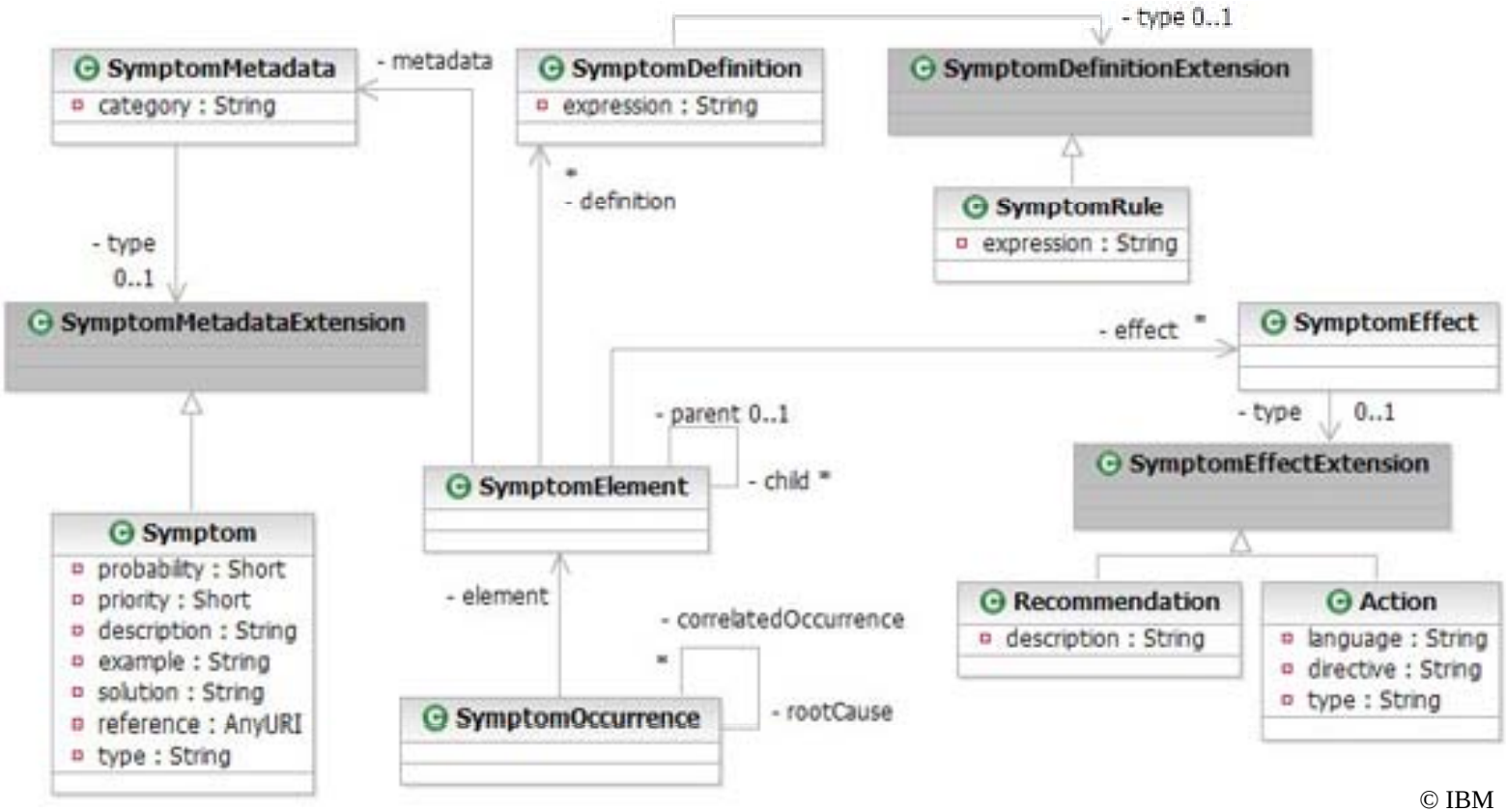
- Timestamp: No standard format and interpretation
- Unknown number representation (decimal or hexadecimal?)
- No token identifier / separator
- No machine readable format specification for the entire log
- Repetitive patterns (multiple reporting of problems)

