SRDS
44th International Symposium on Reliable Distributed Systems (SRDS)
September 29 – October 2, 2025

Nip it in the Bud: Unsupervised KPI Incipient Fault Detection via Dynamic Latent Feature Ensembling

Yanwen Wang, Wenda Tang, Jie Wu

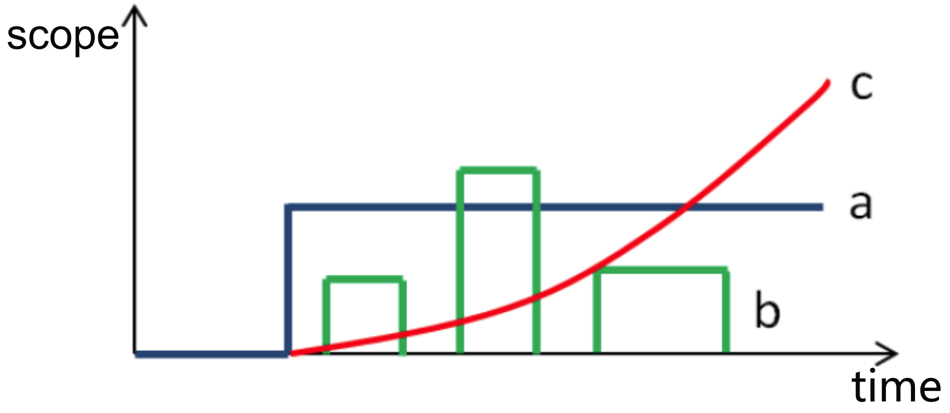
China Telecom Cloud Computing Reserch Institute, Beijing, China

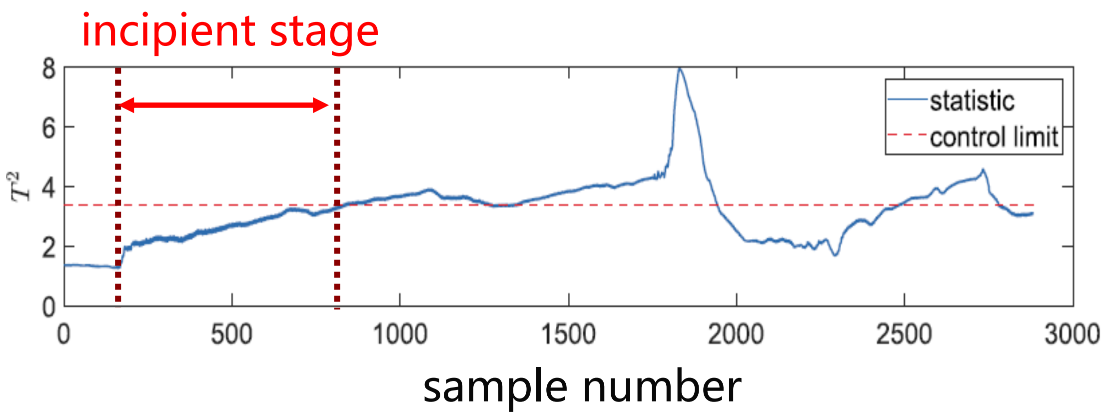
中国电信
CHINA TELECOM

Contents

Introduction of incipient fault and method classification

Feature Description of Incipient Fault



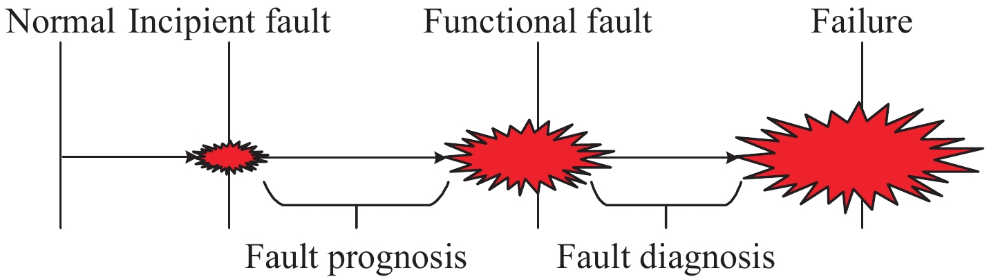


☐ Difficult Problem

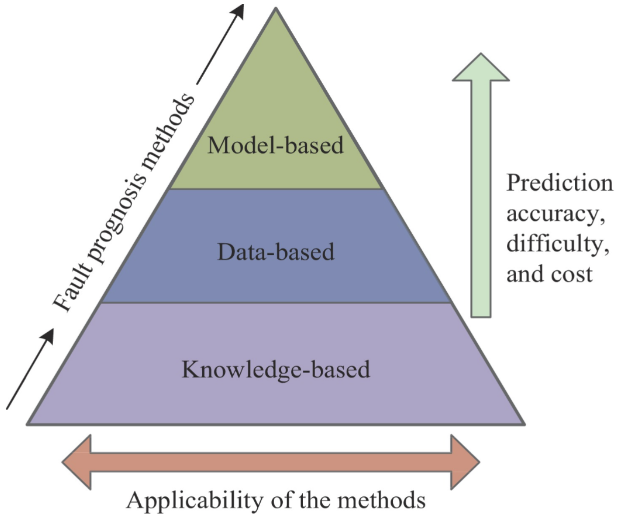
- ✓ Concealment, randomness, small deviation from normal state
- ✓ Computational complexity doubling and data redundancy in data sample dimension expansion
- ✓ Autocorrelation and cross correlation are mixed together and difficult to separate

