AI-Oriented Monitoring Agent in Network Infrastructure of Multicloud Data Centers

Andrii Raichuk and Larysa Globa
Educational and Research Institute of Telecommunication Systems, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Industrial Lane, 2, 03056 Kyiv, Ukraine
raaukr@gmail.com

Keywords: Intelligent Network Monitoring, Data Centers, Event Log Analysis, Log Processing, Network Failures, Automated Diagnostics, AI Assistant.

Abstract: The growing complexity of multicloud data center infrastructures, together with the rapid increase in telemetry streams, event logs, and protocol notifications, makes timely incident detection and fault localization a critical problem for modern telecommunication systems. Conventional monitoring approaches mainly rely on the analysis of large historical logs, which limits their ability to identify short-term correlated failures and support rapid operational decision-making. The aim of this paper is to develop a software-oriented architecture for intelligent monitoring of data center network infrastructure based on localized telemetry buffering, centralized event correlation, and AI-assisted diagnostics. The proposed method combines local short-term telemetry snapshots formed on network and server nodes, critical-event clustering within a limited time window, root cause analysis using correlation of protocol, resource, and trap data, and risk estimation through a machine learning model. As a result, a five-layer monitoring architecture and an incident-processing algorithm were developed, including local agents, temporary telemetry buffers, a central correlation node, a failure knowledge base, and an AI assistant for troubleshooting support. The results show that the proposed approach reduces the amount of data requiring urgent processing, improves the identification of causally related events, and decreases the time needed to interpret critical incidents. The practical relevance of the study lies in the possibility of deploying the system as a software layer over existing monitoring and logging infrastructure without hardware modification. The theoretical relevance lies in the formalization of localized incident-context modeling, critical-event clustering, and AI-assisted root cause identification for multicloud data center environments.

1 INTRODUCTION

Modern data centers and telecommunication networks operate in an environment of continuously growing volumes of telemetry, logs, service events, and different types of messages. In multicloud infrastructures, this problem is intensified by the distributed nature of resources, the asynchrony of events, and complex cause-and-effect relationships between network and server components. Under such conditions, the traditional approach in which an operator or a monitoring system works with large-volume event logs is increasingly proving insufficient for rapid fault localization, since an incident is usually accompanied by an avalanche-like increase in the number of secondary messages, which complicates the identification of the root cause [1]-[5].

In data center network systems, the situation becomes particularly complex when a failure develops not in isolation, but through a cascade of interrelated changes: link degradation leads to the loss of routing protocol sessions, control plane overload, repeated route recalculation attempts, and the appearance of numerous trap messages. Under such conditions, the mere presence of a complete event log does not guarantee rapid diagnostics, because the practical problem lies not in the absence of data, but in their excess and insufficient structure for rapid decision-making [2], [4], [6].

At the same time, modern telecommunication systems are actively developing approaches related to AIOps, automated log analysis, anomaly detection, and root cause identification. Individual studies demonstrate the high efficiency of using machine learning for real-time telemetry processing, deep

learning for log analysis, as well as interpretable root cause analysis models for microservice systems [3], [7]-[9]. However, most such solutions either focus on a single data modality, or require significant computational resources, or do not take into account the operational specifics of the data center network environment, where response time, explainability of conclusions, and the possibility of software deployment without modifying the hardware infrastructure are critical [3], [7], [8],[10].

Given this, the task of building such a monitoring system is relevant, one that, on the one hand, does not duplicate the full analysis of all logs, and on the other hand ensures the rapid formation of incident context exactly when it is most needed. The aim of this work is to develop a software-oriented architecture of localized monitoring in which each network or server element forms a short-term telemetry buffer for the last N minutes, and in the event of a cluster of critical events such snapshots are transmitted to a centralized correlation node, where together with a knowledge base of failures and an AI assistant they are used to accelerate fault localization and troubleshooting. Such an approach makes it possible to reduce analysis time and decrease the volume of data that must be transmitted for processing and for the intelligent support of the engineer during incident response.

