http://doi.org/10.35784/iapgos.9544 *received: 12.04.2026 | revised: 18.05.2026 | accepted: 07.06.2026 | available online: 30.06.2026*

AGGREGATION OF MULTIMODAL LOG AND METRIC STREAMS FOR NEURO-FUZZY ANOMALY DETECTION IN COMPUTER SYSTEMS

Andrii Mishchenko[1], Oleksii Shushura[1], Alona Kolomiiets[2], Andrii Donets[1], Olena Kosaruk[2]
[1]National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine, [2]Vinnytsia National Technical University, Vinnytsia, Ukraine

Abstract. *Ensuring the stable and reliable operation of modern computer systems is a critical challenge. This is typically achieved through the continuous logging of system events and the monitoring of hardware resource metrics (e.g., CPU, RAM). However, conventional monitoring solutions generally analyse these data streams in isolation. Their direct integration is significantly hindered by fundamental differences in their temporal characteristics and measurement scales. For instance, logs are often processed using OpenSearch, while metrics are monitored via Grafana. Consequently, the correlation context is lost, which impedes the identification of the root causes of system anomalies. To overcome these limitations, this paper proposes a novel method that fuses the multimodal input streams of logs and metrics into a unified feature space, specifically designed for subsequent use by neuro-fuzzy systems for advanced anomaly detection. This study presents a mathematical formalization of the problem domain by introducing a unified system of variables and developing an observation space model. The proposed heterogeneous data aggregation method effectively prepares the input space for neuro-fuzzy classifiers. Temporal synchronization between metrics and events is achieved through a sliding window strategy, while min-max normalization is applied to numerical indicators to eliminate feature dominance. Additionally, log processing is implemented by converting unstructured messages into standardized templates, which are then weighted by their criticality level and further analysed using the entropy of the event stream. The proposed approach generates an informative state space characterized by high spatial separability between normal and anomalous system states, making the resulting feature vector highly suitable for the subsequent training of neuro-fuzzy networks. Experimental results demonstrate that the method successfully captures the synchronous correlation between hardware load spikes and the occurrence of critical errors.*

Keywords: anomaly detection, data aggregation, multimodal data, log analysis, performance metrics, neuro-fuzzy systems

AGREGACJA WIELOMODALNYCH STRUMIENI LOGÓW I METRYK NA POTRZEBY NEURONOWO-ROZMYTEGO WYKRYWANIA ANOMALII W SYSTEMACH KOMPUTEROWYCH

Streszczenie. *Zapewnienie stabilnego i niezawodnego działania nowoczesnych systemów komputerowych stanowi kluczowe wyzwanie. Zazwyczaj osiąga się to poprzez ciągłe rejestrowanie zdarzeń systemowych oraz monitorowanie wskaźników zasobów sprzętowych (np. procesora, pamięci RAM). Jednak konwencjonalne rozwiązania monitorujące zazwyczaj analizują te strumienie danych w oderwaniu od siebie. Ich bezpośrednia integracja jest znacznie utrudniona przez zasadnicze różnice w charakterystyce czasowej i skalach pomiarowych. Na przykład logi są często przetwarzane przy użyciu OpenSearch, podczas gdy wskaźniki są monitorowane za pomocą Grafany. W rezultacie traci się kontekst korelacji, co utrudnia identyfikację pierwotnych przyczyn anomalii systemowych. Aby przezwyciężyć te ograniczenia, w niniejszym artykule proponuje się nowatorską metodę, która łączy wielomodalne strumienie danych wejściowych logów i metryk w ujednoliconą przestrzeń cech, zaprojektowaną specjalnie do późniejszego wykorzystania przez systemy neuronowo-rozmyte w celu zaawansowanego wykrywania anomalii. W niniejszym badaniu przedstawiono matematyczną formalizację obszaru problemowego poprzez wprowadzenie ujednoliconego systemu zmiennych oraz opracowanie modelu przestrzeni obserwacyjnej. Proponowana metoda agregacji danych heterogenicznych skutecznie przygotowuje przestrzeń wejściową dla klasyfikatorów neuronowo-rozmytych. Synchronizację czasową między metrykami a zdarzeniami osiąga się za pomocą strategii okna przesuwnego, natomiast do wskaźników liczbowych stosuje się normalizację min-max w celu wyeliminowania dominacji cech. Dodatkowo przetwarzanie logów jest realizowane poprzez konwersję nieustrukturyzowanych komunikatów na standardowe szablony, które są następnie ważone według poziomu krytyczności i dalej analizowane przy użyciu entropii strumienia zdarzeń. Proponowane podejście generuje informacyjną przestrzeń stanów charakteryzującą się wysoką separowalnością przestrzenną między normalnymi i anomalnymi stanami systemu, dzięki czemu wynikowy wektor cech doskonale nadaje się do późniejszego uczenia sieci neuronowo-rozmytych. Wyniki eksperymentalne pokazują, że metoda z powodzeniem uchwyciła synchroniczną korelację między skokami obciążenia sprzętu a występowaniem krytycznych błędów.*

Słowa kluczowe: wykrywanie anomalii, agregacja danych, dane wielomodalne, analiza logów, metryki systemowe, sieci neuronowo-rozmyte

Introduction

Ensuring the reliable operation of computer systems relies not only on correct software implementation and IT infrastructure setup but also on the timely detection of operational anomalies. Modern services are constantly evolving to satisfy client needs; however, in certain cases, this leads to system performance issues, as effectively tracking system logs and service metrics for early anomaly detection and resolution remains a challenge. Therefore, it is common practice to utilize specialized information systems designed to automate log analysis and configure notifications for predefined anomalies. For example, OpenSearch enables the monitoring of system logs and reports problems specified by specialists [1], while Grafana is used to track correct service operation by setting boundary thresholds for metrics like central processing unit (CPU) load or random-access memory (RAM) usage [4]. The primary drawback of such monitoring systems is their inability to natively support multimodal data. Specifically, it is impossible to use them for the concurrent analysis of logs and corresponding metrics within a single timeframe to make an informed decision on whether a given situation represents an anomaly that must be reported to a specialist for resolution.