Introduction of incipient fault and method classification

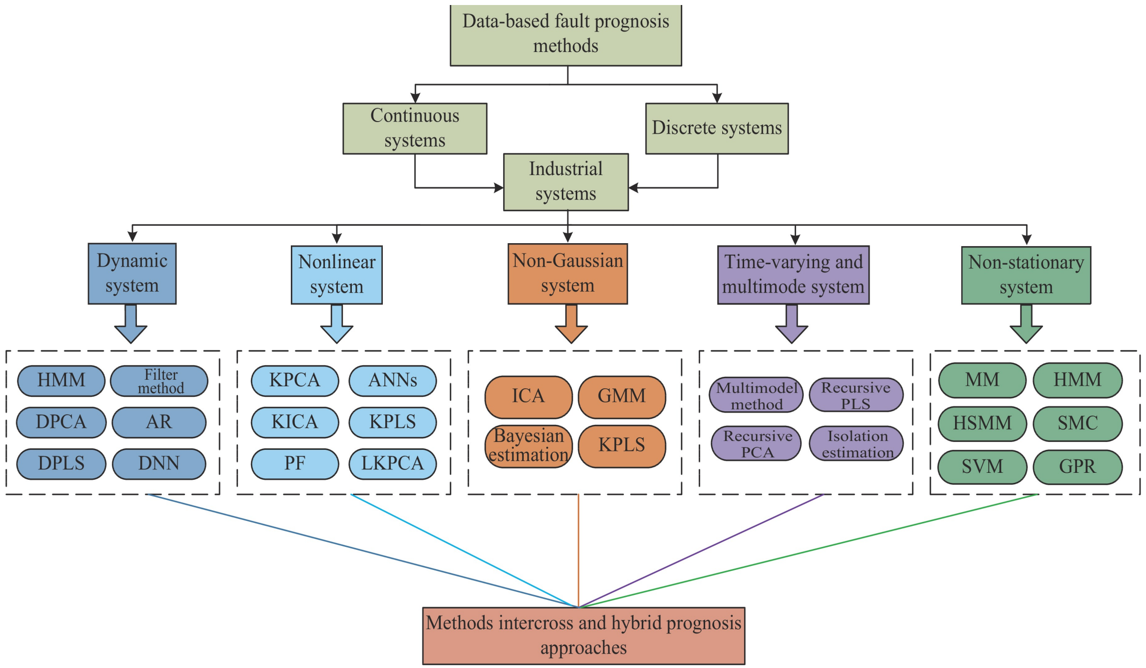
Incipient Fault Detection



Goal: Catch performance degradation early and maintain the reliability of cloud services



