

<https://doi.org/10.31891/csit-2026-3-18>

Svitlana POPERESHNYAK

PhD in Physics and Mathematics, Associate Professor, Associate Professor at the Department of Computer Science and Software Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”
<https://orcid.org/0000-0002-0531-9809>
e-mail: spopereshnyak@gmail.com

Received: 31/07/2026

Accepted: 02/09/2026

Published: 30/09/2026

© Copyright
2026 by the author(s)



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

UDC 004.052.2

EARLY WARNING OF CRITICAL INFORMATION SYSTEM STATES FROM SHORT SEQUENCES

The reliability of modern information systems largely depends on the timely detection of changes that may lead to performance degradation, violation of established response-time requirements, and service unavailability. Conventional monitoring tools primarily register the exceedance of predefined thresholds but do not always make it possible to recognize the preceding development of an adverse trend and distinguish it from a short-term deviation. The objective of this study is to develop an approach to the early warning of critical information system states based on the sequential aggregation of structural estimates obtained from short monitoring-data sequences. The article proposes a procedure for normalizing heterogeneous indicators, forming binary sequences within sliding time windows, and calculating local structural estimates that characterize the frequency, fragmentation, and persistence of adverse observations. To detect developing degradation, the local estimates are sequentially aggregated with consideration of the current deviation level, trend, accumulation, duration, and variability. Based on the resulting aggregated estimate, information-response rules are developed using activation thresholds, confirmation intervals, and hysteresis during system recovery. The proposed approach distinguishes stable operation, an isolated deviation, gradually developing degradation, and a persistent critical state. A simulation-based evaluation was conducted for the corresponding server-infrastructure workload scenarios. The results showed that the sequential aggregation of structural estimates enables persistent adverse changes to be detected before established quality-of-service requirements are violated while limiting responses to isolated short-term fluctuations. Comparison with conventional threshold-based monitoring confirmed the possibility of providing a time interval for preventive actions, including resource scaling, workload redistribution, and restriction of noncritical requests. The proposed procedure can be integrated into information technologies for monitoring server, distributed, and cloud systems and used to support decision-making aimed at preventing critical operating conditions.

Keywords: information technology; forecasting of hazardous states; short random sequences; structural-probabilistic assessment; sequential aggregation; early warning; information systems; system-state monitoring.

Introduction

The reliability of modern information systems largely depends on the operational stability of their server infrastructure. Increasing workloads, limited computational resources, and the accumulation of processing errors may gradually increase response times, reduce system performance, and cause partial or complete service unavailability. The timely identification of developing critical states is therefore essential for maintaining the availability and continuity of information services.

Conventional monitoring approaches generally compare current server metrics with predefined thresholds. Although such approaches can detect an existing violation, they provide limited information about its preceding development. Moreover, an isolated deviation may result from a temporary workload surge and does not necessarily indicate system degradation. A critical state is usually preceded by a sequence of coordinated changes whose individual values may initially remain within acceptable limits.

This limitation makes it necessary to move from detecting separate anomalies to identifying adverse trends before a critical state occurs. Short data sequences allow changes to be assessed with minimal delay, while the sequential aggregation of their structural estimates makes it possible to consider the

direction, accumulation, duration, and variability of observed deviations. Therefore, developing and evaluating an approach under simulated degradation scenarios for the early warning of critical information system states from short sequences of server monitoring data is a relevant scientific and applied problem.

Statement of the problem

Modern information systems generate continuous streams of server monitoring data, but the availability of these data does not itself ensure the timely recognition of developing critical states. Conventional monitoring tools mainly detect threshold violations or already formed anomalies, whereas preceding adverse changes may remain unnoticed. As a result, preventive action is often taken only after system performance has substantially deteriorated.

The use of short data sequences is necessary when a decision must be made with minimal delay. However, many statistical methods require sufficiently long samples and assumptions about the distribution or stationarity of the observed process. Data-driven methods may additionally require representative training sets containing labelled examples of different degradation scenarios. These requirements are difficult to satisfy when the system state changes rapidly and only a limited number of recent observations is available.

Individual estimates obtained from short sequences are also sensitive to random fluctuations and may not reliably distinguish a temporary deviation from a developing degradation process. Therefore, the problem is to identify coordinated and persistent changes across successive short sequences and provide an early warning before server performance reaches a critical level.

Analysis of recent studies and publications

Anomaly detection in server and cloud infrastructures is commonly based on metrics, logs, and traces. Existing statistical, machine-learning, and deep-learning methods primarily focus on recognizing anomalous behaviour and identifying its causes after observable deviations have emerged [1]. A recent systematic mapping of 104 publications confirms the growing use of operational monitoring data for detecting performance degradation and failures in cloud-based systems [2]. However, differences in data acquisition, preprocessing, and experimental conditions complicate the comparison and practical implementation of existing approaches.

Recent studies increasingly represent server monitoring indicators as multivariate time series. Deep-learning approaches use reconstruction, prediction, and representation models to capture complex temporal dependencies [3, 4]. However, such models often require extensive training data, parameter configuration, and considerable computational resources. Online model-based methods also face unresolved problems related to representative datasets, evaluation metrics, and decision-threshold selection [5]. These limitations become particularly important when a system state changes rapidly and a decision must be made from a limited number of recent observations.

Active learning and contrastive variational autoencoders have been applied to account for relationships among multiple indicators in large distributed systems [6]. Online unsupervised methods can process multivariate streams without preliminary labelled training sets [7]. The integration of time-series anomaly detection into DevOps workflows has also confirmed the practical value of monitoring data while demonstrating a strong dependence of detection performance on the selected algorithm, its parameters, and the properties of the monitored system [8]. Nevertheless, these methods mainly classify current observations or intervals as normal or anomalous rather than identify the gradual formation of a critical state.

