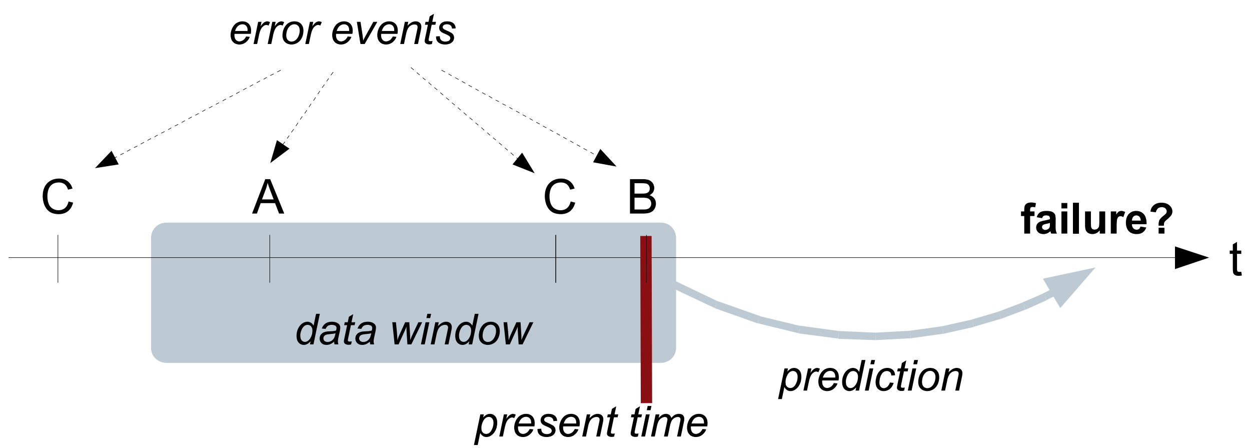
Error Log Processing for Accurate Failure Prediction

Felix Salfner
ICSI Berkeley

Steffen Tschirpke
Humboldt-Universität zu Berlin

Introduction

- Context of work: Error-based online failure prediction:



- Data used:
  - Commercial telecommunication system
  - 200 components, 2000 classes
  - Error- and failure logs

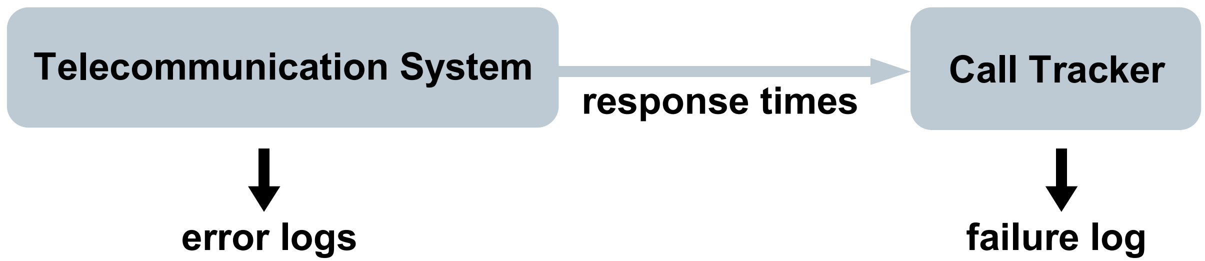
→ In this talk we present the **data preprocessing concepts** we applied to obtain accurate failure prediction results

Contents

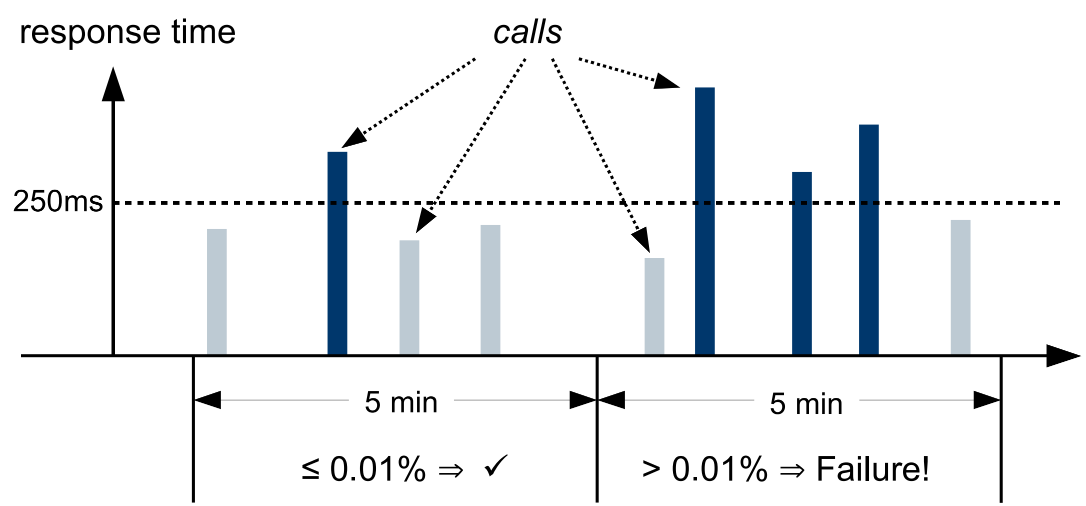
- Key facts on the data
- Overview of online failure prediction and data preprocessing process
- Detailed description of major preprocessing concepts
  - Assigning IDs to Error Messages
  - Failure Sequence Clustering
  - Noise Filtering
- Experiments and Results

Key Facts on the Data

- Experimental setup:

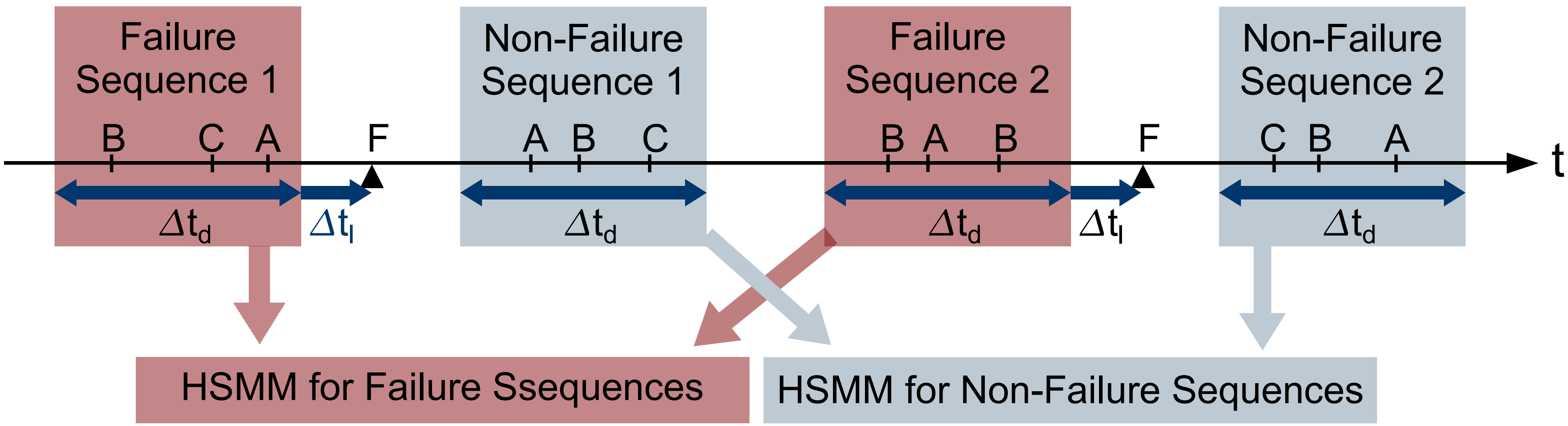


- 200 days of data from a 273 days period
- 26,991,314 error log records
- 1,560 failures of two types
- Failure Definition:
  - If within a 5 min interval more than 0.01% of calls experience a response time > 250ms
  - Performance Failures



Online Failure Prediction

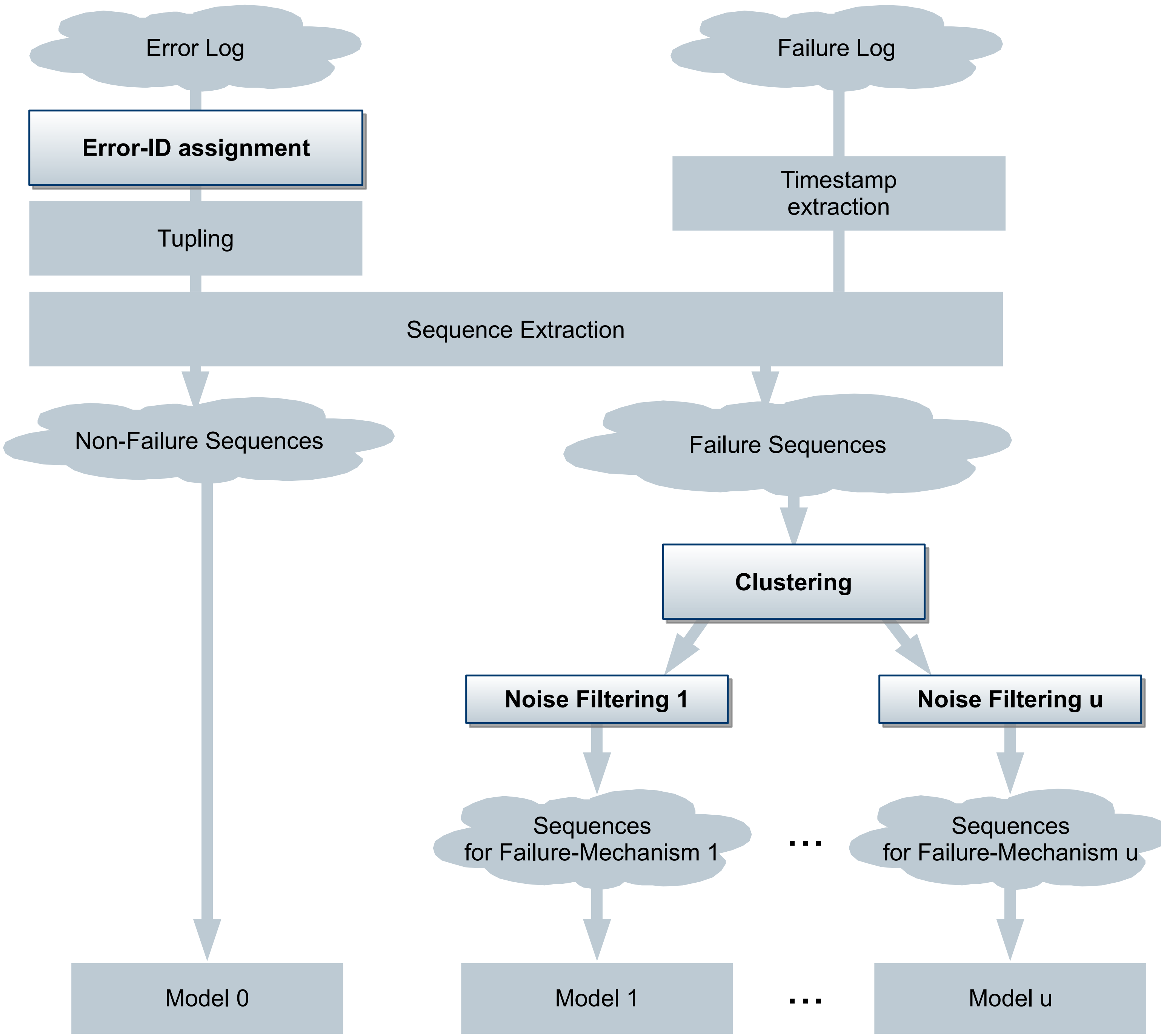
- Approach: Pattern recognition using Hidden Semi-Markov Models



- Objectives for data preprocessing:
  - Create a data set to train HSMM models exposing key properties of system
  - Identify how to process incoming data during runtime

- Tasks:
  - Machine-processable data → **Error-ID assignment**
  - Separate sequences for inherent failure mechanisms → **Clustering**
  - Distinguishing, noise-free sequences → **Noise Filtering**

Training Data Preprocessing



Error ID Assignment

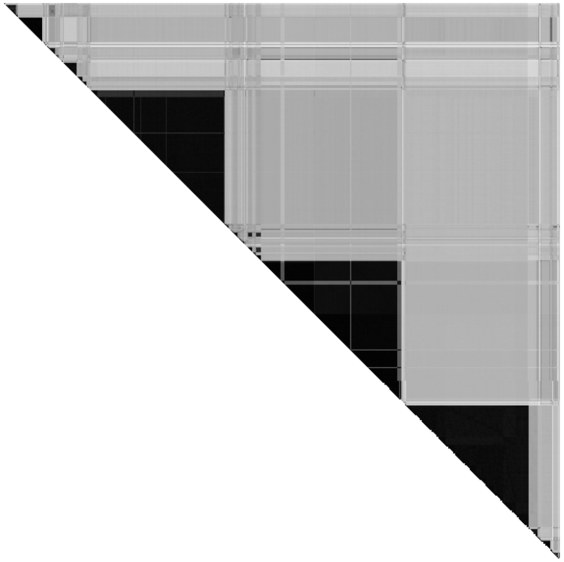
- Problem: Error logs contain no message IDs
  - Example message of a log record:

    `process 1534: end of buffer reached`

  → Task: Assign an ID to message to characterize *what has happened*
- Approach: Two steps:
  - Remove numbers

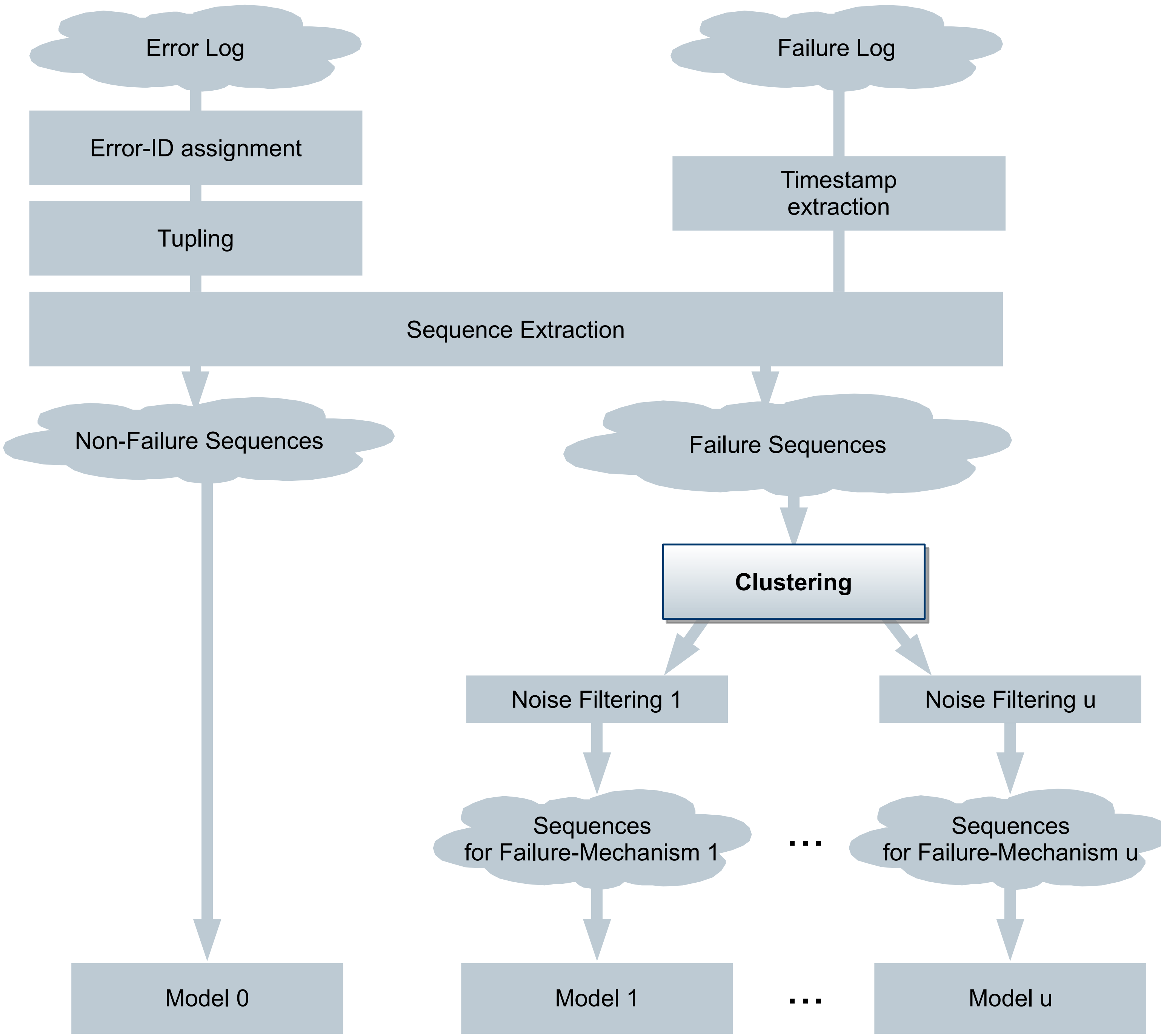
    `process xx: end of buffer reached`

  - ID assignment based on Levenshtein's edit distance with constant threshold



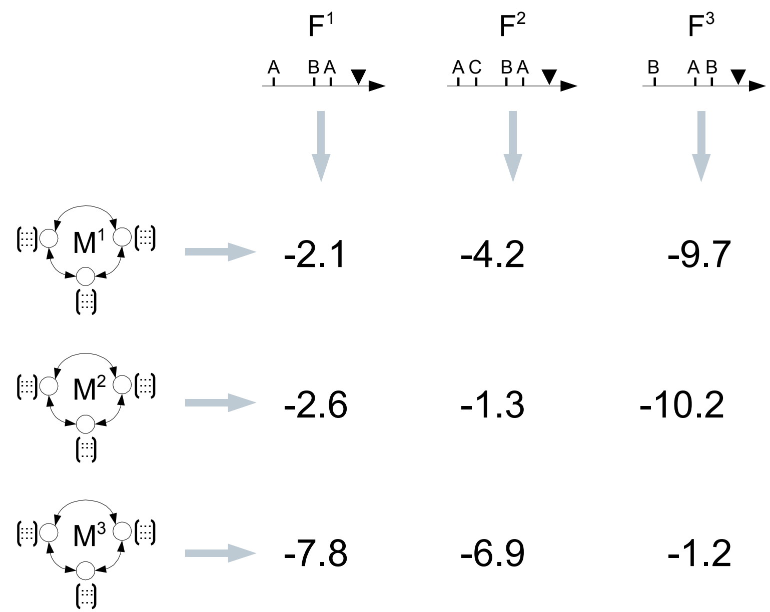
Data	No of Messages	Reduction
Original	1,695,160	
Without numbers	12,533	
Levenshtein	1,435	

Failure Sequence Clustering



Failure Sequence Clustering (2)

- Goal:
  - Divide set of training failure sequences into subsets
  - Group according to sequence similarity
- Approach:
  - Train a small HSMM for each sequence
  - Apply each HSMM to all sequences
  - Sequence log-likelihoods express similarities