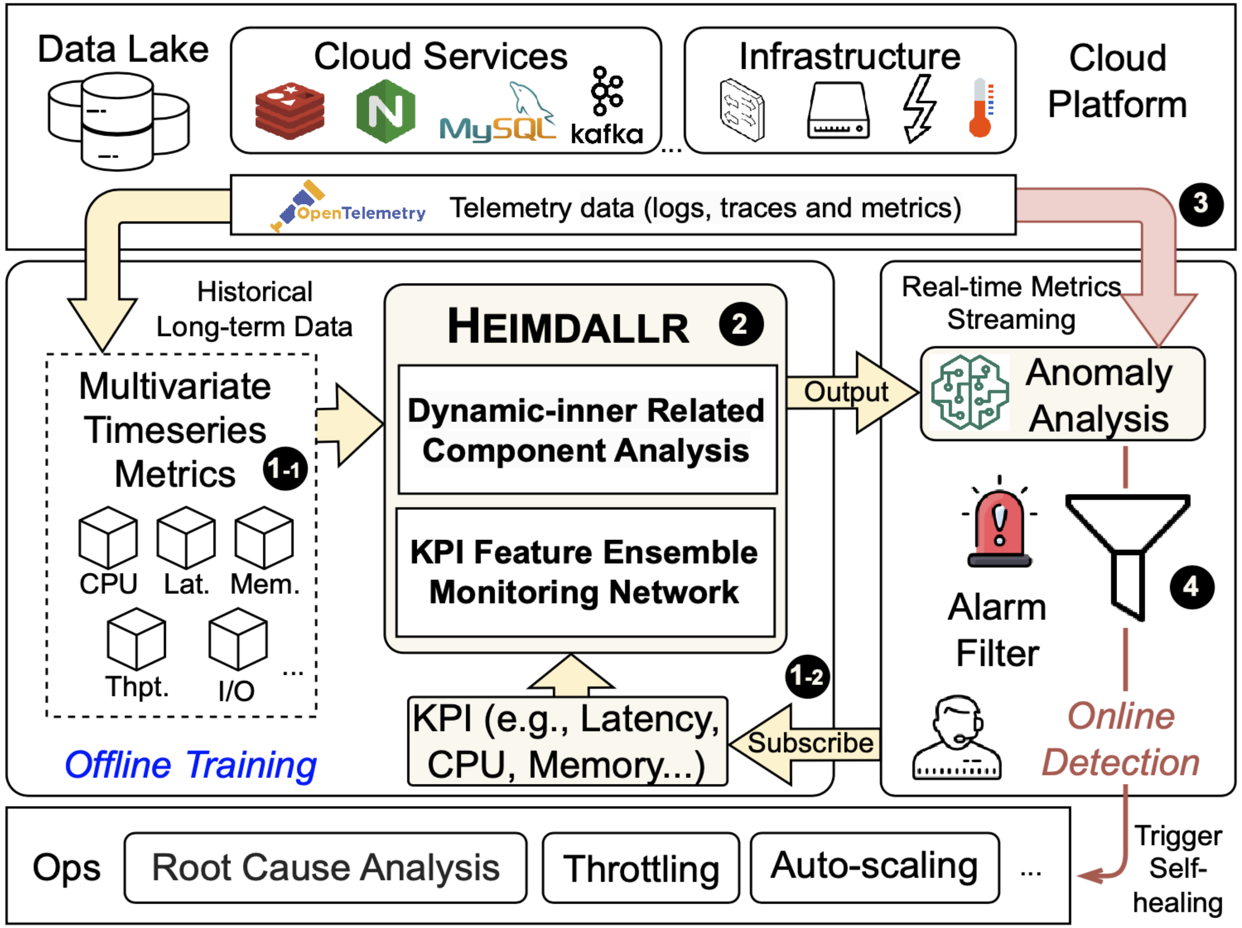
Method Classification



- Rely on oversimplified statistical assumptions
- Depend on large-scale supervised training
- Interference from noise
- The high-dimensional, correlated nature of data

HEIMDALLR Design

Overview of HEIMDALLR

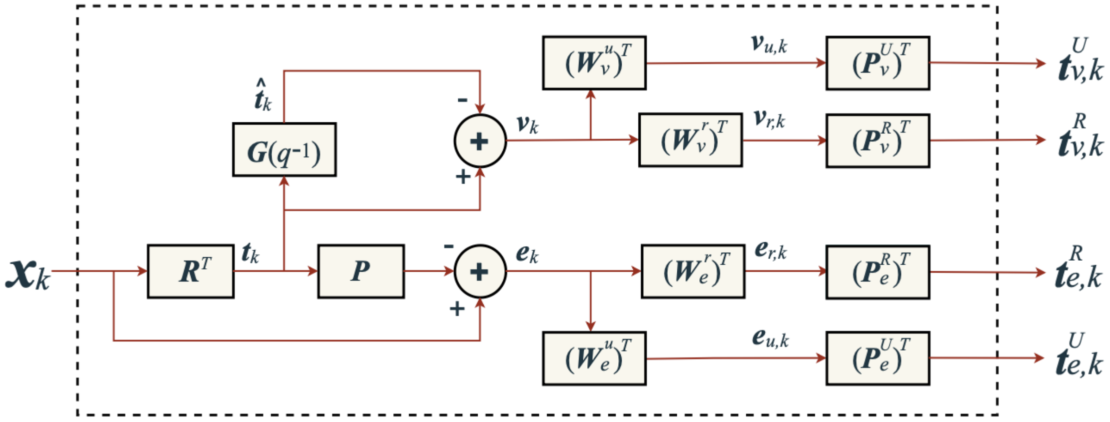


Key Features of HEIMDALLR

- MTS metrics collected from cloud services (e.g., Redis, Nginx) and datacenter infrastructure (e.g., temperature, power).
- HEIMDALLR receives historical data (step ❶-1) from data lakes, as well as KPI metrics (step ❶-2) curated by domain experts.
- In real-time, HEIMDALLR outputs processed metrics (step ❷) to the fault analysis module (step ❸), which detects potential faults.
- Detected anomalies are filtered by an alarm filter (step ❹) before triggering actions (RCA, power throttling, auto-scaling).

Temporal Correlation Mining

Algorithm Structural Diagram



- The dynamic and static residual:

$$\mathbf{v}_k = \mathbf{t}_k - \hat{\mathbf{t}}_k = \mathbf{R}^T\mathbf{x}_k + \sum_{i=1}^{s} \mathbf{\Theta}_i^T \mathbf{t}_{k-i}$$

$$= \mathbf{R}^T\mathbf{x}_k + \mathbf{G}\left(q^{-1}\right)\mathbf{t}_k$$

$$\mathbf{R} = \mathbf{W}\left(\mathbf{P}^T\mathbf{W}\right)^{-1},\ \mathbf{G}\left(q^{-1}\right) = \sum_{i=1}^{s} \mathbf{\Theta}_{s+1-i}^T q^{-i}$$

$$\mathbf{e}_k = \mathbf{x}_k - \mathbf{P}\mathbf{t}_k$$

- The internal model prediction for the k-th moment：

$$\begin{aligned} \hat{u}_k &= \mathbf{x}_k^T\mathbf{w}\beta_1 + \cdots + \mathbf{x}_{k-s+1}^T\mathbf{w}\beta_s \\ &= \begin{bmatrix} \mathbf{x}_k^T & \cdots & \mathbf{x}_{k-s+1}^T \end{bmatrix} (\beta \otimes \mathbf{w}) \end{aligned}$$