The reliability of experimental evaluation remains another important issue. Commonly used point-adjustment procedures may substantially overestimate anomaly-detection performance [9]. Moreover, high classification accuracy does not necessarily mean that an anomaly is detected early enough for preventive action [10]. Therefore, early-warning methods should be evaluated not only by accuracy, precision, recall, and F1-score but also by false warning frequency and the time interval between the warning and the actual critical state.

Structural-probabilistic analysis provides an alternative basis for processing limited observations. Exact probabilistic characteristics have been obtained for statistics of short binary sequences and their joint distributions [11, 12]. However, local estimates calculated for separate short sequences do not directly characterize the direction, accumulation, and duration of changes across successive observation windows. Consequently, there remains a need for an approach that sequentially aggregates local structural estimates, distinguishes isolated deviations from developing degradation, and provides an early warning before server performance reaches a critical level.

Research objectives and tasks

The objective of this study is to develop and evaluate an approach for the early warning of critical information system states based on the sequential aggregation of structural estimates obtained from short server-monitoring sequences. To achieve this objective, the following tasks are addressed:

- to define the components of the sequentially aggregated assessment;
- to establish decision rules for distinguishing stable operation, isolated deviations, developing degradation, and persistent degradation;
- to implement the early-warning procedure using server performance indicators and construct a simulation model with representative degradation scenarios;

- and to compare the proposed approach with conventional threshold-based monitoring in terms of detection accuracy, false warnings, and warning lead time.

Research results

FORMATION OF STRUCTURAL ESTIMATES FROM SHORT SEQUENCES

Let the operational state of the server infrastructure at discrete time t be represented by a vector of monitoring indicators

$$\mathbf{x}(t) = \{x_1(t), x_2(t), \dots, x_m(t)\},$$

where the components may include CPU utilization, memory utilization, response time, request-processing rate, and error rate. Since these indicators have different units and ranges, each indicator is transformed into a dimensionless deviation from its reference value:

$$z_j(t) = d_j \frac{x_j(t) - \mu_j}{\sigma_j}, j = 1, \dots, m,$$

where μ_j and σ_j are the mean value and standard deviation obtained during stable server operation, while $d_j \in \{-1, 1\}$ determines the adverse direction of change. For indicators whose increase denotes deterioration, $d_j = 1$; for indicators whose decrease denotes deterioration, $d_j = -1$. Consequently, positive values of $z_j(t)$ correspond to an adverse deviation regardless of the physical meaning of the indicator.

The monitoring indicators selected for the experimental assessment characterize the use of the principal server resources and the resulting quality of request processing. Their designation, adverse direction of change, and operational interpretation are presented in Table 1.

Table 1.

Server monitoring indicators used for structural assessment

Indicator	Designation	Adverse change	Operational meaning
CPU utilization	$x_1(t)$	Increase	Growth of computational load
Memory utilization	$x_2(t)$	Increase	Accumulation of occupied memory
Response time	$x_3(t)$	Increase	Deterioration of service performance
Request-processing rate	$x_4(t)$	Decrease	Reduction in server throughput
Error rate	$x_5(t)$	Increase	Growth in unsuccessful operations

As shown in Table 1, an increase is considered adverse for most of the selected indicators. Request-processing rate is an exception because its decrease indicates a reduction in server throughput. The direction coefficient d_j makes it possible to represent these heterogeneous changes uniformly: a positive normalized deviation always corresponds to a deterioration in the monitored system state.

The normalized observations are processed within successive short sliding windows. Let $r = 1, 2, \dots, R$ denote the window number, and let t_r be the discrete time index of the first observation included in the r -th window. Then the r -th window is defined as

$$W_r = \{t_r, t_{r+1}, \dots, t_{r+n-1}\},$$

where n is the number of observations within each window and R is the total number of windows. The starting index of the next window is determined as $t_{r+1} = t_r + h$,

where h is the sliding step measured in observations. Thus, for $h < n$, two successive windows overlap by $n - h$ observations. If monitoring data are collected at a constant sampling interval Δt , the time duration of one window is $(n - 1)\Delta t$, while a new structural estimate is produced every $h\Delta t$.

The use of short windows reduces the delay between the onset of an adverse change and its detection. However, an individual observation within such a window is insufficient to distinguish a random fluctuation from a developing degradation process. Therefore, the normalized observations within each window are transformed into a binary sequence and subsequently described by a set of local structural estimates:

$$b_{j,r}(i) = \begin{cases} 1, & z_j(t_r + i - 1) > \theta_j, \\ 0, & z_j(t_r + i - 1) \leq \theta_j, \end{cases} i = 1, \dots, n,$$

where θ_j is the admissible local deviation level. A value of 1 indicates an adverse observation, whereas a value of 0 indicates that the corresponding observation remains within the accepted range.

The internal arrangement of adverse observations is characterized by three local structural estimates. The first estimate represents their relative frequency:

$$q_{j,r} = \frac{1}{n} \sum_{i=1}^n b_{j,r}(i).$$

The second estimate characterizes the number of transitions between acceptable and adverse observations:

$$u_{j,r} = \frac{1}{n-1} \sum_{i=2}^n |b_{j,r}(i) - b_{j,r}(i-1)|.$$

The third estimate represents the relative length of the longest uninterrupted adverse fragment:

$$g_{j,r} = \frac{l_{j,r}^{\max}}{n},$$

where $l_{j,r}^{\max}$ is the maximum number of consecutive unit values in the binary sequence. The estimate $q_{j,r}$ characterizes the overall presence of adverse observations, $u_{j,r}$ reflects their fragmentation, and $g_{j,r}$ describes their local persistence.

These estimates make it possible to distinguish sequences containing the same number of adverse observations but having different internal structures. For example, the sequences 10101000 and 11100000 have the same value $q_{j,r} = 0.375$. However, the first sequence contains alternating isolated deviations, whereas the second contains a continuous adverse fragment. Therefore, their transition and persistence estimates differ, indicating different patterns of system behaviour.

