HUMBOLDT-UNIVERSITÄT ZU BERLIN

Tutorial on

Predictive Algorithms and Technologies

for

Availability Enhancement

Miroslaw Malek
Institut für Informatik
Humboldt-Universität zu Berlin
malek@informatik.hu-berlin.de

International Service Availability Symposium (ISAS2008)
Venue: University of Tokyo, Tokyo, Japan.
May 19, 2008

Predictive Algorithms and Technologies for Availability Enhancement

Miroslaw Malek, Institut für Informatik
Humboldt-Universitaet zu Berlin
malek@informatik.hu-berlin.de

Predicting the future has fascinated people throughout civilizations but until the 20th century it has been more of a magic than science. Ability to predict the future may have a significant impact on wide spectrum of applications ranging from communication systems to health monitoring. With development of dynamic systems research, data mining and computer technology we seem to be better equipped to tackle the prediction problem with a more science-based, goal-oriented approach.

In this tutorial, we focus on predictive algorithms and technologies which may have a major impact on computer systems availability and performance. We first survey long-term and short-term prediction techniques, introduce prediction quality measures, and then demonstrate how the availability of software and hardware systems can be increased by preventive measures which are triggered by short-term failure prediction mechanisms.

We present and evaluate mainly non-parametric techniques which model and predict the occurrence of failures as a function of discrete and continuous measurements of system variables.

We introduce two modelling approaches in detail: Hidden Markov models and a function approximation technique utilising universal basis functions. The presented modelling methods are data driven rather than analytical and can handle large amounts of variables and data. They offer the potential to capture the underlying dynamics of even high-dimensional and noisy systems. Both modelling techniques have been applied to real data of a commercial telecommunication platform. The data includes event-based log files and measured system states. We compare the effectiveness of discussed techniques with other methods in terms of precision, recall, F-measure and cumulative cost. The two methods demonstrate significantly improved forecasting performance compared to alternative approaches such as linear ARMA models. We outline some other methods as well.

Finally, we present a plethora of preventive measures that can be applied once it is established that a failure appears to be imminent.

By using the presented prediction and prevention techniques the system availability may be improved by an order of magnitude.

Miroslaw Malek –Brief Biography

- Miroslaw Malek received the M.Sc. degree in Electrical Engineering in 1970 and the Ph.D. degree in Computer Science in 1975, both from the Technical University of Wroclaw, Poland. He is professor and holder of the Chair in Computer Architecture and Communication at Humboldt University in Berlin since 1994. In 1977, he was a visiting scholar at the Department of Systems Design at the University of Waterloo, Waterloo, Ontario, Canada, then Assistant, Associate and Full Professor at the University of Texas at Austin where he was also a holder of the Bettie Margaret Smith and the Southwestern Bell Professor in Engineering, Malek's research interests focus on dependability, failure prediction, composability and mobility mainly in distributed systems but also in parallel architectures, real-time systems, and interconnection networks. He has participated in two pioneering parallel computer projects, contributed to the theory and practice of parallel network design, developed the comparison-based method for system diagnosis, codeveloped comprehensive WSI and networks testing techniques, proposed the consensus-based framework for responsive (fault-tolerant, real-time) computer systems design and failure prediction methods and has made numerous other contributions, reflected in over 200 publications including five books.

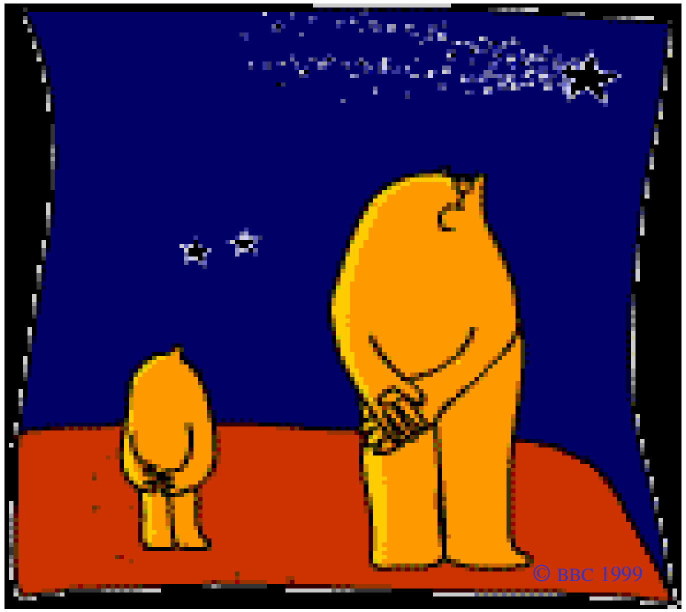
- He has organized, chaired and been a program committee member of numerous IEEE and ACM international conferences and workshops. Among others, he was Program and General Chairman of the Real-Time Systems Symposium in 1984 and 1985, respectively and in 1994 General Chairman of the 24th Fault-Tolerant Computing Symposium, Program Co-chairman of the 22nd Symposium on Reliable Distributed Computing in 2003, Program Chairman and General Chairman of the International Service Availability Symposium in 2004 and 2005, respectively. He serves or served on the editorial boards of various journals, among them the Journal of Parallel and Distributed Computing as well as Journal of Internet Engineering and Real-Time Systems journal. Of 25 Ph.D.'s under his supervision, 10 of them are professors world wide.
- Malek was a Visiting Scientist at Bell Labs in Murray Hill and at IBM's T. J. Watson Research Center, Yorktown Heights, NY. He held the IBM Chair at Keio University in Japan in 1992. He was also a Visiting Professor at Stanford University (1997/1998), New York University (2001) and the Italian National Research Center and Pisa University (2002), Chinese University of Hong Kong (2005) and Universita di Roma "La Sapienza"(2005/6).

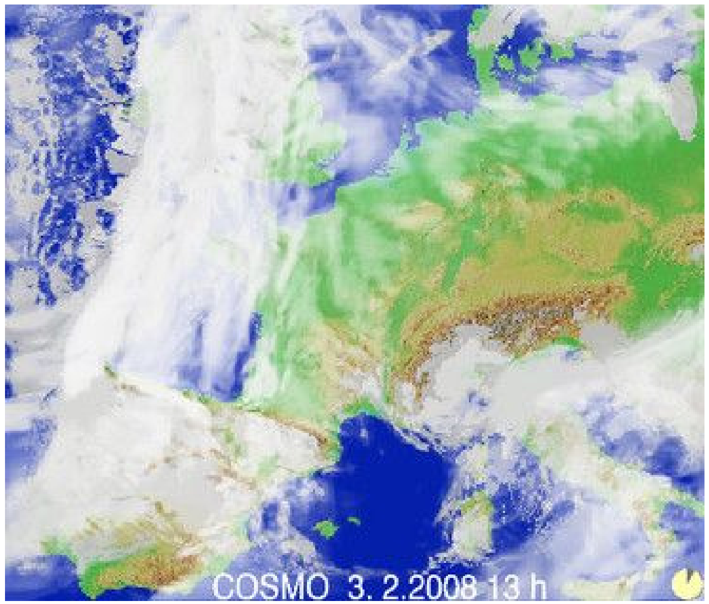
Contents

- Predicting the Future
- Background, Preliminaries and Motivation
- Failure Prediction Taxonomy
- Predictive Algorithms, Methods and Technologies
  - Error Logfiles and Variable Selection
  - Training, Validation and Runtime
  - Algorithms and Methods (RB, DFT, EM, UBF, HSMM)
- Metrics and Evaluation
- Case study, Real System, Real Data
- Summary: Trends, Open Issues and Best Practice

Predicting the Future in Daily Life

- 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 scientists/engineers





... and in Computer Science

- Typical predictions in computing and communication include: branch prediction, scheduling, memory management (e.g., Least Recently Used), performance prediction, reliability modeling, traffic modeling, congestion prediction, etc.

- In this tutorial we focus on:

  failure prediction of computer systems

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
- T. J. Watson (cofounder of IBM) stated that there will be a market for about five computers per year

Further Predictions and Reality

- *"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
- Several other luminous scientists, industrialists and thinkers made incorrect predictions about trains, ships, planes, radio, VLSI,…
- Notable exception: Moore’s Law

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

- **Factor 1:** People
  - 4-Nine-System (A> 0.9999): Downtime approx. 52 min. per year
  - 5-Nine-System: Downtime approx. 5 min per year
  - Human intervention is too time consuming
  - *Automation, e.g., self-repair = self-healing*

- **Factor 2:** Cost:
  - Further availability improvement is expensive
  - Further availability improvement boosts complexity

Motivation

- **Factor 3:** Ever-growing complexity
  - Networks and systems are becoming more complex whether we like it or not
  - The pace of upgrades and changes increases
  - Mobility, virtual and physical dynamics
  - Speed, storage, and bandwidth are constantly on the rise
  - New media and technology

- **Potential Solution:**

- Proactive Failure Handling (failure prediction followed by preventive measures)
- Obviously, in addition to classical redundancy in space / time methods

Further Reasons

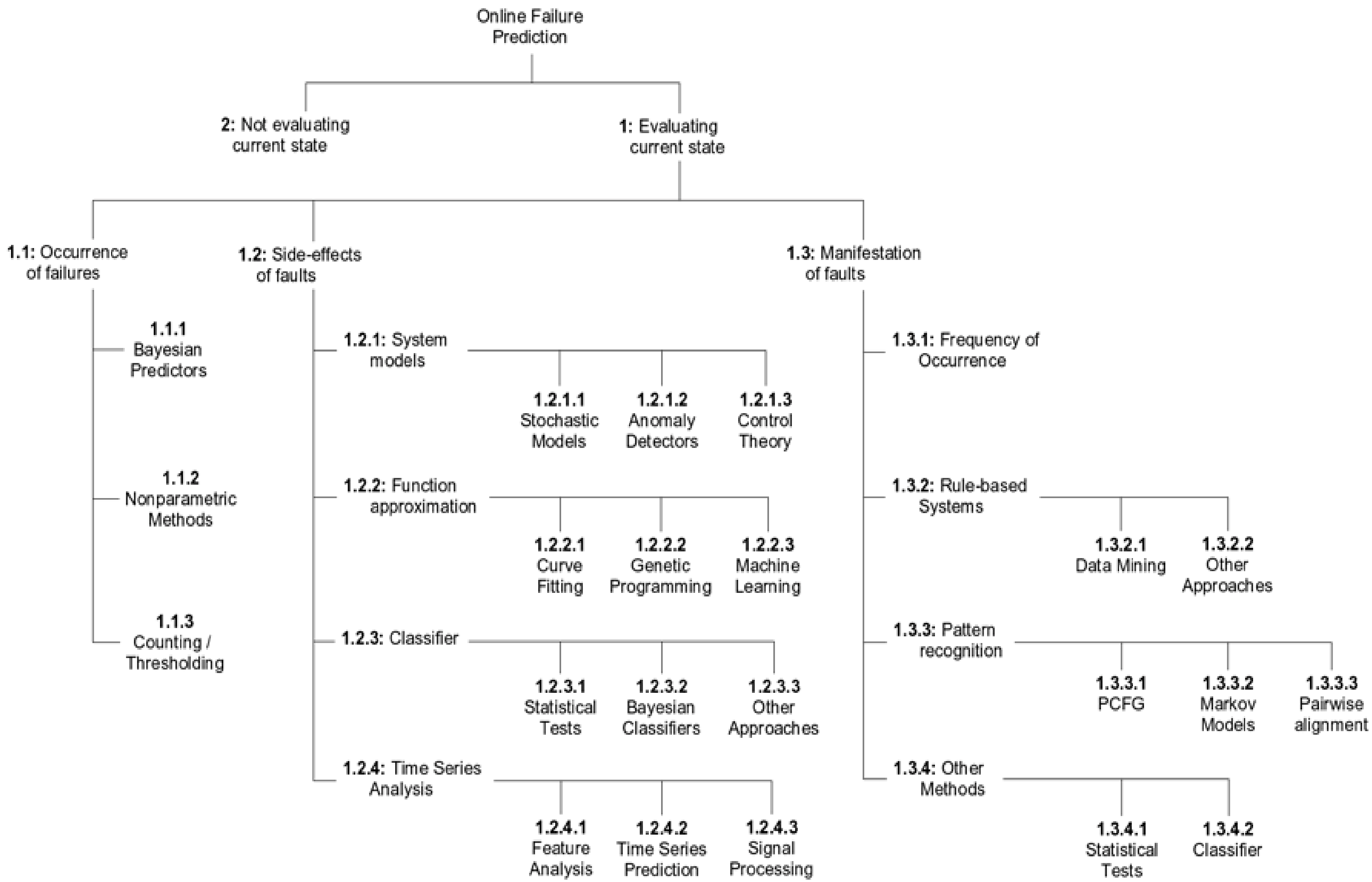
- Knowing the near term future usually helps to prevent a potential disaster or limit the damage
- Classical reliability theory has been mainly used for long term or average behavior predictions and comparative analysis
- Classical reliability theory may help but is not very good for short term prediction due to dynamics, mobility, systems/networks complexity, changing execution environments, upgrades, online repair, etc.