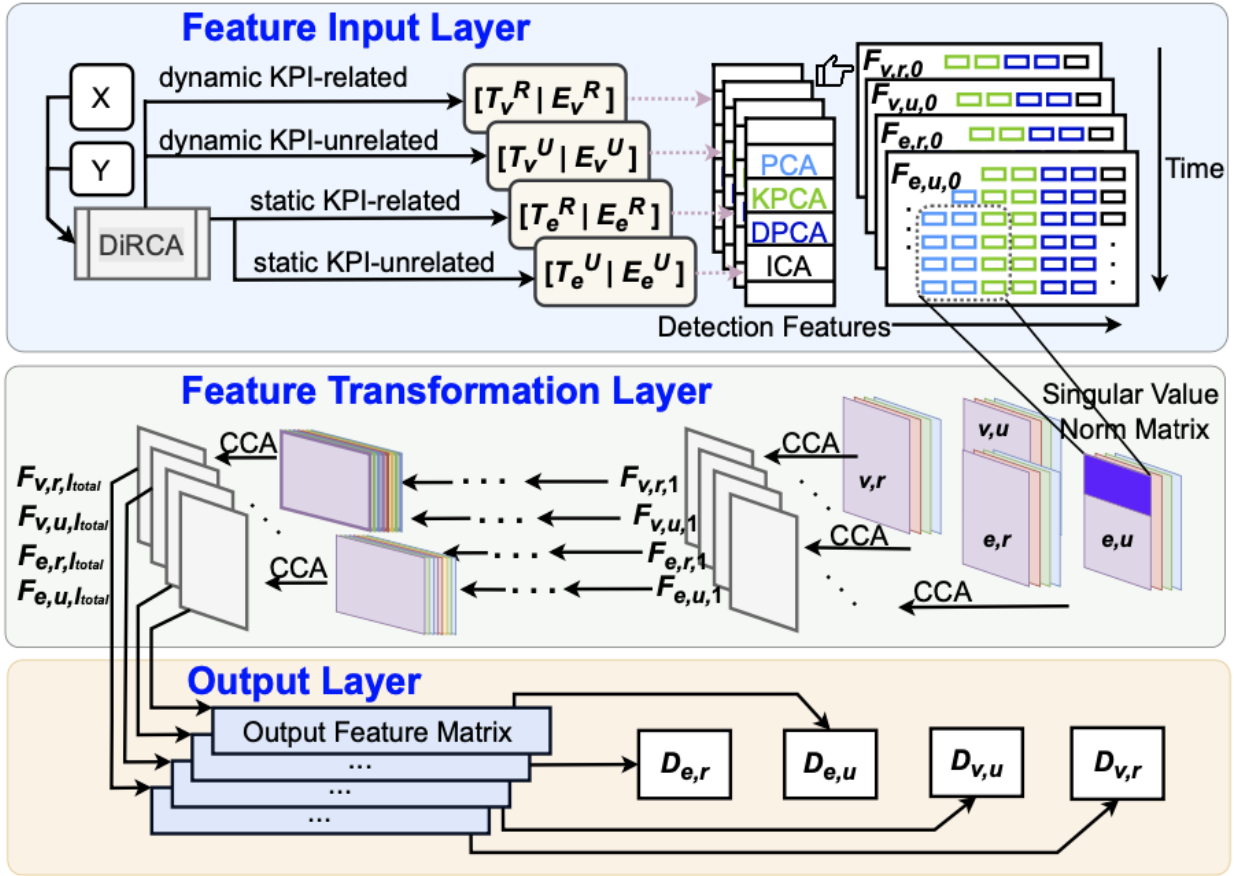
- The objective function：

$$\max_{\mathbf{q},\mathbf{w},\beta} \quad \mathbf{q}^T\mathbf{Y}_s^T\mathbf{Z}_s(\beta \otimes \mathbf{w})$$

$$\text{s.t.} \quad \|\boldsymbol{\beta}\| = 1,\ \|\boldsymbol{w}\| = 1,\ \|\boldsymbol{q}\| = 1$$

- Dynamic latent model is defined over the latent variables, which enables an effective separation of dynamic and static relationships.

KPI Feature Ensemble Monitoring Network (KFEMNet)



- The input feature matrix：

$$
\boldsymbol{F}_{v,r,l-1}^{c_p^l}=\begin{bmatrix}(\boldsymbol{F}_{v,r,l-1})_{1,c_p^l(1)} & \cdots & (\boldsymbol{F}_{v,r,l-1})_{1,c_p^l(g_l)}\\ (\boldsymbol{F}_{v,r,l-1})_{2,c_p^l(1)} & \cdots & (\boldsymbol{F}_{v,r,l-1})_{2,c_p^l(g_l)}\\ \vdots & \ddots & \vdots\\ (\boldsymbol{F}_{v,r,l-1})_{n_{l-1},c_p^l(1)} & \cdots & (\boldsymbol{F}_{v,r,l-1})_{n_{l-1},c_p^l(g_l)}\end{bmatrix}
$$

