Proactive Fault Management: Status and Perspectives

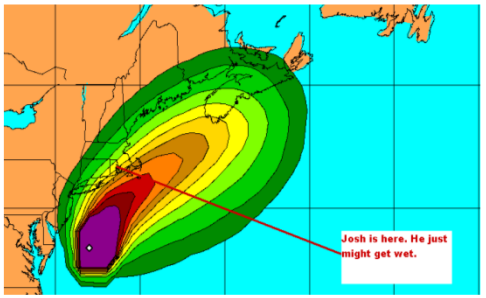
Miroslaw Malek

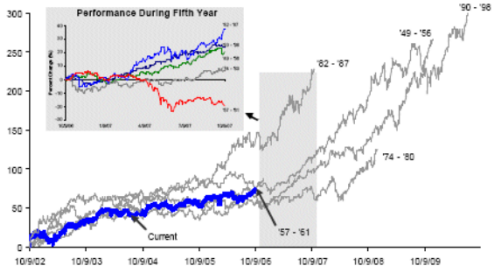
Visegrad, June 28, 2013

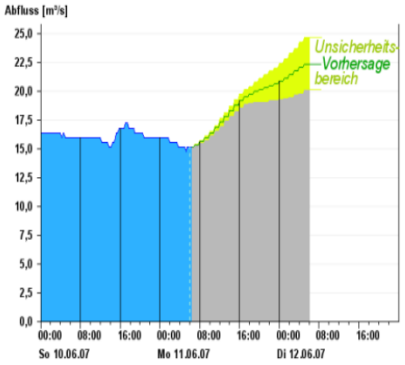
Università della Svizzera italiana
Faculty of Informatics
Advanced Learning and Research Institute ALaRI

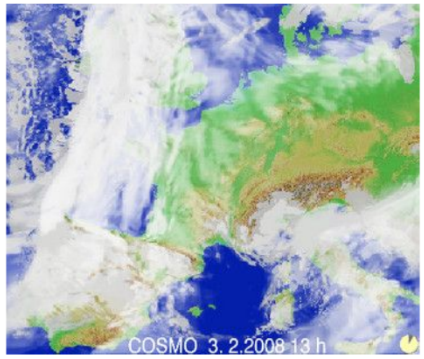
Predicting the Future

- 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, including increasing number of dependability researchers and engineers



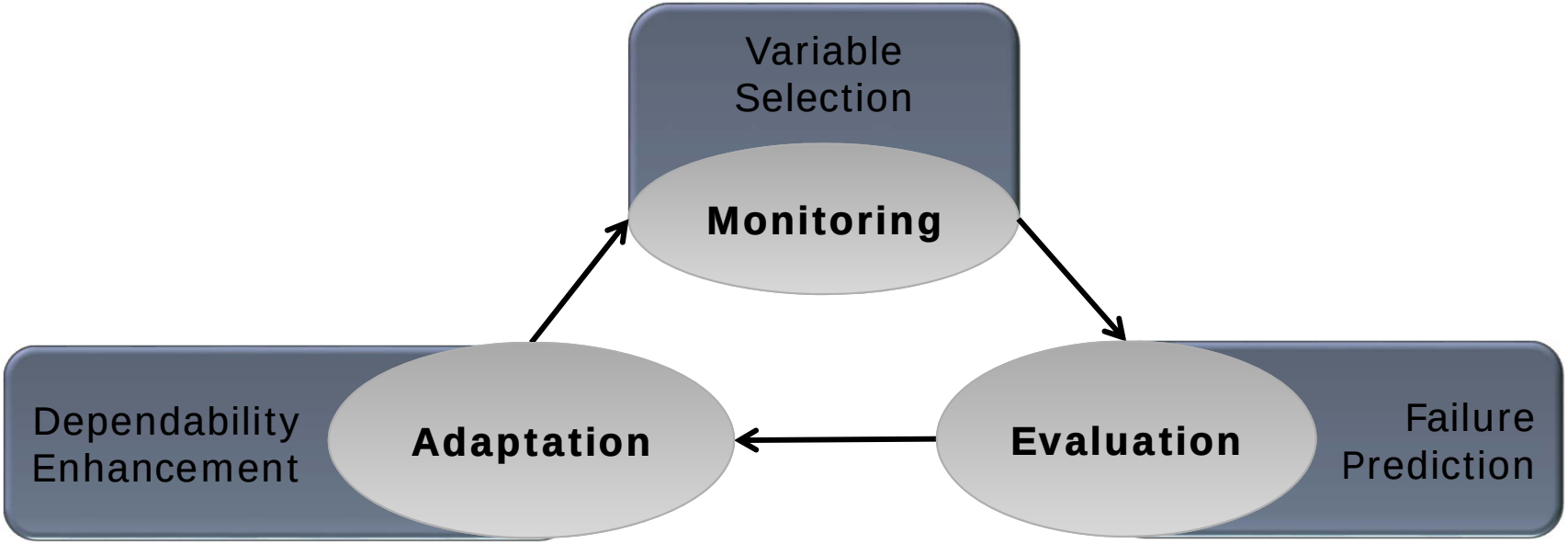






Proactive Fault Management (PFM)

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.



Motivation

- Ever-increasing systems complexity
- Ever-increasing amount of data (big data)
- 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

New Significance with Big Data

- Measuring and predicting the world is becoming a common practice
- People, organizations and machines collect data in unprecented quantities....

Example: 40 GB of monitoring data per day in server cluster processing phone calls

Key problem:

What to collect and how to process it to get the useful information leading to a correct decision?

And so it is with a failure prediction as well.