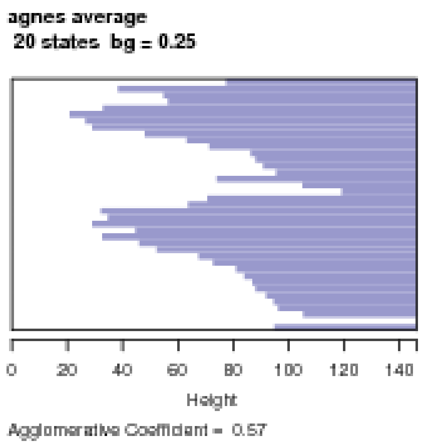


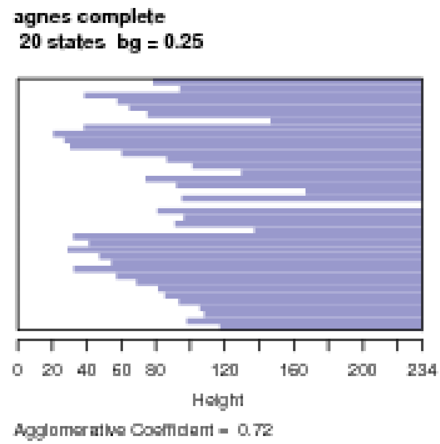
  - Make matrix symmetric by

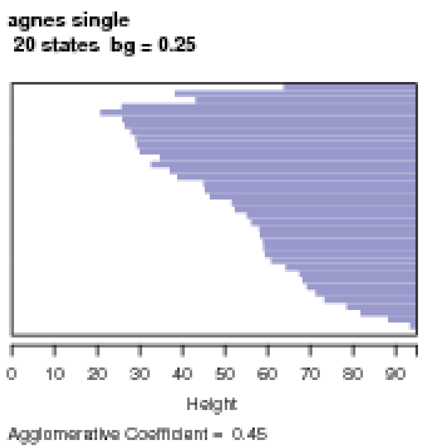
$$D(i,j) = \left| \frac{\log\left[P(F^i|M^j)\right] + \log\left[P(F^j|M^i)\right]}{2} \right|$$

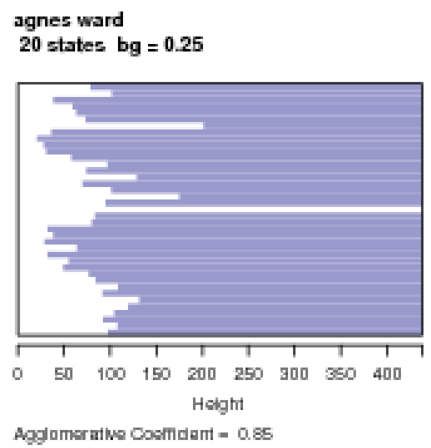
  - Apply standard clustering algorithm

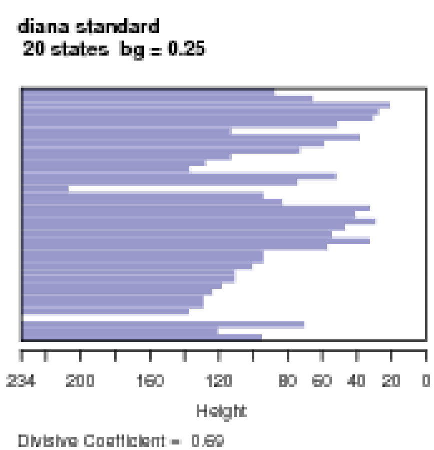
Failure Sequence Clustering (3)