Anomaly detection in modern distributed computer systems is a critical challenge addressed by numerous researchers and practitioners. A survey of 312 DevOps and SRE professionals revealed that over 74% collect both logs and metrics within their systems, nevertheless, current academic literature largely overlooks this practical need for multimodal data analysis [8]. Consequently, over the past few years, the scientific community has witnessed a gradual shift from analysing a single data type to analysing multiple types simultaneously to improve the accuracy of model inferences.

Anomaly detection methods that leverage multiple data types as input significantly enhance detection accuracy and reliability by synergizing the strengths of each modality: logs specify the nature of the problem, while metrics pinpoint the exact time the system deviated from normal behaviour. Identifying anomalies within logs and correlating these findings with metric time series enables the effective isolation of root causes [12]. In practical applications, the approach of combining logs and metrics has been utilized for anomaly detection in data centres. The results demonstrated that the joint analysis of diverse data sources substantially increases the number of successful detections, particularly for complex anomalies that are difficult to identify when data streams are analysed in isolation [6].

Beyond large-scale infrastructures such as data centres, leveraging the correlation between metrics and system logs is equally crucial for ensuring the reliable operation of micro-service architectures. Recent studies have developed approaches to identify the root causes of failures using multimodal data, revealing that abrupt fluctuations in metric data frequently correlate with anomalous log scores [15]. This provides further evidence that developing formal models capable of fusing these two distinct data streams remains a highly relevant research objective.

A significant portion of contemporary research focuses on enhancing methods for processing time series and system logs independently. For example, a study on deep multivariate time series anomaly detection (MTSAD) provides a taxonomy of modern architectures that enable the modelling of both temporal and inter-variable dependencies [13]. However, most of the reviewed models are characterized by a limited ability to provide explanations for the obtained results.

In the domain of log analysis, the MLAD model has been introduced. It leverages semantic relational reasoning to facilitate anomaly detection within multi-system environments. This approach effectively resolves the "identical labels" issue and demonstrates an enhanced capacity for knowledge transfer across distinct systems; however, the underlying semantic processing requires substantial computational resources [14].

In contrast to complex deep learning models, neuro-fuzzy systems offer the distinct advantage of ensuring transparency in the decision-making process [7]. A notable example of the successful application of hybrid fuzzy networks is the detection of cyber anomalies [2]. The underlying rule sets are formulated based on massive datasets. Employing this approach facilitates the development of expert systems characterized by both high accuracy and interpretable operational logic.

The capacity of models to adapt to system dynamics during continuous monitoring is a critical feature that researchers strive to achieve. In addressing this challenge, Evolving Fuzzy Systems (EFS) have garnered significant interest. A notable solution for log monitoring within INFN data centres employs sliding time windows combined with an evolving Gaussian fuzzy classifier (eGFC). It has been demonstrated that this approach enables monitoring systems to dynamically update their structure in real time, thereby ensuring continuous adaptation to shifts in the underlying data distribution [3].

Another solution for system protection and diagnostics is a proposed approach utilizing quantum autoencoders (QAE-u8) for intrusion detection in IoT networks; however, this method entails a distinct trade-off between detection accuracy and the consumption of CPU and RAM resources [11]. Furthermore, to secure web applications and detect anomalies, a hybrid method for analysing typical user behaviour has been introduced. Built upon statistical modelling (KDE) and deep learning (LSTM), this approach enhances the discrimination between actual cyberattacks and the atypical actions of legitimate users, thereby significantly reducing the false positive rate [10].

Recent advancements in multimodal anomaly detection primarily rely on large-scale deep learning architectures. Methods leveraging Transformer-based models, representation learning, and contrastive approaches have demonstrated remarkable accuracy in aligning heterogeneous telemetry data, such as logs and metrics. However, these methods exhibit significant practical limitations. Modern deep neural networks often suffer from 'black-box' opacity, failing to provide the strict interpretability required by system administrators to determine the root cause of an alert, while simultaneously imposing substantial computational costs [4].

An analysis of recent studies indicates that while effective methods exist for processing logs [3, 14] and metrics [13] independently, alongside emerging attempts to combine them [6, 15], the majority of these approaches either rely on computationally intensive deep learning models – which are often unsuitable for real-time application – or restrict their monitoring scope exclusively to log data. Furthermore, although study [3] proposes the application of evolving fuzzy systems, its primary focus remains confined solely to the analysis of system records.

Consequently, it can be concluded that there is an urgent need to develop novel approaches for the formalization and pre-processing of multimodal data, specifically integrating both logs and metrics. The subsequent application of these pre-processed data streams within evolving neuro-fuzzy systems is highly promising, as it ensures real-time adaptation to dynamic system behaviour, guarantees high accuracy in anomaly detection, and provides clear, interpretable explanations for the obtained results.

1. Research objectives

The objective of this study is to develop approaches for the formalization and pre-processing of multimodal data – encompassing system logs and performance metrics – for their subsequent integration into evolving neuro-fuzzy models aimed at anomaly detection in computer systems.

To achieve this objective, the following specific tasks must be addressed:

- formally describe the input data by defining the variables for hardware metrics (CPU, RAM) and system logs using the mathematical framework of set theory,
- develop a method for fusing these multimodal inputs into a unified feature vector, thereby enabling the resulting data to be processed by neuro-fuzzy models,
- implement the proposed method and validate it on a real-world dataset to evaluate the efficacy of the data preparation process.

2. Proposed methodology

A critical stage in addressing the anomaly detection problem within computer systems using neuro-fuzzy networks – such as ANFIS or Evolving Fuzzy Systems (EFS) – is the unification of heterogeneous data.

