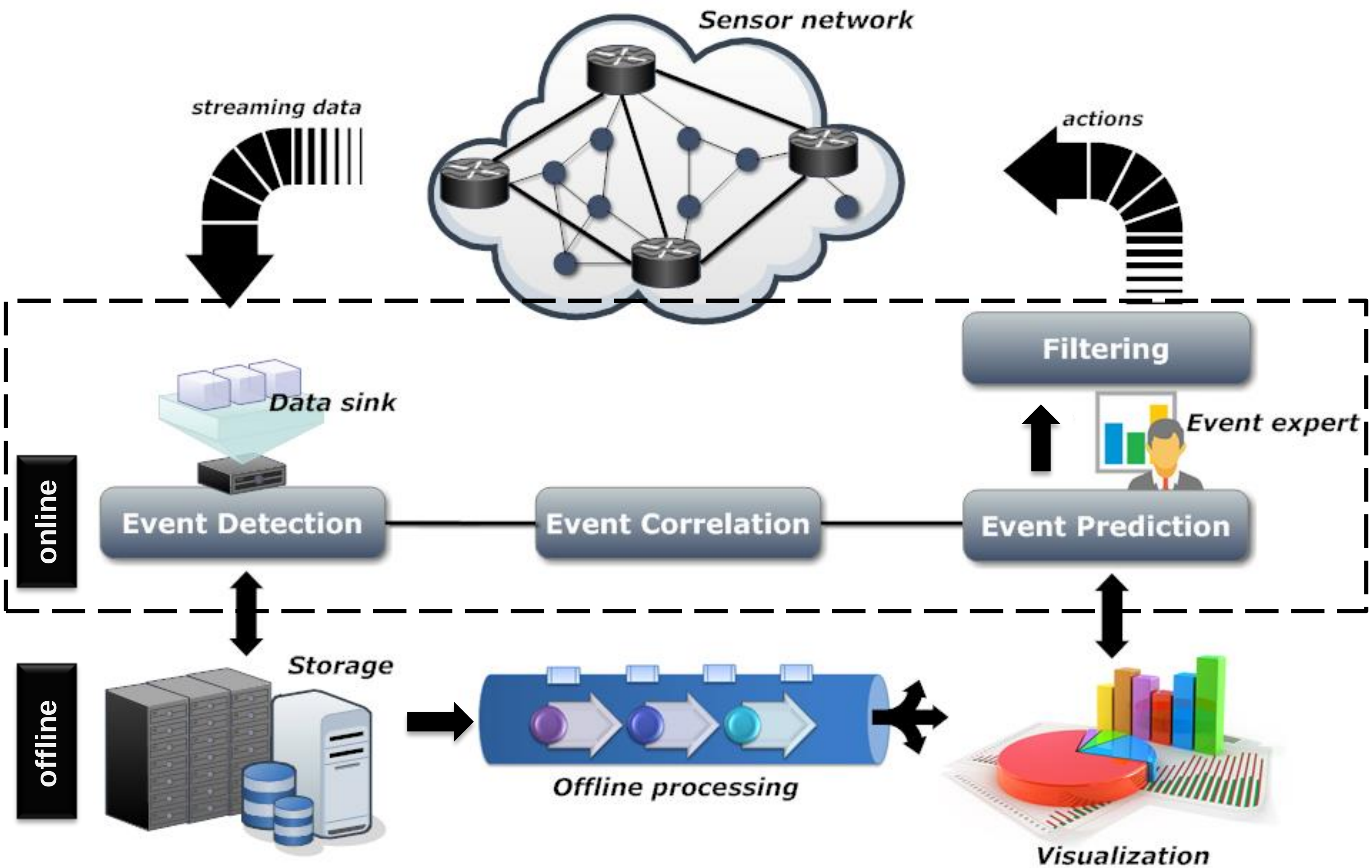
# Event Management in Multivariate Streaming Sensor Data