Error Log Example

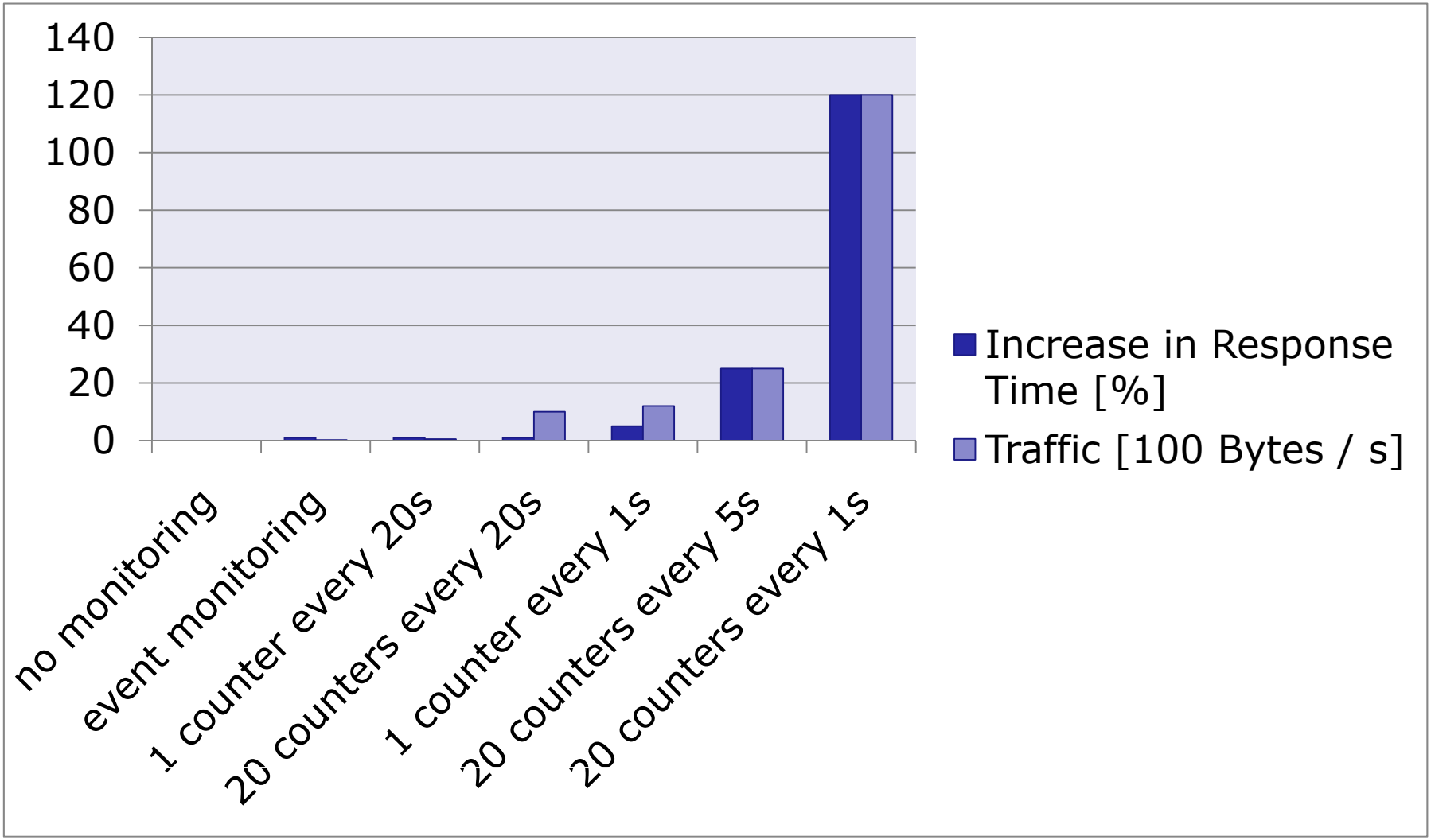
```
2004/02/09-19:26:13.634089-29836-00010-LIB_ABC
    ANOPTK#0243546463464346|0555553456|0000000000000-
    4456547457434-2.3.1|356546346|0001
2004/02/09-19:26:13.634089-29836-00010-LIB_ABC
    src=APPLICATION sev=SEVERITY_MINOR
2004/02/09-19:26:13.634089-29836-00010-LIB_ABC
    unknown value specified in Context 000256
```