For each monitoring indicator, a local structural-estimate vector is formed:

$$\mathbf{e}_{j,r} = (q_{j,r}, u_{j,r}, g_{j,r}).$$

The estimates obtained for all m indicators are combined into the structural representation of the server state within window W_r :

$$\mathbf{E}_r = \{\mathbf{e}_{1,r}, \mathbf{e}_{2,r}, \dots, \mathbf{e}_{m,r}\}.$$

Thus, each observation window is represented not only by the magnitudes of server indicators but also by the frequency, fragmentation, and persistence of adverse changes. The sequence $\mathbf{E}_1, \mathbf{E}_2, \dots$ forms the input for the subsequent aggregation procedure, which determines whether the identified changes are isolated or constitute a developing degradation trend.

The complete procedure for transforming the initial monitoring data into local structural estimates is shown in Fig. 1. It includes the normalization of heterogeneous indicators, the formation of successive short windows, binary representation of adverse observations, and calculation of structural characteristics.

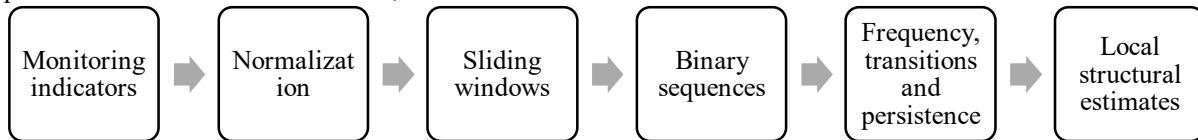


Fig. 1. Formation of structural estimates from a short server-monitoring sequence

As illustrated in Fig. 1, the transformation is performed separately for each monitoring indicator, whereas the resulting local estimates are subsequently combined into the structural representation $\mathbf{E}_r$ of the server state. This representation retains information about the frequency, fragmentation, and persistence of adverse observations within each short window. The sequence of vectors $\mathbf{E}_1, \mathbf{E}_2, \dots$ constitutes the input data for the sequential aggregation procedure described in the next subsection.

SEQUENTIAL AGGREGATION OF LOCAL STRUCTURAL ESTIMATES

The structural estimates obtained within one window characterize the current arrangement of adverse observations but do not indicate whether the detected change is isolated or persists over time. Therefore, the local estimates are first combined within each window and then sequentially aggregated across successive windows.

For the j -th monitoring indicator, the local structural score is defined as

$$a_{j,r} = \alpha_q q_{j,r} + \alpha_g g_{j,r} + \alpha_c q_{j,r}(1 - u_{j,r}),$$

where $\alpha_q, \alpha_g, \alpha_c \geq 0$ and $\alpha_q + \alpha_g + \alpha_c = 1$.

The components $q_{j,r}$ and $g_{j,r}$ represent the frequency and persistence of adverse observations, respectively. The term $q_{j,r}(1 - u_{j,r})$ characterizes their structural coherence. Multiplication by $q_{j,r}$ prevents a sequence containing no adverse observations from obtaining a high coherence score merely because it has no transitions.

The local scores for all monitoring indicators are combined into the current structural state estimate:

$$A_r = \sum_{j=1}^m \omega_j a_{j,r},$$

where $\omega_j \geq 0$ and $\sum_{j=1}^m \omega_j = 1$.

Thus, $A_r \in [0,1]$: values close to zero correspond to stable operation, whereas increasing values indicate a greater frequency, coherence, and persistence of adverse observations. Equal weights may be used when the relative importance of the monitoring indicators has not been established.

Because an isolated workload surge may also produce a high value of A_r , the ordered sequence

$$A_1, A_2, \dots, A_r$$

is used to calculate five temporal components. Their meanings are summarized in Table 2.

Table 2.

Components of the sequentially aggregated assessment			
Component	Designation	Characterized property	Influence on S_r
Current level	L_r	Severity of adverse changes in the current window	Positive
Trend	T_r	Growth of estimates across recent windows	Positive
Accumulation	C_r	Repeated relevant deviations considering previous windows	Positive
Duration	D_r	Continuity of the adverse state	Positive
Variability	V_r	Instability of recent structural estimates	Negative

As shown in Table 2, the level, trend, accumulation, and duration components provide complementary evidence of developing degradation. Variability is used as a penalty because irregular changes are more likely to correspond to temporary fluctuations than to a stable adverse process.

The current-level component is equal to the estimate obtained for the current window: $L_r = A_r$.

The trend component represents the mean positive change over the last p transitions:

$$T_r = \frac{1}{p} \sum_{k=0}^{p-1} \max(0, A_{r-k} - A_{r-k-1}), r > p.$$

Only positive differences contribute to T_r , since a decrease in the structural estimate does not indicate further degradation.

The accumulation component is updated recursively:

$$C_r = \min \left[1, \rho C_{r-1} + (1 - \rho) \frac{\max(0, A_r - \theta_0)}{1 - \theta_0} \right], C_0 = 0,$$

where $\rho \in [0,1]$ is the memory coefficient and $\theta_0 \in [0,1]$ is the minimum relevant level of the current structural estimate. Repeated values above θ_0 increase C_r , whereas a return to stable operation gradually reduces the accumulated effect.

To characterize the continuity of adverse changes, the number of consecutive windows exceeding θ_0 is determined as

$$d_r = \begin{cases} d_{r-1} + 1, & A_r > \theta_0, \\ 0, & A_r \leq \theta_0, \end{cases} d_0 = 0.$$

The normalized duration component is

$$D_r = \min \left(1, \frac{d_r}{d_{\max}} \right),$$

where $d_{\max}$ is the number of consecutive adverse windows sufficient to obtain the maximum duration estimate.