2.1. Formalization of input data

Modern computer systems exhibit a complex and dynamic architecture comprising various components $C = \{c_1, c_2, \dots, c_q\}$ (e.g., microservices, databases, and network gateways) that may evolve during system operation. Constructing a unified model for the entire system S is highly challenging due to the problem of high data dimensionality; therefore, this study adopts a decomposition approach to the monitoring task. The object of analysis is a distinct target node $c \in C$, whose operation is observed in continuous time $t \in [0, +\infty)$. It is assumed that anomaly detection methods are applied independently to each component.

To establish a monitoring process for the current state of the system S, a feature space O is introduced. This space is formulated by aggregating two heterogeneous data streams:

- the metric stream (M): generated through the regular measurement of quantitative performance indicators of a component, such as CPU or RAM utilization, and network ingress and egress,
- the log stream (L): comprising asynchronous system records generated by each individual component upon a change in its internal state.

While each of these data streams can independently indicate the onset of system anomalies, leveraging them concurrently yields superior accuracy and facilitates a more comprehensive root-cause analysis. Therefore, anomaly detection relies on a multimodal feature space, formally expressed as:

$$O = O_M + O_L \tag{1}$$

2.2. Processing of performance metrics

Modern computer systems are highly complex mechanisms whose reliable operation depends on a multitude of factors. Consequently, effective monitoring requires the simultaneous analysis of multiple performance metrics. Let N denote the total number of metric types measured at the target node. The measurement vector at time t is defined as $\mathbf{m}(t) \in \mathbb{R}^N$, which can be formally expressed as:

$$m(t) = [m_1(t), m_2(t), \dots, m_N(t)]^T \tag{2}$$

To enhance the semantic expressiveness of the model, the metrics measured within the system are best partitioned into subsets based on their respective resource types:

- $M_{\text{cpu}} \subset M$: Central Processing Unit (CPU) metrics, such as utilization percentage or active load,
- $M_{\text{mem}} \subset M$: Random Access Memory (RAM) metrics, such as memory usage percentage and swap file utilization,
- $M_{\text{io}} \subset M$: Input/Output (I/O) metrics, such as disk read/write speeds and network bandwidth.

Although metrics are represented numerically, they encompass a variety of physical units of measurement, most commonly percentages, bytes, and milliseconds. This heterogeneity necessitates a data pre-processing step to prevent features with larger numerical ranges from dominating the learning process of the neuro-fuzzy network. To address this issue, it is proposed to normalize each metric $m_i(t)$ into a dimensionless space [0,1] using the Min-Max scaling formula:

$$\widehat{m_i}(t) = \frac{m_i(t) - m_i^{min}}{m_i^{max} - m_i^{min}} \tag{3}$$

where m_i^{min} and m_i^{max} denote the boundary values, which can be determined via two approaches:

- predefined limits based on the component's specification (e.g., 0% and 100% for CPU utilization),
- dynamic extrema calculated over a historical time window T_{hist}.

A fundamental distinction between metrics and logs lies in their temporal characteristics. Let t denote the continuous time axis. The metric stream M is not updated continuously but rather at discrete, fixed intervals; thus, it can be formally defined on a regular time grid T_M:

$$T_M = \{t_0 + n \cdot \Delta t_m \mid n \in N\} \tag{4}$$

where t_0 is the initial measurement time, n is the measurement index, and Δt_m represents the sampling period (e.g., 10 or 60 seconds).

2.3. Processing of system logs

Conversely, the log stream L represents a stochastic point process defined over a set of random time instances $T_L = \{t_1, t_2, \dots, t_j\}$, where each occurrence t_j is independent of the regular measurement grid T_M.

Consequently, the problem of constructing the state spaces reduces to the challenge of mapping the stochastic process L onto the regular grid T_M, or a similar temporal structure. Achieving this necessitates the application of specific aggregation operators.

Each log message $l_j \in L$ can be formally defined as a tuple:

$$l_j = \langle t_j, s_j, msg_j \rangle \tag{5}$$

where $t_j \in \mathbb{R}^+$ is the timestamp of the event occurrence ($t_j \leq t_{j+1}$), $s_j \in S_{levels}$ represents the severity level of the event (e.g., S_{levels} = {DEBUG, INFO, WARN, ERROR, FATAL}), and msg_j is the unstructured textual description of the event, comprising a sequence of characters or tokens.

Processing the log messages msg_j represents the primary challenge in developing the proposed method, as they inherently intertwine static text with dynamic variables (e.g., server or database IP addresses, transaction or request identifiers, and execution delays). Therefore, to derive a feature space suitable for input into an ANFIS, factorization of the message space is required. To achieve this, we define an abstraction operator $\mathcal{P}$ that maps the set of all possible log messages onto a finite set of event templates $\Omega = \{E_1, E_2, \dots, E_K\}$:

$$\mathcal{P}(msg_j) \rightarrow E_p \tag{6}$$

where $E_p \in \Omega$; E_p is the constant part of the message, that is, its template.

Employing this approach enables the transformation of each log message l_j into a pair $\langle t_j, id(E_p) \rangle$, where $id(E_p)$ represents the unique numerical identifier of the event template. Consequently, this facilitates a transition from unstructured text analysis to the analysis of discrete event time series.

2.4. The proposed multimodal data aggregation method

Building upon the formalized characteristics presented above, it is imperative to develop a method capable of fusing and temporally synchronizing the multimodal data. A critical challenge in processing multimodal data is their fundamentally different temporal origins: the metrics M constitute regular time series, whereas the system logs L represent a stream of discrete events. To fuse these distinct data streams into a unified phase space O, a novel method is proposed, founded on a sliding window mechanism coupled with the semantic weighting of events.

To resolve the issue of temporal synchronization, let ΔT denote the sliding window size and δ represent the stride (step size). At a given time instance t_k, the temporal interval $W_k = [t_k - \Delta T, t_k]$ is evaluated. The primary objective of the proposed method is to map all multimodal data points captured within the interval W_k onto a single, unified feature vector $\mathbf{x}_k$.