WSDM Event Format (WEF)

- Log standard by Web Services Distributed Management (WSDM) approved by OASIS



Impact of Monitoring

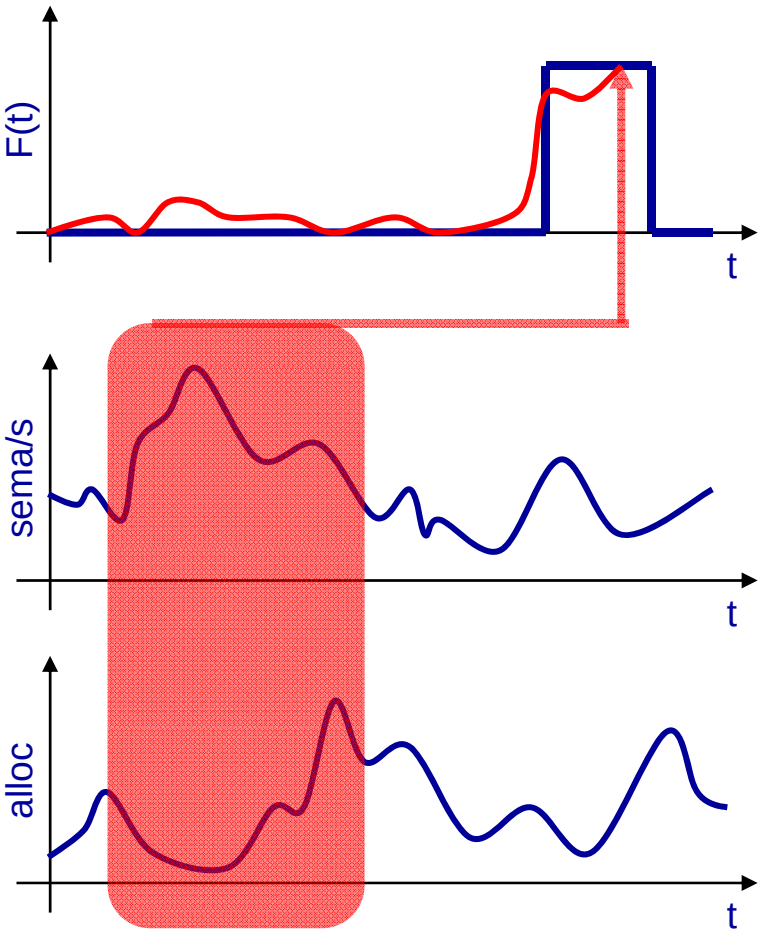


Contents

- Runtime Monitoring
- Error Logs
- Variable Selection
- Online Failure Prediction Taxonomy
- Online Failure Prediction Techniques
- **A Case Study**
- Taking Action
- Summary

Universal Basis Functions (UBF)

- Tailored to periodic measurements: e.g.,
  - Semaphore operations per second or minute
  - Allocated OS-kernel memory
- Function approximation approach: Express target value as function of input variables
- Examples for target values:
  - Availability
  - Memory consumption