# Event management lifecycle in SN

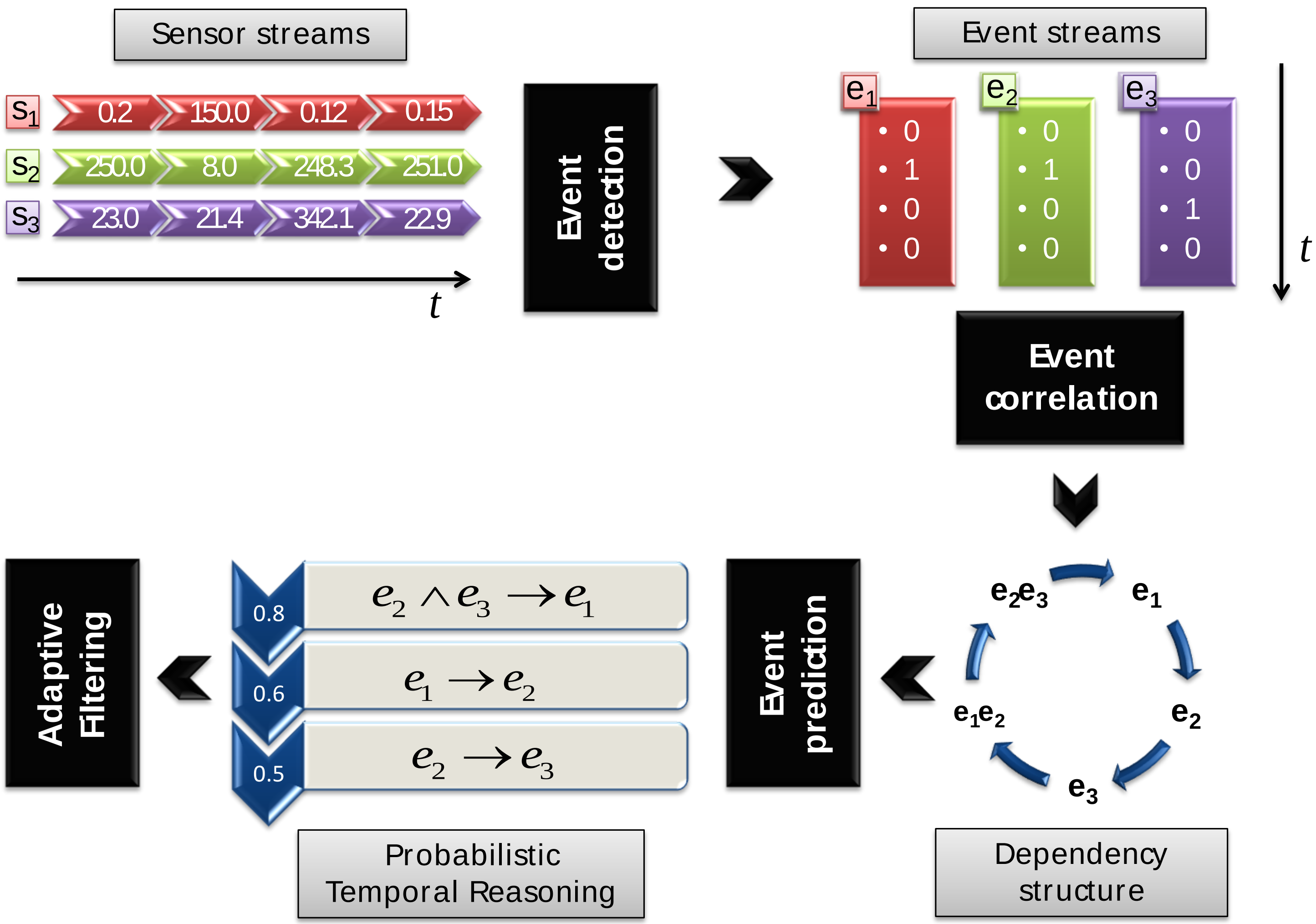


# What is an event?

- A. Change detection
  - Continuous monitoring of sensor streams
  - Abrupt change on value distribution

  (0.12, 1.11, 1.09, 2566.04, …)
  - Algorithms
    - Single variate regression
      - CUMSUM, Schewart Controller
    - Multi-variate regression
- B. Predefined conditions checked in real-time
- Events are stored as a ***binary table***

# Online event processing



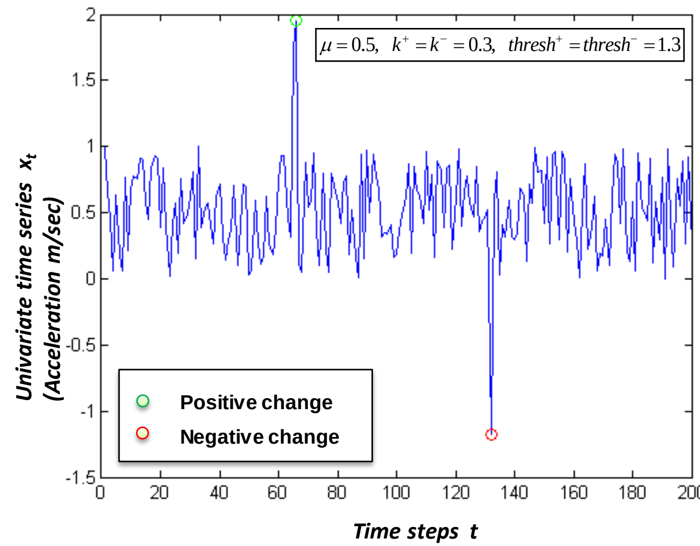
# Change Det: Cumulative Sum (CUSUM)

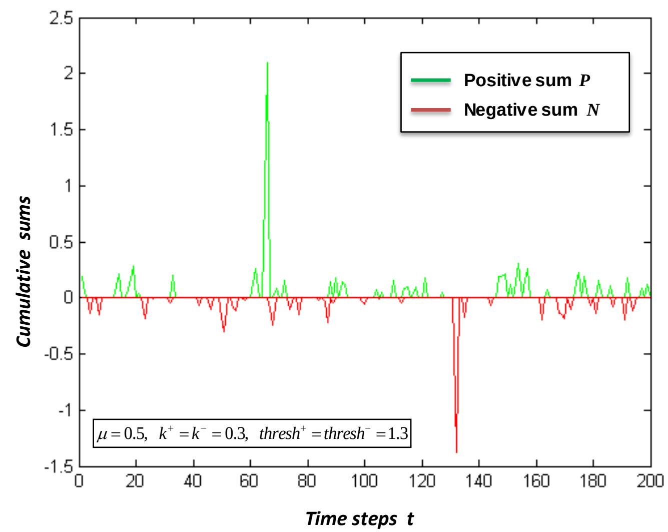
**ALGORITHM 1.** Cumulative Sum (CUSUM)