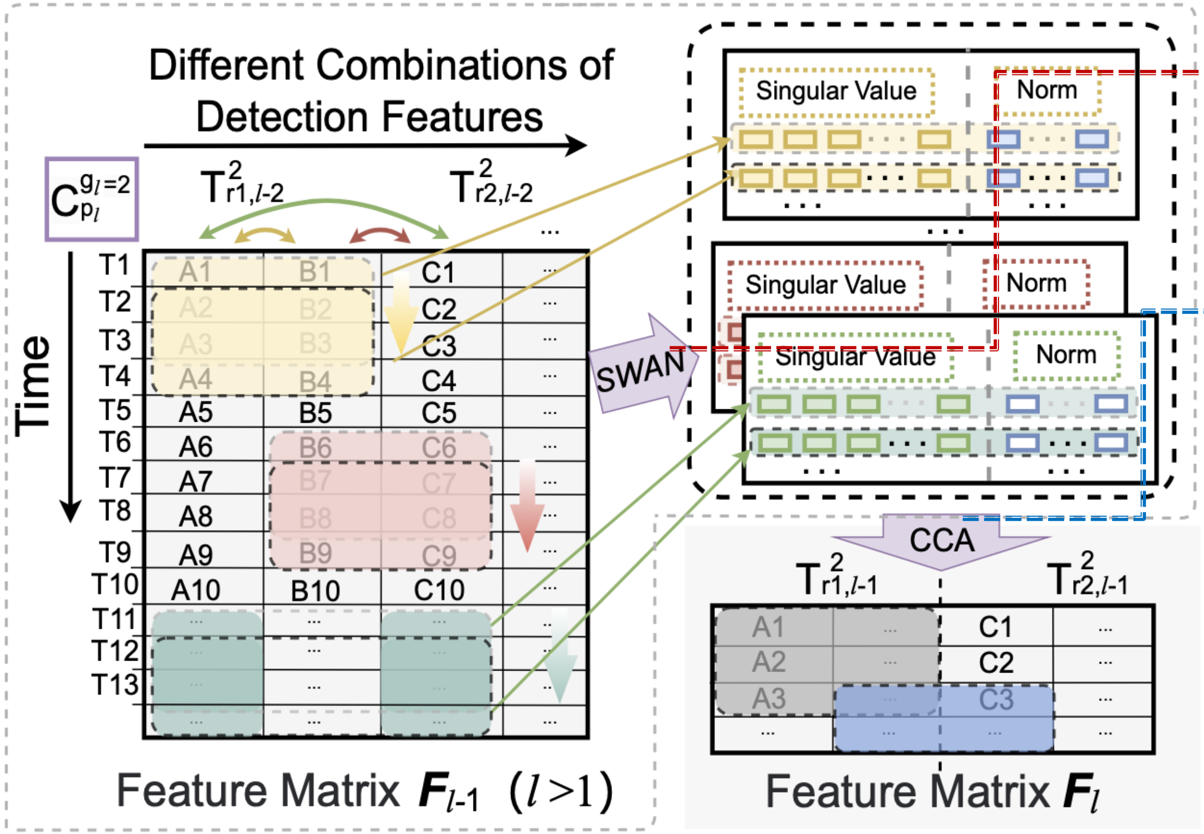
- The detection statistic：

$$
D_{v,r,k}=\left\|\Psi_{v,r}^{-1}\left(\boldsymbol{\sigma}_{v,r,k}-\bar{\boldsymbol{\sigma}}_{v,r}\right)\right\|_{\rho}
$$

- KFEMNet is constructed for the four subspaces to further **mine deep information**.

KPI Feature Ensemble Monitoring Network (KFEMNet)

Feature Transformation Layer



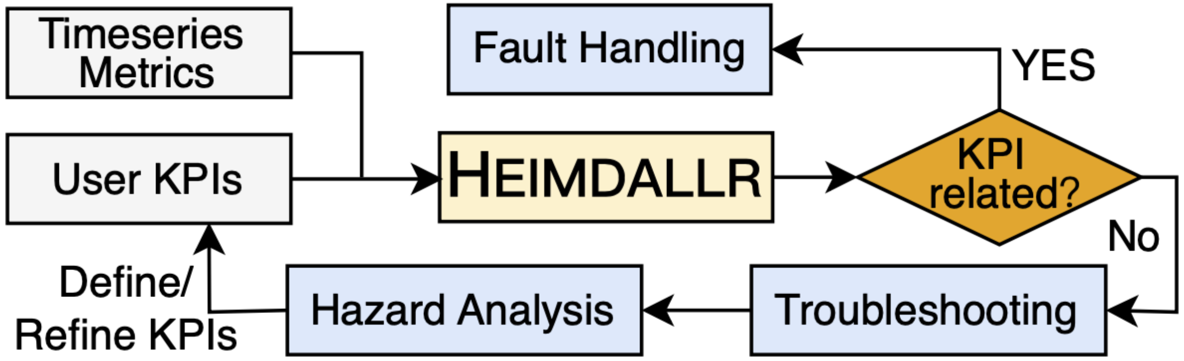
$$\gamma^{c_p^l}_{v,r,l-1}(k)=\Big[\sigma^{c_p^l}_{v,r,1}(k),\sigma^{c_p^l}_{v,r,2}(k),\cdots,\sigma^{c_p^l}_{v,r,g_l}(k),\ \|\bar{F}^{c_p^l}_{v,r,l-1}(k)\|_1,\|\bar{F}^{c_p^l}_{v,r,l-1}(k)\|_\infty\Big]^T\in\mathbb{R}^{g_l+2}$$

$$(\varsigma_i,\vartheta_i)=\underset{\varsigma_i^T\Sigma_u\varsigma_j=0,\vartheta_i^T\Sigma_r\vartheta_j=0,i\neq j}{\arg\max}\frac{\varsigma_i^T\Sigma_{ur}\vartheta_i}{\sqrt{\varsigma_i^T\Sigma_u\varsigma_i\cdot\vartheta_i^T\Sigma_r\vartheta_i}}$$

The **fault monitoring logic** of HEIMDALLR：

- If $D_{v,r}$ or $D_{e,r}$ exceeds the limit, a KPI-related fault is detected.
- If $D_{v,r}$ and $D_{e,r}$ are within the limit, but $D_{v,u}$ or $D_{e,u}$ exceeds the limit, a KPI-unrelated fault is detected.
- If none of the detection statistics exceed the limit, no fault is detected.

The flowchart of HEIMDALLR



Main points:

1. Unsupervised training – uses only normal data, no need for predefined fault models.
2. Beyond supervised limits – avoids reliance on scarce labeled fault data.
3. KPI-related vs. KPI-unrelated faults – addresses a key real-world challenge.
4. Iterative KPI refinement – detects hidden issues and improves KPI design over time.