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

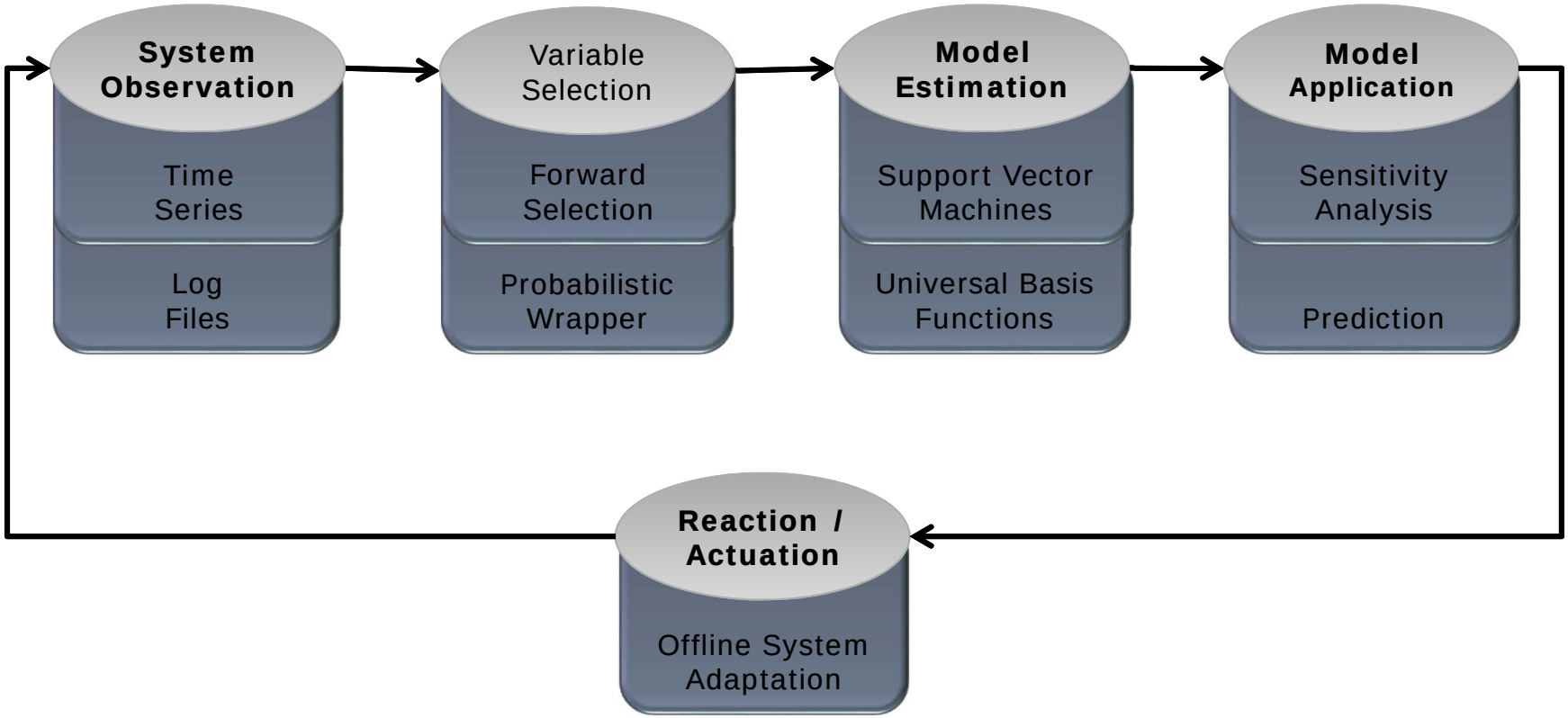
The Philosophy

- 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)

How to Get There (Off-Line Loop)?



[Hoffmann, Trivedi, Malek 2007]

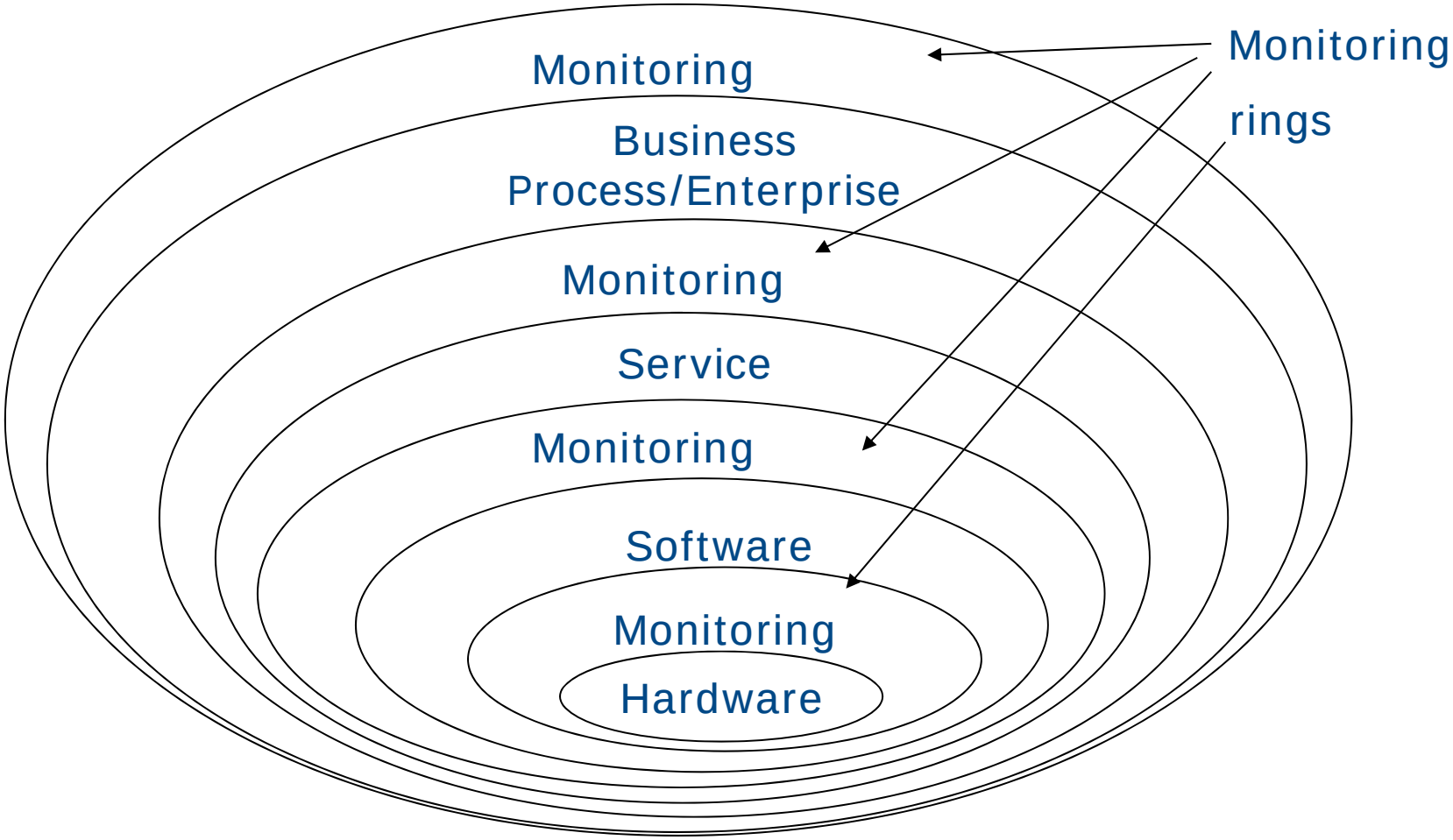
Contents

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

Runtime (Online) Monitoring

- Runtime monitoring is an observation of system variables for a given purpose such as diagnosis or failure prediction
- Fundamental questions:
  - What variables to monitor?
  - How and where 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?
  - What is the impact on performance?
- **Remember:** monitoring is not free and never complete

Monitoring – At What Level?



Types of Data Sources

- Error Logs (Logfiles)
  - Lack of uniformity
  - Standards are emerging
  - Redundancy (in some cases a problem is reported many times; in one case we have seen it over 60,000 times)
- System Activity Reporter (SAR) data
  - Over 4500 parameters can be monitored by Windows and up to 60,000 in a small multiserver system
  - 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
- Summary and Perspectives