**Input:** univariate time series $x_t$, target value $\mu$, above-tolerance $k^+$, below-tolerance $k^-$, above-threshold $thres^+$, below-threshold $thres^-$

**Output:** above detection signal $s^+$, below detection signal $s^-$

$P \leftarrow 0$;
$N \leftarrow 0$;
$t \leftarrow 1$;
**while** ( *true* )
    $s^+ \leftarrow 0$;
    $s^- \leftarrow 0$;
    $P \leftarrow \max\left(0, x_t - \left(\mu + k^+\right) + P\right)$;
    $N \leftarrow \min\left(0, x_t - \left(\mu - k^-\right) + N\right)$;
    **if** ( $P > thres^+$ ) **then**
        $s^+ \leftarrow 1$;
        $P \leftarrow 0$;
        $N \leftarrow 0$;
    **end**
    **if** ( $N < -thres^-$ ) **then**
        $s^+ \leftarrow 1$;
        $P \leftarrow 0$;
        $N \leftarrow 0$;
    **end**
    $t \leftarrow t+1$;
**end**



# Change Det: Shewhart controller

**ALGORITHM 2.** Shewhart Control Chart

**Input:** univariate time series $x_t$ , tightness $k$

**Output:** detection signal $s$

$\bar{x}_0 \leftarrow 0$ ;

$\sigma_0 \leftarrow 0$ ;

$t \leftarrow 1$ ;

**while** ( *true* )

  $\bar{x}_t \leftarrow \bar{x}_{t-1} + \dfrac{x_t - \bar{x}_{t-1}}{t}$ ;

  $\sigma_t \leftarrow \sqrt{\dfrac{1}{t}\left((t-1)\cdot\sigma_{t-1}^2 + (x_t - \bar{x}_t)(x_t - \bar{x}_{t-1})\right)}$ ;

  $UCL_t \leftarrow \bar{x}_t + k\cdot\sigma_t$ ;

  $LCL_t \leftarrow \bar{x}_t - k\cdot\sigma_t$ ;

  **if** (( $x_t > UCL$ ) **or** ( $x_t < LCL$ )) **then**

    $s \leftarrow 1$ ;

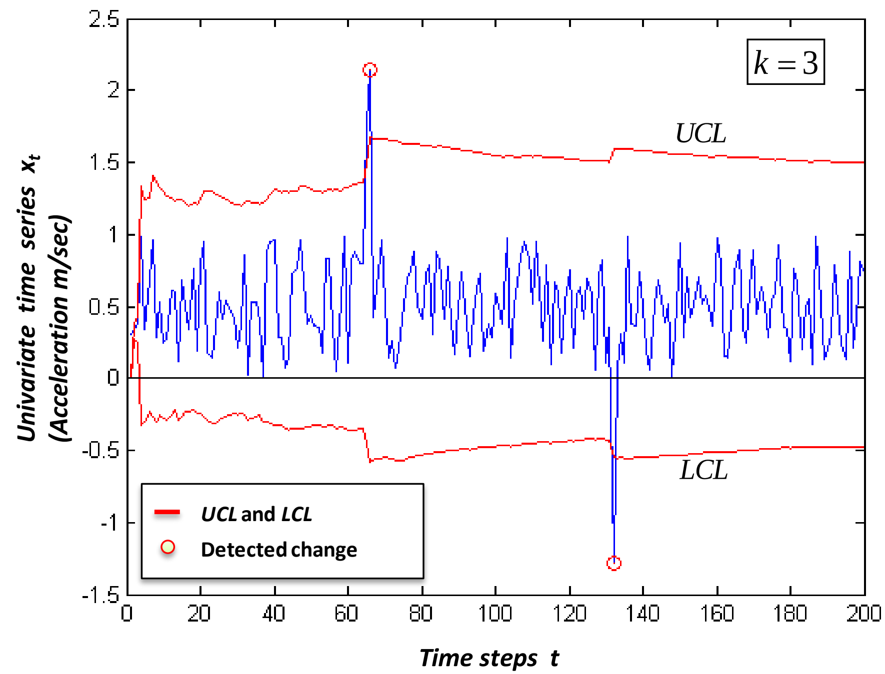
  **else**

    $s \leftarrow 0$ ;

  **end**

  $t \leftarrow t+1$ ;

**end**



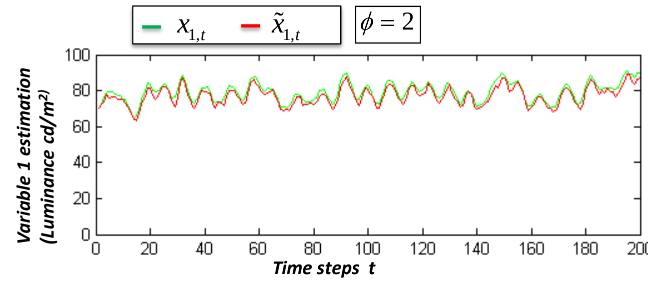
# Change Det: Multivariate Autoregressive Model (MAR)