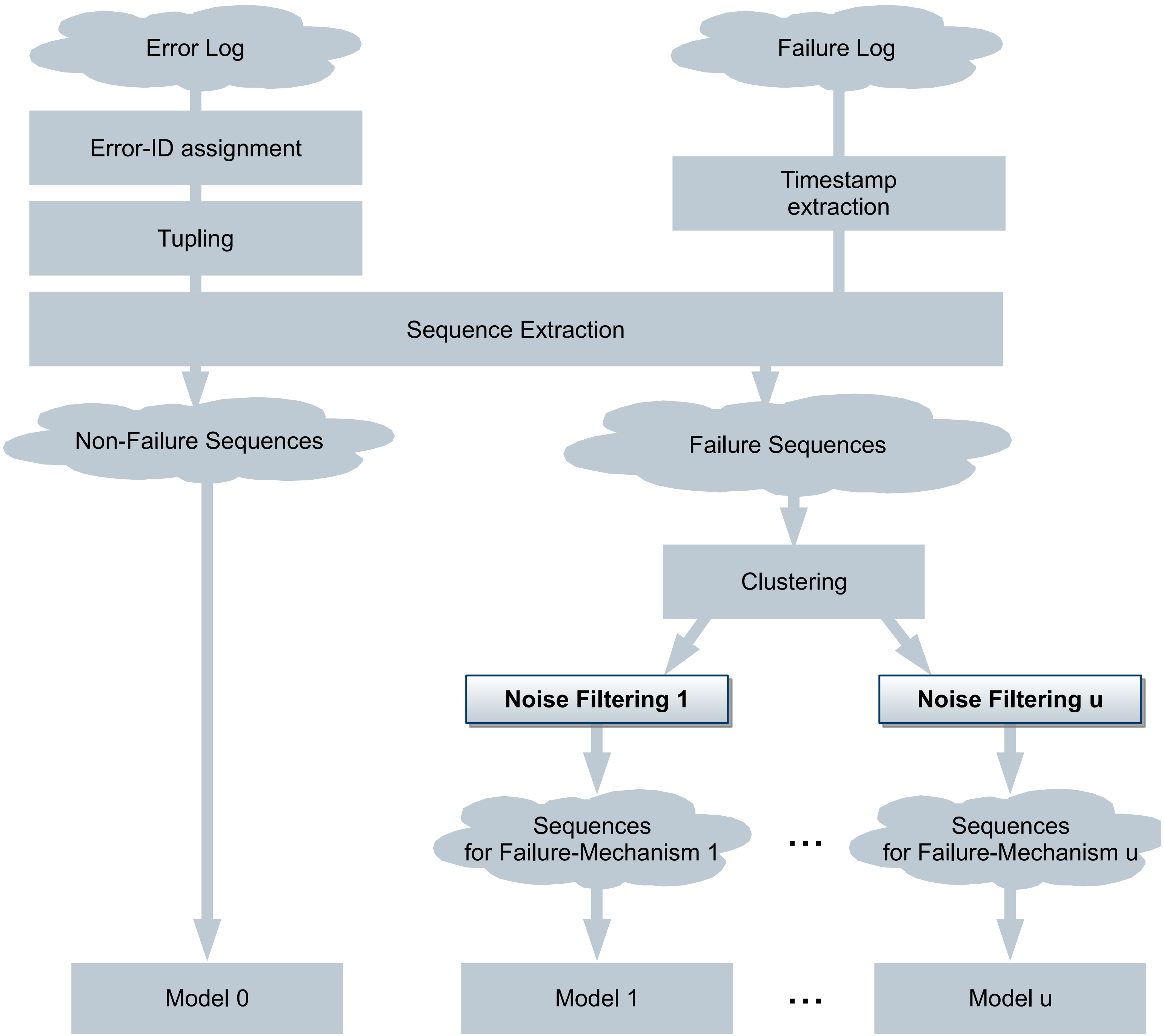






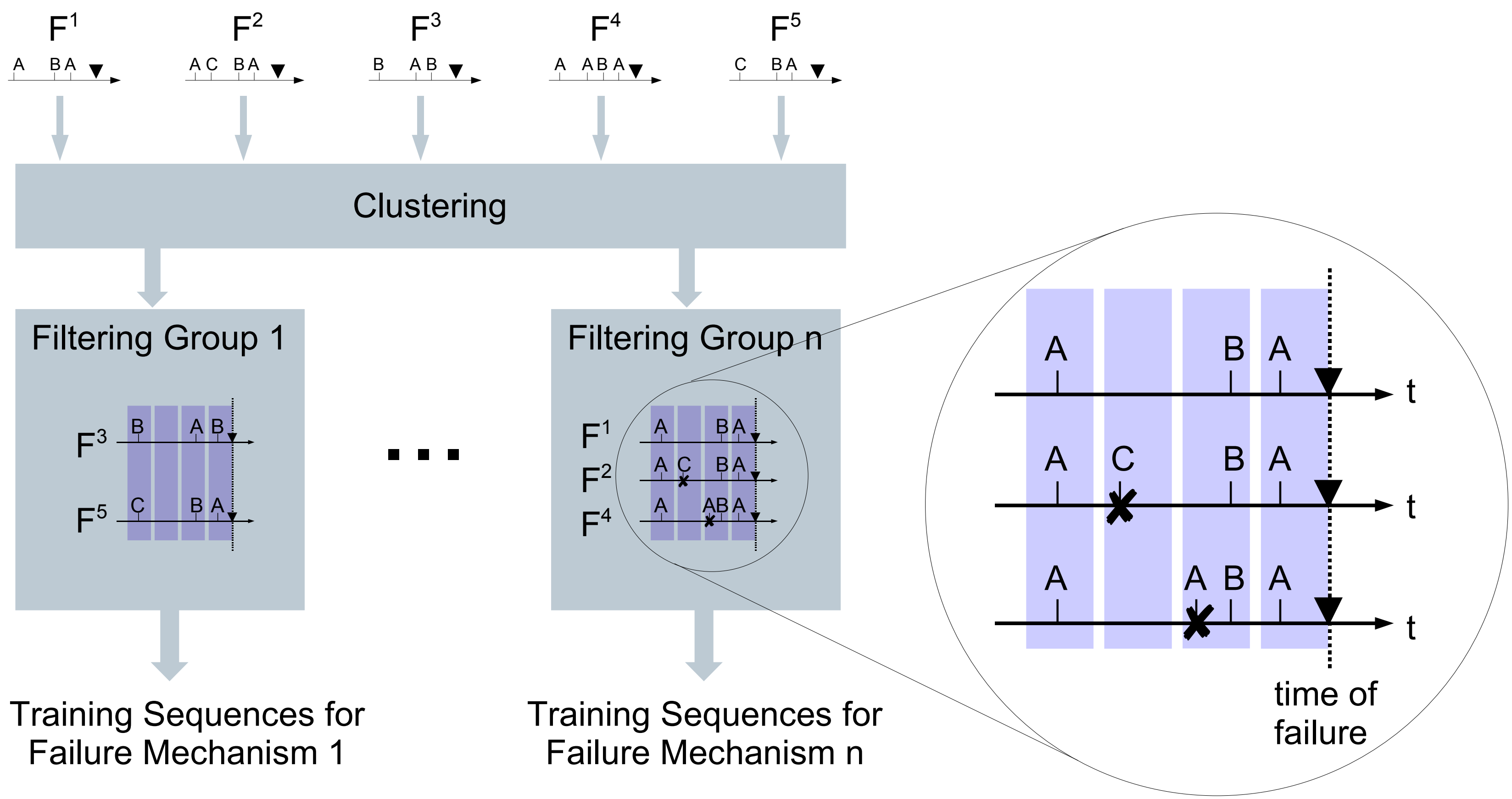


Noise Filtering



Noise Filtering (2)

- Problem: Clustered failure sequences contain many unrelated errors
  → Main reason: parallelism in the system
- Assumption: Indicative events occur more frequently prior to a failure than within other sequences
  → Apply a statistical test to quantify what “more frequently” is



Noise Filtering (3)

- Testing variable derived from χ^2 goodness-of-fit test:

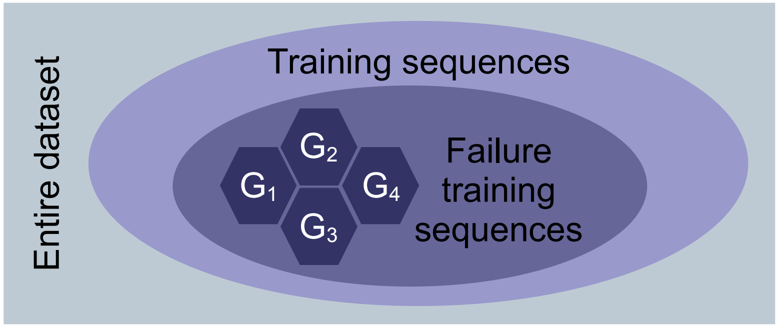
$$X_i = \frac{n_i - n\,\hat{p}_i^0}{\sqrt{n\,\hat{p}_i^0}}$$

n_i denotes the number of occurrences of error e_i
n denotes the total number of errors in the time window.
$\hat{p}_i^0$ denotes the prior probability of occurrence of error e_i