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
  (binary, octal, 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
```

Contents

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

Variable Selection

- What are the most indicative variables to use for failure prediction?
- There are thousands of 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:

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

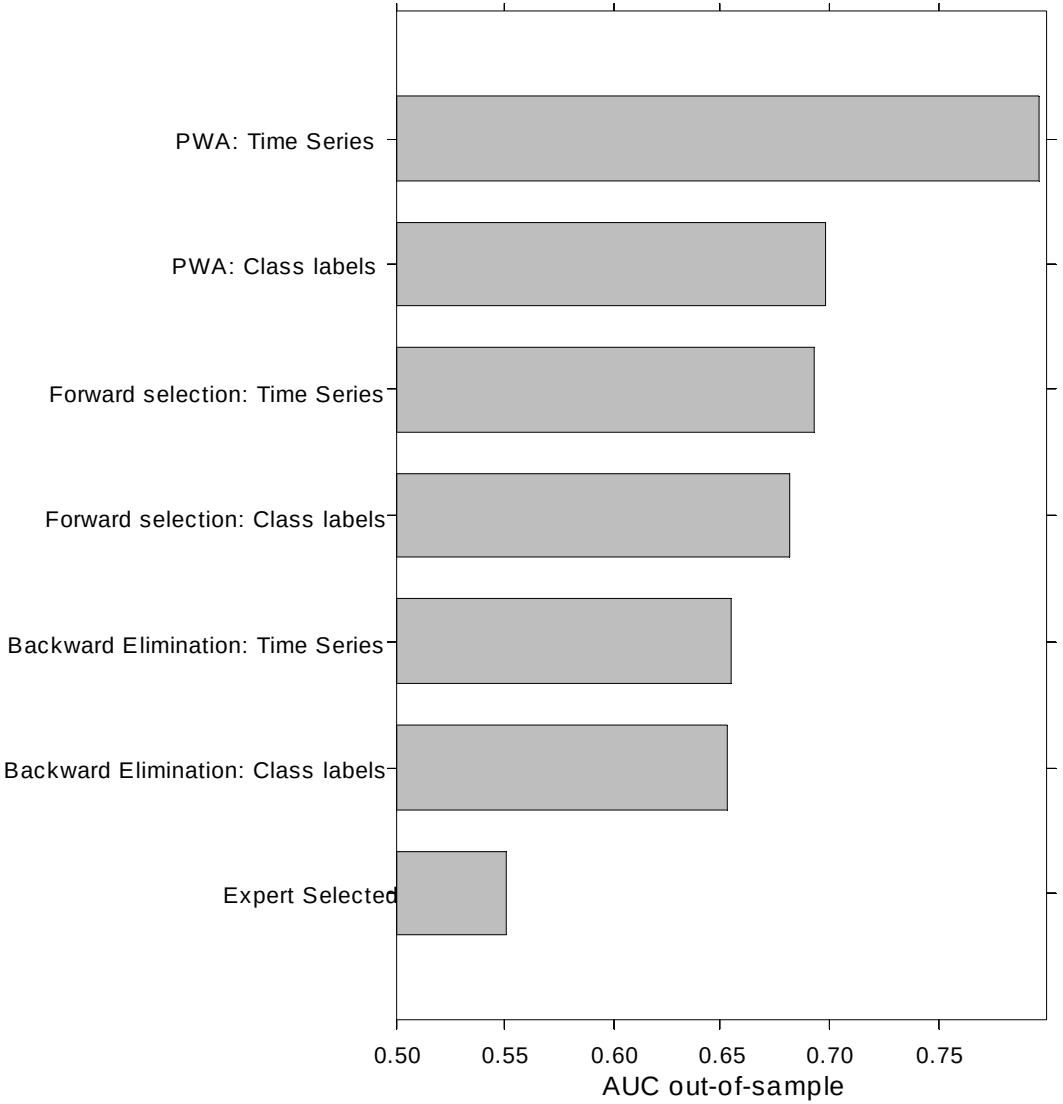
- Combinatorial explosion!

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

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

Four Ways of Detecting Faults

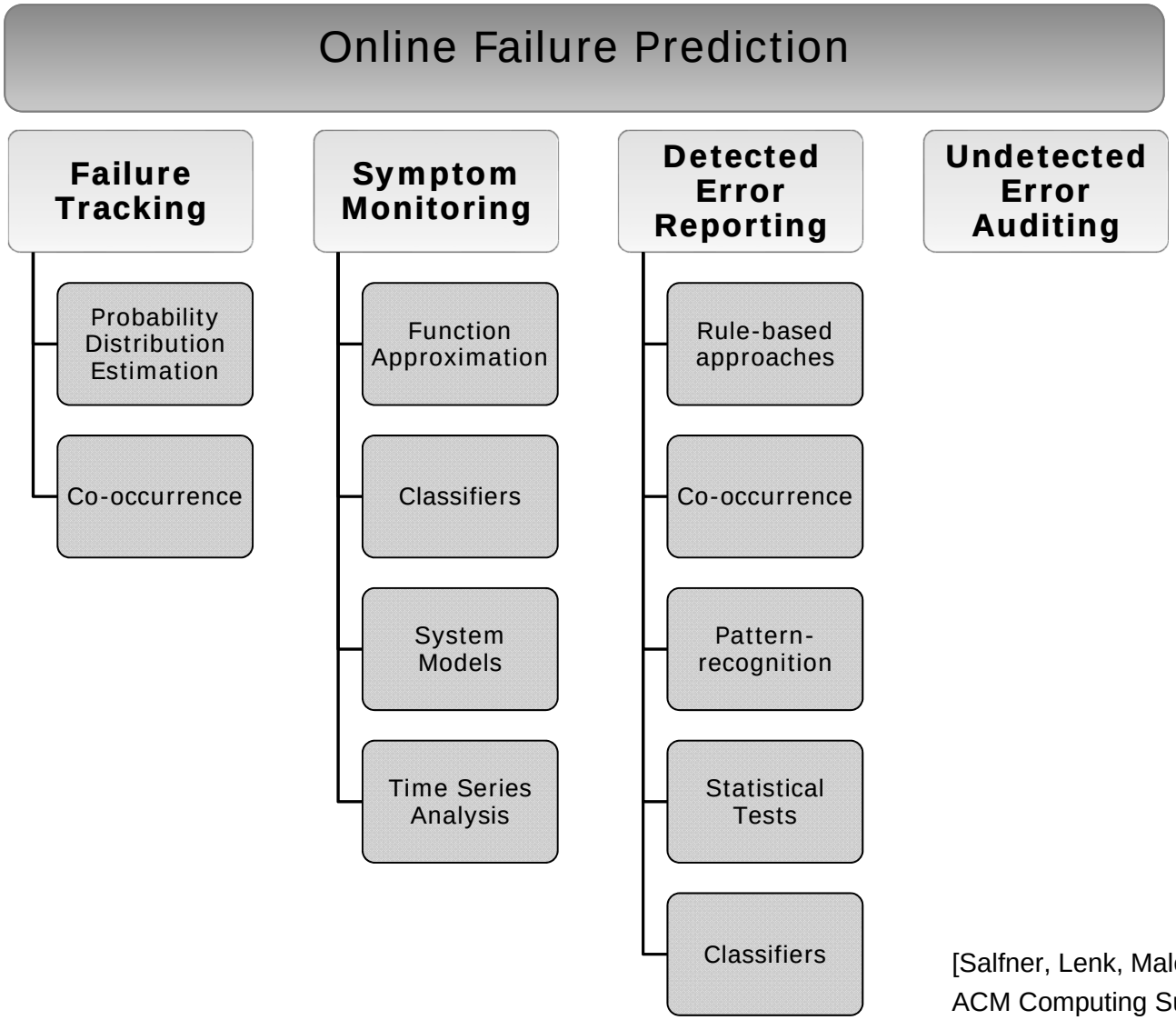
(1) The system can be *audited* in order to actively search for *faults*, e.g., by testing on checksums of data structures, etc.

(2) System variables 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,
ACM Computing Surveys, 2010]

Contents

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

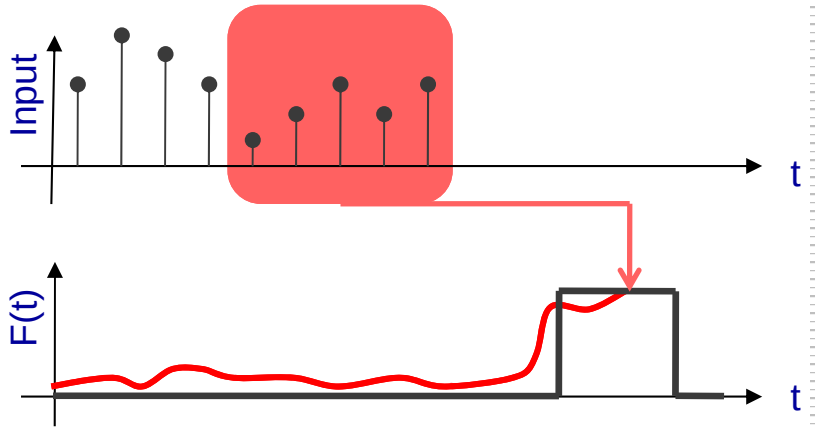
Online Failure Prediction - Definition

- 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***.