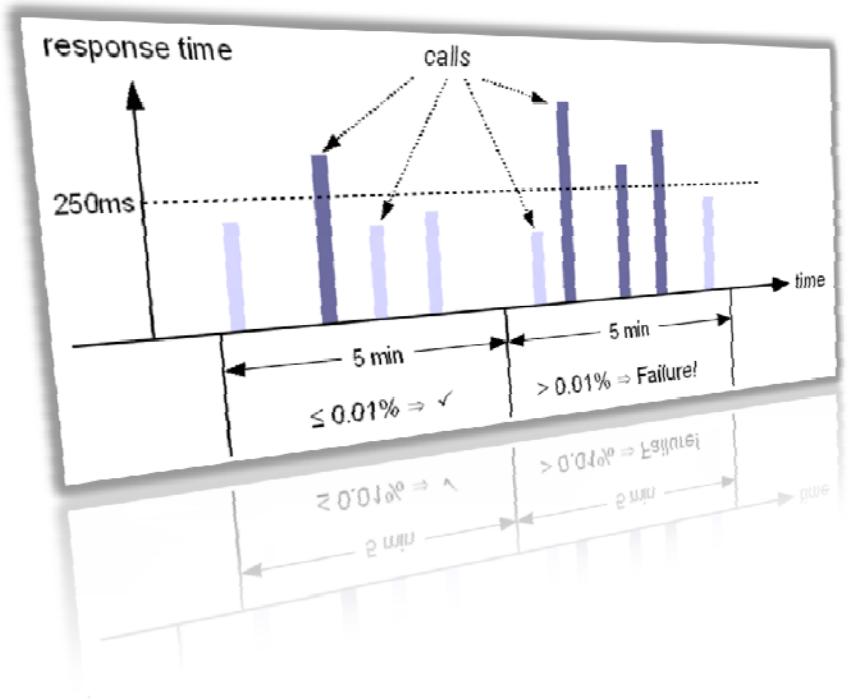


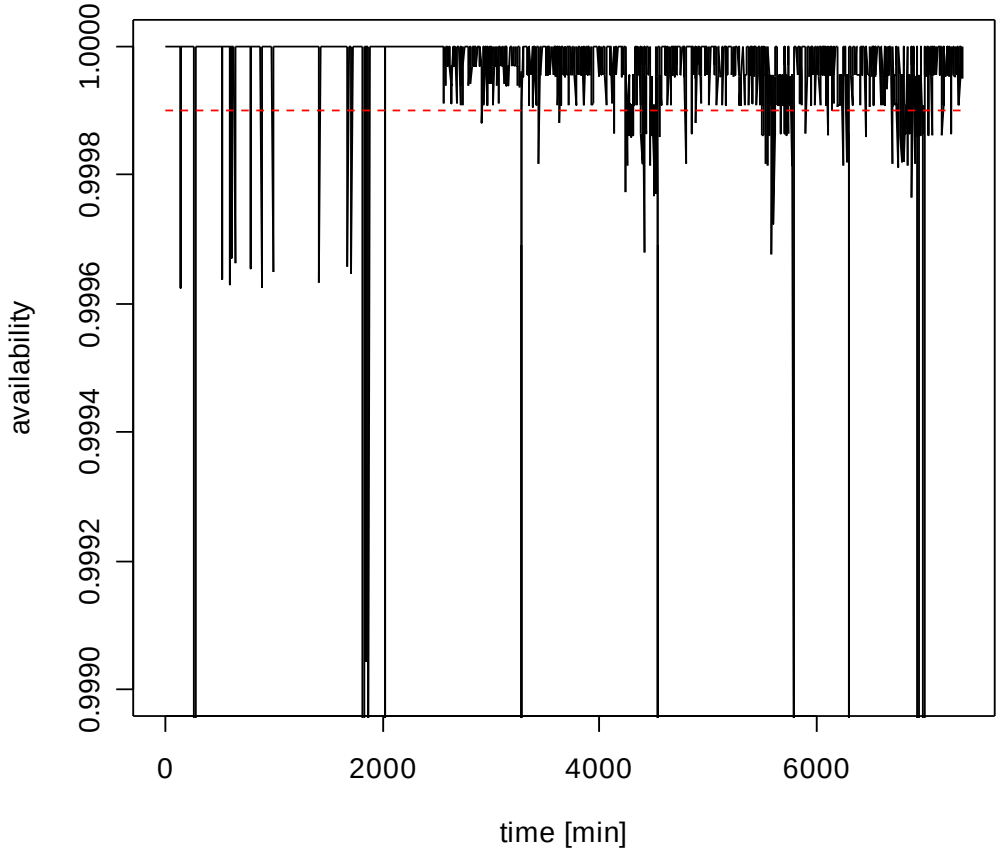
[Hoffmann, Malek 2006; Hoffmann 2005]

Case Study

- Commercial telecommunication platform
- Platform implements service control functions
  - Examples: billing, SMS, pre-paid services
- 400-10,000 service requests per minute
- Distributed and component based system
  - 1.5+ million lines of code
  - 2000+ classes and 200+ components
  - Two nodes (up to eight)