Variability is calculated from the last $p + 1$ estimates:

$$V_r = \min \left[1, 2 \sqrt{\frac{1}{p+1} \sum_{k=0}^p (A_{r-k} - \bar{A}_r)^2} \right],$$

$$\text{where } \bar{A}_r = \frac{1}{p+1} \sum_{k=0}^p A_{r-k}.$$

Because $A_r \in [0,1]$, the coefficient 2 normalizes the standard deviation relative to its maximum value of 0.5. A high value of V_r indicates unstable alternation of the recent structural estimates.

The full temporal aggregation is activated after $p + 1$ window estimates have been formed. During initialization, the response level remains INFO; therefore, trend and variability are not evaluated from an incomplete history.

The resulting sequentially aggregated estimate is determined as

$$S_r = [\beta_L L_r + \beta_T T_r + \beta_C C_r + \beta_D D_r - \beta_V V_r]_0^1,$$

where

$$[x]_0^1 = \min(1, \max(0, x)), \beta_L, \beta_T, \beta_C, \beta_D, \beta_V \geq 0,$$

and the weights of the positive components satisfy

$$\beta_L + \beta_T + \beta_C + \beta_D = 1.$$

The coefficient β_V determines the penalty applied to unstable changes. The clipping operator ensures that $S_r \in [0,1]$.

The sequence of operations used to form S_r is shown in Fig. 2.

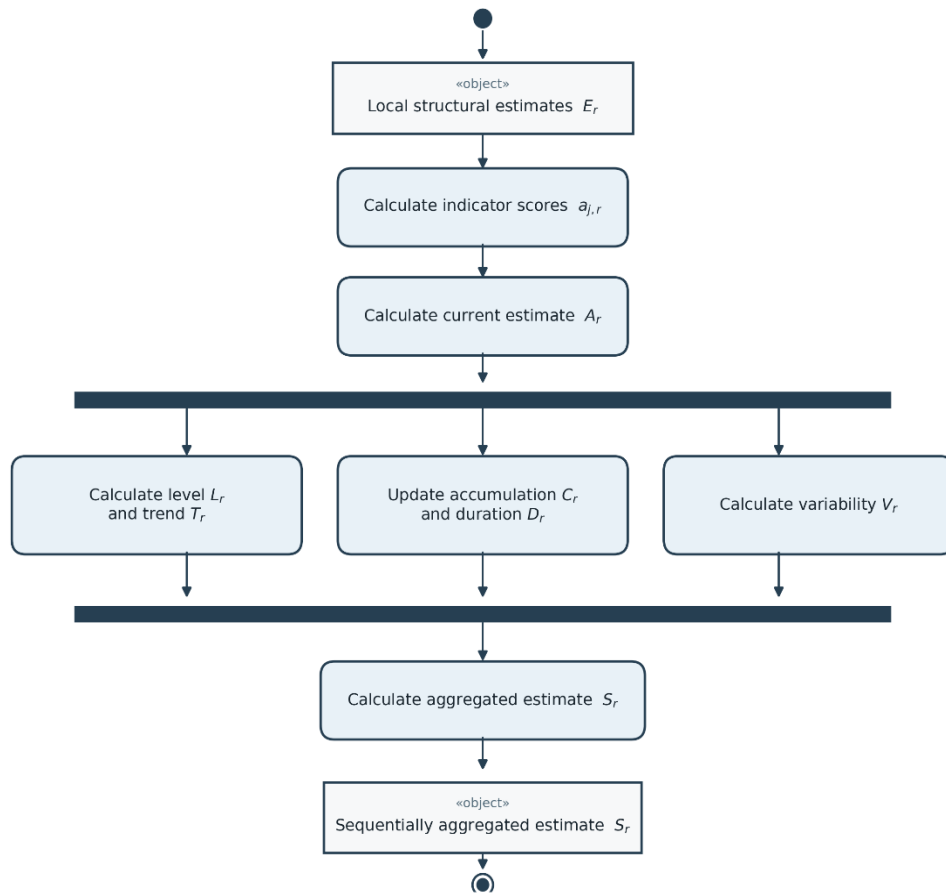


Fig. 2. Sequential aggregation of local structural estimates

As illustrated in Fig. 2, the first aggregation stage combines the structural characteristics of all monitoring indicators within the current window. The second stage accounts for the development of these estimates across successive windows. Therefore, an isolated increase in A_r does not necessarily result in a sustained increase in S_r , whereas coordinated and persistent adverse changes increase its trend, accumulation, and duration components.

The ordered sequence

$$S_1, S_2, \dots, S_R$$

forms the basis for distinguishing stable operation, temporary deviations, developing degradation, and persistent degradation using the decision procedure presented in the next subsection.

EARLY-WARNING DECISION PROCEDURE

The sequentially aggregated estimate S_r is transformed into three information response levels: INFO corresponds to stable operation or isolated deviations, WATCH indicates developing degradation, and ALERT represents a persistent critical tendency requiring intervention. The initial response level is INFO.

The activation and recovery thresholds satisfy

$$0 \leq \theta_W^{\text{off}} < \theta_W^{\text{on}} < \theta_A^{\text{off}} < \theta_A^{\text{on}} \leq 1.$$

Here, θ_W^{on} and θ_A^{on} activate WATCH and ALERT, whereas θ_W^{off} and θ_A^{off} provide hysteresis during recovery. A transition is confirmed only when its condition is satisfied for a specified number of consecutive windows: k_W for WATCH, k_A for ALERT, and k_R for recovery. The decision rules are presented in Table 3.

Table 3.