Solutions:

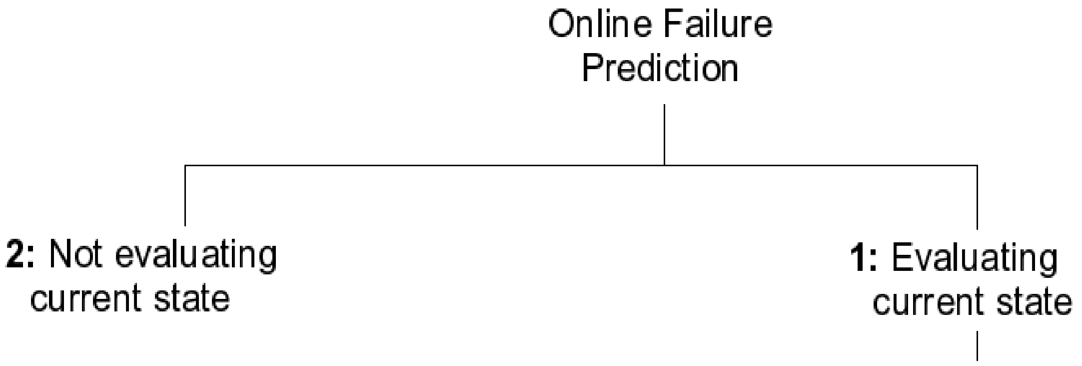
Runtime monitoring and failure prediction

Dynamic models capable of reconfiguration at runtime

Prediction Methods Taxonomy



Taxonomy: 1st Level



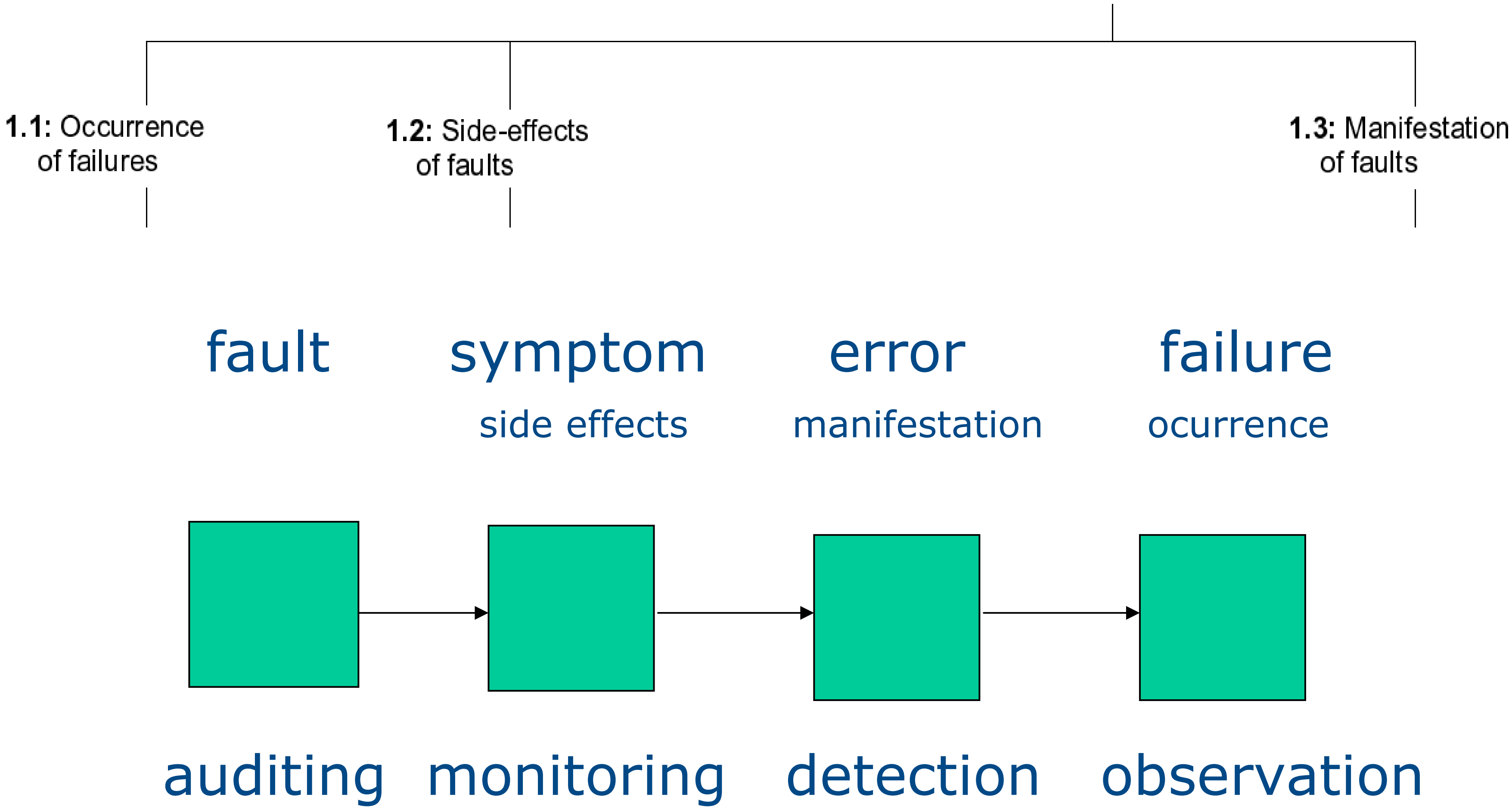
- Is the current system state evaluated?

- YES: monitoring-based methods (Branch 1)
- NO: methods based on, e.g., system architecture or lifetime probability distributions (Branch 2)

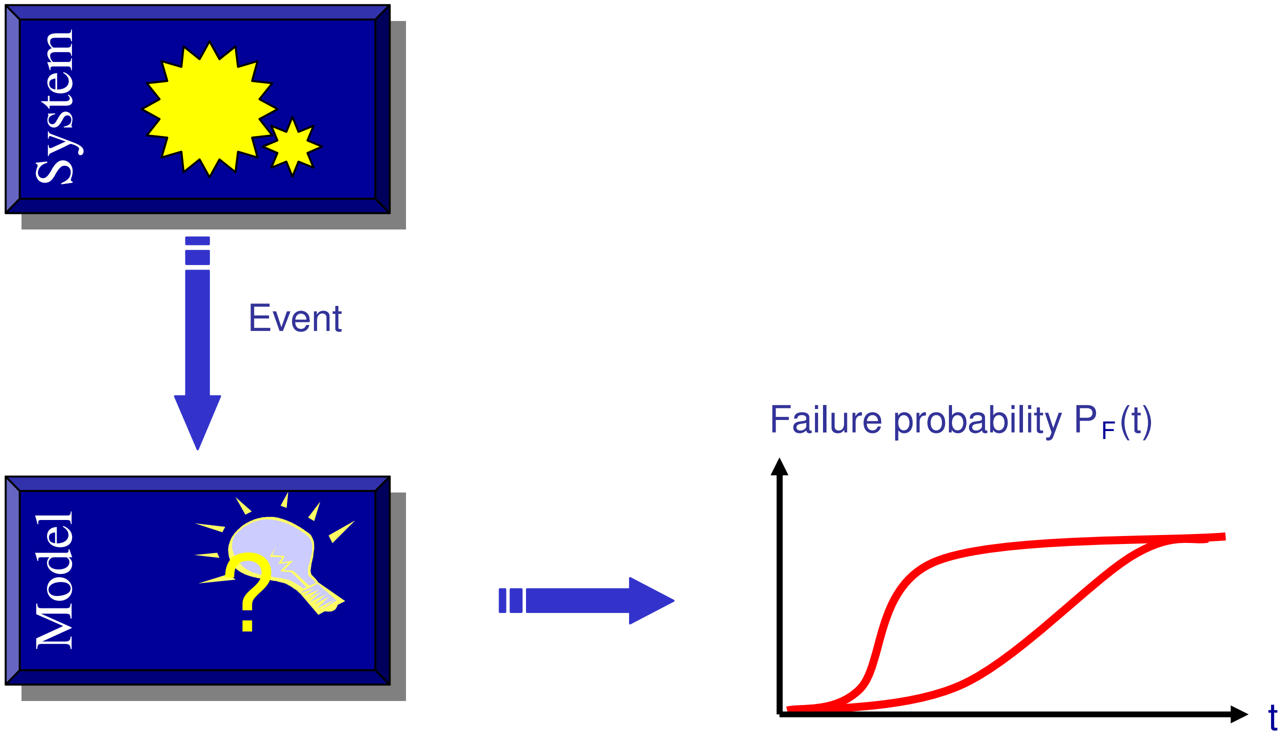
What is a failure?

- Fault
  - incorrect state of hardware or software
- Symptom
  - observed out-of-norm parameter behavior
- Error
  - manifestation of a fault observed by a fault detector
- Failure
  - occurs when the delivered service deviates from the specified service: failures are caused by errors