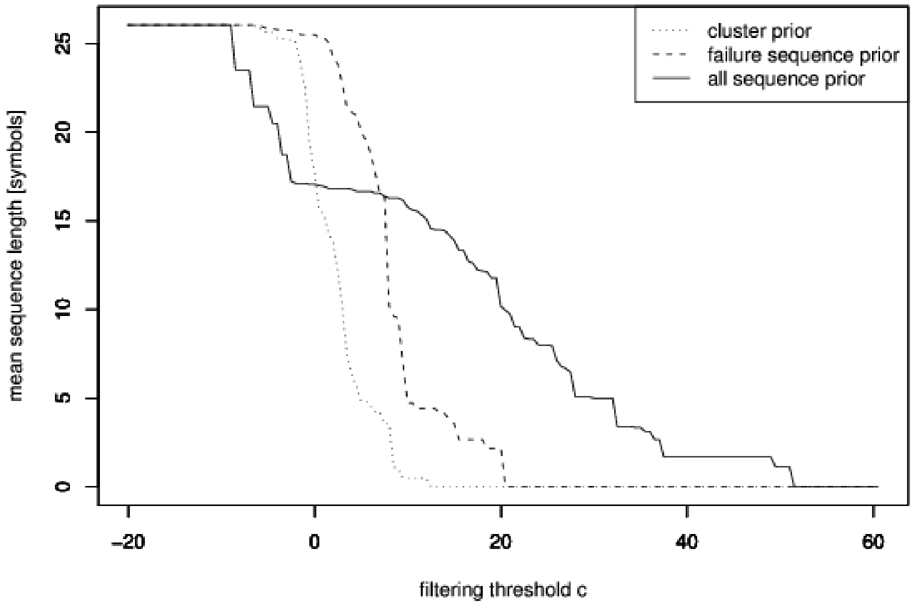
- Keep events in the sequence if

$$X_i > c$$

- Three ways to estimate priors $\hat{p}_i^0$ from training data set



- Results

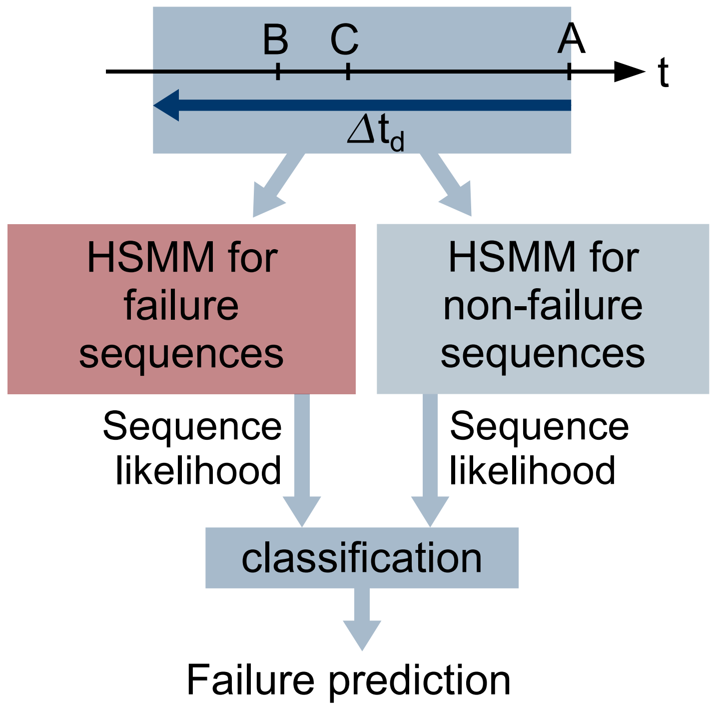


Experiments and Results

- Objective: Predict upcoming failures as accurate as possible
- Metric used: F-Measure:
  - Precision: relative number of correct alarms to total number of alarms
  - Recall: relative number of correct alarms to total number of failures
  - F-Measure: harmonic mean of precision and recall
- Failure prediction is achieved by comparing sequence likelihood of an incoming sequence computed from failure and non-failure models
- Classification involves a customizable decision threshold

→ Maximum F-Measure

Data	Max. F-Measure	Relative Quality
Optimal Results	0.66	100%
Without grouping	0.5097	77%
Without filtering	0.3601	55%



Conclusions

- We have presented the data preprocessing techniques that we have applied for online failure prediction in a commercial telecommunication system
- The presented techniques include:
  - Assignment of IDs to error messages using Levenshtein's edit distance
  - Failure sequence clustering
  - Noise filtering based on a statistical test
- Using error and failure logs of the commercial telecommunication system, we showed that elaborate data preprocessing is an essential step to achieve accurate failure predictions