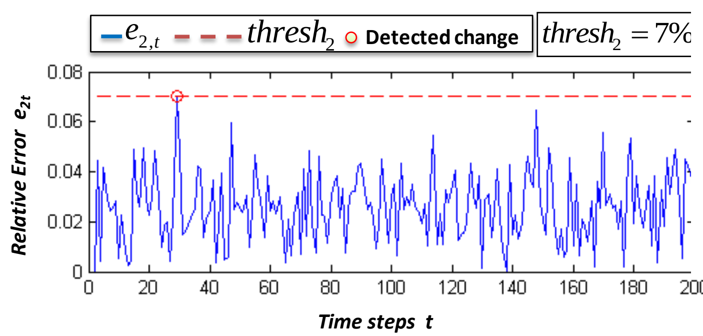
**ALGORITHM 3.** MAR-based change detection

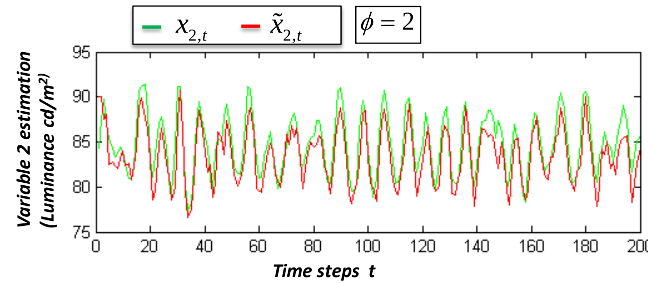
**Input:** multivariate time series $\boldsymbol{x}_t = (x_{1,t}, \ldots, x_{n,t})$, number of training samples $k$, thresholds $(thresh_1, \ldots, thresh_n)$

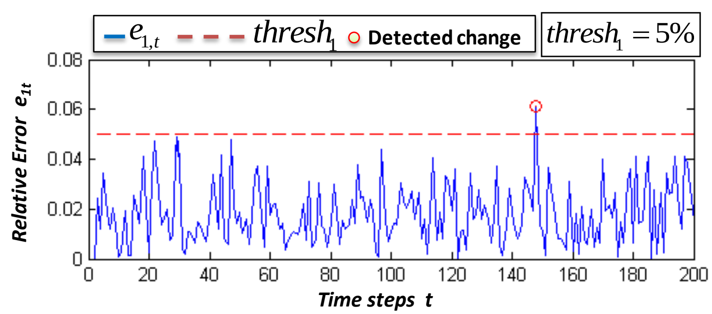
**Output:** detection signal $s$

Estimate the model $\langle c, \phi, \boldsymbol{\Pi}_1, \boldsymbol{\Pi}_2, \ldots, \boldsymbol{\Pi}_\phi \rangle$ that fits the training data $\{\boldsymbol{x}_t\}$, $\forall t \in [1,k]$
$t \leftarrow k+1$;
**while** ( *true* )
  $\tilde{\boldsymbol{x}}_t \leftarrow c + \boldsymbol{\Pi}_1 \boldsymbol{x}_{t-1} + \ldots + \boldsymbol{\Pi}_\phi \boldsymbol{x}_{t-\phi}$;
  **for** $i \leftarrow 1$ **to** $n$
    $e_{i,t} \leftarrow \frac{\| x_{i,t} - \tilde{x}_{i,t} \|}{\| x_{i,t} \|}$;
    **if** $(e_{i,t} > thresh_i)$ **then**
      $s \leftarrow 1$;
    **else**
      $s \leftarrow 0$;
    **end**
  **end**
**end**



# Event Correlation Engine (ECE)

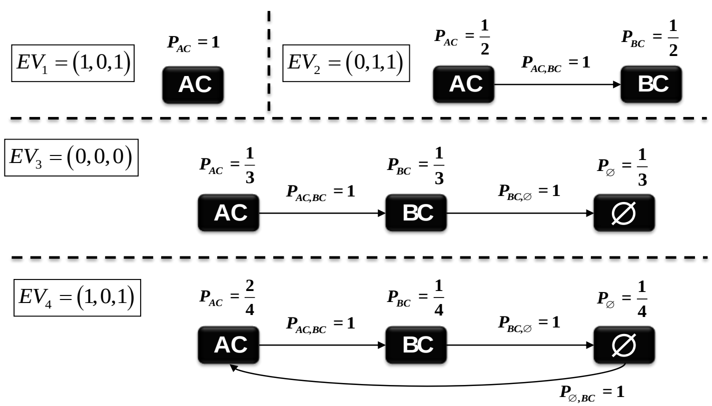
- Data-driven approach (no pre-defined model)
- Facilitates decision-making process
  - Post-analysis of events (***offline mode***)
    - Root Cause – RC determination, Cause analysis
    - Explanation (the sequence of events that led to an event triggering)
  - Prediction (***online mode***)
    - Predict system behavior in the near future (events that will be possibly triggered – coming with a probability value)