Decision rules for information response levels

Current level	Confirmed condition	New level
INFO	$S_r \geq \theta_W^{\text{on}}$ for k_W windows	WATCH
INFO or WATCH	$S_r \geq \theta_A^{\text{on}}$ for k_A windows	ALERT
WATCH	$S_r < \theta_W^{\text{off}}$ for k_R windows	INFO
ALERT	$S_r < \theta_A^{\text{off}}$ for k_R windows	WATCH
Any level	No transition condition is confirmed	Current level retained

If the WATCH and ALERT conditions are confirmed simultaneously, ALERT has priority. Consequently, rapid degradation may produce a direct transition from INFO to ALERT. Recovery is performed gradually to prevent unstable switching near the decision thresholds.

The warning time t_{warning} is defined as the time of the last observation in the window in which the WATCH condition is first confirmed. If the independently defined critical state begins at t_{critical} , the warning lead time is

$$T_{\text{lead}} = t_{\text{critical}} - t_{\text{warning}}.$$

A positive T_{lead} indicates that the developing degradation was detected before the service-level condition was violated.

SIMULATION SETUP AND DEGRADATION SCENARIOS

The proposed procedure was evaluated using model-generated monitoring sequences representing a server application with a reverse proxy, an application service, and a database. The conceptual environment comprises a workload-generation component, a simulated server host, a monitoring component, and an analysis component. CPU utilization, memory utilization, 95th-percentile response time, request-processing rate, and error rate were generated at a constant sampling interval and processed by the proposed procedure. The information flows are shown in Fig. 3.

For indicator j , scenario s , and realization ℓ , each simulated observation was obtained by combining the stable reference level μ_j , the deterministic scenario-dependent component $\Delta_{j,s}(t)$, and a zero-mean random disturbance $\varepsilon_j^{(\ell)}(t)$ with standard deviation σ_j . The values were restricted to the physical range of each indicator. A common workload component produced coordinated changes among the indicators, while independently generated disturbances provided variability across repeated realizations. Scenario S_0 used a zero scenario-dependent component, S_1 contained a temporary pulse, S_2 used a gradual adverse increase, and S_3 combined an increase with a persistent high-load plateau.

As shown in Fig. 3, the workload generator defines the profile of each simulated scenario. The server-state model converts this profile into coordinated resource and service-quality indicators, and the monitoring component forms time-series observations. The analysis component performs normalization, binary-sequence formation, structural estimation, and sequential aggregation. The resulting estimate S_r is transformed into the INFO, WATCH, or ALERT response level. The principal processing parameters are presented in Table 4.

The components in Fig. 3 represent the conceptual target architecture and were not deployed as a physical experimental environment in the present study.

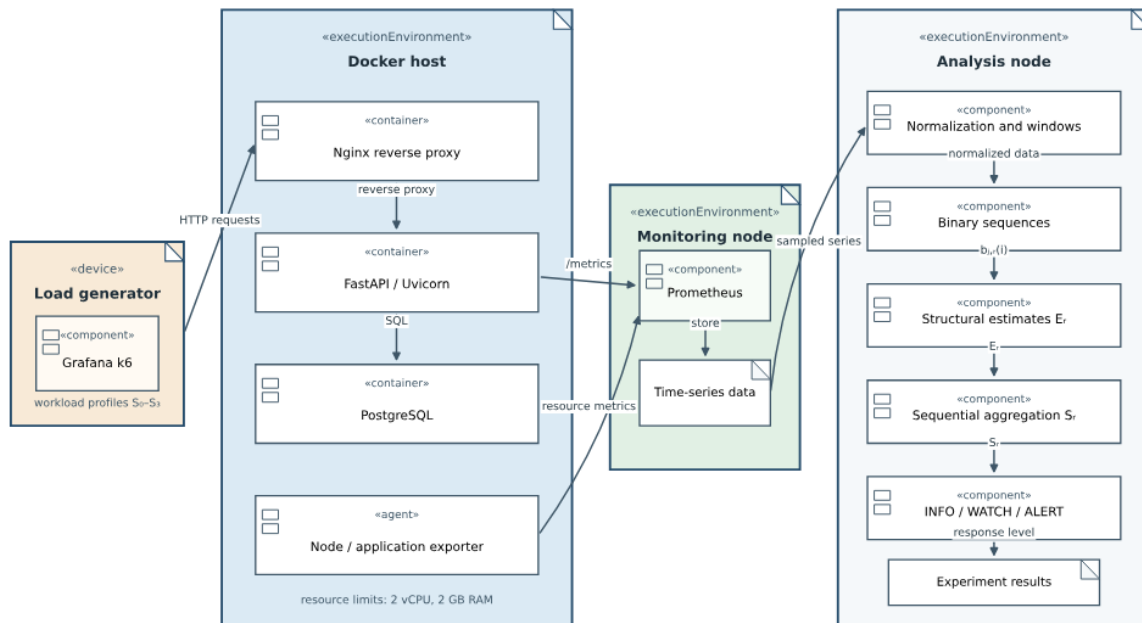


Fig. 3. Conceptual architecture of the simulation and early-warning environment

The parameters in Table 4 were selected using separate simulated stable-operation and calibration sequences and remained unchanged during evaluation. With $n = 12$ and $\Delta t = 5$ s, each window covered 55 s between its first and last observations, and a new estimate was produced every 15 s.

A 20-min baseline sequence was generated under stable conditions and used to calculate the reference values μ_j and σ_j . Four simulated workload scenarios were then evaluated, as summarized in Table 5.

Each scenario comprised 30 independent realizations generated with different random disturbances. The parameters of the deterministic scenario components and the decision procedure remained unchanged across all realizations.

The actual critical-state onset was determined independently of S_r . It was registered when at least one of the following conditions was satisfied for three consecutive observations: $RT_{95}(t) > 500$ ms or $ER(t) > 5\%$, where

$RT_{95}(t)$ is the 95th-percentile response time and $ER(t)$ is the HTTP error rate. This independent criterion enabled the calculation of warning lead time and comparison with conventional threshold-based monitoring.