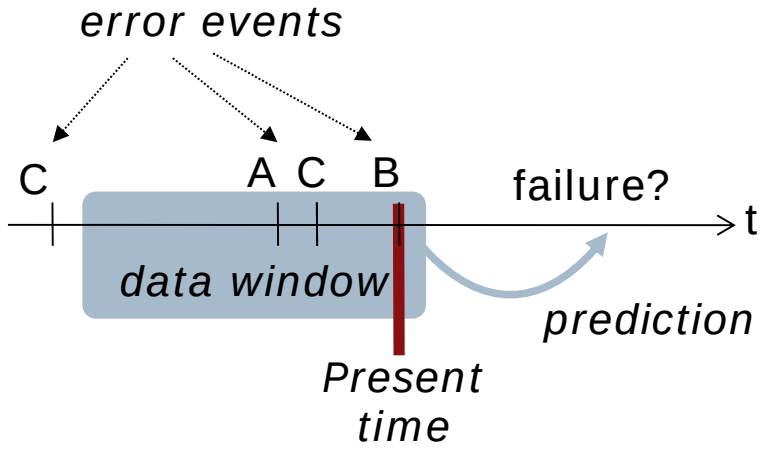
- The output of online failure prediction can either be
  - a decision that a failure is imminent or not, or
  - some continuous measure evaluating how failure-prone the current situation is.

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

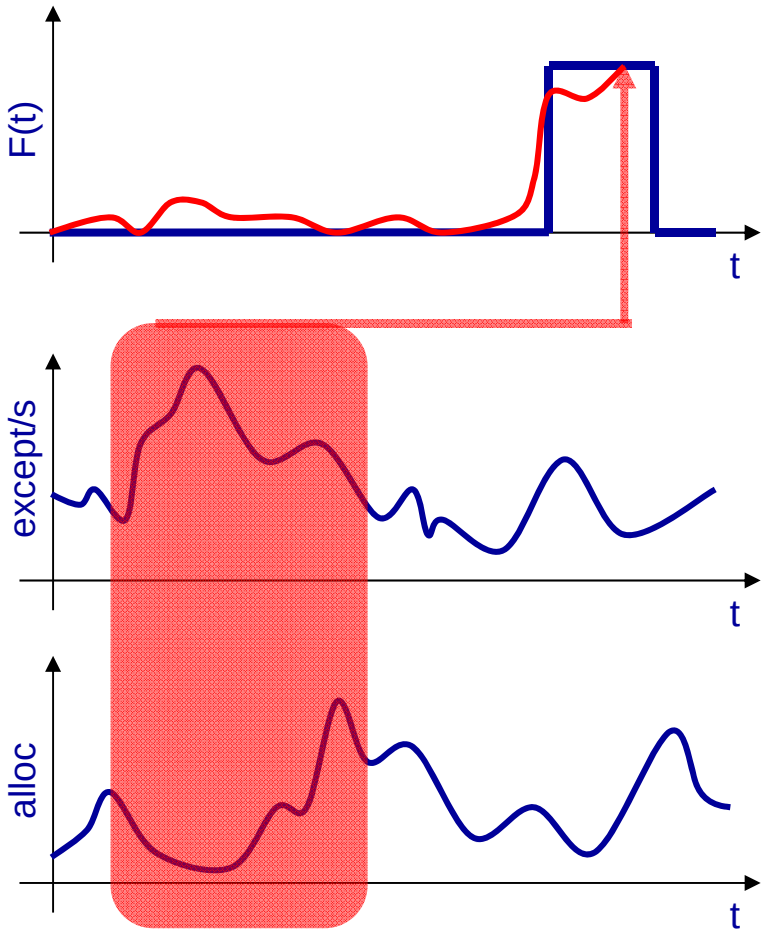


Prediction Techniques Examples

1. *Universal Basis Functions (UBF)*
2. *Hidden Semi-Markov Model (HSMM)*
3. Eventset method

Universal Basis Functions (UBF)

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



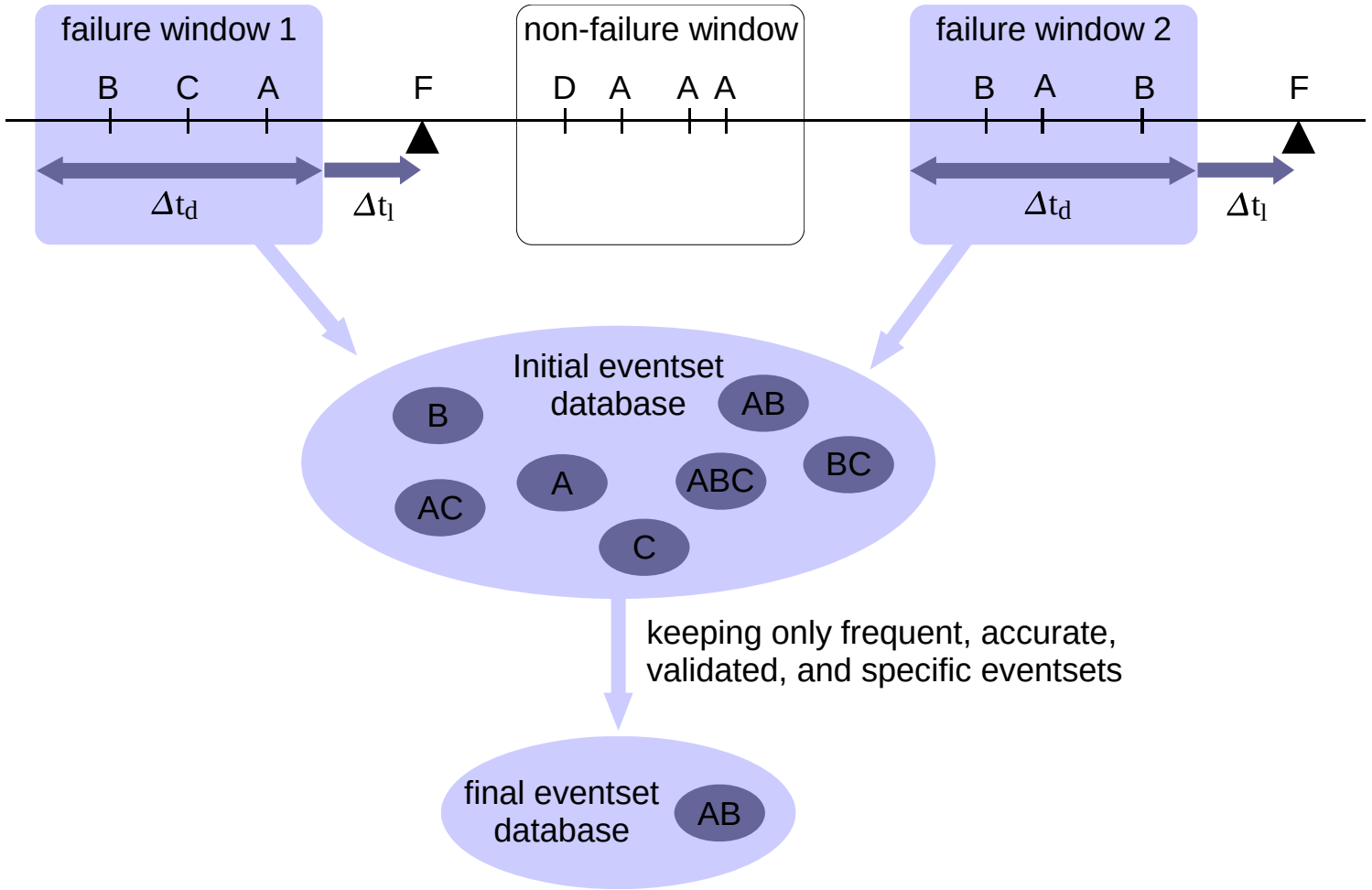
[Hoffmann, Malek 2006; Hoffmann 2005]

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

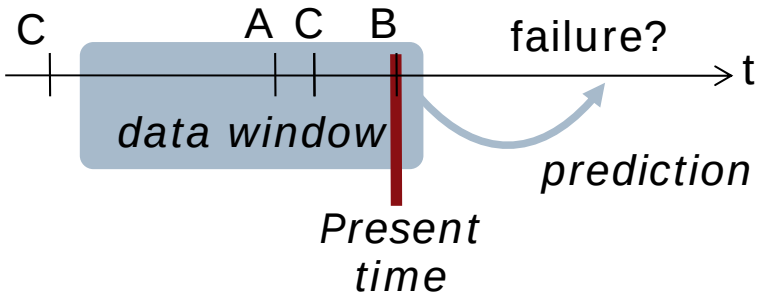
[Vilalta, Ma 2002]

Event-set Method



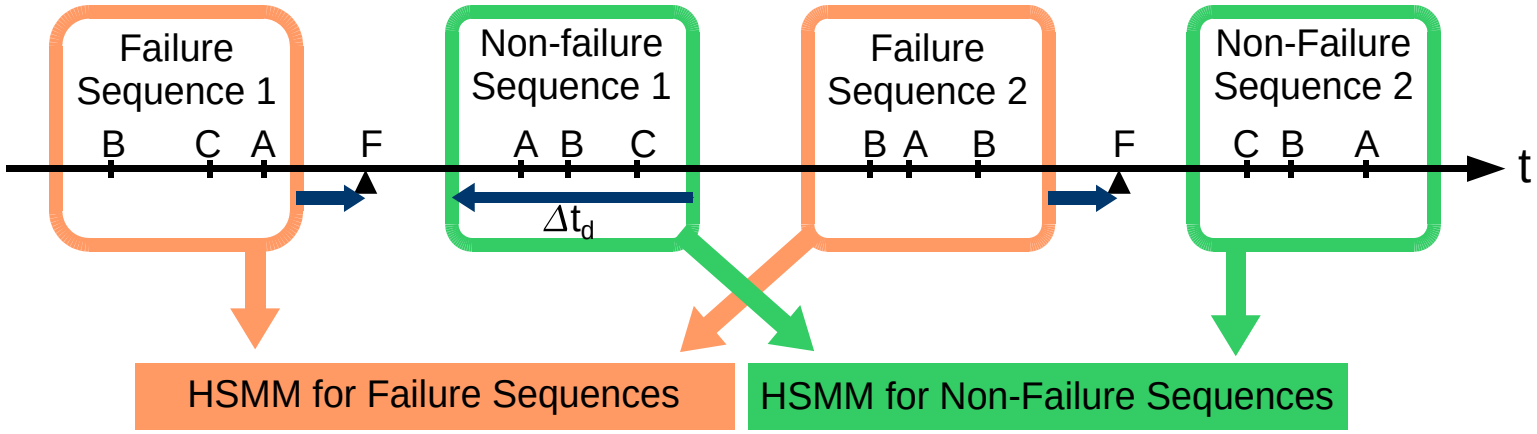
Hidden Semi-Markov Model Prediction

- 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

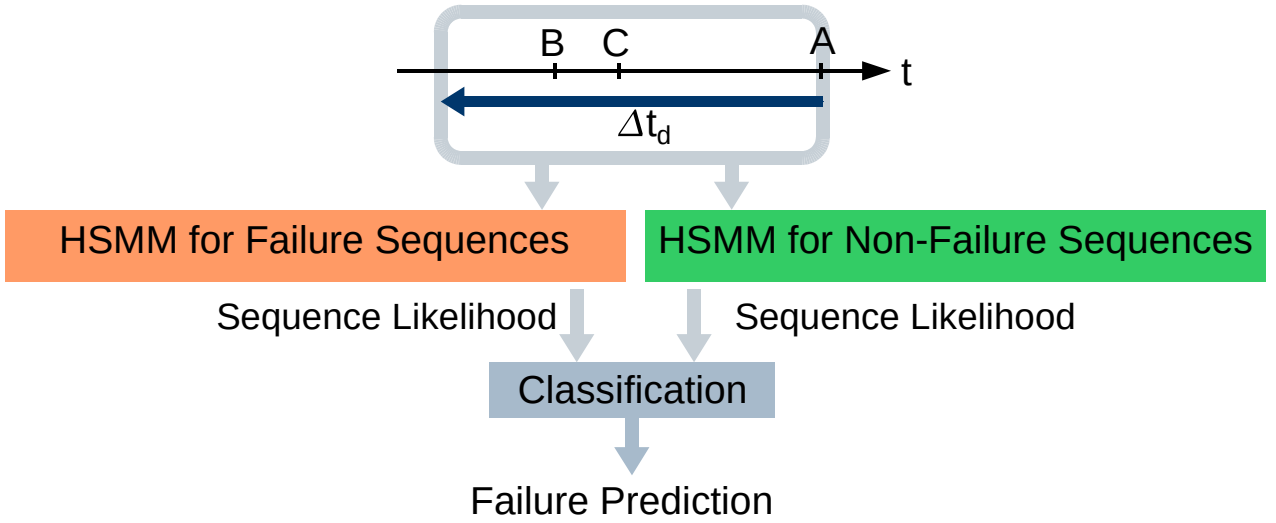


Machine Learning: Two Steps

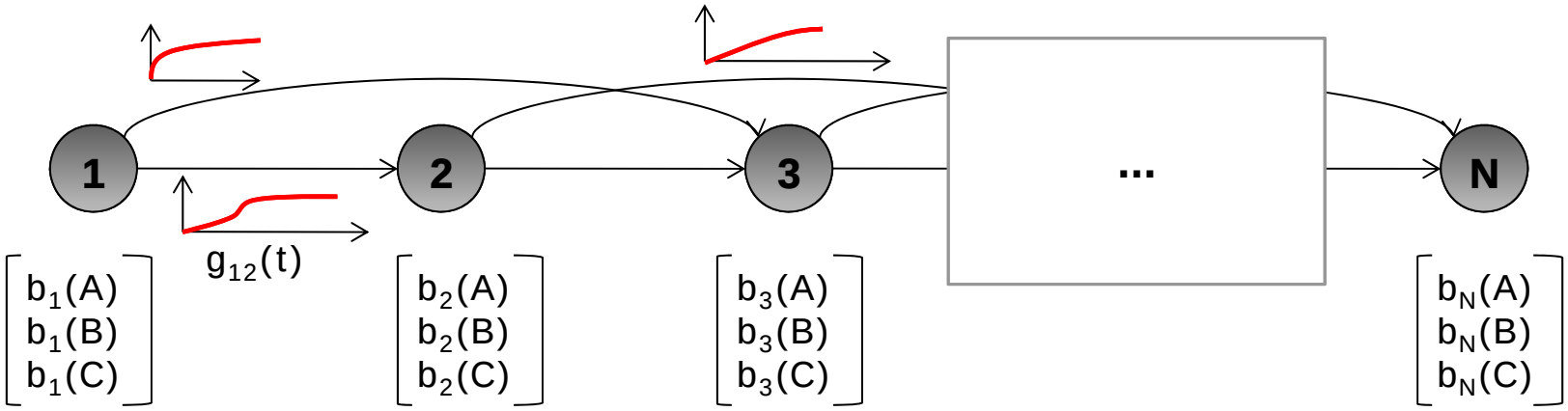
1.Training: Fit model parameters to training data