The process of constructing the feature vector $\mathbf{x}_k$ encompasses three sequential stages. First, the metrics must be discretized and aggregated. Because the arrival frequency of metrics from disparate sources may fluctuate or vary, they must be consolidated into a single representative value over the interval W_k. To this end, an aggregation function F_{agg} is employed, calculating an aggregated value $\mu_i(t_k)$ for each metric $m_i \in M$. Given that the primary objective is anomaly detection, the maximum function is deemed the most informative, as it allows for the effective capture of load spikes:

$$\mu_i(t_k) = \max_{t \in W_k} \widehat{m_i} \tag{7}$$

This stage yields a fixed-length metric vector:

$$\mathbf{v}_M(t_k) = [\mu_1(t_k), \dots, \mu_N(t_k)] \tag{8}$$

The second stage involves the semantic vectorization of the system logs. Its primary objective is to transform the subset of logs $L_k = \{l_j \mid t_j \in W_k\}$ into a numerical representation. To achieve this, a Weighted Event Frequency approach is proposed, as opposed to the traditional Bag-of-Words (BoW) method, which solely accounts for the occurrence frequency of event templates. The proposed approach is highly advantageous because it enables the explicit incorporation of each message's severity level into the resulting vector.

Let $\Omega = \{E_1, \dots, E_K\}$ denote the dictionary of event templates. The log vector $\mathbf{v}_L(t_k)$ has a dimensionality of K, where each element $v_{L,p}$ corresponds to the template E_p and is computed as follows:

$$v_{L,p}(t_k) = \sum_{l_j \in L_k} \mathbb{I}\big(\mathcal{P}(msg_j) = E_p\big) \cdot w(s_j) \tag{9}$$

where $\mathbb{I}\big(\mathcal{P}(msg_j) = E_p\big)$ is a binary indicator function (yielding 1, if the template matches, and 0 otherwise), and $w(s_j)$ is a weighting function that maps the severity level s_j to a specific numerical coefficient (e.g., $w(\text{INFO}) = 0.1, w(\text{ERROR}) = 1$).

The weighting coefficients for severity levels are defined as a heuristically structured non-linear scale that reflects the architectural principles of logging protocols. These values are selected to balance the model's sensitivity to operational risks: the low weight of INFO (0.1) minimizes the impact of routine, low-entropy operations; the intermediate weight of WARN (0.5) identifies potential deviations that, while non-critical, require attention; and the maximum weight of ERROR (1) ensures an aggressive system response to confirmed failures. Utilizing this progression within a unified feature space allows for the effective filtering of background noise from the underlying data.

To ensure the compatibility of the derived vector with the neuro-fuzzy network, a normalization step is imperative. Techniques such as logarithmic scaling or Min-Max normalization are recommended to prevent the dominance of excessively large values, which typically manifest during cascading system failures or log storms.

During complex system failures, the total volume of logs may not increase and can remain within normal limits; however, the combinations of messages shift as warnings and errors emerge. To address the challenge of detecting structural changes within the message stream, the application of the Shannon information entropy metric is proposed. To this end, within the current time window t_k, it is necessary to determine the probability of occurrence for each pattern $E_p \in \Omega$ using the ratio of the specific pattern's frequency to the total number of events in the window $|L_k|$:

$$P(E_p) = \frac{\sum_{l_j \in L_k} \mathbb{I}(\mathcal{P}(msg_j)=E_p)}{|L_k|} \quad (10)$$

Using the probabilities derived from (10), the information entropy of the current state $H(t_k)$ is calculated, which characterizes the degree of chaos in the log generation process:

$$H(t_k) = -\sum_{p=1}^{K} P(E_p) \log_2 P(E_p) \quad (11)$$

where K is the total number of patterns in the dictionary Ω.

The final stage involves integrating the two vectors and the entropy metric into a unified feature space. To form the final system state vector x_k, a concatenation operation is performed on the metric and log vectors:

$$\mathbf{x}_k = \mathbf{v}_M(t_k) \oplus \mathbf{v}_L(t_k) \oplus H(t_k) \quad (12)$$

where $\mathbf{v}_M(t_k)$ represents the aggregated metric vector at time t_k, $\mathbf{v}_L(t_k)$ denotes the weighted log vector at the same time instance, , and $H(t_k)$ is the entropy of the log stream.

The dimensionality of the resulting feature space is $D = N + K$, where N denotes the number of performance metrics and K represents the total number of event templates.

The proposed method demonstrates high computational efficiency, making it strictly suitable for large-scale and high-load systems. The time complexity for processing a single temporal window evaluates to $O(|L_k| \cdot d)$, where $|L_k|$ is the number of events in the window and d is the maximum log message length. Crucially, because the temporal aggregation window ($\Delta T = 1$ minute) and the template dictionary are fixed, the memory complexity remains $O(1)$. Memory usage is bounded only by the maximum throughput within a single minute, rather than total system uptime. Furthermore, the framework's decentralized node-level decomposition naturally supports horizontal scalability. In distributed environments, each microservice independently executes its own $O(|L_k| \cdot d)$ aggregation pipeline, explicitly preventing central processing bottlenecks during anomaly-induced 'log storms'.

2.5. Architectural integration with neuro-fuzzy systems

While the primary contribution of the proposed pipeline is the mathematical fusion of heterogeneous data streams, its objective is to process input data for the subsequent operation of a neuro-fuzzy classifier, such as an adaptive neuro-fuzzy inference system (ANFIS). The unified feature vector – comprising peak-preserved metrics and severity-weighted log event counts – provides crisp numerical inputs that resolve the problem of temporal alignment.