Computational complexity analysis

Method	Indicators	Single-point computational complexity (simplified)
DPLS	2	$\mathcal{O}(mda + a^2 + mr^2)$
DiPLS	2	$\mathcal{O}(ma + a^2 + m^2)$
DTPLS	4	$\mathcal{O}(2mda + 2ma^2 + 2m^2 + ma + a^2)$
DTPLS$_{+net}$	2	$\mathcal{O}(2wm^2 + 2wma + 2w \log w)$
HEIMDALLR$_{-net}$	4	$\mathcal{O}(4ma^2 + 4ma + 4m^2)$
HEIMDALLR	4	$\mathcal{O}(4wmda + 4wma + 4wm^2 + 4w \log w)$

Symbol description: m – number of process variables, a – number of latent variables, d – time delay, w – sliding window width.

Comparison of computational complexity of each method for **single point online detection.**

HEIMDALLR adds modest online cost but maintains real-time efficiency while delivering **superior accuracy** and **reliability**.

Performance Metrics

- Three main **detection performance evaluation indicators**：

$$\mathrm{FDR} = N_{\text{alarm, fault}} / N_{\text{sample, fault}} \times 100$$
$$\mathrm{FAR} = N_{\text{alarm, normal}} / N_{\text{sample, normal}} \times 100$$
$$\mathrm{FMR} = N_{\text{alarm, wrong-type}} / N_{\text{sample, fault}} \times 100$$

Symbol	**Meaning (Note: "#" denotes "number of".)**
$N_{\text{alarm, fault}}$	# alarms in fault period
$N_{\text{sample, fault}}$	# samples in fault period
$N_{\text{alarm, normal}}$	# alarms in normal period
$N_{\text{sample, normal}}$	# samples in normal period
$N_{\text{alarm, ,wrong-type}}$	# alarms incorrectly triggered as another type of fault

Variable definitions for FDR, FAR, and FMR.

From a practical perspective, an ideal KPI-related **fault monitoring algorithm** should possess the following capabilities

- High FDR when faults occur.
- Low FAR when no faults occur, as well as low FMR of fault types.
- Good tracking ability for KPI changes.

Synthetic Experiments

- A synthetic dataset designed to mimic complex system behaviors.

$$\begin{cases} \boldsymbol{t}_k = \mathbf{A}\boldsymbol{t}_{k-1} + \boldsymbol{c} + \boldsymbol{v}_k \\ \boldsymbol{x}_k = \mathbf{P}_{inner}\boldsymbol{t}_k + \boldsymbol{e}_k \\ \boldsymbol{y}_k = \mathbf{B}_1\boldsymbol{t}_k + \mathbf{B}_2\boldsymbol{e}_k + \boldsymbol{f}_k \end{cases}$$

✓ Case 1: Starting from the 1001st sampling, t_k experiences **a ramp drift** with a slope of 0.002.

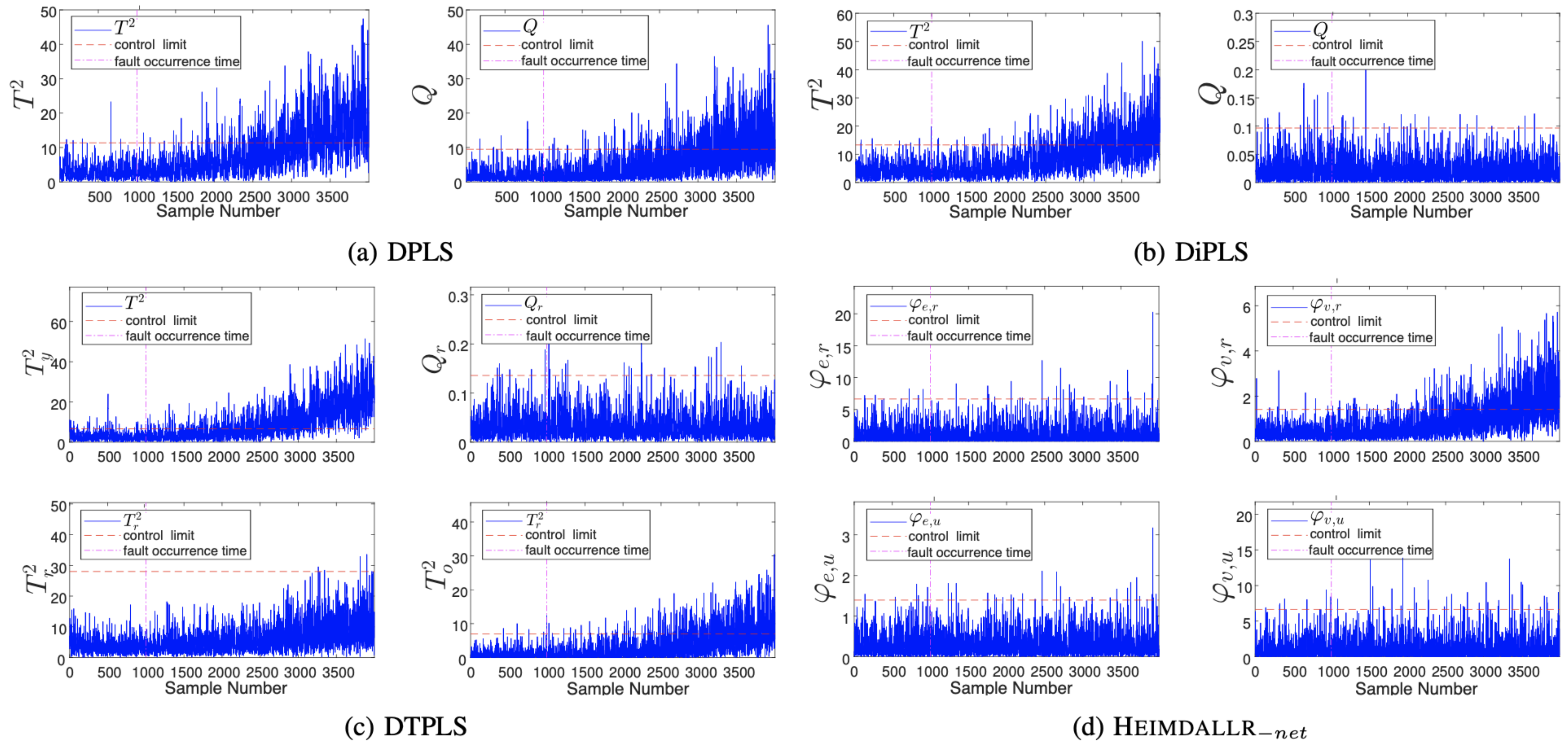
✓ Case 2: After the 1001st sampling moment, a fault that occurs in the static relationship part is introduced.