2.Prediction: Processing of runtime measurements



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 $g_{ij}(t)$

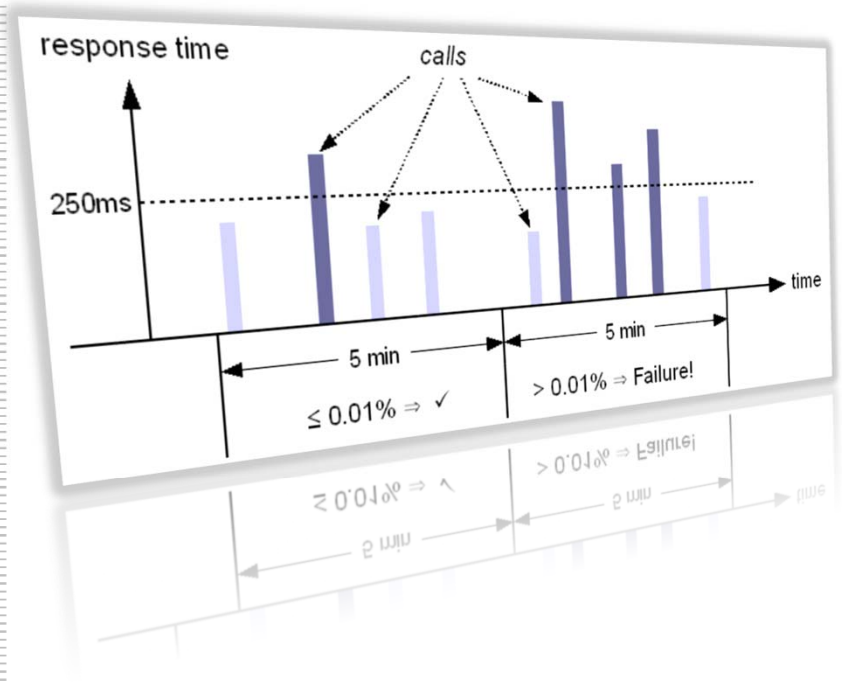
Contents

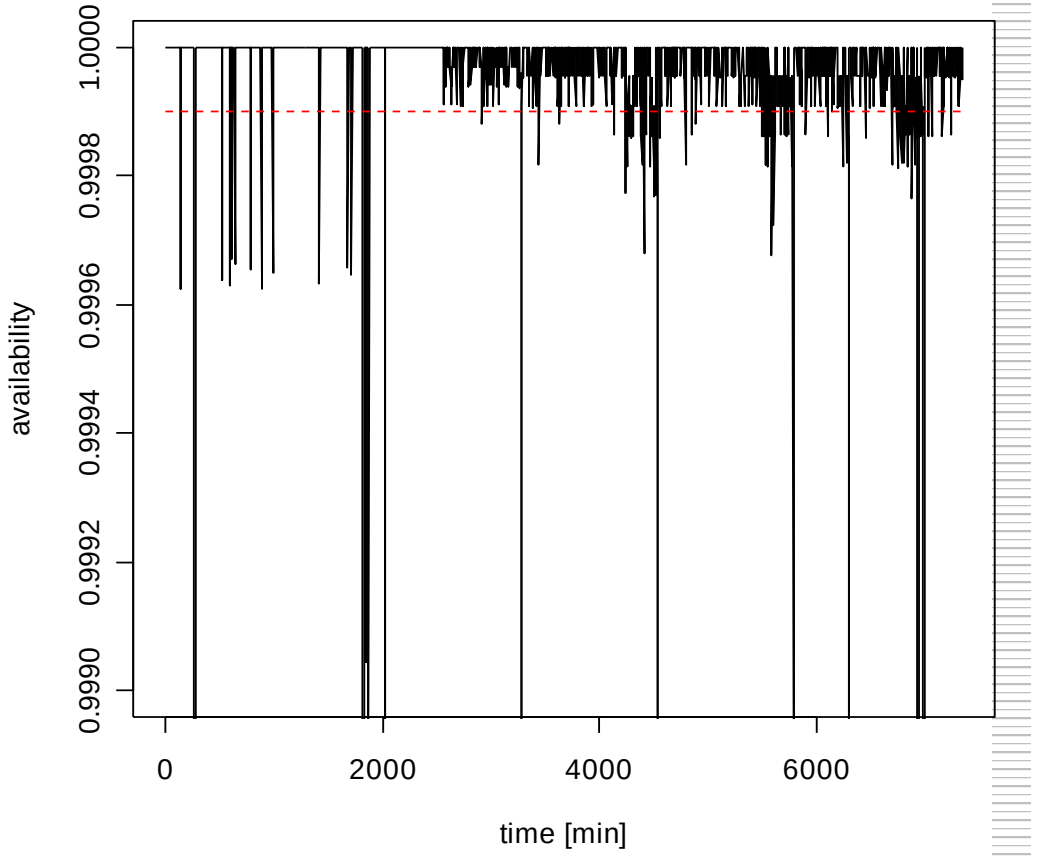
- Runtime Monitoring
- Error Logs
- Variable Selection
- Online Failure Prediction Taxonomy
- Online Failure Prediction Techniques
- **A Case Study**
- Summary

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)

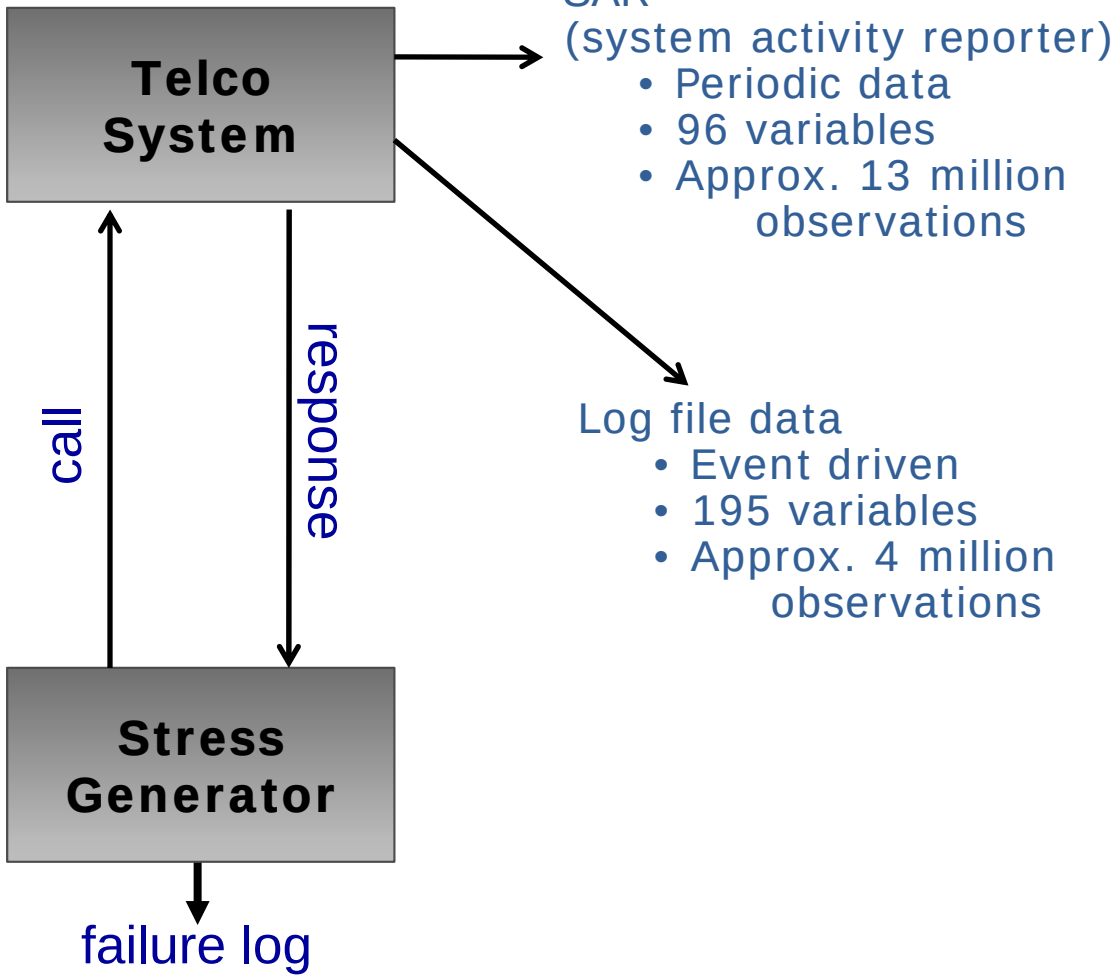
Definition of Failures





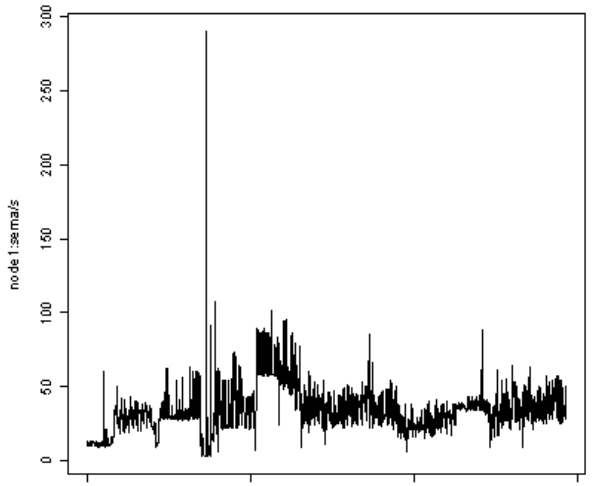
$$A = \frac{\#\,\text{calls with response time} \leq 250ms}{\text{total}\,\#\,\text{of calls}}$$

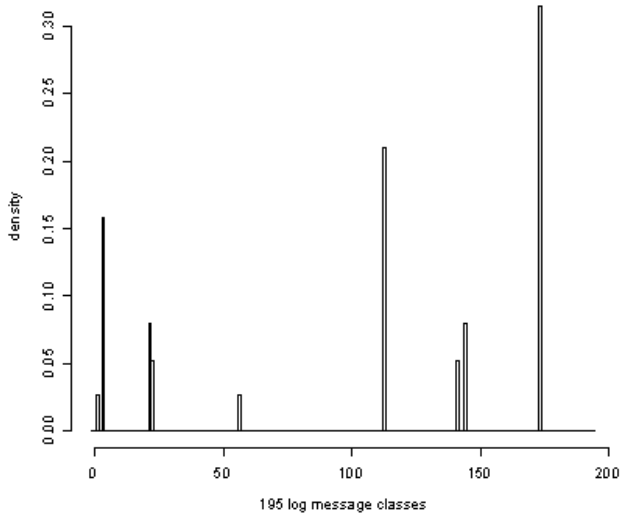
Experimental Setup



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

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





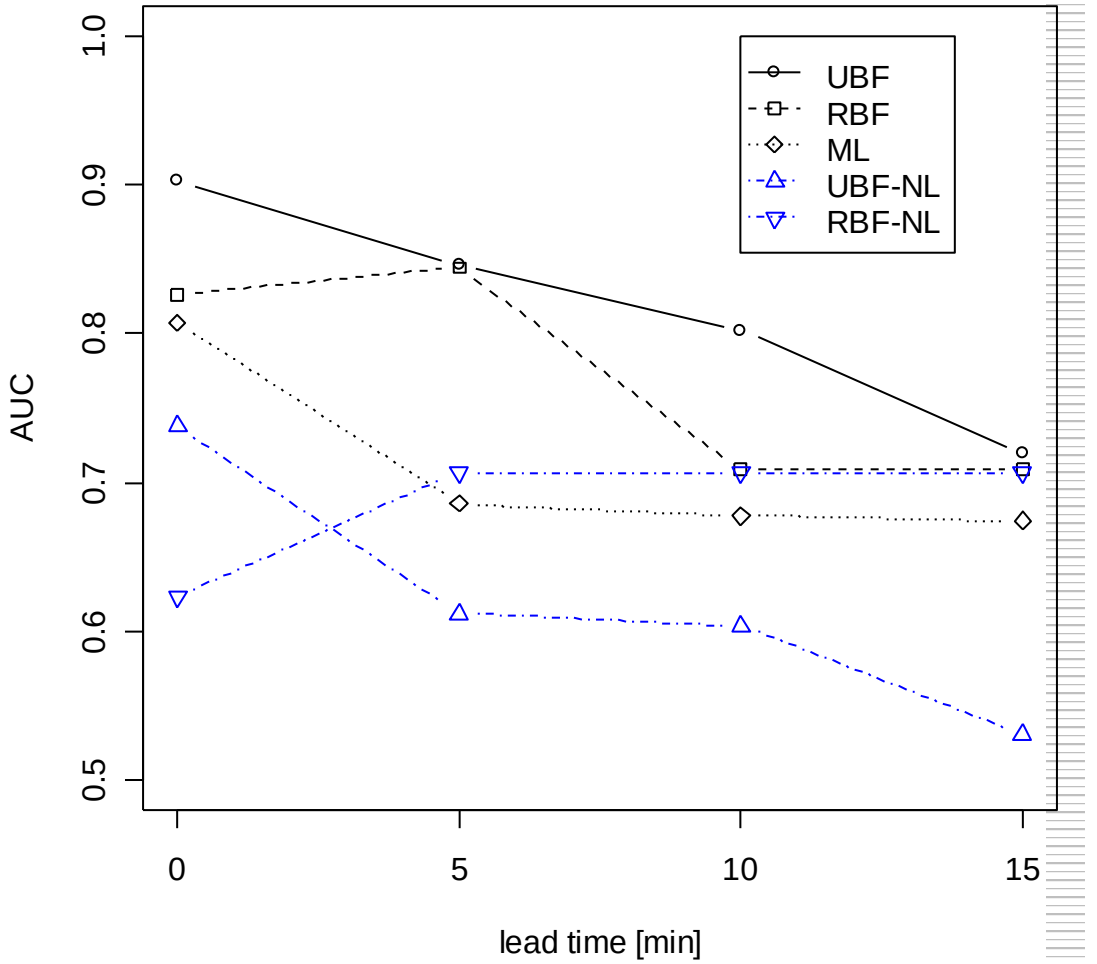
Precision, Recall and other Metrics

contingency table	True failure	True success	Sum
Failure alarm	Correct alarm	False alarm	# Alarms
No warning	Missing alarm	Correct no-alarm	# No-Alarms
Sum	# Failures	# Successes	# Total

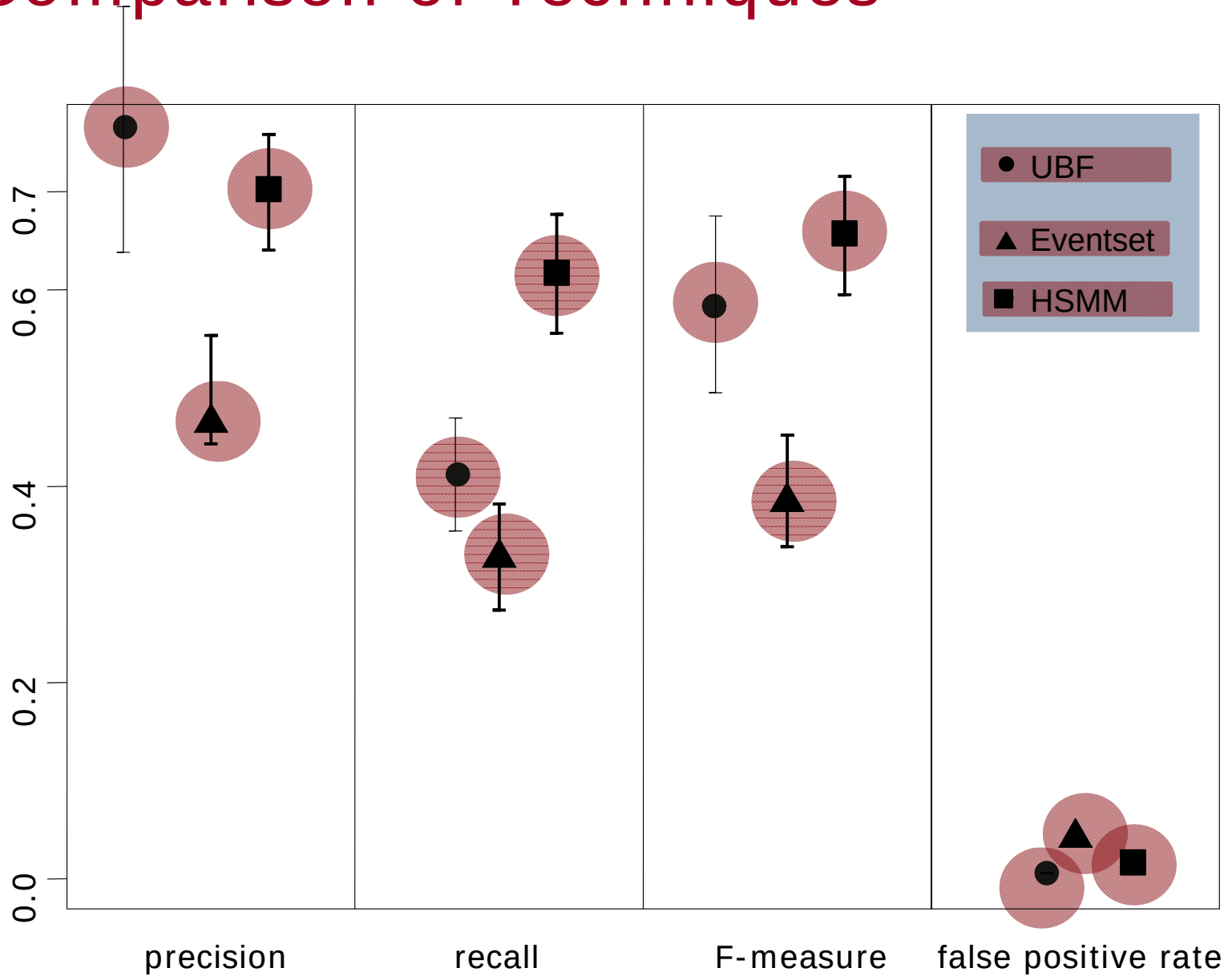