Within the overall architecture, the resulting vector is planned to be passed to the fuzzification layer, where metric values and discrete log event frequencies are mapped to linguistic variables using membership functions. Subsequently, the adaptive weighting mechanism of the neuro-fuzzy architecture utilizes the neural network's learning phase to autonomously tune the weights of the fuzzy inference rules. By providing a mathematically stable and non-diluted feature space, the aggregation pipeline ensures that the downstream neuro-fuzzy model can effectively learn complex, non-linear correlations between hardware resource exhaustion and software execution errors, free from the interference of temporal baseline noise.

3. Experimental evaluation

To evaluate the efficacy of the proposed method, an experimental study was conducted using a synthetic dataset that simulates the operation of a server node over a 60-minute period. The experimental scenario encompasses intervals of normal system operation alongside an injected anomaly, which is characterized by a sudden spike in workload and a subsequent service failure. This anomalous event is explicitly introduced between the 30th and 35th minutes of the simulation.

The experimental setup utilizes the following input parameters:

- a sliding window size of $\Delta t = 1$ minute,
- Severity level weights configured as w(INFO) = 0.1, w(WARN) = 0.5, and w(ERROR) = 1,
- Input data streams comprising CPU and RAM utilization time series, alongside a system log stream (generating up to 50 events per minute).

The execution of the aggregation algorithm resulted in the formation of the final feature matrix X, an excerpt of which is presented in Table 1.

Table 1. Fragment of the feature matrix (metrics are normalized, log frequencies are weighted)

Time	CPU Load	RAM Usage	Info Logs	Warn Logs	Error Logs	Log Entropy
14:25	0.235	0.212	0.7	1	0	0.76
14:26	0.254	0.212	0.5	0	0	0
14:27	0.278	0.204	0.6	0.5	0	0.592
14:28	0.208	0.242	0.4	0	0	0
14:29	0.235	0.228	0.4	0.5	0	0.722
14:30	0.686	0.252	0.8	0	0	0
14:31	0.67	0.283	0.9	0	0	0
14:32	0.658	0.224	1	0	0	0
14:33	0.648	0.184	0.7	0	0	0
14:34	0.672	0.256	0.9	0	0	0
14:35	0.648	0.237	0.8	0	0	0
14:36	0.661	0.237	0.7	0	0	0
14:37	0.61	0.208	1	0	0	0
14:38	0.659	0.216	1.1	0	0	0
14:39	0.687	0.22	1.1	0	0	0

Table 1 presents a feature matrix fragment capturing an interval of elevated hardware load (CPU 0.686 at 14:30) occurring under nominal system conditions. Despite the significant increase in CPU utilization, the system remains in a stable, non-anomalous state. The corresponding software logs confirm this status: Error Logs and Warn Logs remain at zero, and the Log Entropy is 0, indicating that the system's execution paths are consistent with standard operation. This fragment validates the pipeline's robustness against false positives, demonstrating its ability to distinguish between harmless, high-load baseline fluctuations and true system failures that necessitate administrative intervention.

To empirically justify the aggregation strategy, a comparative analysis was conducted at the optimal 1-minute window. The empirical tests revealed a critical operational trade-off: the median operator achieved a higher overall F1-Score (0.929) compared to the maximum function (0.867), as it effectively acted as a low-pass filter, smoothing out baseline telemetry noise. However, as visualized in Figure 1, this statistical smoothing comes at an unacceptable operational cost. Median aggregation systematically discards true sub-minute transient anomalies (e.g., brief RAM spikes or sudden CPU deadlocks) as statistical outliers. In production AIOps environments, masking a short but fatal resource exhaustion event is a catastrophic failure of the monitoring system. Consequently, the maximum function is explicitly retained as the necessary architectural choice. It mathematically guarantees the preservation of critical peak amplitudes, prioritizing the strict visibility of transient faults over minor statistical smoothing.

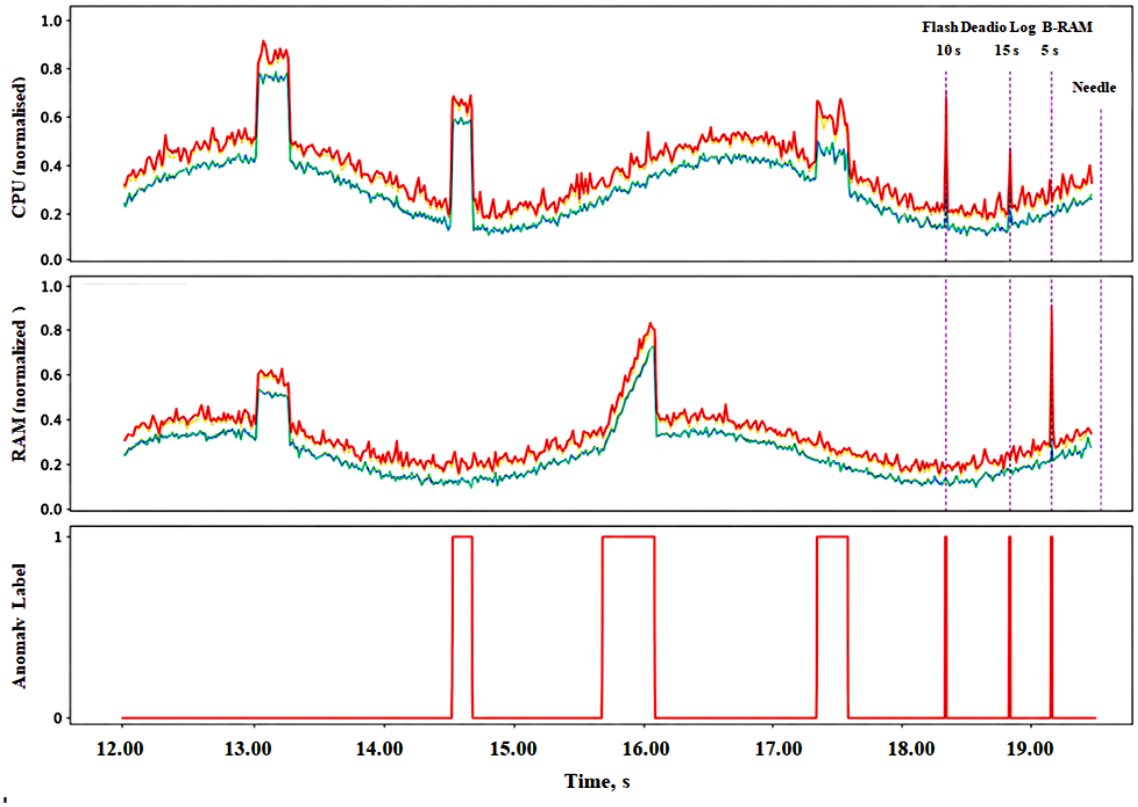

Fig. 1. Signal dilution effect of statistical aggregation methods on sub-minute transient anomalies