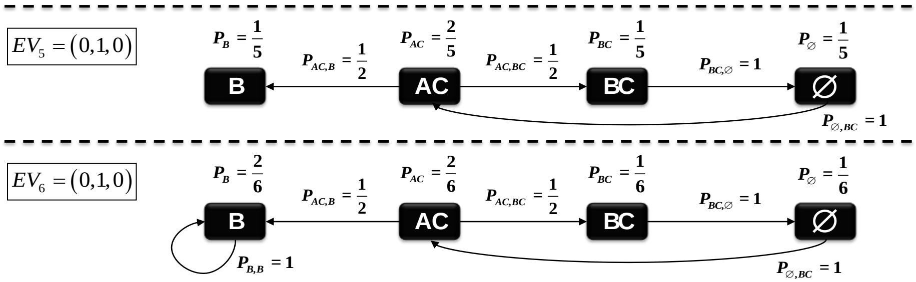
# Event Correlation: Stepwise approach

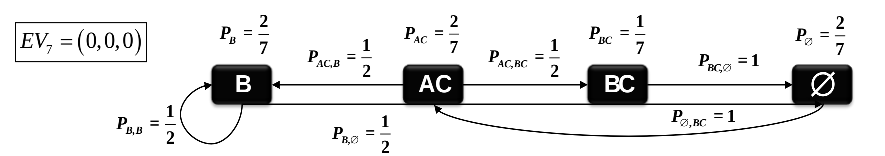
$$(\forall t \in N)(\forall I_l, I_m \subseteq I)\ P_{I_l I_m}^{\overline{1:t}} = P\left(I_m \mid I_l, \overline{1:t}\right) =$$

$$\frac{N\left(I_l^{t-1}, I_m^t \mid \overline{1:t}\right)}{N\left(I_l \mid \overline{1:t-1}\right)} = \frac{\sum_{k=2}^{t} \prod_{e_i \in I_l, \bar{e}_j \in \bar{I}_l} \prod_{e_p \in I_m, \bar{e}_q \in \bar{I}_m} e_i^{k-1} \cdot \bar{e}_j^{k-1} \cdot e_p^k \cdot \bar{e}_q^k}{\sum_{k=1}^{t-1} \prod_{e_i \in I_l, \bar{e}_j \in \bar{I}_l} e_i^k \cdot \bar{e}_j^k}$$