- Precision: fraction of correct alarms: $\text{precision} = \frac{\text{correct alarms}}{\text{total \# of alarms}}$
- Recall: fraction of predicted failures: $\text{recall} = \frac{\text{correct alarms}}{\text{total \# of failures}}$
- F-measure: harmonic mean: $\text{F} - \text{measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$
- False positive rate: $\text{false positive rate} = \frac{\text{false positives}}{\text{\# of successes}}$

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
- **Summary**

Secret of Success: Variable Selection

- Correct selection may decrease model error (mean square error) up to an order of magnitude
- Focus on variable selection to improve model quality is critical (choosing the right variables is more important than choosing the right type of modeling technique); Examples:

  a) sem/sec b) OS memory allocation c) response time, d) swap space, e) physical memory used f) load
- Strong nonlinearities and changing dynamics detected in data favor nonlinear modeling techniques over linear

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 machine learning and 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

Perspectives

- Proactive Fault Management methods and technologies have the potential to significantly improve system availability (even by an order of magnitude or more)
- Systems must be monitored and automatically triggered by failure prediction to avoid failures or speed up diagnosis and repair
- *PFM can* significantly not only increase availability but also reduce runtime cost in comparison to classical methods
- Predictive technologies are and will be used in all walks of life such as health condition prediction (e.g. heart failure prediction), energy management, traffic jams elimination, etc.

Acknowledgment

I would like to acknowledge the contributions of **Felix Salfner** and **Günther Hoffmann** to development and presentation of Hidden Markov Models and UBF techniques and **Steffen Tschirpke**'s to measurements.

References

[Hoffmann 2005] Hoffmann, G.: *Failure Prediction in Complex Computer Systems: A Probabilistic Approach*, Ph.D. Thesis, Shaker Verlag GmbH, Germany;