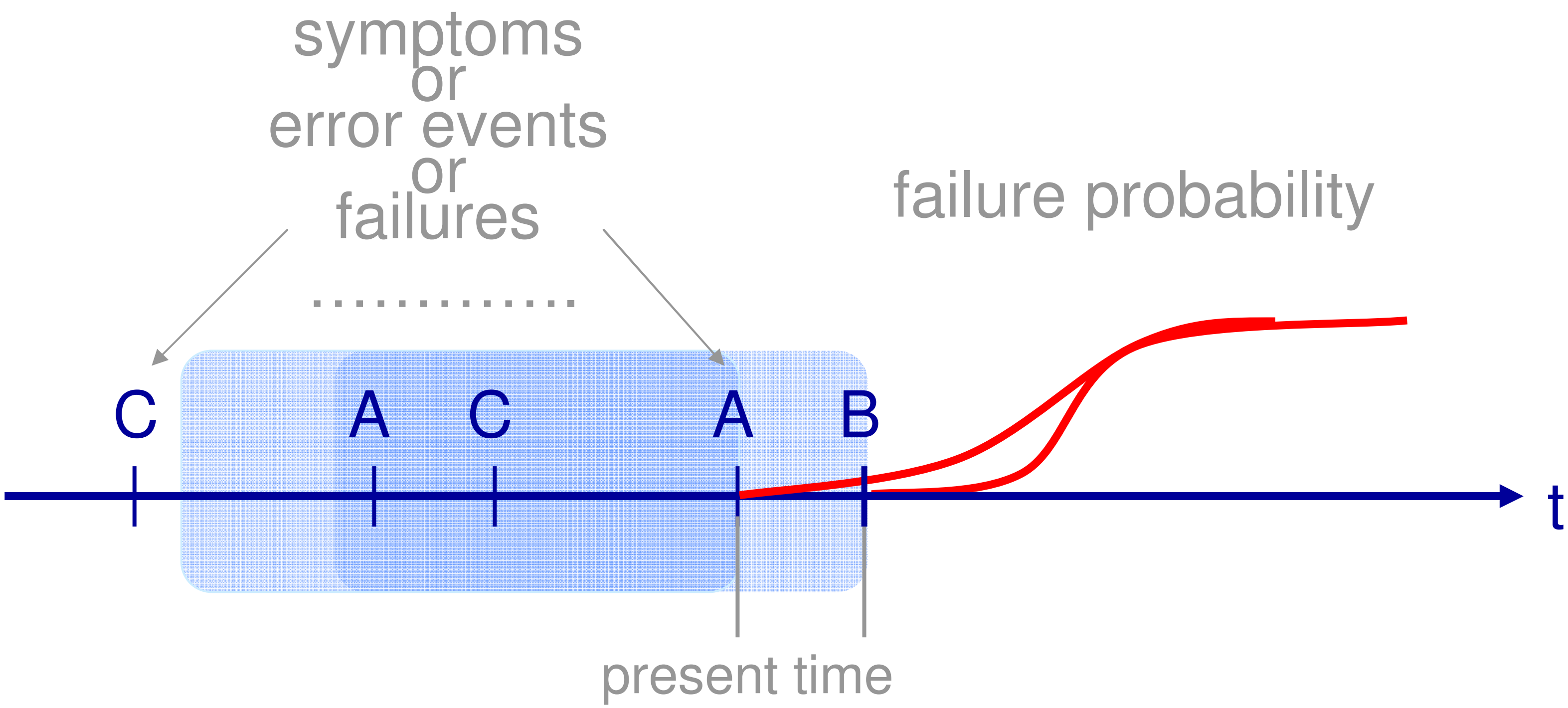
Taxonomy: 2nd Level



Problem Statement

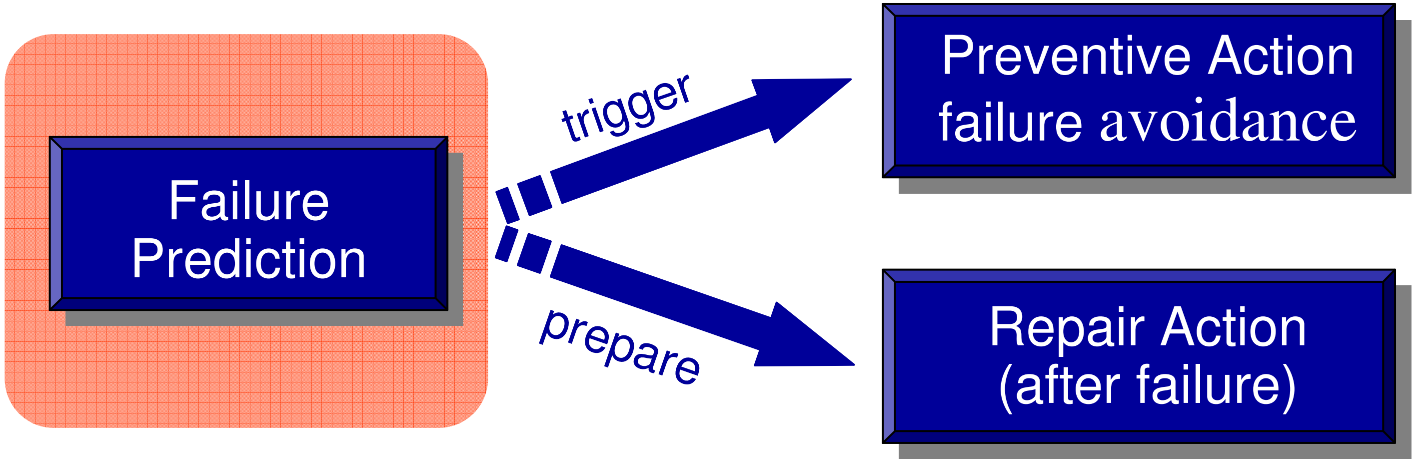


Problem Statement (refined)



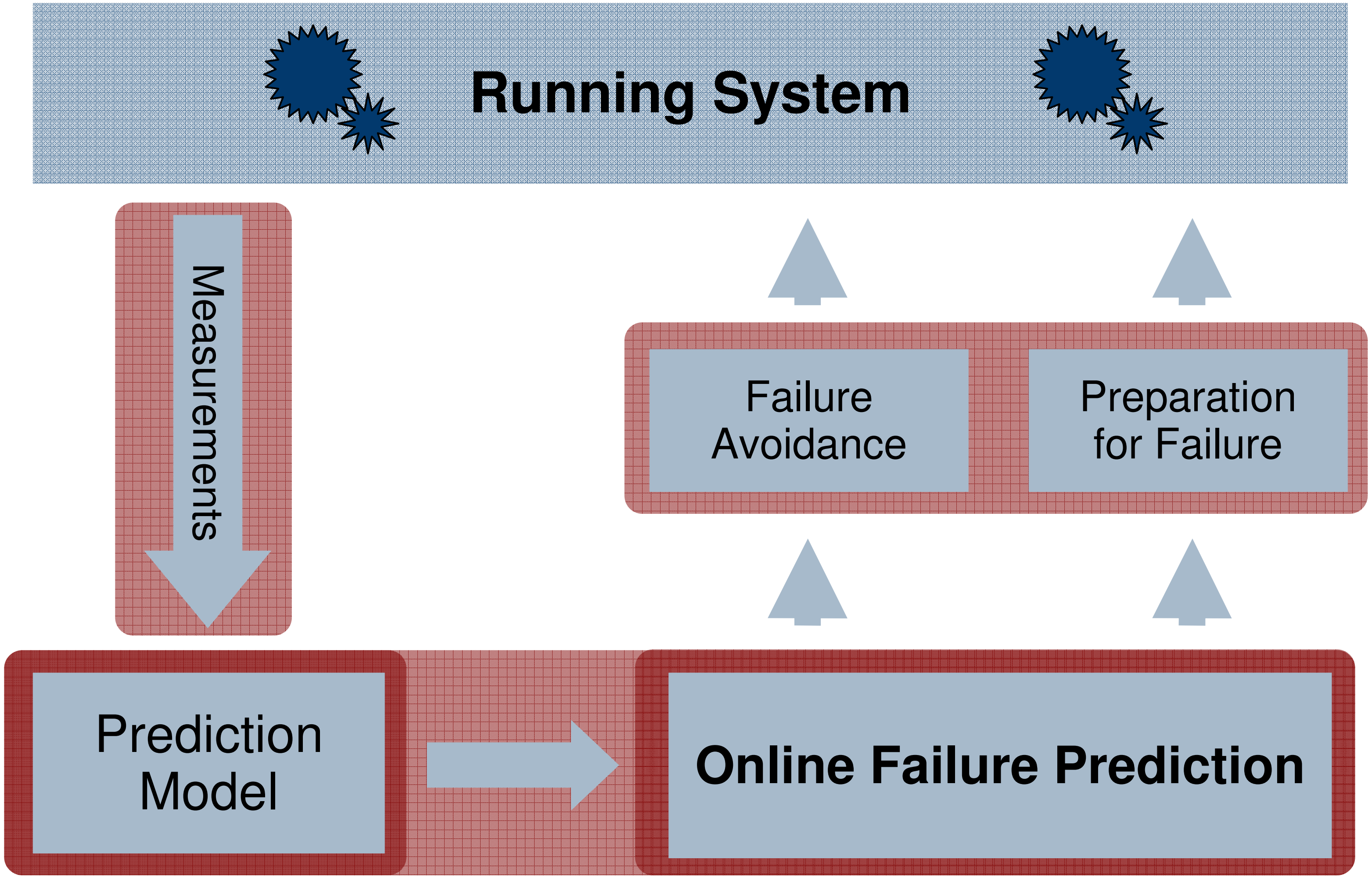
- Compute failure probability

Proactive Fault Management

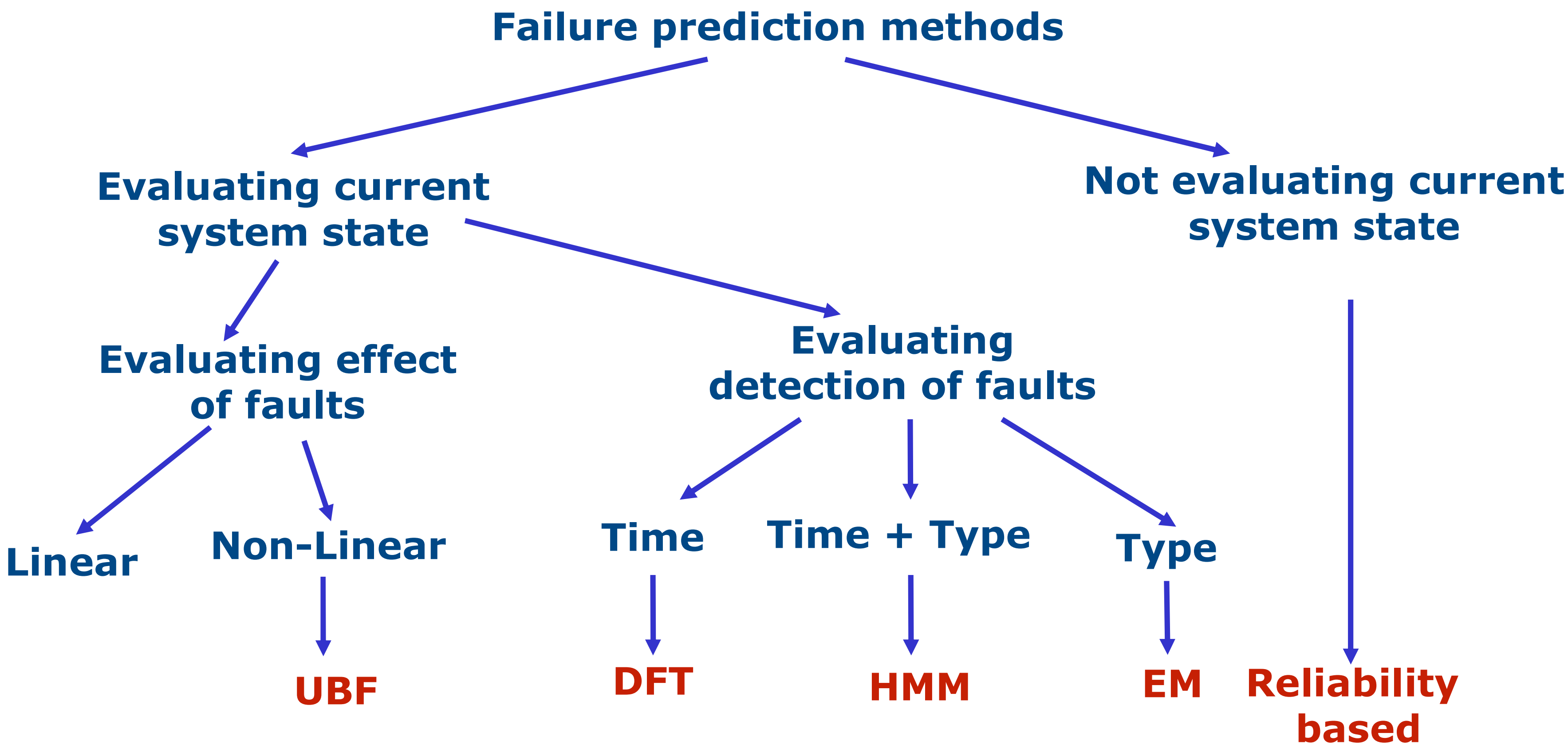


- Today‘s tutorial focuses on failure prediction

Proactive Fault Management



Failure Prediction Methods (in this tutorial)



UBF – Universal Basis Functions (Humboldt)

HSMM – Hidden Semi-Markov Chains

DFT – Dispersion Function Technique (CMU)

EM - Evenset Method (IBM)

Reliability-based Prediction

- Failure probability is directly related to reliability:

$$F(t) = P[T \leq t] = 1 - R(t)$$

- We use a standard model assuming a Poisson failure process:

$$F(t) = 1 - e^{-\lambda t} \quad \lambda = \frac{1}{MTTF}$$

where MTTF is estimated from the data

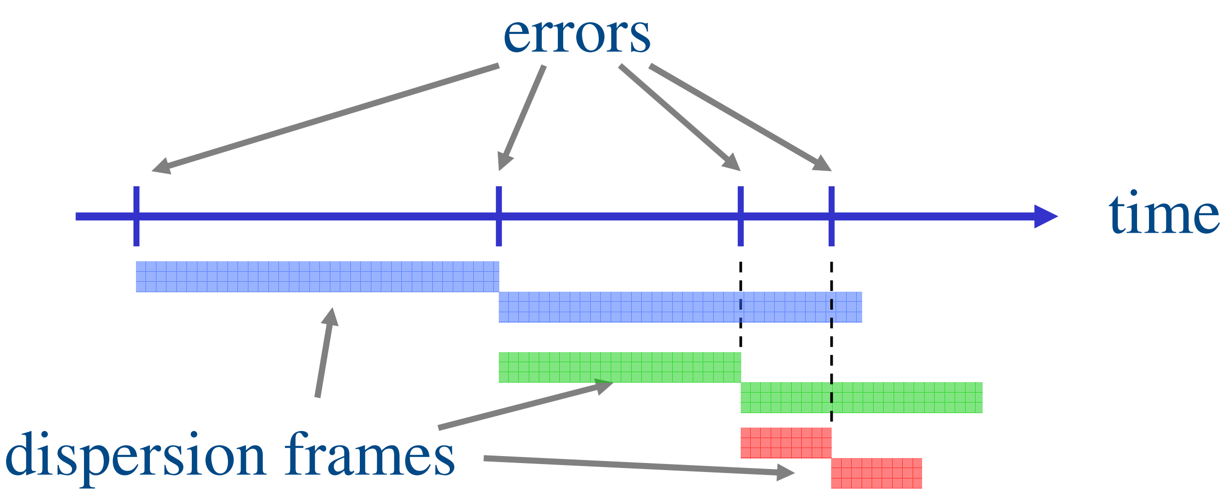
- Time to next failure: Median of the distribution

$$T = \frac{1}{\lambda} \ln(2)$$

- After each failure that occurs in the test data set, the timer is reset and the next failure is predicted at time T in the future.