| A | B | C |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |

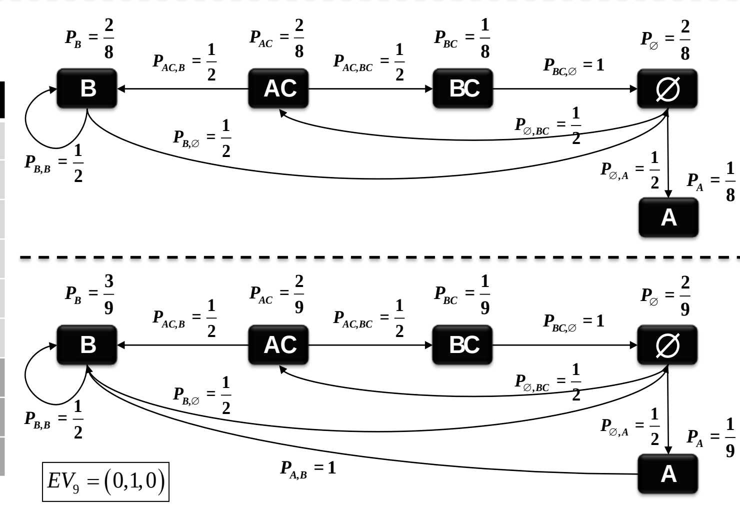


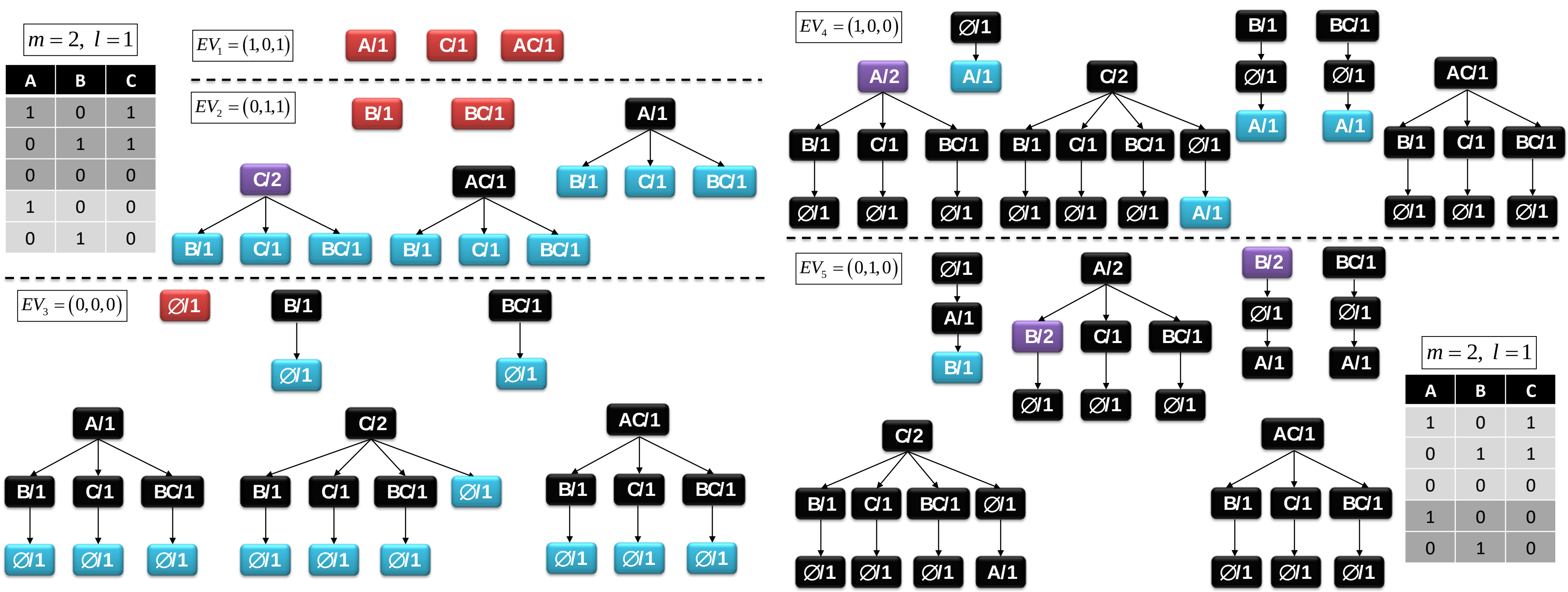

$EV_8 = (1,0,0)$

| A | B | C |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |



# Event Correlation: Variable-order approach

- Similar to the previous approach, but now multiple Markov models are considered
- Markov-chains of order *1,…,m* are combined to predict event sequences of length up to *m*



# Event Correlation: Sliding-window approach

$$\alpha_v^t = \begin{cases} 1 & \text{if } \prod_{e_i \in I_v} e_i^t = 1 \\ 0 & \text{otherwise} \end{cases}$$

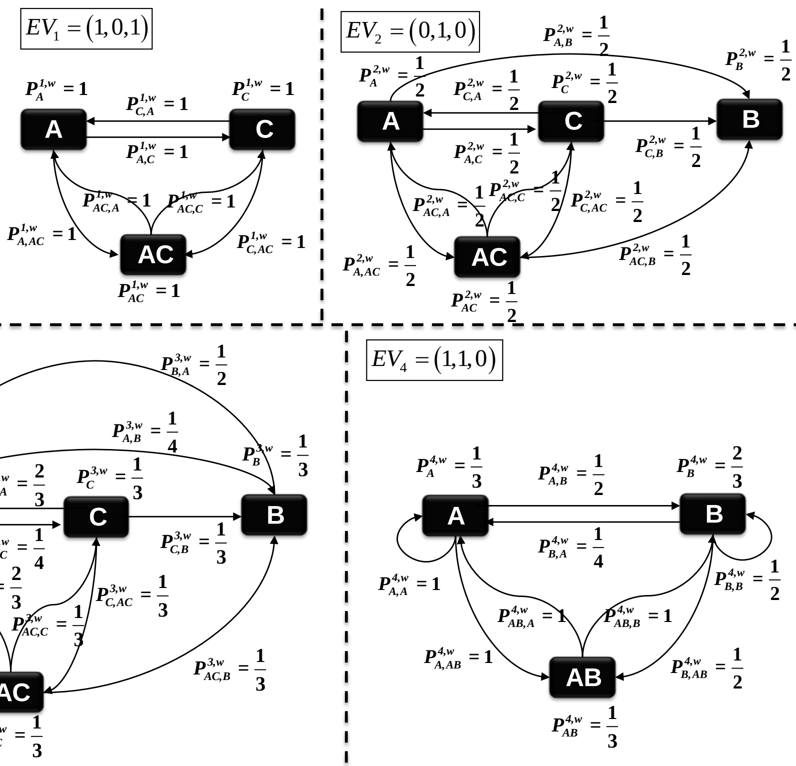
$$\beta_{uv}^{t_1,t_2} = \begin{cases} 1 & \text{if } \prod_{e_i \in I_u, e_j \in I_v} e_i^{t_1} \cdot e_j^{t_2} = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$(\forall u, v \in V) \quad N_{uv}^{t,w} = \sum_{i=t-w+1}^{t} \sum_{j=i}^{t} \beta_{uv}^{i,j}$$

$$(\forall u, v \in V) \quad P_{uv}^t = \frac{N_{uv}^{t,w}}{\sum_{j=t-w+1}^{t} \alpha_u^j \cdot (t-j+1)}$$

$w = 3$

| A | B | C |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



# Dependency Graph

- Probabilistic Directed Graph (cycles are possible)
- *V* (nodes): attributes (in our case ***sensors***)
- *E* (edges): event transition
- $P_{ij}$: conditional probabilities of event pairs $V_i - V_j$
- Parameters
  - *w*: look-ahead window
  - $p_c$: cutoff probability
  - *a*: aging factor
- Formal transformation into “if…then” rules (SWRL)
- Possible execution inside a probabilistic rule engine

# Event prediction

- Event prediction by probabilistic temporal reasoning
  - *probabilistic temporal rules* [Shakarian et al. 2011]