FARs(%) , FDRs(%) and FMR(%) of Different Methods in Case 1 of Synthetic Dataset.

Methods	Statistics	FARs (%)	FDRs (%)
DPLS	T^2	0.50	29.24
DiPLS	T^2	0.90	21.90
DTPLS	T_y^2	6.81	42.37
	Q_r	0.80	0.83
DTPLS$_{+net}$	D_r	0.90	28.88
HEIMDALLR$_{-net}$	$\varphi_{e,r}$	0.80	0.87
	$\varphi_{v,r}$	0.80	24.33
HEIMDALLR	$D_{e,r}$	**0.00**	2.07
	$D_{v,r}$	**0.00**	**71.96**

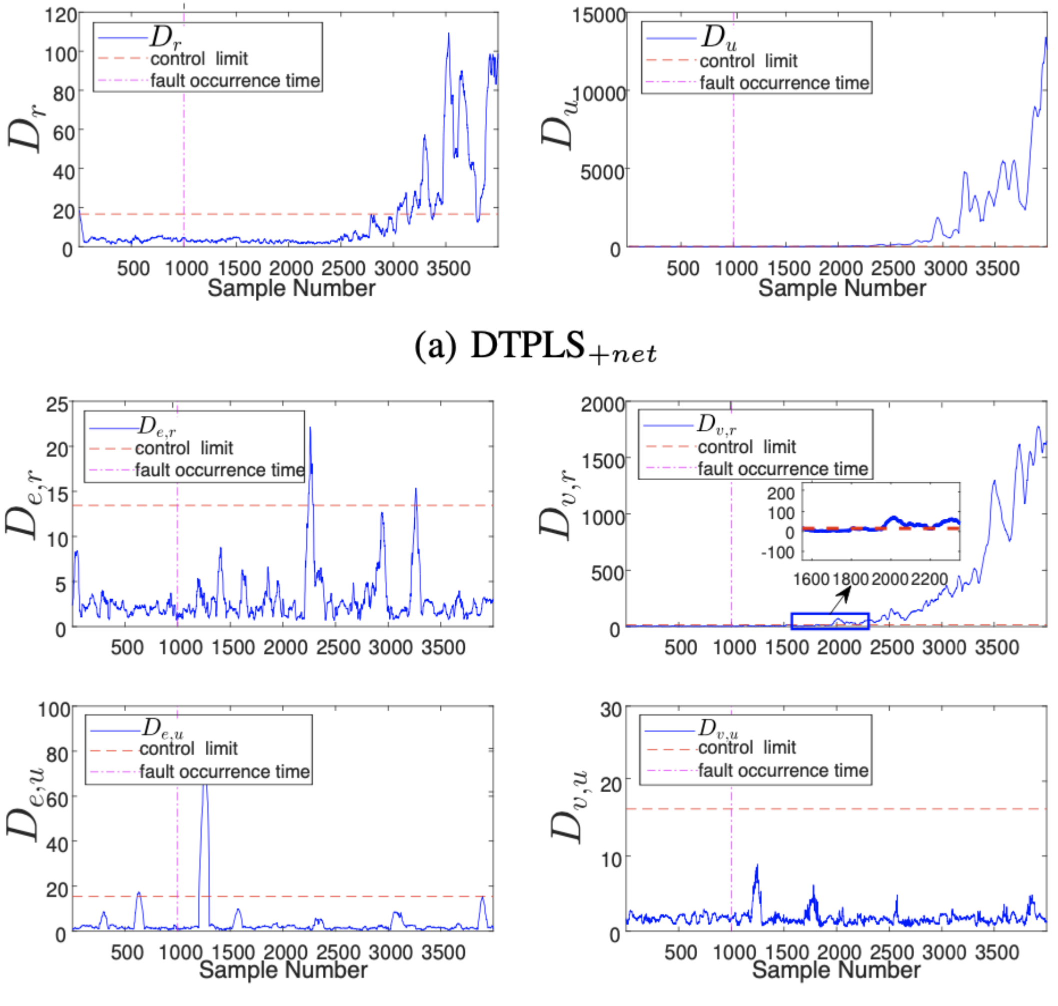
Case	DPLS	DiPLS	DTPLS	DTPLS $+net$	HEIMDALLR $-net$	HEIMDALLR
1	8.33	0.87	3.02	28.50	0.41	**0.12**

Synthetic Experiments



Monitoring Results of DPLS, DiPLS, DTPLS and HEIMDALLR$_{-net}$ in Case 1 of Synthetic Dataset.