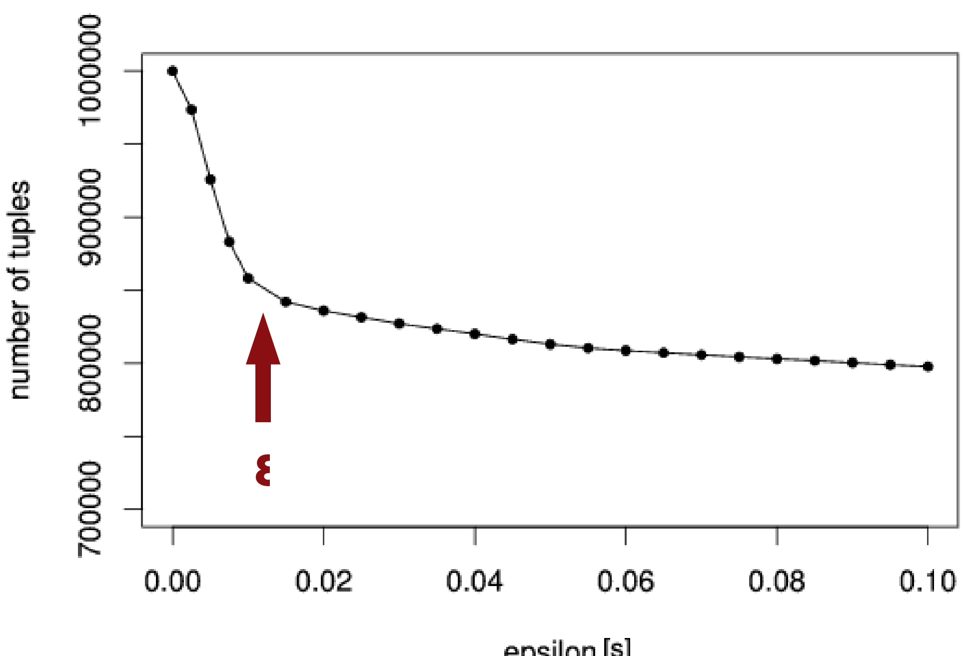
Backup

Tupling

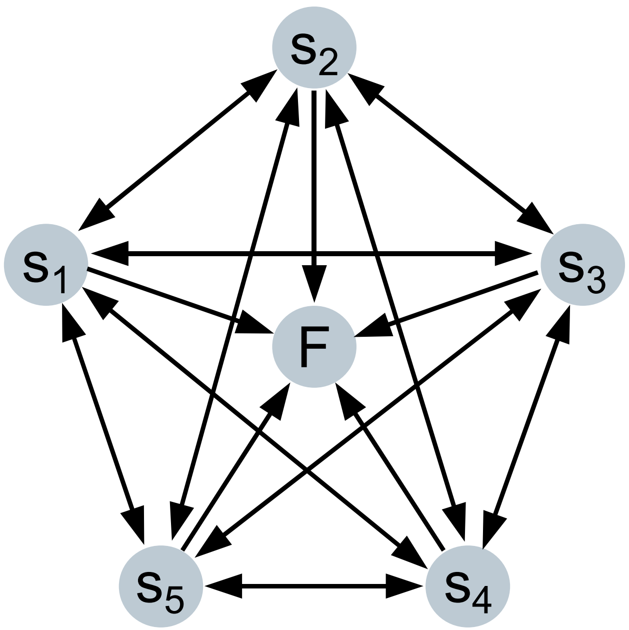
- Goal: Remove multiple reporting of the same issue
- Approach:

  Combine messages of the same type if they occur closer in time to each other than a threshold ε.

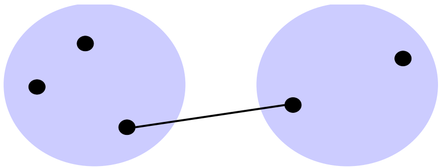
- Problem:
  - Determine the threshold value ε
  - Solution suggested by Tsao and Siewiorek: Observe the number of tuples for various values of ε and apply the “elbow rule”



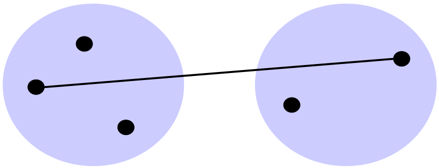
HSMM Model Structure for Failure Sequence Clustering