To determine the optimal aggregation interval, a sensitivity analysis evaluated window sizes of 1, 5, 15 and 30 minutes (ΔT). As illustrated in Figure 2, results revealed a strict inverse correlation between window size and detection efficacy. Extended intervals (e.g., 15 or 30 minutes) cause "temporal collision" where distinct events are erroneously grouped, mathematically diluting brief, critical anomalies. Testing confirmed the 1-minute window as optimal (F1-Score: 0.867), balancing granular signal preservation with sufficient multimodal context. Regarding other architectural parameters, the temporal stride was set equal to the window size to prevent data leakage during cross-validation, and the number of log templates remains deterministically bounded by the semantic diversity of the data stream. Furthermore, the severity weights follow a fixed non-linear scale to filter routine background noise. Rather than manual sensitivity tuning, the dynamic optimization of these baseline weights is explicitly delegated to the downstream neuro-fuzzy network, which will autonomously adapt them during its learning phase.

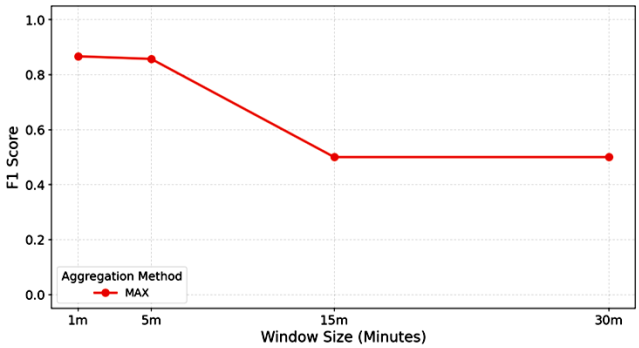

Fig. 2. Effect of aggregation window size on anomaly detection F1-Score using the MAX operator

To rigorously quantify the effectiveness of the proposed multimodal aggregation pipeline, a Random Forest classifier was utilized to evaluate the unified feature vectors. In accordance with standard evaluation protocols for anomaly detection, performance was measured using Precision, Recall, F1-Score, and ROC-AUC metrics. Furthermore, to demonstrate the explicit value of multimodal fusion, the proposed combined method was evaluated against two established ablation baselines: a Metrics-Only approach and a Logs-Only approach (Table 2). The quantitative results clearly demonstrate the advantage of the proposed fusion pipeline. While the isolated baselines suffered from lower detection accuracy due to incomplete system context (F1-Scores of 0.815 and 0.643, respectively), the Proposed Combined Vector achieved a maximum F1-Score of 0.867 and a ROC-AUC of 0.997. This confirms that mathematically aligning logs and metrics into a single feature space significantly enhances the classifier's ability to detect complex, cross-domain anomalies that isolated methods miss.

Table 2. Quantitative evaluation of the proposed multimodal fusion method against isolated data baselines

Feature Set	Precision	Recall	F1-Score	ROC-AUC
Metrics Only	1	0.688	0.815	0.971
Logs Only	0.75	0.563	0.643	0.865
Proposed Combined	0.929	0.813	0.867	0.997

Figure 3 presents a graph depicting the temporal correlation between the physical load of the system (normalized CPU utilization) and the semantic load of the log files (weighted error logs). As observed in the graph, the system experiences periods of normal elevated workload (e.g., around 13:00 and 17:30) where CPU utilization spikes occur without any accompanying log errors. Conversely, the plot distinctly captures transient anomalies of varying natures. At approximately 18:20, a critical hardware exhaustion event occurs, marked by a sudden, isolated CPU utilization spike reaching the maximum value of 1.0. Later, at 18:50, a severe software-level failure emerges, characterized by a massive surge in weighted error logs accompanied by a moderate elevation in CPU load (0.48). An additional, smaller log anomaly is captured at exactly 16:00. This visualization confirms that the proposed multimodal feature space accurately captures and differentiates between harmless baseline load fluctuations, isolated hardware constraints, and critical software-level errors.

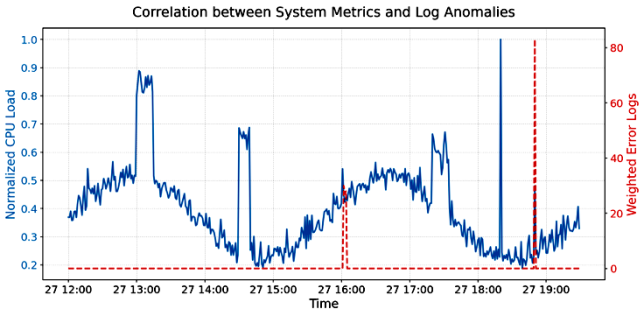

Fig. 3. Temporal dependence between the CPU metric and the weighted number of errors

Figure 4 presents a heat map that visualizes the temporal evolution of the system's state vector.