Table 4.

Parameters of the simulation-based evaluation

Parameter group	Parameter	Value
Monitoring	Sampling interval Δt	5 s
Short sequences	Window length n	12 observations
Short sequences	Sliding step h	3 observations
Binarization	Local threshold θ_j	2
Local estimate	$(\alpha_q, \alpha_g, \alpha_c)$	(0.35, 0.40, 0.25)
Indicator aggregation	ω_j	0.20
Sequential aggregation	$p, \rho, \theta_0, d_{\max}$	4, 0.80, 0.25, 4
Component weights	$(\beta_L, \beta_T, \beta_C, \beta_D)$	(0.30, 0.20, 0.25, 0.25)
Variability penalty	β_V	0.15
WATCH thresholds	$(\theta_W^{\text{on}}, \theta_W^{\text{off}})$	(0.45, 0.35)
ALERT thresholds	$(\theta_A^{\text{on}}, \theta_A^{\text{off}})$	(0.70, 0.58)
Confirmation	(k_W, k_A, k_R)	(3, 2, 3)
Prediction horizon	H_p	300 s

Table 5.

Simulated workload scenarios

Scenario	Workload profile	Purpose
S_0	Constant load of 20 virtual users	Stable operation
S_1	Increase from 20 to 100 users for 60 s	Isolated deviation
S_2	Linear increase from 20 to 160 users over 20 min	Developing degradation
S_3	Increase to 160 users followed by constant high load	Persistent degradation

SIMULATION RESULTS AND COMPARATIVE EVALUATION

The simulation-based evaluation included 30 independent realizations for each scenario and 120 realizations in total. Scenarios S_0 and S_1 contained no critical states, whereas all runs of S_2 and S_3 reached the service-level criterion.

A response was considered timely if WATCH or ALERT was activated no more than $H_p = 300$ s before the critical-state onset. A response in a noncritical realization, or a response outside the specified prediction horizon, was treated as a false warning for the run-level evaluation.

Detection of Developing Degradation. The dynamics of the monitoring indicators during a representative developing-degradation realization are presented in Fig. 4.

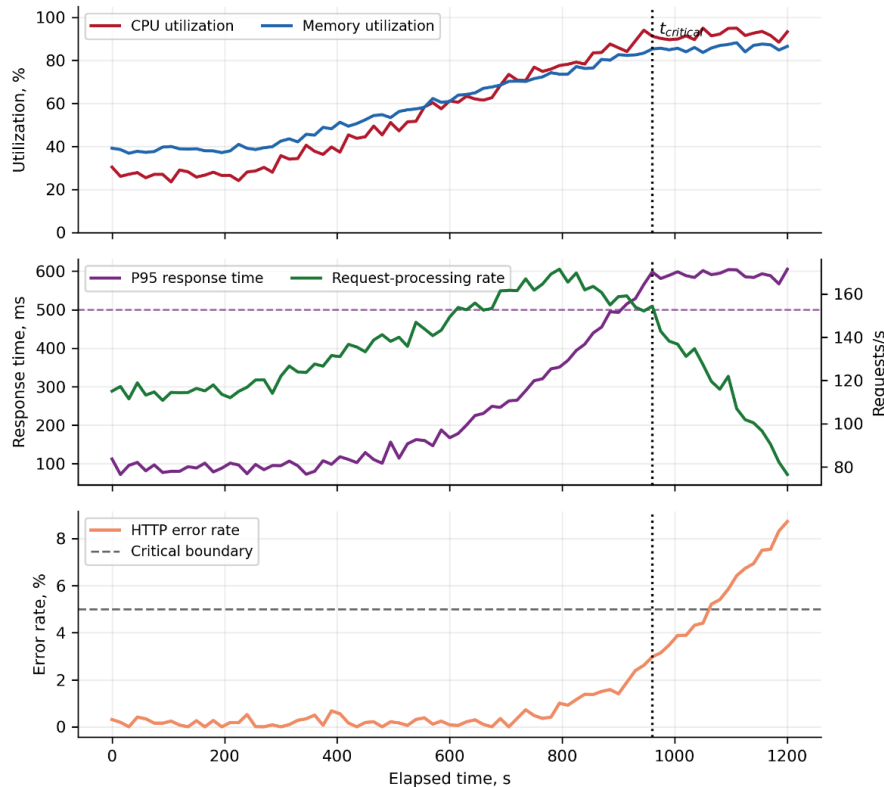


Fig. 4. Dynamics of simulated server-monitoring indicators under developing degradation

As shown in Fig. 4, increasing workload caused coordinated growth in CPU and memory utilization, followed by higher response time and error rate and reduced request-processing rate. However, the actual critical state was registered only at $t_{critical}$, when the service-level condition remained violated for three consecutive observations.

The effect of sequential aggregation is demonstrated in Fig. 5.

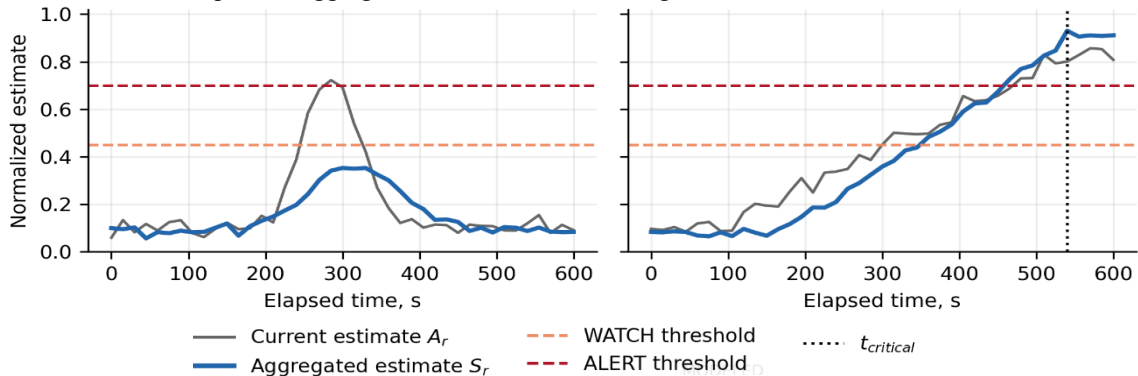


Fig. 5. Current and sequentially aggregated estimates: (a) isolated deviation; (b) developing degradation