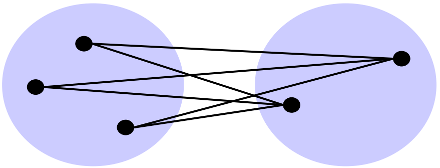
Cluster Distance Metrics



Single linkage

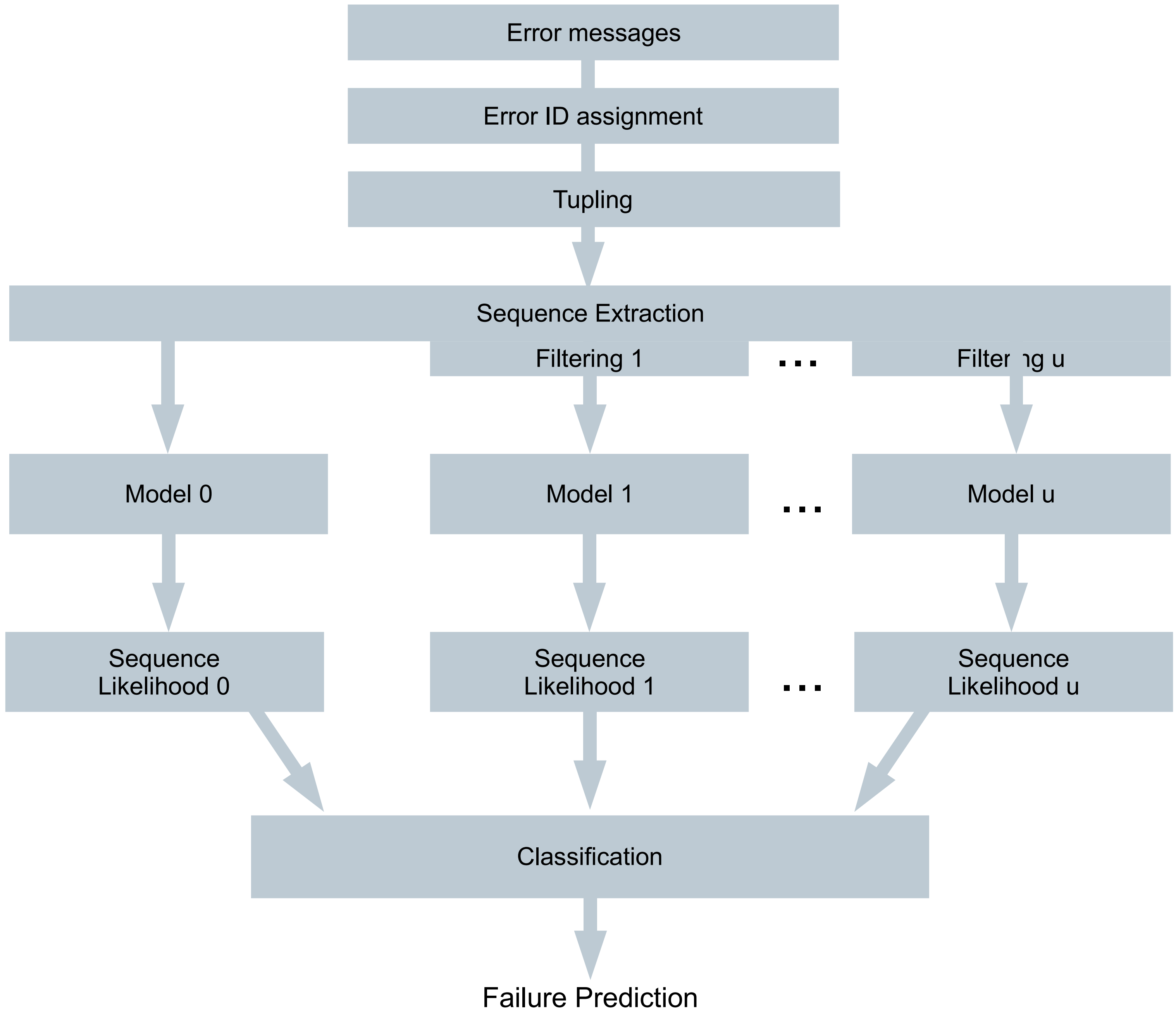


complete linkage

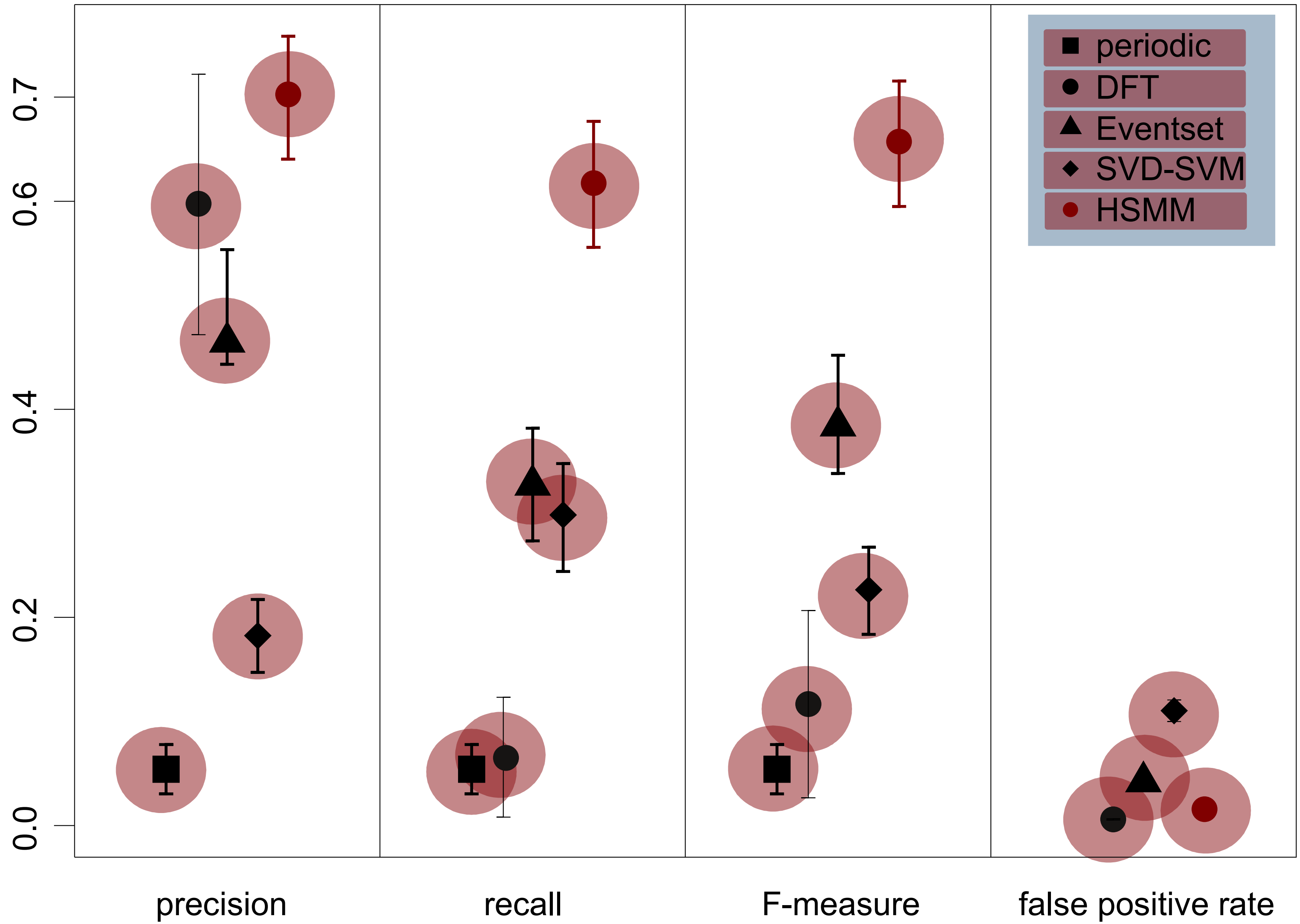


Average linkage

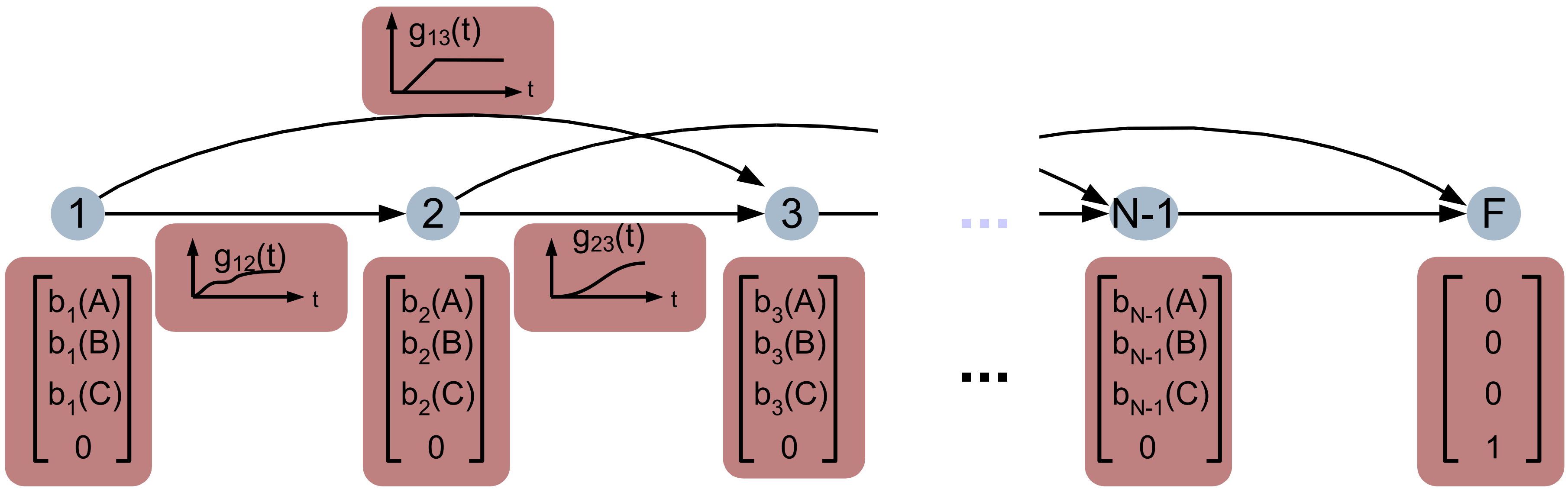
Online Failure Prediction



Comparison of Techniques



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 $g_{ij}(t)$

Proactive Fault Management