    $$A \rightarrow B : [t, p]$$

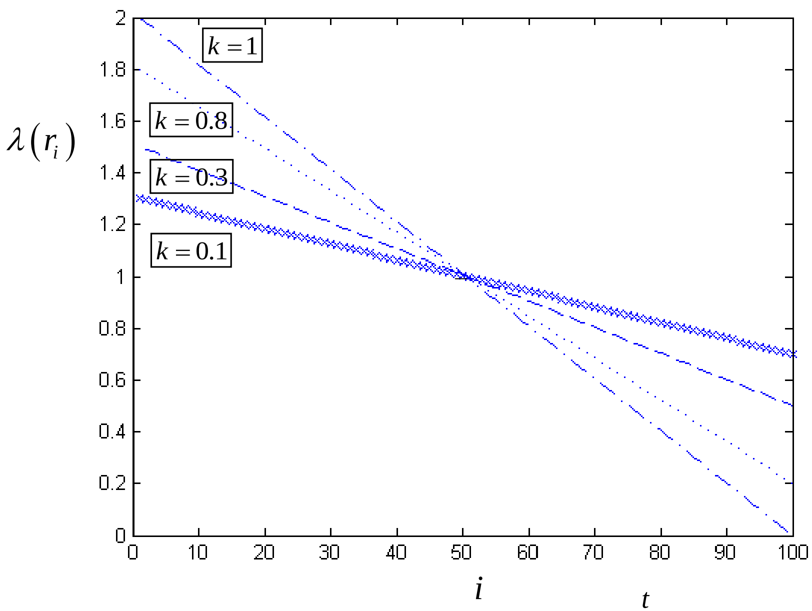
  - *A,B* are (ground) formulae consisting of (ground) atoms and typical logic programming operators for conjunction, disjunction and negation while the (ground) formula *B* is annotated with a probability value *p* and a time unit *t*.

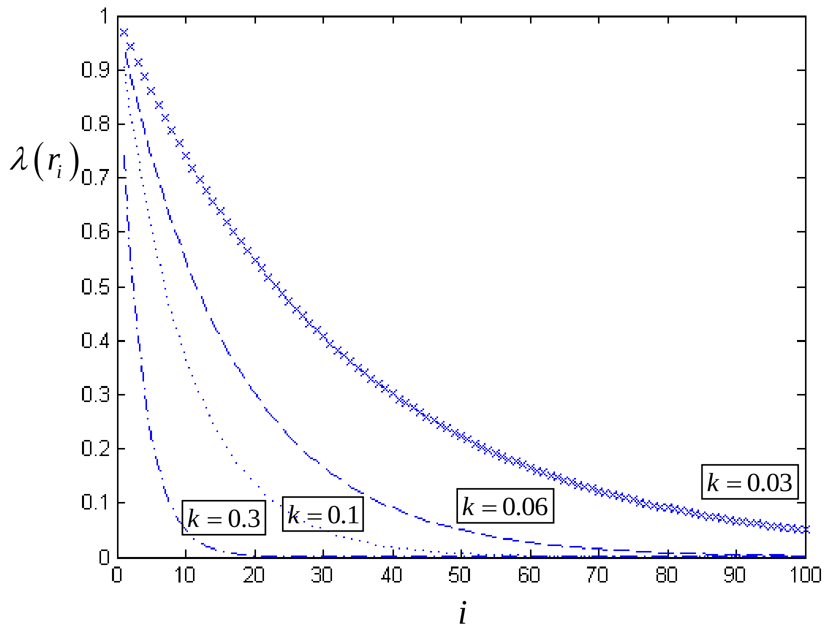
# Adaptive filtering of rules

- Use of *aging* or *decay function* $\lambda(r_t) = f(t)$
- $f(t)$ : Linear or exponential degradation