Dispersion-Frame Technique (DFT)

- Classic technique for error-log analysis (Lin & Siewiorek, 1990)
- Evaluates the time of error occurrence



- Applies a set of heuristic rules evaluating the number of errors within successive dispersion frames

Eventset method (Vilalta et al)

- This approach is inspired by data-mining techniques trying to identify sets of events that are indicative of the occurrence of failures
- The technique is based on sets of events preceding a target event (events correspond to errors and target events to failures)
- The goal of the method is to set up a rule-based failure prediction system containing a database of indicative eventsets

Eventset method (Vilalta et al)

- Each time an error occurs, the current set of events Z is formed from all errors that have occurred until t (present time)

- The database DB of indicative eventsets is then checked whether Z is a subset of any D in DB. If so, a failure warning is raised

- The database of indicative eventsets is algorithmically built by procedures known from data mining

- Going through the training dataset, two initial databases are formed: a failure database containing all eventsets preceding failures and a non-failure database with eventsets that occurred between the failures (please note that the database also contains all subsets of eventsets that occurred)

Eventset method (Vilalta et al)

- In the training data, which has – at least theoretically – cardinality of the power set, minimum support thresholding together with branch and bound techniques are used to limit the cardinality.
- The goal is to reduce the database in several steps such that only most indicative eventsets stay in it.
- To achieve it, all eventsets that have support less than a minimum threshold are filtered out.

Real World – IBM Blue Gene Series

Basic idea:

- The use of event logs containing reliability, availability and serviceability data from IBM’s systems to predict memory, network and application I/O failures
- Three prediction algorithms based on failure occurrence before non-fatal events and the spatial skewness of failure occurrence have been developed.
- A three-step filtering algorithm is applied to extract and categorize failure events as well as to perform temporal and spatial compression.
- Failure prediction is based upon temporal and spatial failure characteristics and correlations between fatal and non-fatal events.

Results, application areas, experiments

- The prediction methods were applied to BlueGene/L, a supercomputer with 128K processors. 37% of all network failures and 48% of all application I/O failures were predicted by using temporal compression.

- If network and application I/O failure events were merged, 370 out of 687 failures (54%) were predicted.

- By monitoring the following five jobs after the occurrence of a non-fatal event, 82% of the total fatal failures can be predicted.

- Futhermore, if five jobs after observing a fatal failure are checked, 9.5% additional fatal failures can be caught.

Thesis

- Dependencies in software/hardware lead to error patterns
- Systems fail only under special conditions
- Special conditions reflect special patterns

➤ Failure prediction by recognition of special patterns

Sources of Data

Where do we get the special patterns from?

- 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
  - 20 - 4200 parameters can be monitored
  - Up to five can usually be measured and processed in real time for 1-5 minutes prediction

Three-Step Approach

1. **Data acquisition**
   – Logfiles
   – Variable selection
2. **Failure prediction**
   – White-box approach
   – Black box approach (Günther Hoffmann)
   – Hybrid (grey box) approach (Felix Salfner)
3. **Preventive Actions**

Data Acquisition: Variable Selection

- What are the right variables to use for modeling?
- There are about 20 to 4200 parameters (SAR - System Activity Reporter) and up to hundreds of fault classes (from log files) per node

  For n nodes: m = p x f x n variables

- If we analyze r of them at a time then

  we have m choose r problem

- 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

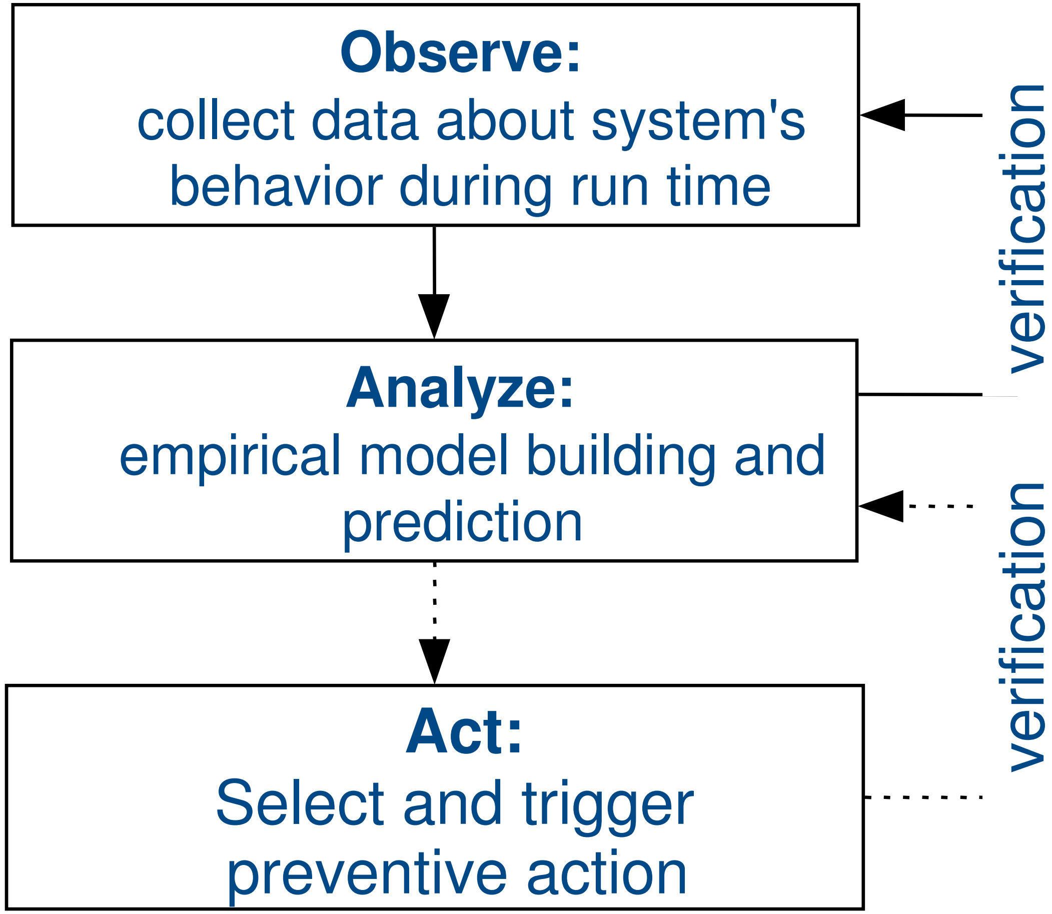
Variable Selection: Initial Results

Using wrapper approach and experts we have selected two variables (42 parameters, 195 fault classes, 2 nodes):

- *alloc* – the amount of memory (in bytes) that KMA (Kernel Memory Allocator) has allocated to OS
- *sema/s* – the number of semaphore operations per second

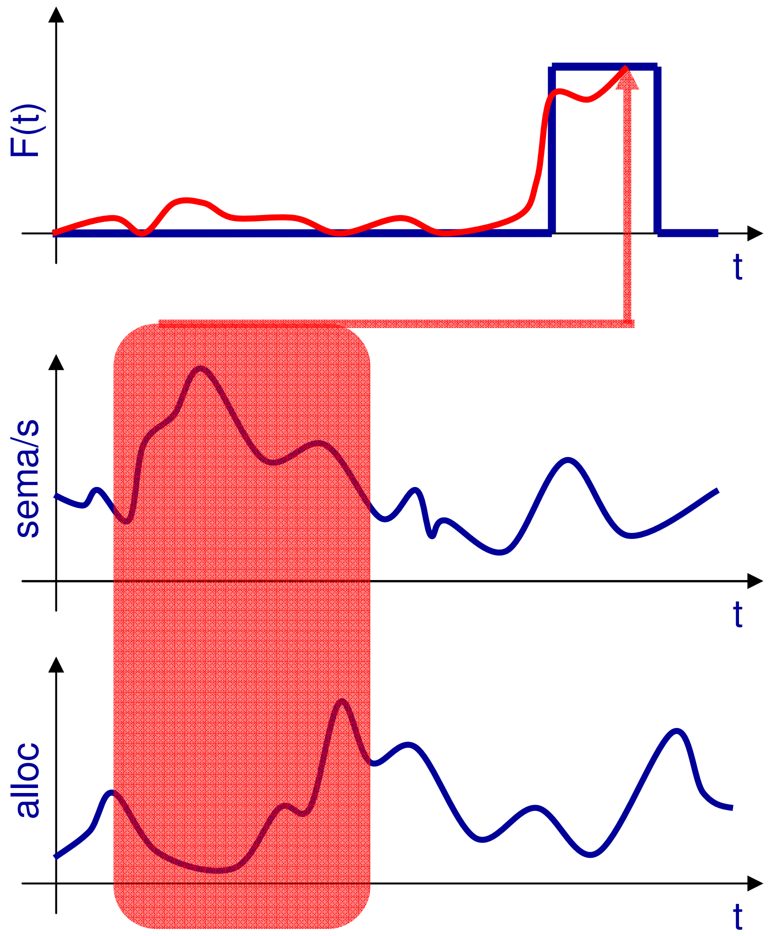
Surprise: Including log files did not improve prediction quality with Universal Basis Function!

Black Box - Empirical Modeling



Universal Basis Functions

- Tailored to equidistant measurements: e.g.,
  - Semaphore operations per second or minute
  - Allocated OS-kernel memory
- Function approximation: Failure probability as function of system variables
- Universal Basis Functions:
  - Linear combination of nonlinear kernel functions
    - o Gaussian
    - o Sigmoide functions, …



Universal Basis Functions (UBF)

Find underlying function by minimizing, e.g., mean-square-error

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

Linear combination of kernel functions G

e.g.

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i(\mathbf{x}) * G_i(\mathbf{x}, \mathbf{t}_i)$$

Replace fixed Gaussian by flexible domain specific kernel

$$G(\mathbf{r}, \boldsymbol{\omega}') = \boldsymbol{\omega}' \Phi_1(\mathbf{r}) + (1 - \boldsymbol{\omega}') \Phi_2(\mathbf{r})$$

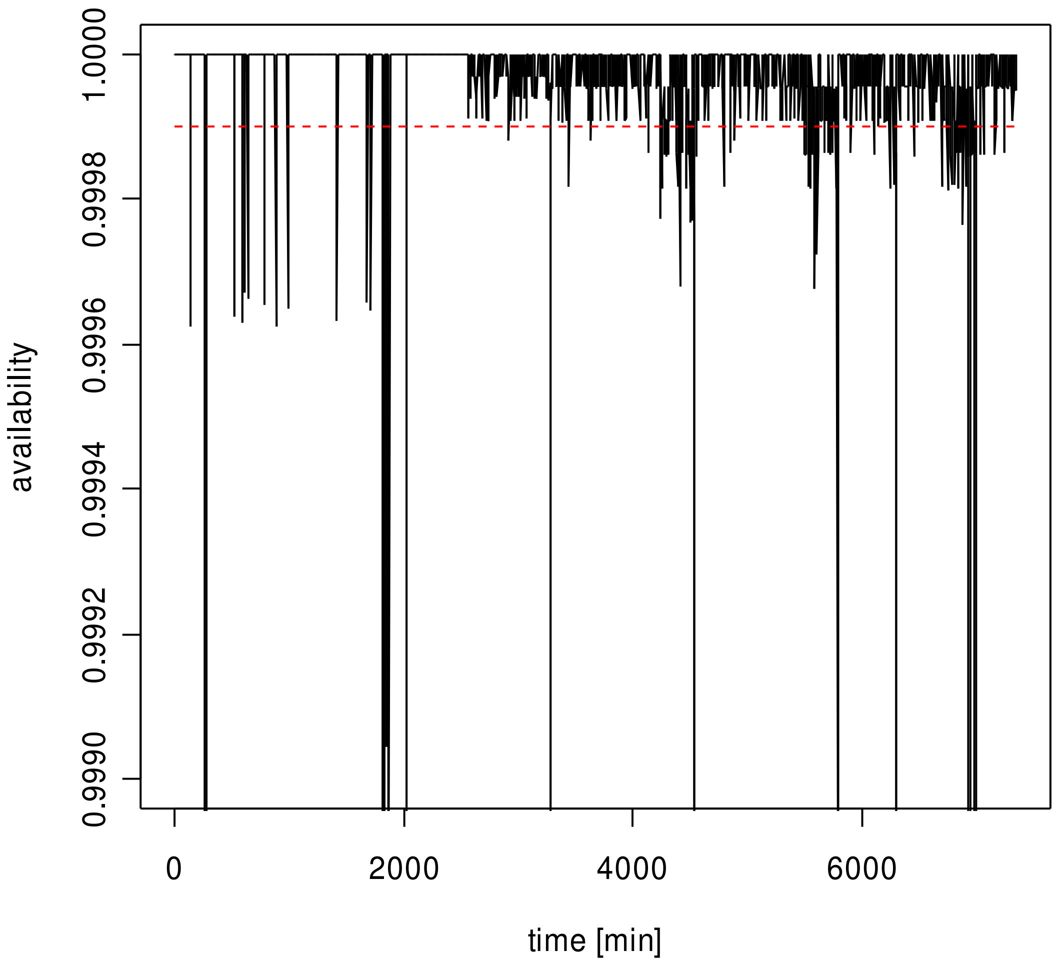
Case Study: Multiserver Support System

- Distributed and component based system
  - 1.5+ million lines of code
  - 2000+ classes and 200+ components
  - handles value added services in GSM/GPRS networks (e.g., billing, SMS, pre-paid services)
  - 400-10,000 service requests per minute
  - Two nodes (up to eight)