[Hoffmann, Malek 2006] Hoffmann, G. and Malek M.: *Call Availability Prediction in a Telecommunication System: A Data Driven Empirical Approach* In: 25th IEEE Symposium on Reliable Distributed Systems (SRDS 2006), Leeds, UK, October 2006

[Hoffmann, Trivedi, Malek, 2007] Hoffmann, G., Trivedi, K.S., Malek, M*.: A Best Practice Guide to Resources Forecasting for the Apache Webserver,* IEEE Transactions on Reliability, 56(4), Dec. 2007

[Salfner, Malek 2007] F. Salfner, M. Malek *Using hidden semi-markov models for effective online failure prediction* In: Proceedings of 26th IEEE International Symposium on Reliable Distributed Systems (SRDS), 2007

[Salfner 2008] Salfner, F.: *Event-based Failure Prediction: An Extended Hidden Markov Model Approach*; Dissertation.de, Berlin, 2008. Available at www.rok.informatik.hu-berlin.de/Members/salfner

[Salfner, Lenk, Malek, CSUR] Salfner, F., Lenk, M., and Malek, M. *A Survey of Online Failure Prediction Methods*; ACM Computing Surveys, 2010

[Vilalta, Ma 2002] Vilalta, R. and Ma, S. *Predicting rare events in temporal domains*. In Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02).