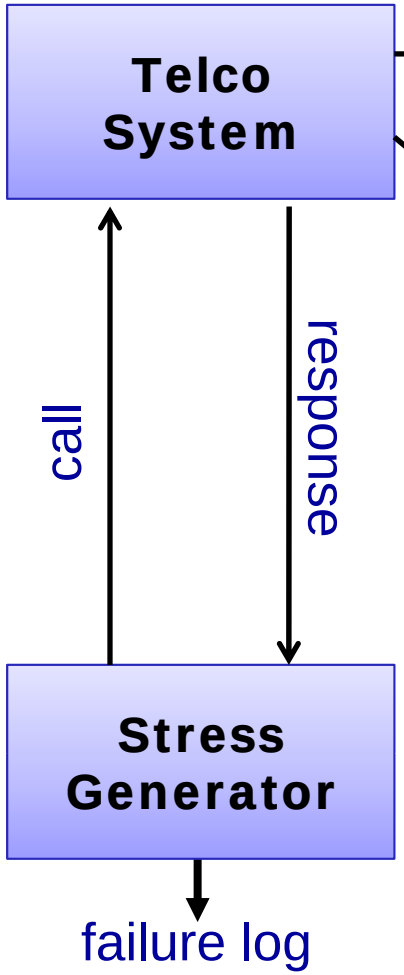
Definition of Failures



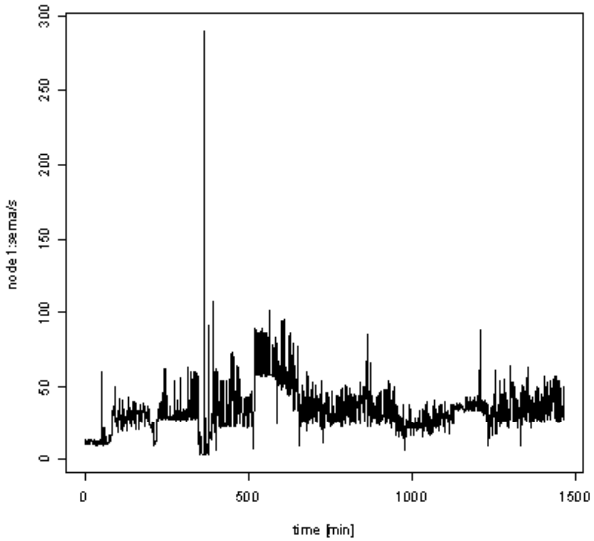


$$A = \frac{\#\,\text{calls with response time} \leq 250ms}{\text{total}\,\#\,\text{of calls}}$$

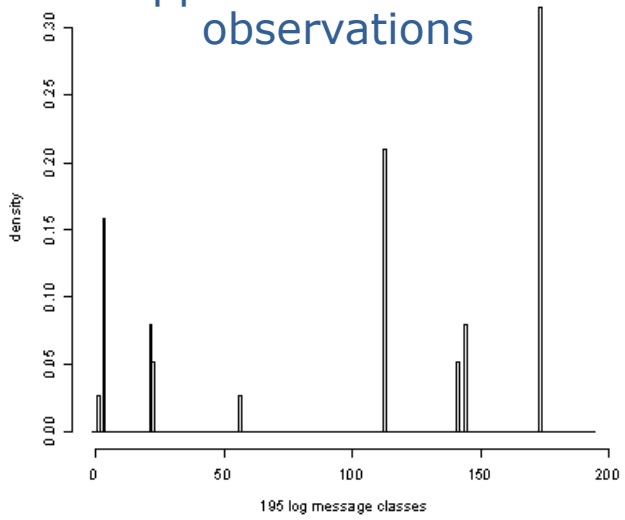
Experimental Setup



SAR
(system activity reporter)
- Periodic data
- 96 variables
- Approx. 13 million observations