→ **Model availability and predict failures**

Case Study: Multiserver Support System

- Call Availability
  - $A_c(\Delta t) = 1 - (n_f / n_c)$
- Interval: Δt = 5 minutes
- Failure:
  - $A_c(\Delta t) < 0.9999$
- Reformulation as classification task
- Variables
  - *alloc*
  - *sema/s*

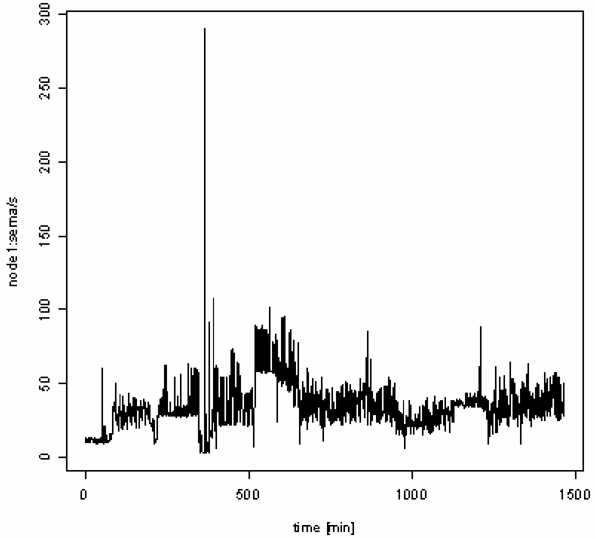


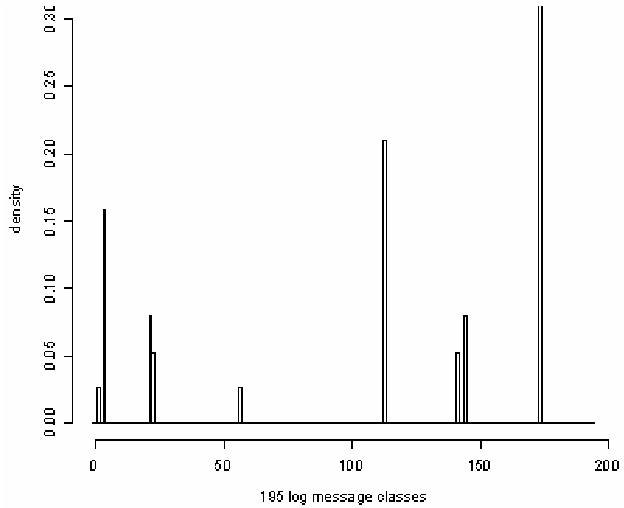
Case Study: Multiserver Support System

- Two types of data
  - SAR (system activity reporter)
    - o Time driven equidistant
    - o 96 variables
    - o Approx. 13 million observations
  - Log file data
    - o Event driven
    - o 390 variables
    - o Approx. 4 million observations

- 486 variables in total

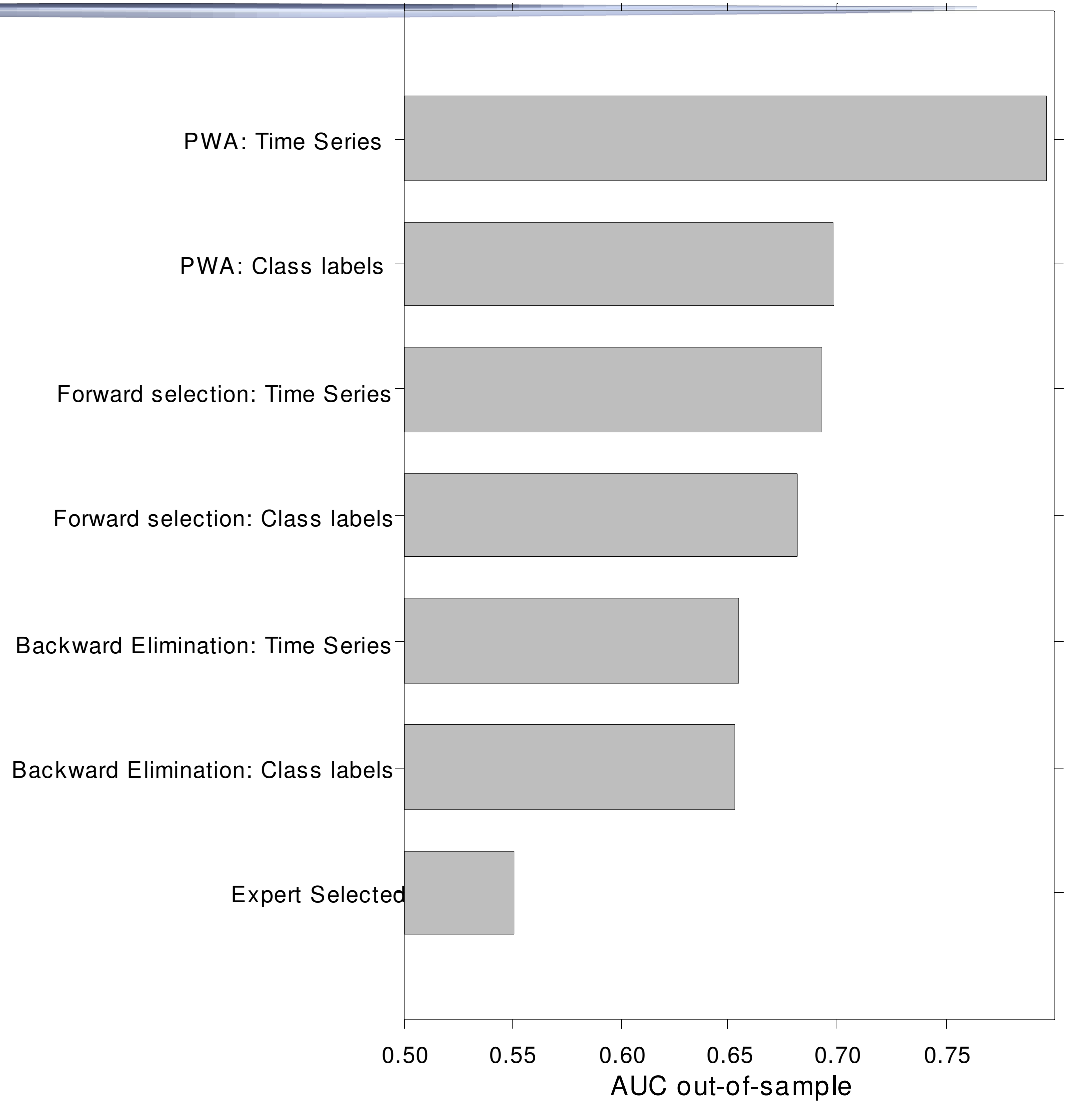
- Ten days observation period





Variable Selection

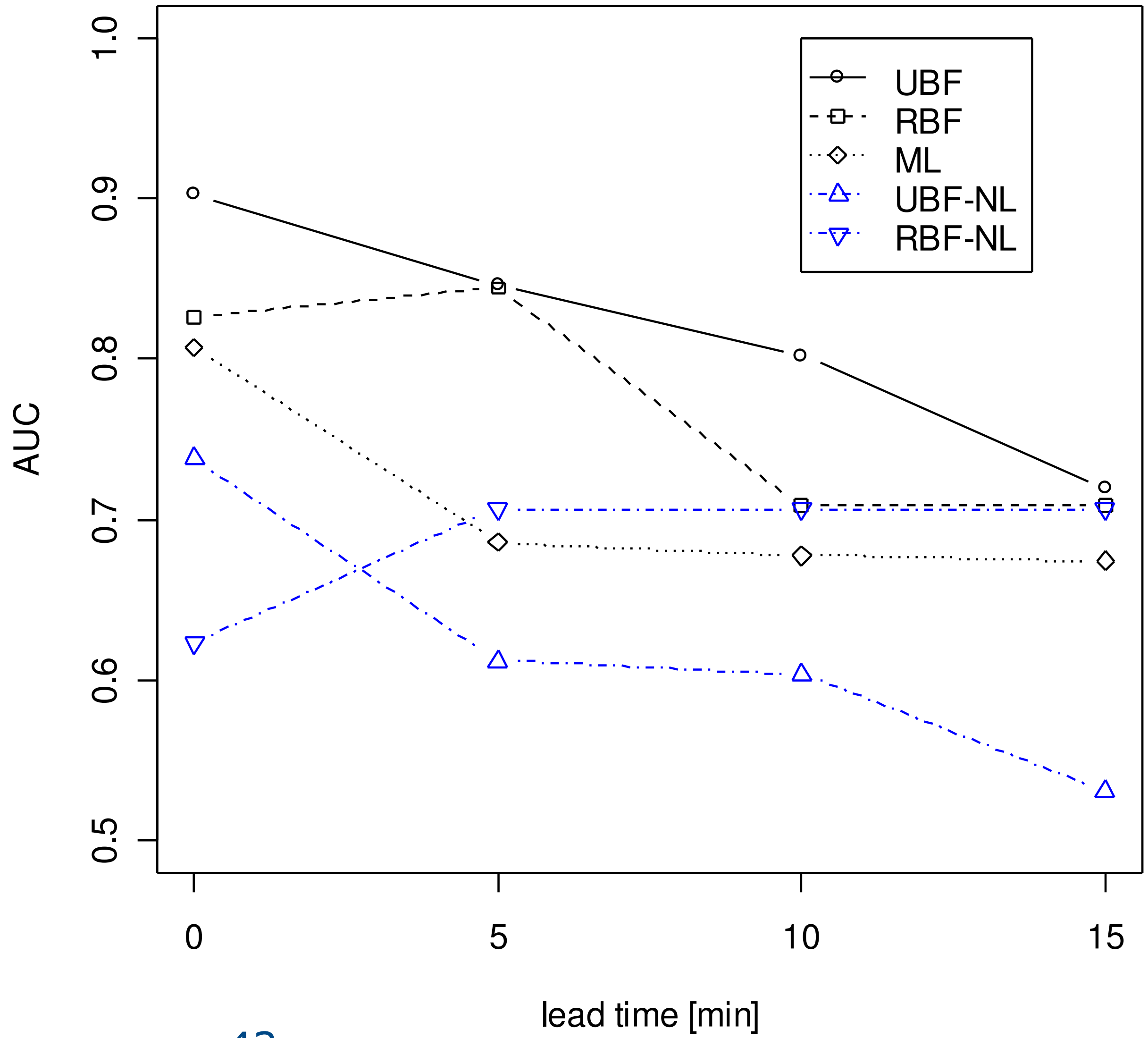
- Benchmarked four techniques
  - Forward selection
  - Backward elimination
  - Expert selected
  - PWA (Prob. Wrapper)
- Variables
  - *alloc*
  - *sema/s*
- PWA performs best on time series *and* class label data



Failure Recognition and Prediction

- Mean AUC values for 0,5,10,15 minutes into the future (lead time)