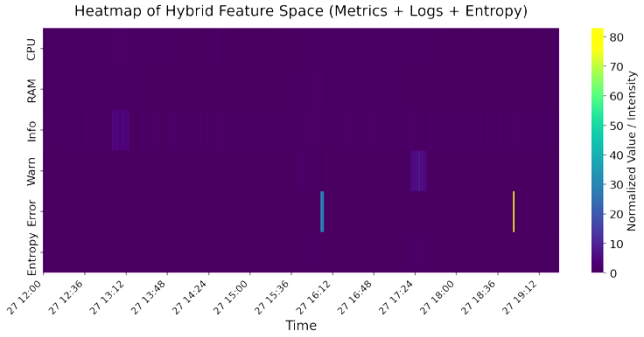

Fig. 4. Heat map of the hybrid feature space

Figure 4 provides a visualization of the constructed feature space. The lighter regions indicate areas with high values of the feature vector $\mathbf{x}_k$. The region of anomalous activity is clearly discernible in the center of the temporal scale, confirming that the proposed aggregation method ensures a synchronous representation of the correlation between the increased workload (metrics) and the occurrence of critical errors (logs). Consequently, the resulting feature vector is highly suitable for subsequent anomaly detection utilizing a neuro-fuzzy network.

The experimental results demonstrate that the proposed aggregation method successfully transforms heterogeneous raw data into a structured vector space while preserving critical information regarding anomalous system behaviour for subsequent analysis by a fuzzy network. Although the dimensionality of the resulting vector is low, it proves highly sufficient for accurate incident detection, thereby satisfying the computational efficiency prerequisites inherent to neuro-fuzzy networks.

4. Discussion

While the quantitative results demonstrate the superiority of the multimodal aggregation pipeline, a critical analysis of its boundary conditions reveals specific operational limitations and potential failure modes.

The primary analytical strength of the proposed fusion vector lies in its cross-domain validation. In isolated metric monitoring, a sudden CPU spike might be falsely flagged as an anomaly when it is merely a scheduled database backup. Conversely, isolated log monitoring might flag a flurry of WARN messages that are harmless network retries. By enforcing a unified feature space, the pipeline requires cross-domain consensus, inherently generalizing well across different microservices by abstracting raw telemetry into a standardized vector.

Despite this robustness, the architecture is susceptible to two primary failure modes. First, the pipeline is strictly bound by temporal synchronization. Because the multimodal fusion relies on overlapping time windows ($\Delta T = 1$ minute), it assumes precise timestamp alignment. If a host machine experiences Network Time Protocol synchronization failure or severe clock drift, the sliding window will erroneously fuse disjointed logs and metrics, severely corrupting the feature vector and causing false negatives. Second, the pipeline is vulnerable to semantic drift. The log template dictionary is deterministically bounded by the training corpus. A major software update that introduces entirely novel logging paradigms or unstructured formats will result in unmatched log events, degrading the log-feature inputs until the template dictionary is recalculated.

Finally, as noted in the methodology, the linear computational complexity $O(|L_k| \cdot d)$ and $O(1)$ memory bounds ensure the pipeline can scale horizontally in high-load environments without centralized bottlenecks. However, applying this pipeline in a real-world production environment will introduce non-deterministic baseline noise and overlapping multi-fault scenarios that are not present in controlled datasets. Addressing these complex, overlapping failure states is the explicit objective of the downstream neuro-fuzzy classifier, which will utilize these aggregated vectors to map non-linear correlations in future real-world deployments.

5. Conclusions

Thus, this study addresses the pressing problem of multimodal data pre-processing for subsequent application in neuro-fuzzy networks aimed at anomaly detection within distributed computer systems. A mathematical formalization of the problem domain is presented, establishing a unified system of variables for heterogeneous data and developing a comprehensive model of the observation space. This approach enables a transition from analysing continuous system metrics and stochastic textual events in isolation to operating within a single, unified mathematical state space of the system.

A novel method for multimodal data aggregation is proposed, enabling the fusion of raw log and metric streams into a single feature vector of fixed dimensionality. The application of a sliding window approach mitigates the temporal synchronization challenge: metrics and asynchronous events are consolidated within a unified interval and projected onto a compact feature vector. Min-max normalization was applied to the dynamic range of metrics to transform all values into a standardized [0,1] interval. Concurrently, to convert the input log stream into numerical features, severity-based categorization was applied, assigning weights according to the severity level of the events. Furthermore, evaluating the overall chaos level through entropy calculation facilitated the detection of hidden and atypical anomalies.

To validate the performance of the proposed method, empirical evaluation utilizing a Random Forest classifier was conducted against isolated baselines. The quantitative results confirmed the effectiveness of the proposed fusion vector, achieving a maximum F1-Score of 0.867 and a ROC-AUC of 0.997, significantly outperforming metric-only and log-only approaches. Furthermore, heat maps and temporal correlation visualizations provided clear supporting evidence of the method's capability to cluster anomalous states accurately, validating the suitability of the final vector for downstream classifiers.

Despite the optimistic results, there are certain limitations inherent to the current evaluation. The validation relies on a controlled, synthetic dataset to establish a mathematical ground truth, which does not fully capture the non-deterministic baseline noise of real-world systems. Furthermore, the use of a statically fixed temporal window may be overly rigid for highly dynamic workloads. Consequently, specific directions for future research focus on three areas: deploying the pipeline in real-world, high-load production environments to test robustness against overlapping multi-fault scenarios; implementing adaptive parameter tuning to dynamically adjust the aggregation window size based on real-time event frequency; and fully integrating this pre-processing pipeline with advanced deep learning models, specifically adaptive neuro-fuzzy inference systems (ANFIS), to map complex, non-linear correlations in real-world anomaly detection.