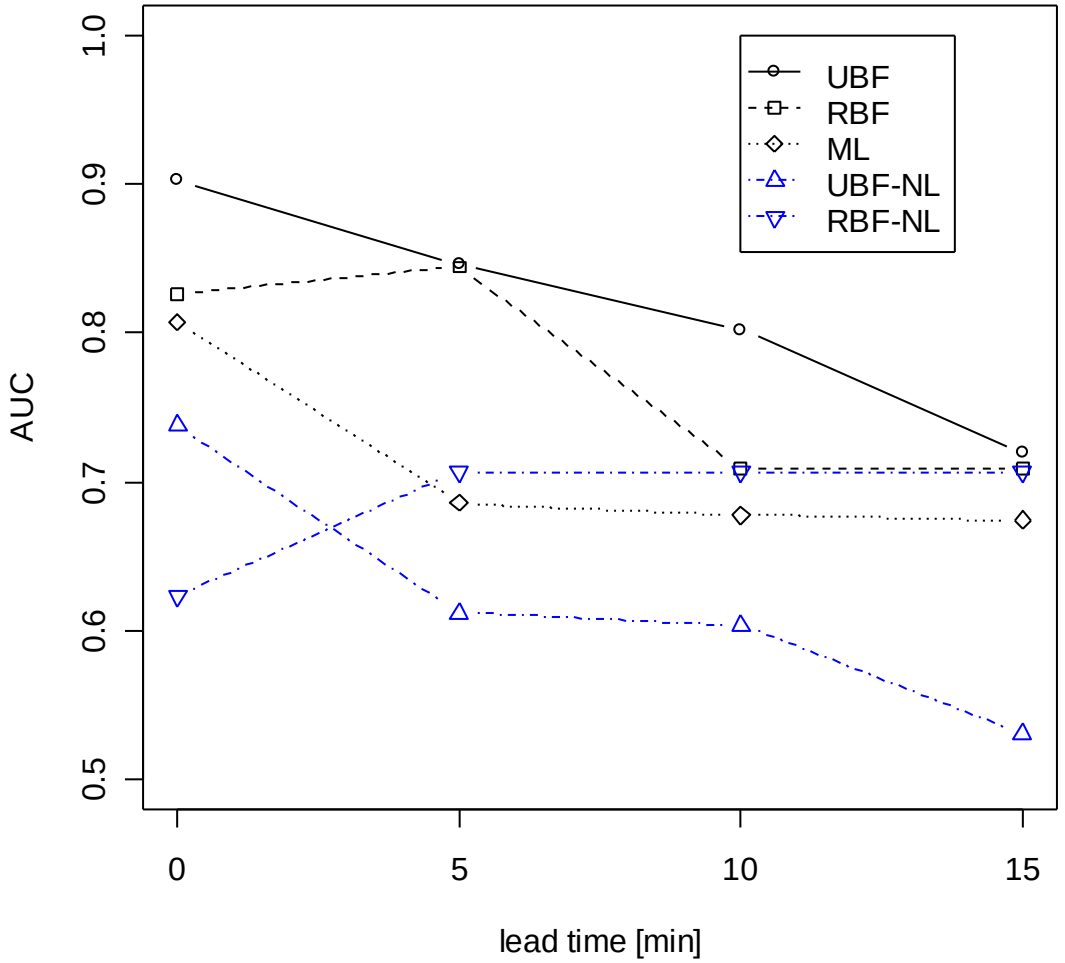


Log file data
- Event driven
- 390 variables
- Approx. 4 million observations

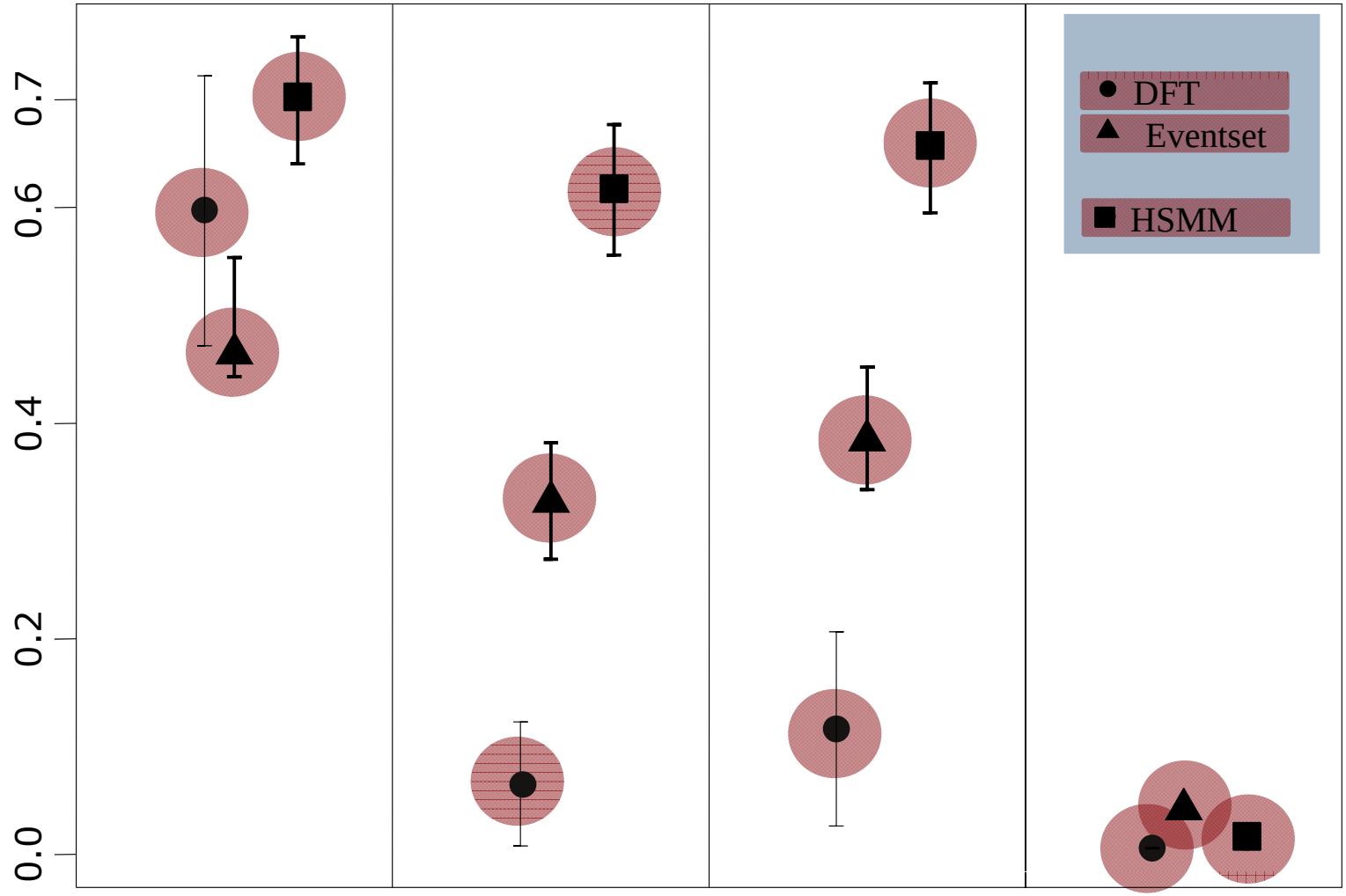


Results for UBF

- Plotting recall over false positive rate yields Receiver Operating Characteristic (ROC) curve
- We use the Area Under ROC Curve (AUC) for comparison
- A perfect predictor results in AUC = 1.0
- Results: Mean AUC values for 0,5,10,15 minutes predictions into the future
- Comparison of UBF with Maximum Likelihood (ML), Radial Basis Functions (RBF) and non-linearities (NL)