- Statistical significance established based on *t*-testing



Unique Aspects

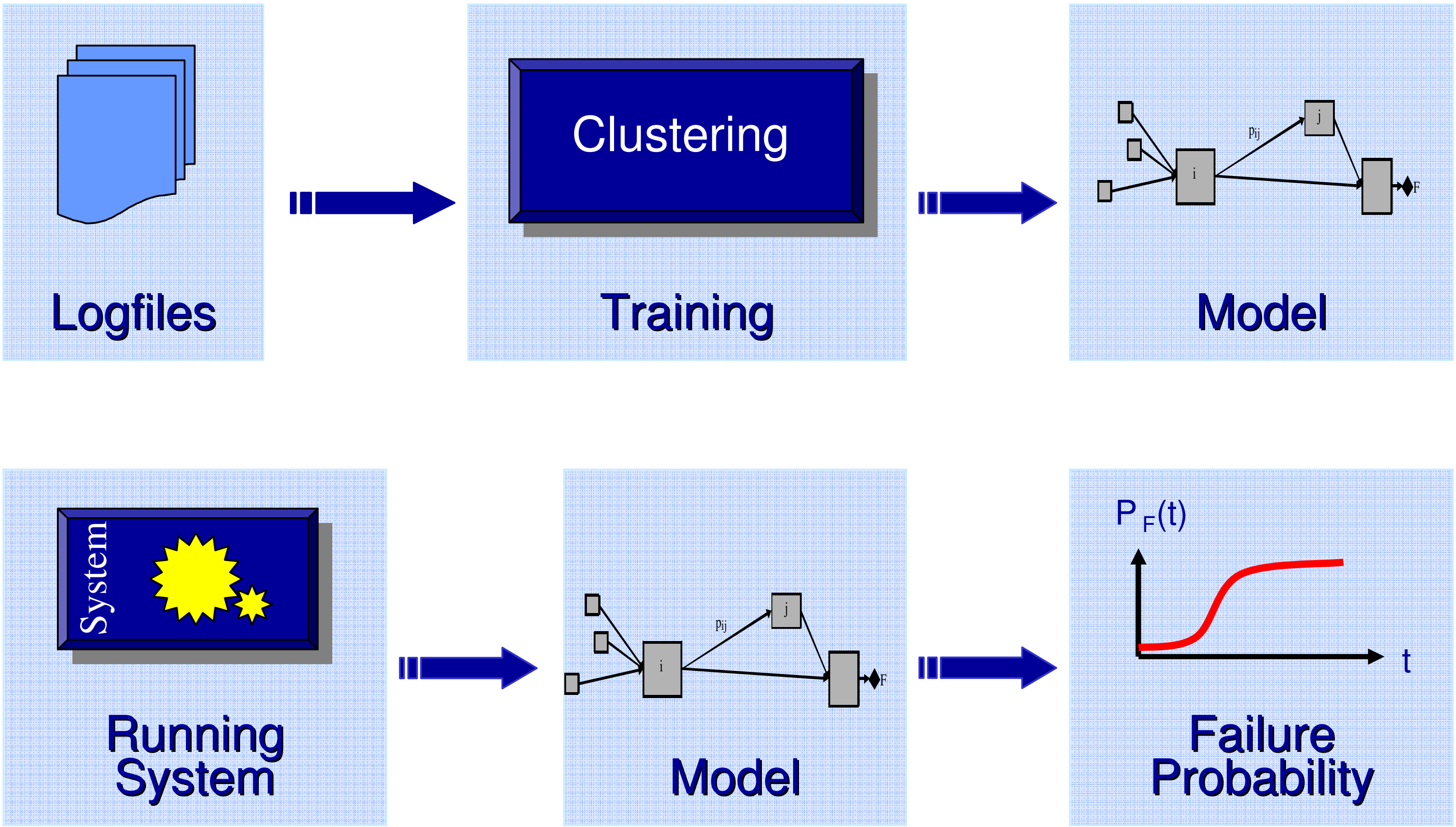
1. Nonlinear modeling approach: Universal Basis Functions (UBF’s)
   + Extended traditional Gaussian RBF modeling framework
   + UBF’s significantly outperform traditional modeling techniques
   + UBF’s retain the transparency and elegancy of linear equations
   + Black box modeling technique, thus nontrivial to interpret

2. Variable selection technique: Probabilistic Wrapper Approach (PWA)
   + Integrated and extended traditional wrapper and filter approaches
   + PWA significantly outperforms common selection techniques
   + PWA allows for transparently including expert domain knowledge

3. Applications
   - Component level:
     - resource forecasting
   - System level:
     - failure prediction,
     - feature detection,
     - sensitivity analysis and cost / benefit analysis
     - performance prediction

Machine Learning Approach using Hidden Semi-Markov Models (HSMM's)

- Since systems are too complex to identify patterns by hand
  ⮙ Machine learning approach:



Prediction of Failures

- **Equivalent to computation of first passage time into failure state.**

$$P_F(t) = \sum_i F_{iF}(t)\pi_i$$

where π_i is the probability of being in state i

- **Computation with standard technique: first step analysis**

$$F_{iF}(t) = G_{iF}(t) + \sum_{k \neq F} \int_0^t dG_{ik}(x) F_{kF}(t-x)$$

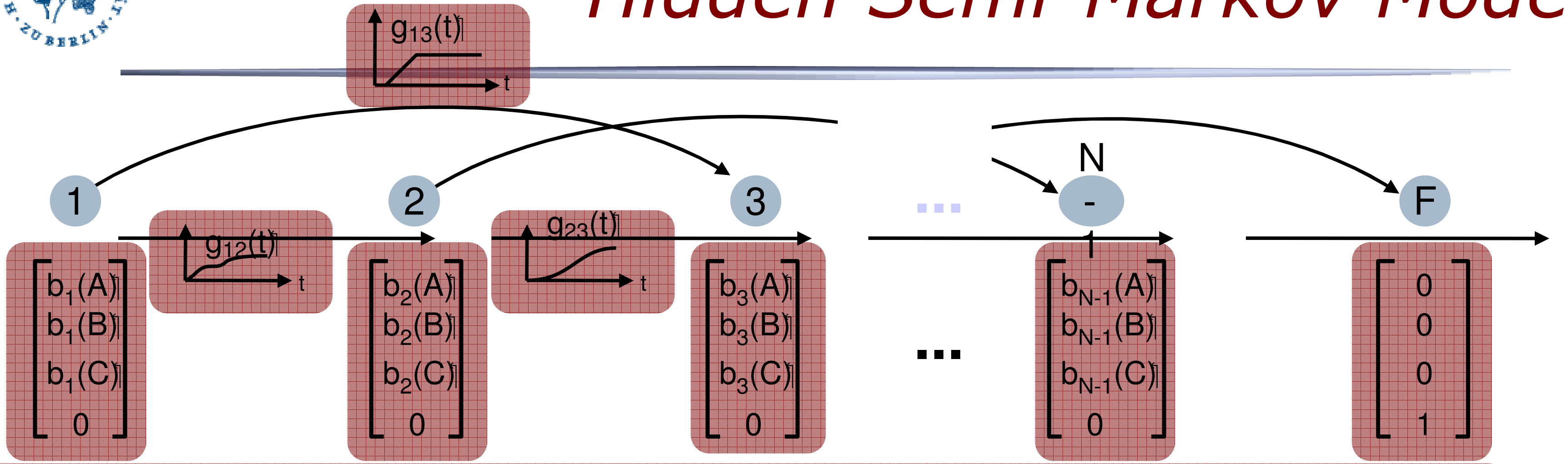
Involves convolution which can be computed analytically using Laplace transformation.

$F_{iF}(t)$ **is precomputed during training period**

Approach

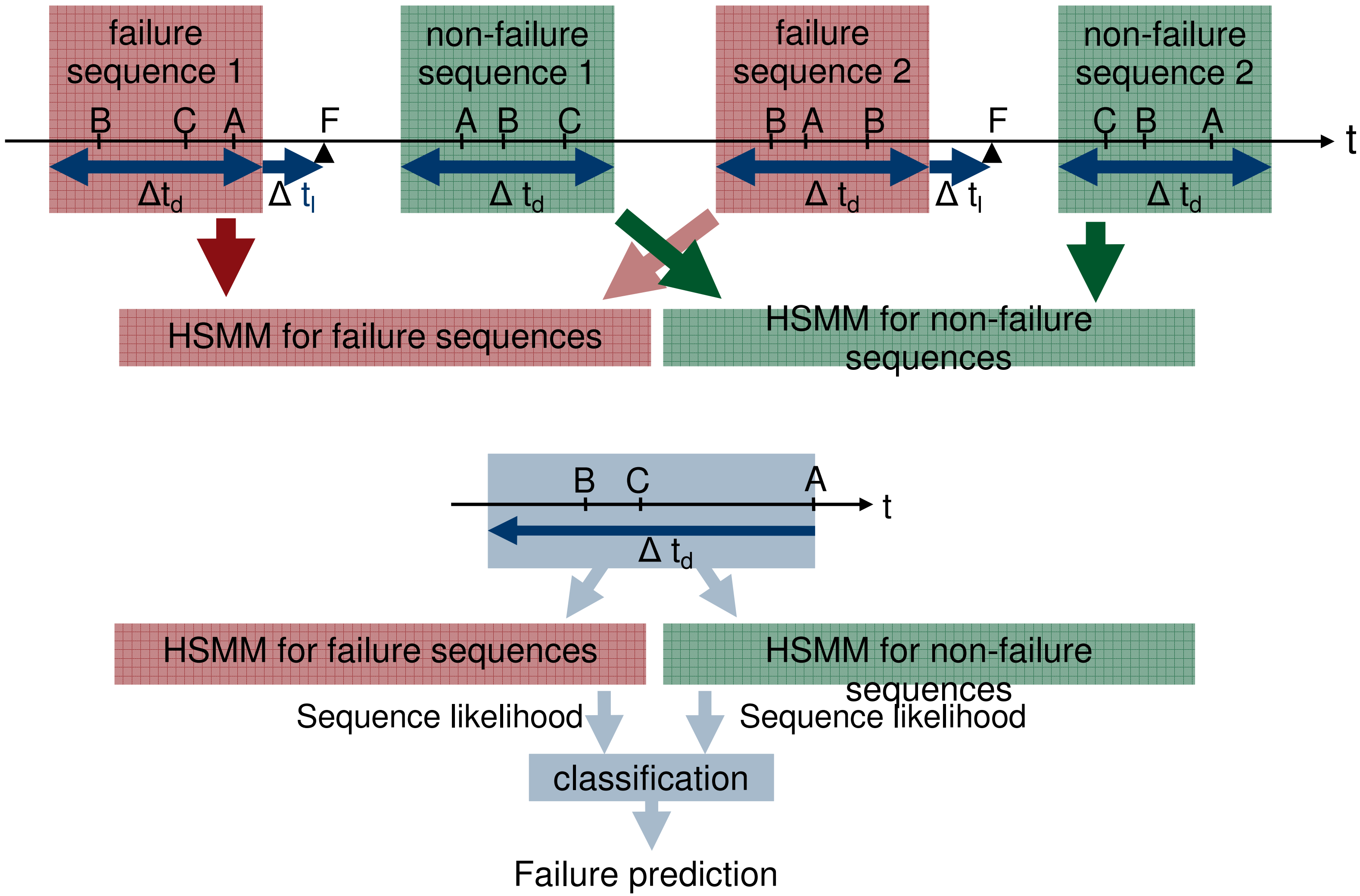
- Pattern recognition to identify symptomatic patterns - standard tool: **Hidden Markov Models**
- Identify symptomatic patterns
  - Algorithmically
  - From recorded training data
  - ✦ **Machine learning**
- Additional assumption:
  - **Time** between events is decisive (temporal sequence analysis)
  - Standard Hidden Markov Models need to be extended
  - ✦ Development of a **Hidden Semi-Markov Model** (HSMM)

Hidden Semi-Markov Model



- Discrete time Markov chain (DTMC)
  - States (1,..., N-1,F)
  - Transition probabilities
- Hidden Markov Model (HMM)
  - Each state can generate (error) symbols (A,B,C,F)
  - Discrete probability distribution of symbols per state $b_i(X)$
- Hidden Semi-Markov Model (HSMM)
  - Time-dependent transition probabilities $g_{ij}(t)$

Machine Learning: Two Steps Training and Prediction



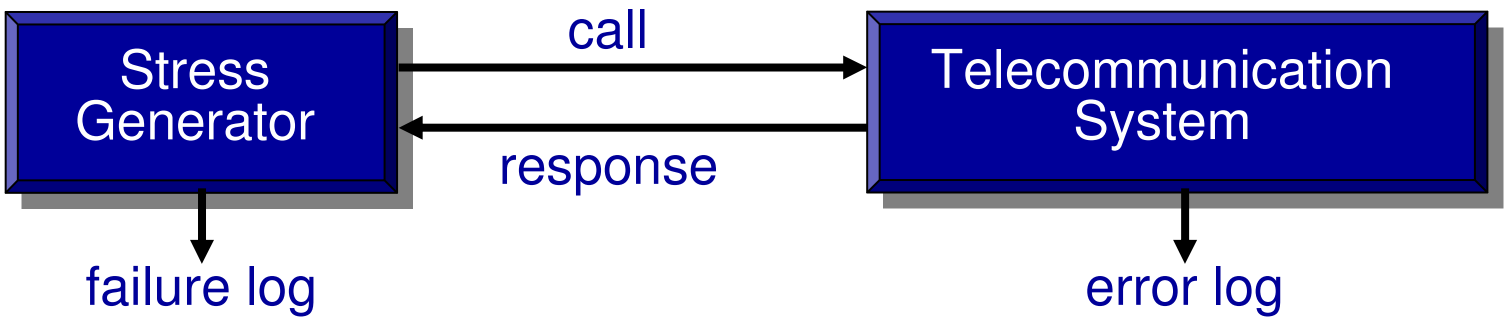
Training phase

- Goal:
  - Fit HSMM parameters to training sequences
- Objective function:
  - Maximizing sequence likelihood
    - Probability that HSMM can generate a given sequence
- Algorithm:
  - iterative improvement
  - modified Baum-Welch algorithm
- Result: Two HSMMs
  - One represents characteristics of failure sequences
  - One represents characteristics of non-failure sequences