In scenario S_1 , the current estimate A_r increased temporarily but returned to its initial range after the workload surge ended. The absence of continued accumulation and duration prevented S_r from reaching the WATCH threshold. In scenario S_2 , repeated adverse changes caused a persistent increase in S_r , resulting in WATCH activation before the actual critical-state onset. The interval between the first confirmed WATCH and $t_{critical}$ was used to calculate warning lead time.

The results obtained for all scenarios are summarized in Table 6.

Table 6.

Results obtained for the modelled scenarios

Scenario	Runs	Critical states	Timely warnings	False warnings	Missed warnings	T_{lead} , mean $\pm$ SD, s
S_0 : stable operation	30	0	–	1	–	–
S_1 : isolated deviation	30	0	–	3	–	–
S_2 : developing degradation	30	30	28	0	2	229.1 $\pm$ 45.2
S_3 : persistent degradation	30	30	29	0	1	248.0 $\pm$ 38.6
Total	120	60	57	4	3	238.7$\pm$42.7

As shown in Table 6, 57 of the 60 critical states were preceded by a timely warning. Four false warnings occurred in the noncritical scenarios, while three critical states were not identified within the specified prediction horizon. The pooled mean lead time calculated over the timely warnings was 238.7 $\pm$ 42.7 s, where SD denotes the sample standard deviation.

Comparison with Threshold-Based Monitoring. Each simulated realization was assigned one run-level outcome. A timely response was classified as a true positive, a response in a noncritical realization as a false positive, a noncritical realization without a response as a true negative, and a critical realization without a timely response as a false negative. Precision, recall, F1-score, and false-positive rate were calculated from these outcomes.

The proposed procedure produced 57 true positives, four false positives, 56 true negatives, and three false negatives. Threshold-based monitoring registered all 60 critical and 60 noncritical realizations correctly because the same service-level boundaries were used to define the critical-state onset. Its perfect classification is therefore expected by construction, and the comparison primarily concerns warning lead time. The results are presented in Table 7.

Table 7.

Comparison with threshold-based monitoring

Method	Precision	Recall	F1-score	False-positive rate	T_{lead} , mean $\pm$ SD, s
Threshold-based monitoring	1.000	1.000	1.000	0.000	0
Proposed procedure	0.934	0.950	0.942	0.067	238.7 $\pm$ 42.7

Threshold-based monitoring generated a response only after the critical criterion had been confirmed and therefore provided no warning lead time. The proposed procedure had an F1-score of 0.942 and provided an average warning lead time of approximately four minutes. Its principal advantage lies in enabling preventive action rather than improving retrospective classification against a threshold-defined reference condition.

DISCUSSION OF THE RESULTS

The results demonstrate the difference between registering an existing critical state and identifying its preceding development. Threshold-based monitoring detected the service-level violation without delay but provided no time for preventive action. Sequential aggregation formed a timely warning in 57 of the 60 critical realizations, with a mean lead time of 238.7 ± 42.7 s.

In the isolated-deviation scenario, a temporary increase in A_r did not produce a persistent increase in S_r . Under developing degradation, repeated adverse changes increased the trend, accumulation, and duration components and activated WATCH before the critical boundary was reached. The procedure therefore distinguished a short workload surge from a persistent adverse process.

Early detection was accompanied by four false warnings and three missed warnings. Increasing activation thresholds or confirmation intervals could reduce false responses but would also shorten warning lead time. The results are limited to the selected simulation assumptions and workload profiles; application to other infrastructures requires parameter recalibration and experimental validation.

Conclusions and prospects for further research

The objective of this study was to develop and evaluate an approach for the early warning of critical information system states based on short sequences of server-monitoring data. To achieve this objective, a procedure for forming local structural estimates was proposed, and their sequential aggregation was implemented by considering the current level, trend, accumulation, duration, and variability of adverse changes.

Decision rules were established to distinguish stable operation, isolated deviations, developing degradation, and persistent degradation. The proposed procedure was evaluated using simulated server-workload scenarios and compared with conventional threshold-based monitoring. The results confirmed that sequential aggregation distinguishes temporary workload surges from persistent adverse processes and can form a warning before the service-level criterion is violated. Therefore, the stated objective and research tasks were achieved.

Further research will focus on experimental validation using real server measurements, adaptive selection of aggregation parameters and decision thresholds, and evaluation under software faults, memory leaks, database contention, and network degradation. The approach should also be investigated in distributed, cloud, and cyber-physical systems with dynamically changing workloads.

ADDITIONAL INFORMATION

DECLARATION ON THE USE OF GENERATIVE ARTIFICIAL INTELLIGENCE TOOLS

During the preparation of this article, ChatGPT (OpenAI) was used for language and stylistic editing, refinement of selected formulations, and structuring of the manuscript. The artificial intelligence tool was not used to generate the source data, conduct experimental studies, perform calculations, or produce scientific results. The final content of the article, methodological provisions, analysis and interpretation of the results, and conclusions were formulated, verified, and approved by the authors.