References

[1] Chansarkar, A. (2025). OpenSearch at Scale: Architecting High-Performance Distributed Search Solutions for Enterprise Data Retrieval. *World Journal of Advanced Research and Reviews*, *26*(2), 2088–2095. https://doi.org/10.30574/wjarr.2025.26.2.1851

[2] De Campos Souza, P. V., Guimarães, A. J., Rezende, T. S., Silva Araujo, V. J., & Araujo, V. S. (2020). Detection of Anomalies in Large-Scale Cyberattacks Using Fuzzy Neural Networks. *AI*, *1*(1), 92–116. https://doi.org/10.3390/ai1010005

[3] Decker, L., Leite, D., Giommi, L., & Bonacorsi, D. (2020). Real-Time Anomaly Detection in Data Centers for Log-based Predictive Maintenance using an Evolving Fuzzy-Rule-Based Approach. *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 1–8. https://doi.org/10.1109/FUZZ48607.2020.9177762

[4] Elradi, M. D. (2025). Prometheus & Grafana: A Metrics-focused Monitoring Stack. *Journal of Computer Allied Intelligence*, *3*(3), 28–39. https://doi.org/10.69996/jcai.2025015

[5] Gui, J., Ma, Z., Zhou, H., Su, Y., Zhang, M., Yu, K., & Wu, X. (2025). Deep anomaly detection of temporal heterogeneous data in AIOps: A survey. *Frontiers of Information Technology & Electronic Engineering*, *26*(9), 1551–1576. https://doi.org/10.1631/FITEE.2400467

[6] Liu, X., Liu, Y., Wei, M., & Xu, P. (2024). LMGD: Log-Metric Combined Microservice Anomaly Detection Through Graph-Based Deep Learning. *IEEE Access*, *12*, 186510–186519. https://doi.org/10.1109/ACCESS.2024.3481676

[7] Loboda, P., Starovit, I., Shushura, O., Havrylko, Y., Saveliev, M., Sachaniuk-Kavets'ka, N., Neprytskyi, O., Oralbekova, D., & Mussayeva, D. (2023). Ventilation control of the new safe confinement of the Chornobyl nuclear power plant based on neuro-fuzzy networks. *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, *13*(4), 114–118. https://doi.org/10.35784/iapgos.5375

[8] Ma, X., Li, Y., Keung, J., Yu, X., Zou, H., Yang, Z., Sarro, F., & Barr, E. T. (2024). *Practitioners' Expectations on Log Anomaly Detection* (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2412.01066

[9] Miroshnyk, M., Shmatkov, S., Strilets, V., & Zats, O. (2025). Investigation of computer systems to detect intrusions and network anomalies. *Bulletin of V.N. Karazin Kharkiv National University, Mathematical Modeling. Information Technology. Automated Control Systems*, (65), 67–82. https://doi.org/10.26565/2304-6201-2025-65-06

[10] Rishniak, M., & Opirskyy, I. R. (2025). Hybrid Behavioural Analysis Method for Early Detection of Anomalous Activity in Web Applications. *Advances in Cyber-Physical Systems*, *10*(2), 178–183. https://doi.org/10.23939/acps2025.02.178

[11] Savenko, B., Kashtalian, A., Lysenko, S., & Savenko, O. (2023). Malware Detection By Distributed Systems with Partial Centralization. *2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, 265–270. https://doi.org/10.1109/IDAACS58523.2023.10348773

[12] Viola, L., Ronchieri, E., & Cavallaro, C. (2022). Combining Log Files and Monitoring Data to Detect Anomaly Patterns in a Data Center. *Computers*, *11*(8), 117. https://doi.org/10.3390/computers11080117

[13] Wang, B., Zang, R., Guo, H., Zhang, S., Cao, S., Di, D., & Li, Z. (2025). Towards Multi-System Log Anomaly Detection. *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, 83–91. https://doi.org/10.18653/v1/2025.acl-industry.8

[14] Wang, F., Jiang, Y., Zhang, R., Wei, A., Xie, J., & Pang, X. (2025). A Survey of Deep Anomaly Detection in Multivariate Time Series: Taxonomy, Applications, and Directions. *Sensors*, *25*(1), 190. https://doi.org/10.3390/s25010190

[15] Wang, L., Zhao, N., Chen, J., Li, P., Zhang, W., & Sui, K. (2020). Root-Cause Metric Location for Microservice Systems via Log Anomaly Detection. *2020 IEEE International Conference on Web Services (ICWS)*, 142–150. https://doi.org/10.1109/ICWS49710.2020.00026

M.Sc. Andrii Mishchenko
e-mail: mishchenko.andrii.02@gmail.com

Master of Computer Science, Ph.D. student, Department of Digital Technologies in Energy, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine. Author of 1 scientific publications.

https://orcid.org/0009-0000-2100-8376

Prof. Oleksii M. Shushura
e-mail: leshu@i.ua

Doctor of Technical Sciences, professor of the Department of Digital Technologies in Energy, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine. Author of over 150 publications, including 1 monographs, 7 textbooks, more than 60 scientific articles in professional journals, of which 4 are in the scientometric databases Scopus and Web of Science.

https://orcid.org/0000-0003-3200-720X

D.Sc. Alona Kolomiiets
mail: alona.kolomiets.vnt@gmail.com

Doctor of Pedagogical Sciences associate professor, professor of Department of Higher Mathematics, Vinnytsia National Technical University.
Research interests: professional pedagogy, the fundamentalization of the mathematical training of future bachelors of technical specialties, mathematical modelling, methods of statistical analysis of experimental data.

https://orcid.org/0000-0002-7665-6247

Ph.D. Andrii G. Donets
e-mail: ogurman72@gmail.com

Candidate of Physical and Mathematical Sciences, associate professor of the Department of Digital Technologies in Energy, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine. Author of over 40 publications, including 1 monographs, more than 20 scientific articles in professional journals, of which 2 are in the scientometric databases Scopus and Web of Science.

https://orcid.org/0000-0002-8122-051X

Ph.D. Olena Kosaruk
e-mail: lena.kosaruk@vntu.edu.ua

Ph.D., associate professor, Faculty of Management and Information Security, Vinnytsia National Technical University, Vinnytsia, Ukraine. Author and co-author of more than 30 scientific publications.
Scientific interests of the leader: fuzzy modelling, digital economy, dual education, development of soft and hard skills.

https://orcid.org/0000-0003-1346-2944