Online Prediction

- Analyze errors that have occurred
- Compute sequence likelihoods P(S)
  - Similarity between observed sequence and training sequences
  - Compute for both HSMMs
- Classification:
  - Based on Bayes' decision theory
  - Analyzes the ratio of sequence likelihoods
  - Decision is based on a threshold

Experimental Setup



- Log data analysis of a commercial telecommunication system
- Failure definition:
  - 5 min average call availability drops below 0.9999
  - A call is considered failed if connection setup time exceeds 250 ms

Evaluation Metrics (Selection)

- Precision
  - Ratio of correct failure warnings to all failure warnings
- Recall
  - Ratio of correct failure warnings to all failures
- F-Measure
  - Harmonic mean of precision and recall
- False positive rate (FPR)
  - Fraction of false-positive warnings to all non-failures

Precision, Recall and F-measure

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

- Precision: fraction of correct alarms:

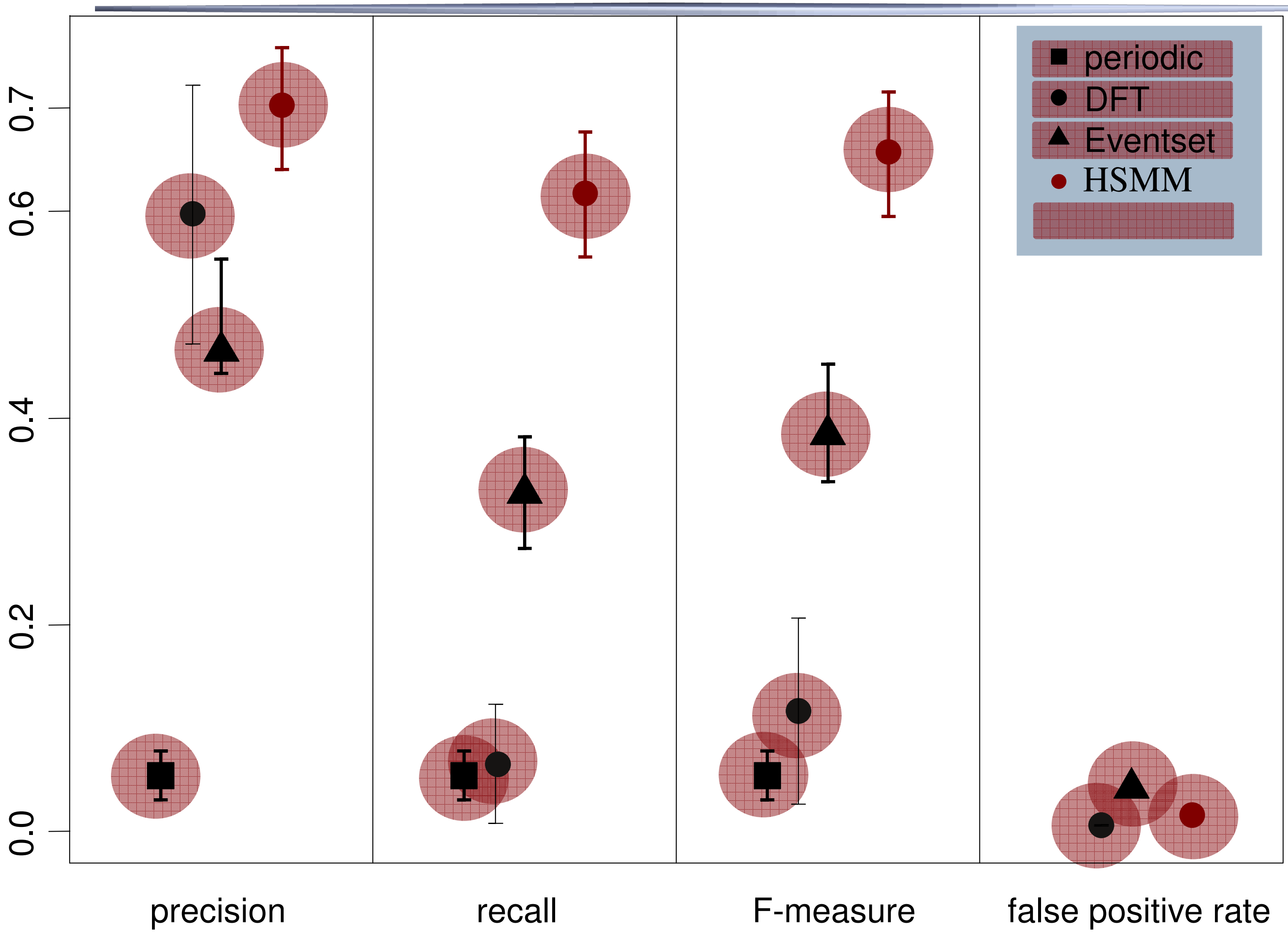
$$\text{precision} = \frac{\text{correct alarms}}{\text{sum of alarms}}$$

- Recall: fraction of predicted failures:

$$\text{recall} = \frac{\text{correct alarms}}{\text{sum of failures}}$$

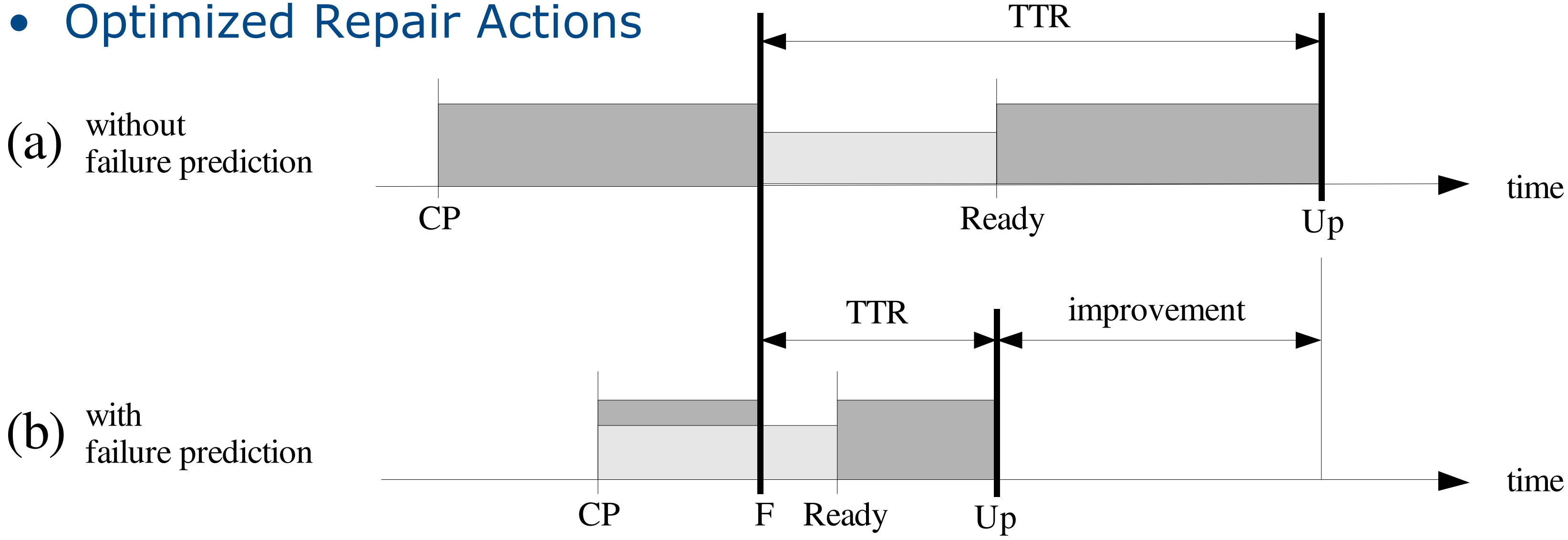
- $$\text{F-measure} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

Comparison of Techniques



Methods to Handle Failures

- Failure Prevention
  - Preventive restart, e.g., rejuvenation, process or job restart
  - Preventive failover, e.g., failure sensitive load balancing
  - State clean-up, e.g., garbage collection
  - Adaptive checkpointing
  - Relaxation, e.g., connection redirection (load lowering)
  - Reboot
- Optimized Repair Actions



Cumulative Cost

- Assign cost to:
  - **Correct alarms:**

    A failure was predicted and could have been prevented

    (in our example 10 cost units)
  - **False alarms:**

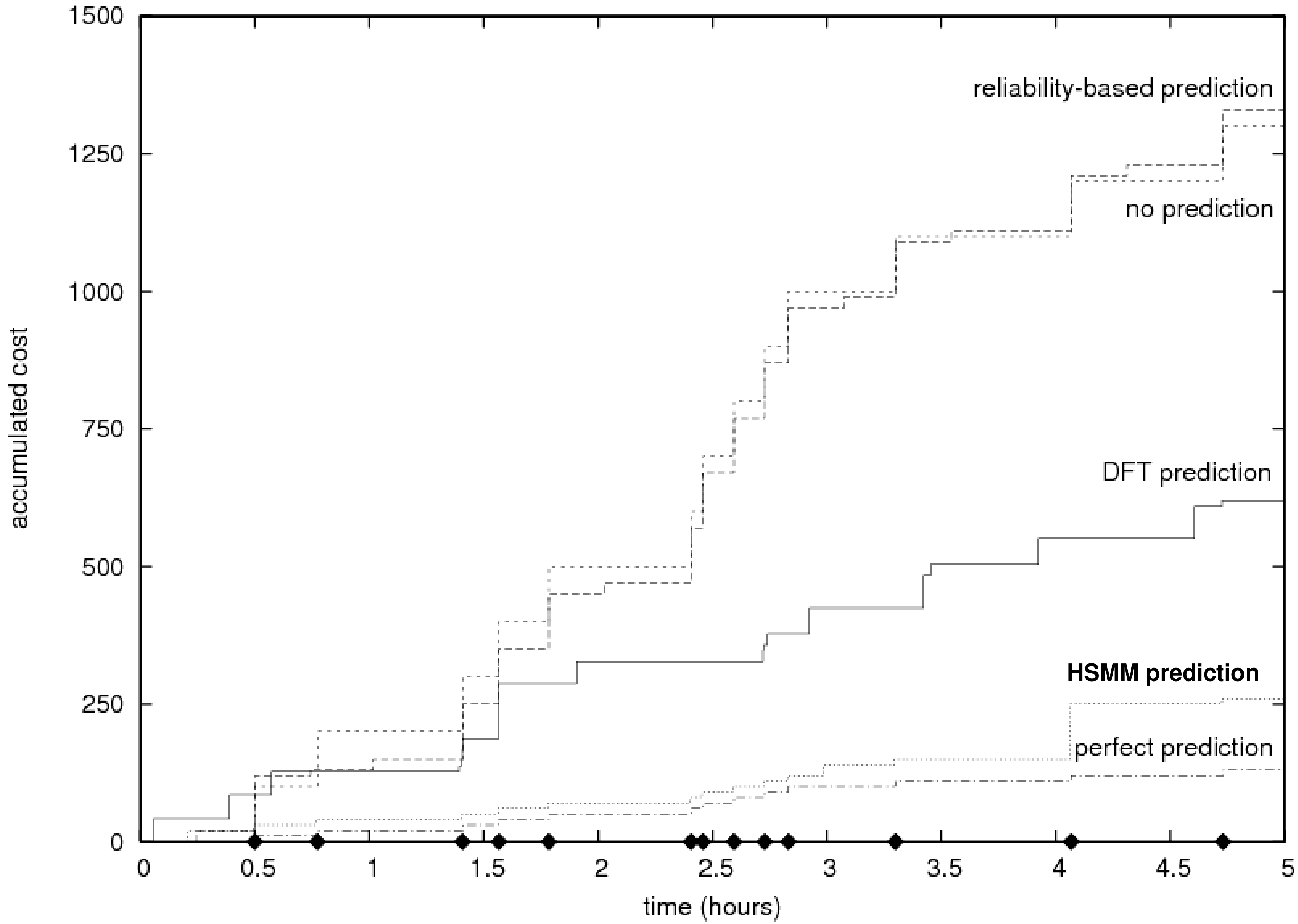
    Preventive actions are performed in vain