REFERENCES

1. Soldani J., Brogi A. Anomaly detection and failure root cause analysis in (micro)service-based cloud applications: A survey. *ACM Computing Surveys*. 2022. Vol. 55, No. 3. Article 59. P. 1–39. DOI: <https://doi.org/10.1145/3501297>
2. Hrusto A., bin Ali N., Engström E., Wang Y. Monitoring data for anomaly detection in cloud-based systems: A systematic mapping study. *ACM Transactions on Software Engineering and Methodology*. 2026. Vol. 35, No. 4. Article 91. P. 1–36. DOI: <https://doi.org/10.1145/3744556>
3. Li G., Jung J. J. Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*. 2023. Vol. 91. P. 93–102. DOI: <https://doi.org/10.1016/j.inffus.2022.10.008>
4. Zamanzadeh Darban Z., Webb G. I., Pan S., Aggarwal C. C., Salehi M. Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*. 2024. Vol. 57, No. 1. Article 15. P. 1–42. DOI: <https://doi.org/10.1145/3691338>
5. Correia L., Goos J.-C., Klein P., Bäck T., Kononova A. V. Online model-based anomaly detection in multivariate time series: Taxonomy, survey, research challenges and future directions. *Engineering Applications of Artificial Intelligence*. 2024. Vol. 138. Article 109323. DOI: <https://doi.org/10.1016/j.engappai.2024.109323>
6. Li Z., Zhao Y., Geng Y., Zhao Z., Wang H., Chen W., Jiang H., Vaidya A., Su L., Pei D. Situation-aware multivariate time series anomaly detection through active learning and contrast VAE-based models in large distributed systems. *IEEE Journal on Selected Areas in Communications*. 2022. Vol. 40, No. 9. P. 2746–2765. DOI: <https://doi.org/10.1109/JSAC.2022.3191341>
7. Bäßler D., Kortus T., Gühring G. Unsupervised anomaly detection in multivariate time series with online evolving spiking neural networks. *Machine Learning*. 2022. Vol. 111. P. 1377–1408. DOI: <https://doi.org/10.1007/s10994-022-06129-4>

8. Kånåhols G., Hasan S., Strandberg P. E. Integrating time series anomaly detection into DevOps workflows. *IEEE Access*. 2025. Vol. 13. P. 46459–46477. DOI: <https://doi.org/10.1109/ACCESS.2025.3550665>

9. Kim S., Choi K., Choi H.-S., Lee B., Yoon S. Towards a rigorous evaluation of time-series anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2022. Vol. 36, No. 7. P. 7194–7201. DOI: <https://doi.org/10.1609/aaai.v36i7.20680>

10. Puccetti T., Ceccarelli A. Detection latencies of anomaly detectors: An overlooked perspective? *Proceedings of the 2024 IEEE 35th International Symposium on Software Reliability Engineering*

(ISSRE). Tsukuba, Japan, 2024. P. 37–48. DOI: <https://doi.org/10.1109/ISSRE62328.2024.00015>

11. Masol V. I., Popereshnyak S. V. Checking the randomness of bits disposition in local segments of the (0, 1)-sequence. *Cybernetics and Systems Analysis*. 2020. Vol. 56, No. 3. P. 513–520. DOI: <https://doi.org/10.1007/s10559-020-00267-0>

12. Masol V., Popereshnyak S. Joint distribution of some statistics of random bit sequences. *Cybernetics and Systems Analysis*. 2021. Vol. 57, No. 1. P. 139–145. DOI: <https://doi.org/10.1007/s10559-021-00337-x>

Світлана ПОПЕРЕШНЯК

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

РАННЄ ПОПЕРЕДЖЕННЯ КРИТИЧНИХ СТАНІВ ІНФОРМАЦІЙНИХ СИСТЕМ ЗА КОРОТКИМИ ПОСЛІДОВНОСТЯМИ

Надійність сучасних інформаційних систем значною мірою залежить від своєчасного виявлення змін, які можуть призвести до погіршення продуктивності, порушення встановленого часу відповіді та недоступності сервісів. Традиційні засоби моніторингу переважно фіксують перевищення заданих порогів, однак не завжди дають змогу розпізнати попередній розвиток небезпечної тенденції та відрізнити її від короткочасного відхилення. Метою дослідження є розроблення підходу до раннього попередження критичних станів інформаційних систем на основі секвенційного агрегування структурних оцінок коротких послідовностей даних моніторингу. У статті запропоновано процедуру нормування різнорідних показників, формування бінарних послідовностей у ковзних часових вікнах та обчислення локальних структурних оцінок, що характеризують частоту, фрагментованість і стійкість несприятливих спостережень. Для виявлення розвитку деградації локальні оцінки послідовно агрегуються з урахуванням поточного рівня відхилення, їхнього тренду, накопичення, тривалості та варіативності. На основі отриманої агрегованої оцінки розроблено правила інформаційного реагування з використанням порогів активації, інтервалів підтвердження та гістерезису під час відновлення системи. Запропонований підхід передбачає розмежування стабільного функціонування, ізольованого відхилення, поступового розвитку деградації та тривалого критичного стану. Імітаційне оцінювання проведено для відповідних сценаріїв зміни навантаження серверної інфраструктури. Результати показали, що секвенційне агрегування структурних оцінок дає змогу виявляти стійкі несприятливі зміни до моменту порушення встановлених вимог до якості обслуговування та водночас обмежувати реакцію на окремі короткочасні коливання. Порівняння з традиційним пороговим моніторингом підтвердило можливість формування часового інтервалу для виконання превентивних дій, зокрема масштабування ресурсів, перерозподілу навантаження або обмеження некритичних запитів. Запропонована процедура може бути інтегрована до інформаційних технологій моніторингу серверних, розподілених і хмарних систем та використана для інформаційної підтримки прийняття рішень щодо попередження критичних режимів їх функціонування.

Ключові слова: інформаційна технологія; прогнозування небезпечних станів; короткі випадкові послідовності; структурно-ймовірнісне оцінювання; секвенційне агрегування; раннє попередження; інформаційні системи; моніторинг стану систем.