$$\lambda(r_i) = -\frac{2k}{n-1}(i-1)+k+1$$

$$\lambda(r_i) = \exp(-ki)$$

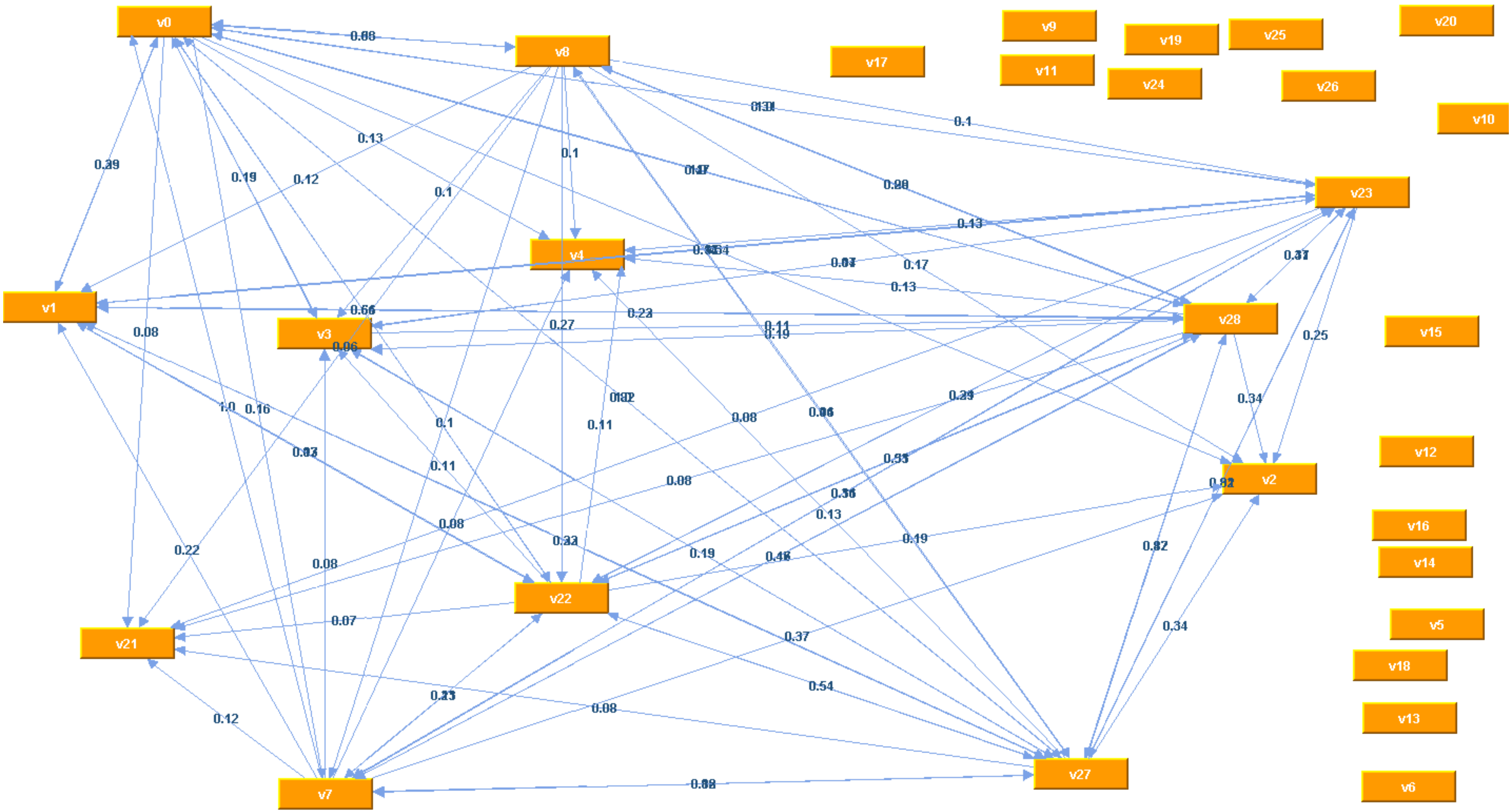


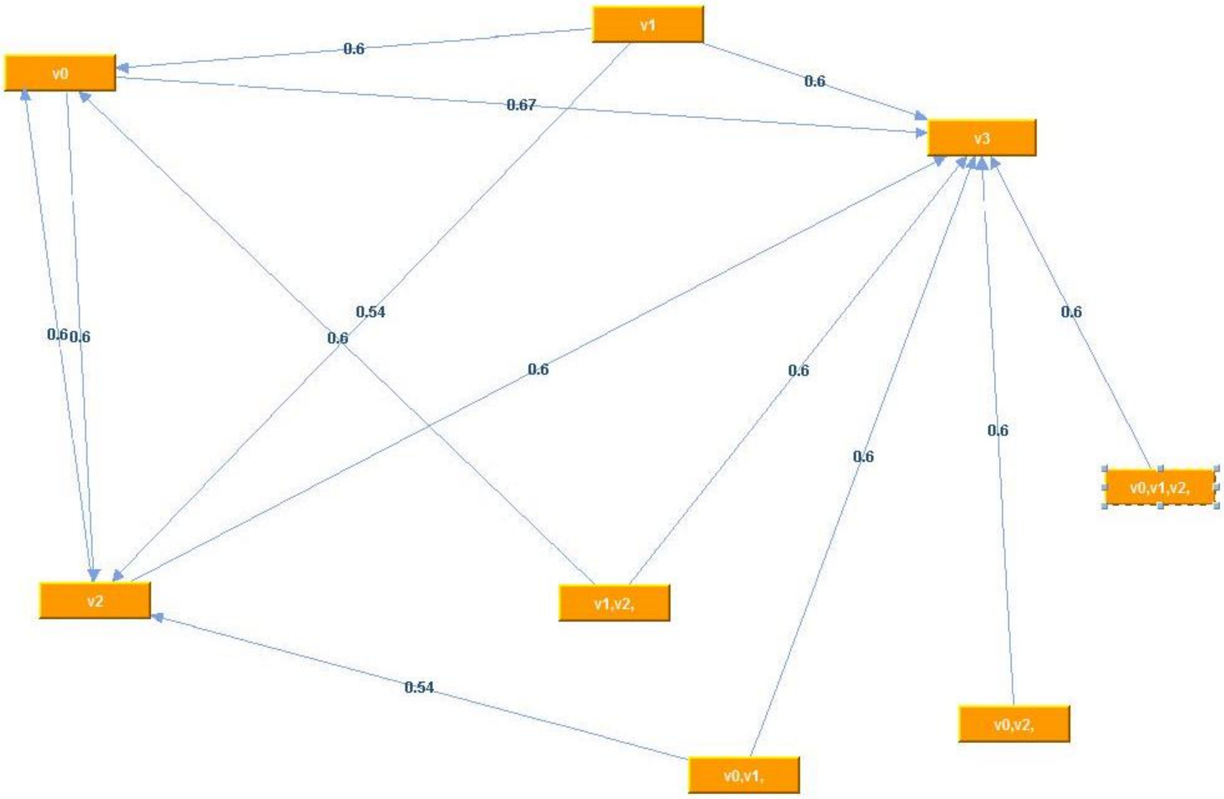

$$p_{r,\Delta t} = \frac{\sum_{i=t-\Delta t+1}^{t} \lambda(r_i)\cdot p_{r,i}}{\sum_{i=t-\Delta t+1}^{t} \lambda(r_i)}$$

*Rules probability*

# Complete Graph



# Extract Useful Correlations




- Implementation technologies
  - *Java, Oracle DB, JGraphT lib for visualization*