    (in our example 20 cost units)
  - **Missing alarms:**

    Failure is not predicted: worst case

    (in our example 100 cost units)
- Based on log data, we are able to compute cumulative runtime cost

Resulting Cumulative Cost



The Gains Formula

$$A_{PFH} = A_{orig} + k * \frac{MTTR}{MTTF + MTTR}$$

- k depends on:
  - Precision (# correct alarms/all alarms)
  - Recall (# correct alarms/ all failures)
  - Probability that a failure can be avoided
  - Probability that failures occur due to actions
  - Mean improvement in MTTR
- Range of valid values for k:

$$k \in \left[-\frac{MTTF}{MTTR}, 1\right]$$

An Example

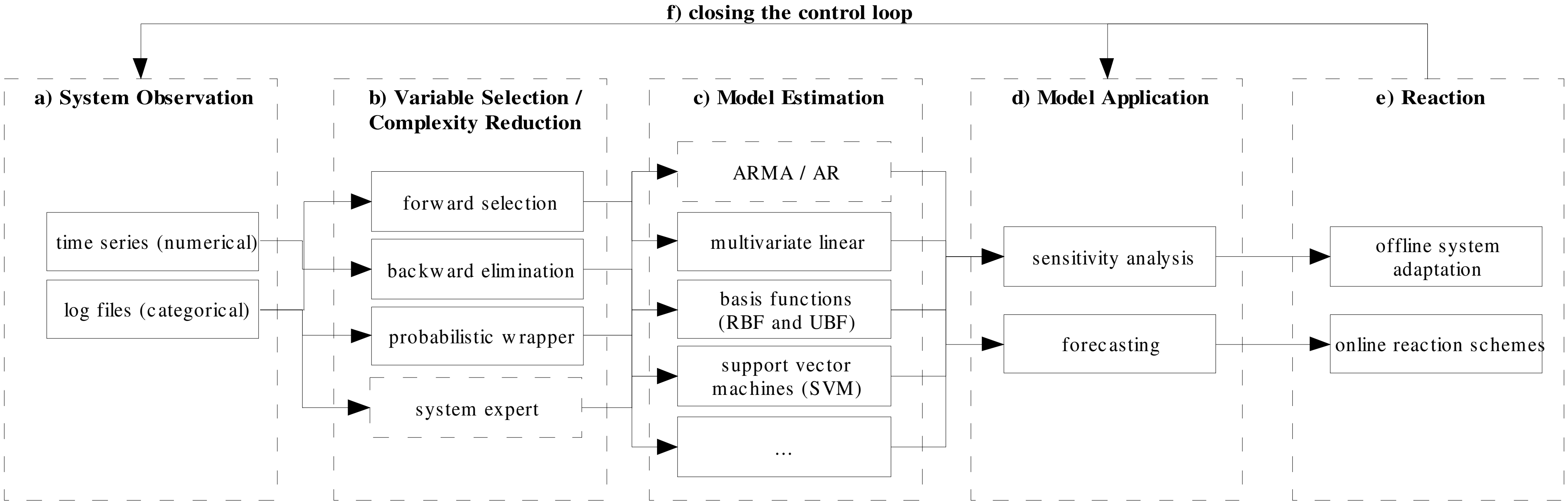
- Assume 4-Nines System
- Correctness of Failure Prediction:
  - Alarms are correct in 85% of all cases (precision)
  - 90% of all failures are predicted (recall)
- Performance of the methods:
  - Correct alarms: Prevention probability = 95% (P_p)
  - Correct alarms: Improved repair time = 0.5 * MTTR
  - False alarms: Probability of extra failures = 15% (P_e)
- Availability of original system:
  - **A_{orig} = 0.9999**
- Availability with proactive failure handling:
  - **A_{pfh} = 0.9999903**

➤ **Order of magnitude improvement!**

Trends and Observations

- Variables should be selected based on the actual model quality rather than based on linear correlation
- 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

Summary - Best-Practice Guide

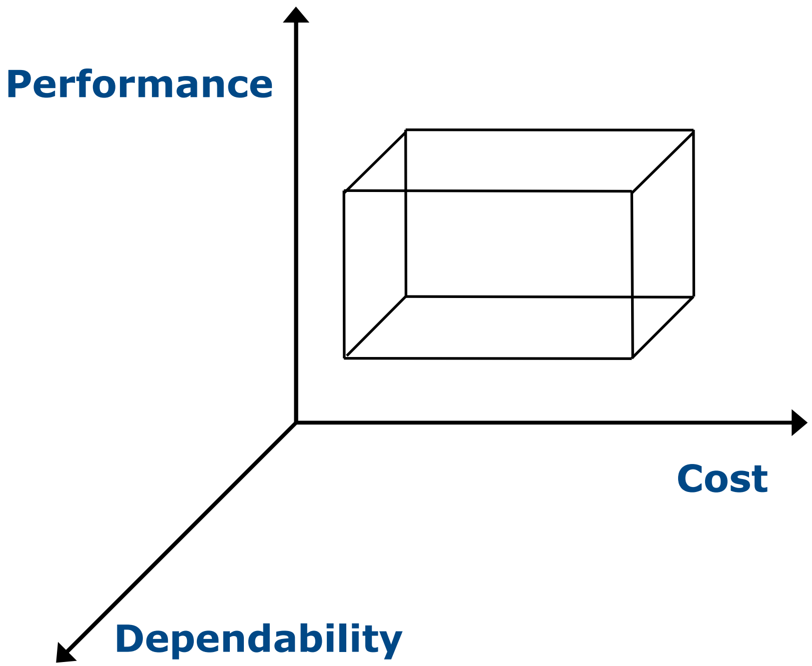


- Focus on variable selection as a major driver of model quality
- Optimal reaction schemes based on cost function
- More field data needed:
  - for reference and benchmarking purposes
  - to bridge the gap between academic approaches and industry

Remember to Be ***One Step Ahead***

- The objective is: to anticipate hostile actions, system malfunctions and crashes
- The key is **proactive failure management** - seamless failure and attack avoidance techniques by using **runtime monitoring and prediction**
- By using the presented techniques the system dependability can be significantly increased and specifically availability may be improved by an order of magnitude or more.

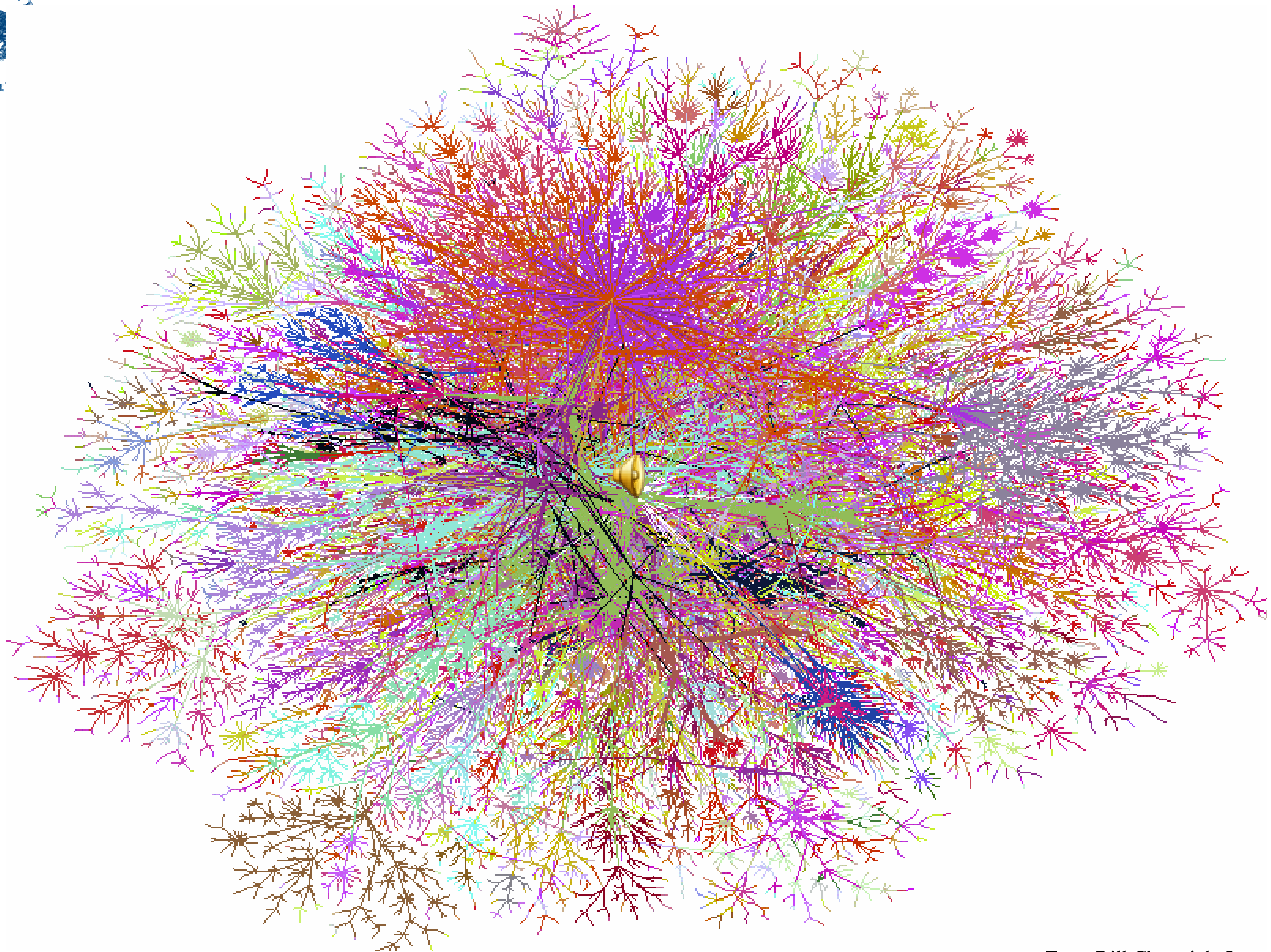
Adding a Third Dimension



Acknowledgment

I would like to acknowledge the contributions of my graduate students. My thanks go to **Günther Hoffmann** for development and presentation of UBF technique, **Felix Salfner** for development and presentation of HSMM method, **Steffen Tschirpke** for measurements and **Maren Lenk** for contributing to general prediction taxonomy.

We‘ve only just begun…



From Bill Cheswick, Lumeta Corp.

References

- Liang, Y., Zhang, Y., Sivasubramaniam, A., Jette, M., and Sahoo, R. 2006. Bluegene/l failure analysis and prediction models. In *IEEE Proceedings of the International Conference on Dependable Systems and Networks (DSN 2006)*. 425–434.
- Lin, T.-T. Y. and Siewiorek, D. P. 1990. Error log analysis: statistical modeling and heuristic trend analysis. *IEEE Transactions on Reliability 39,* 4 (Oct.), 419–432.
- Hoffmann, G. A. and Malek, M. 2006. Call availability prediction in a telecommunication system: A data driven empirical approach. In *Proceedings of the 25th IEEE Symposium on Reliable Distributed Systems (SRDS 2006)*. Leeds, United Kingdom.
- Hoffmann, G. A., Trivedi, K. S., and Malek, M. 2007. A best practice guide to resource forecasting for computing systems, *IEEE Transaction on Reliability 57, 6* (Dec).

References (continued)

- Salfner, F., Lenk, M., and Malek, M. 2007. A Survey of Online Failure Prediction Methods, manuscript submitted for publication.
- Salfner, F. and Malek, M. 2007. Using hidden semi-Markov models for effective online failure prediction. In *IEEE Proceedings on 26th International Symposium on Reliable Distributed Systems (SRDS 2007)*.
- Siewiorek, D.P., and Swarz R. S., *Reliable Computer Systems Design and Evaluation*. The Digital Press, 2nd edition, 1992
- Vilalta, R., Apte, C. V., Hellerstein, J. L., Ma, S., and Weiss, S. M. 2002. Predictive algorithms in the management of computer systems. *IBM Systems Journal 41,* 3, 461–474.
- Vilalta, R. and Ma, S. 2002. Predicting rare events in temporal domains. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*. IEEE Computer Society, Washington, DC, USA, 474–482.