Comparison of Techniques



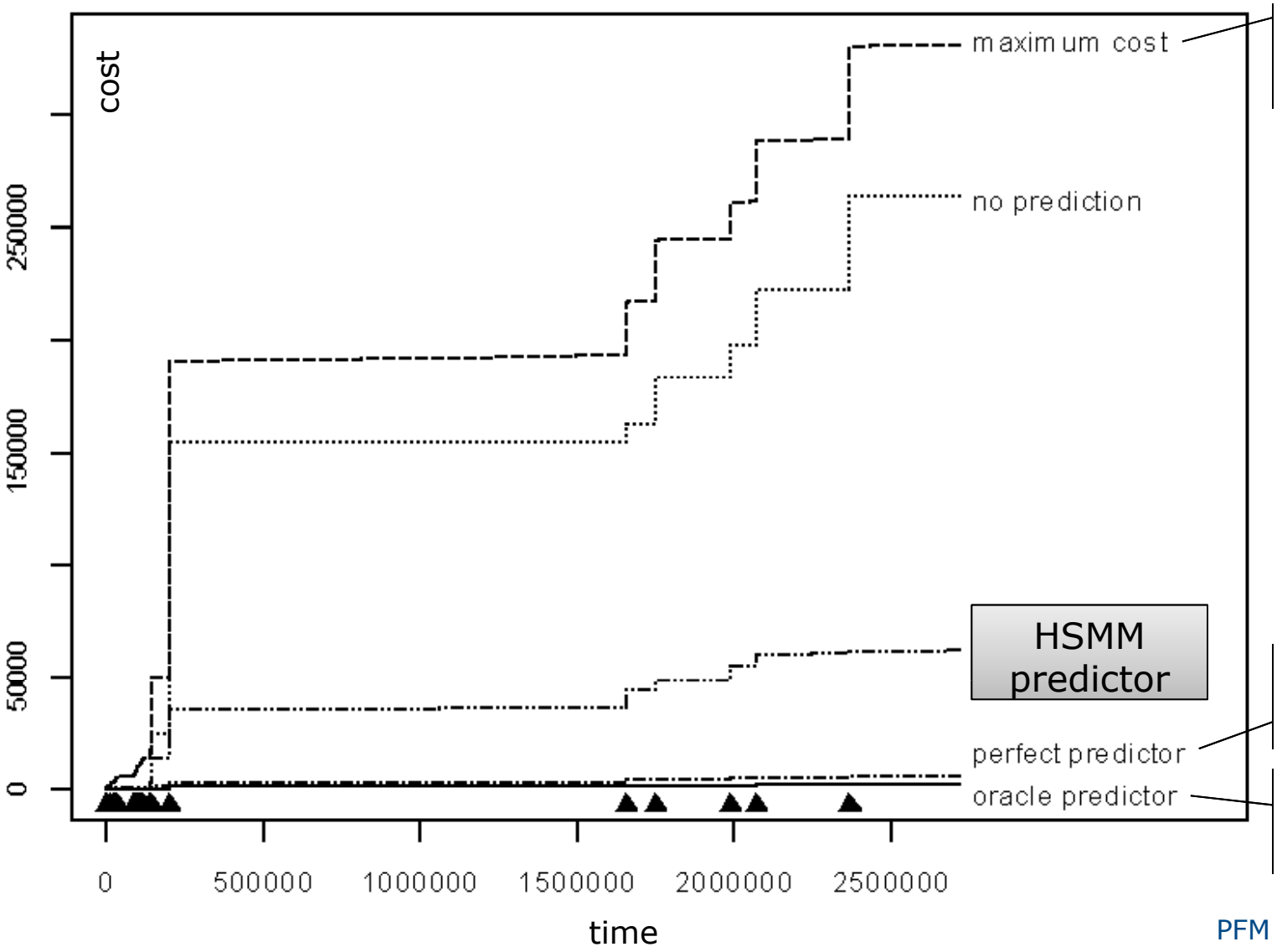
Contents

- Runtime Monitoring
- Error Logs
- Variable Selection
- Online Failure Prediction Taxonomy
- Online Failure Prediction Techniques
- A Case Study
- Taking Action
- **Summary**

Accumulated Runtime Cost

- Assign cost to:
  - Correct no-alarms:
    - Correctly predicted that no failure is imminent, no action performed
    - Cost of 1 cost unit
  - Correct alarms:
    - A failure was predicted and a proactive action has been carried out
    - Cost of 10 cost units
  - False alarms:
    - Preventive actions are performed in vain
    - Cost of 20 cost units
  - Missing alarms:
    - True Failure is not predicted: worst case
    - Cost of 100 cost units
- Based on log data, we plot accumulated runtime cost

Resulting Cumulative Cost



Only FP and FN predictions

Only TP and TN predictions

Only TP predictions

Research Issues

- Runtime Monitoring
  - Overhead vs. completeness / usefulness
  - Raw data vs. extracting information
- Root Cause Analysis / Diagnosis
  - What methods from traditional diagnosis can be used proactively?
- Prediction-driven actions
  - What recovery methods use failure prediction most effectively?
- Decision strategies
  - Models and methods to decide upon or schedule actions
- Accuracy
  - Accuracy of predictive diagnosis and prediction techniques
  - Success probabilities, performance impact of recovery and preventive maintenance actions
- Analysis
  - How sensitive is PFM to system changes
  - What methods from control theory can be applied?
  - Models for Proactive Fault Management
  - PFM economics

Key Choices for Effective PFM

1) Monitoring: what variables, when, how and at what level

2) Failure Prediction: model, method and effectiveness measures

3) Failure Avoidance or Recovery: when, how and at what level

4) Closing the Loop: learn, refine, tune and apply again