Synthetic Experiments



(a) DTPLS$_{+net}$

(b) HEIMDALLR

FARs(%) , FDRs(%) and FMR(%) of Different Methods in Case 2 of Synthetic Dataset.

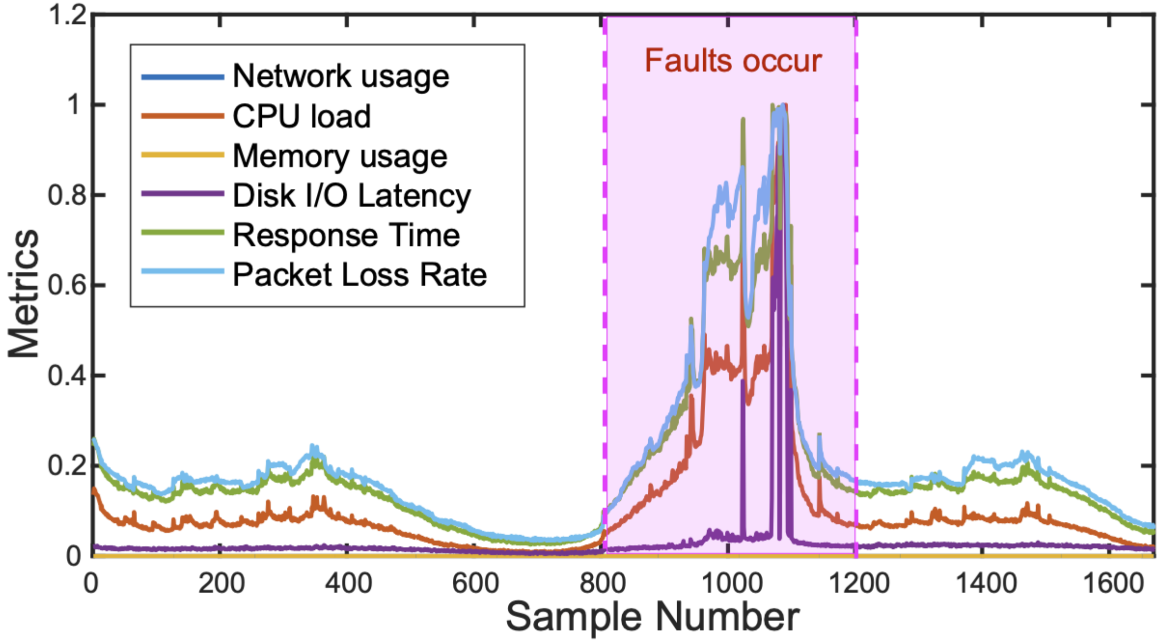
Methods	Statistics	FARs (%)	FDRs (%)
DPLS	Q	1.00	2.33
DiPLS	Q	1.00	4.97
DTPLS	T_r^2	0.70	0.70
	T_o^2	0.60	0.57
DTPLS$_{+net}$	D_u	2.40	78.06
HEIMDALLR$_{-net}$	$\varphi_{e,u}$	1.40	10.90
	$\varphi_{v,u}$	0.60	1.17
HEIMDALLR	$D_{e,u}$	3.00	**99.27**
	$D_{v,u}$	0.00	0.00

Case	DPLS	DiPLS	DTPLS		DTPLS $+net$	HEIMDALLR $-net$		HEIMDALLR	
	T^2	T^2	T_y^2	Q_r	D_r	$\varphi_{e,r}$	$\varphi_{v,r}$	$D_{e,r}$	$D_{v,r}$
2	0.63	2.73	7.73	1.77	40.51	1.50	1.53	0.00	4.93

Incipient FD Benchmark Platform

- **Server Machine Dataset (SMD)**

A public sensor-based time series dataset collected from the **real server machines** of a large Internet company.



The trend chart of the 6 merics over 27 hours of SMD.

FARs(%) , FDRs(%) and FMR(%) of Different Methods for SMD.

Methods	Statistics	FARs (%)	FDRs (%)
DPLS	T^2	1.42	58.36
DiPLS	T^2	0.00	64.95
DTPLS	T_y^2	2.47	17.30
	Q_r	0.00	67.86
DTPLS$_{+net}$	D_r	9.89	88.43
HEIMDALLR$_{-net}$	$\varphi_{e,r}$	0.48	60.97
	$\varphi_{v,r}$	1.24	1.93
HEIMDALLR	$D_{e,r}$	0.00	**94.38**
	$D_{v,r}$	0.00	52.41

DPLS	DiPLS	DTPLS	DTPLS $+net$	HEIMDALLR $-net$	HEIMDALLR
8.25	9.41	0.29	3.16	1.25	**0.00**

Conclusions

- Timely KPI trend monitoring is crucial for cloud reliability, yet early subtle anomalies remain extremely difficult to detect amid noise and high-dimensional correlations.
- HEIMDALLR is designed to capture temporal interdependencies in MTS data and to enhance interpretability.
- HEIMDALLR achieves heightened sensitivity to incipient faults, enabling effective KPI-related incipient FD without the need for labeled data.
- Extensive empirical validation confirms its superior detection performance and robustness across diverse scenarios.

Thank you！
Any questions?

Email: wangyw22@chinatelecom.cn