2 ANALYSIS OF RELATED RESEARCH

In the scientific literature, the problem of intelligent monitoring is divided into several interrelated areas. The first area covers fault management models in telecommunication networks. The survey in [1] on machine learning for network fault management shows that ML approaches are actively used for failure classification, anomaly detection, and failure prediction, while the results largely depend on the completeness of the training datasets, the quality of features, and the specifics of the application domain. An important conclusion of this line of research is that most solutions optimize a separate stage of the fault management chain - detection, classification, or prediction - but more rarely integrate these functions into a single operational mechanism for improving network fault tolerance [1], [11].

The second area concerns the analysis of event logs and system logs. A survey of deep learning methods for anomaly detection in log data demonstrates that modern methods based on LSTM, Transformer, and other sequential architectures significantly outperform classical statistical approaches in detecting complex patterns of anomalous behavior [2]. At the same time, the same survey emphasizes that log-oriented models depend on the quality of parsing, are highly sensitive to changes in message formats, and often transfer poorly to new environments without additional retraining [2]. Newer models, including LogLLM, attempt to reduce dependence on rigid template parsing and use the semantics of messages based on large language models, but this increases system complexity and the cost of operation [6]. Therefore, although AI-based log analysis is developing rapidly, the problem of rapid operational interpretation of an incident in a real data center environment remains open [2], [6].

The third area is related to real-time processing and streaming telemetry. In [3], anomaly detection in the telemetry of server farms is performed, showing that real-time operation is a critical requirement, that is, high model accuracy loses practical value if the detection time does not allow the operator to react in a timely manner. The authors propose special pre-processing of telemetry to reduce anomaly detection delay and prove that speed is a necessity rather than merely a derivative of classification accuracy [3]. For the architecture proposed in this paper, this observation is fundamental: the monitoring system should focus not on the maximum accumulation of historical data, but on the formation of a short yet informative incident context.

The fourth area covers root cause analysis in distributed systems and microservices. The LogRCA (Log Root Cause Analysis) and Chain-of-Event studies show that it is insufficient to simply isolate anomalous log lines in order to identify the root cause; it is much more important to find the minimal causally related set of events that explains the failure [4], [7]. It is especially important that the Chain-of-Event model uses causal relationships between events from multimodal sources and emphasizes the need for interpretability of the result for the engineer [7]. A survey of RCA (Root Cause Analysis) in microservices also confirms that one of the main challenges remains the integration of logs, metrics, traces, and domain expert information into a single data block for obtaining analysis results [5]. Thisconclusion directly leads to the need to use a knowledge base of failures and its processing by an AI assistant in the proposed architecture [4], [5], [7].

The fifth area is associated with the application of AI and its suitability for use in data centers. A study devoted to the automation of data center operations shows that even accurate ML models for predictive maintenance and other tasks lose part of their

practical value without an interpretative explanation for the operator [8]. This means that the engineer needs not only a signal about a danger, but also a reasoned supporting instruction: what exactly happened, which parameters are decisive, and which actions should be performed first. Therefore, within the proposed approach, the AI assistant is considered not as an autonomous mechanism for making decisions instead of the engineer, but as a tool for the intelligent support of diagnostics [8].

In summary, it can be concluded that modern studies sufficiently cover individual components of the problem - anomalies, correlation, RCA, AI support, and prediction - but do not fully take into account the specifics of the data center network environment, where large event logs effectively turn into a bottleneck during a failure. In most papers [1]-[5], [7], [8], [11], [12], the approach in which analysis starts not from a full archive but from a compact localized buffer of the node's latest state is insufficiently disclosed; situations of planned maintenance, when the interpretation of events must be deliberately changed or temporarily suppressed, are also rarely modeled. It is precisely these gaps that form the scientific novelty of the proposed study.

3 PROBLEM STATEMENT

The scientific problem considered in this work lies in the contradiction between the growth in the volume of events in the network infrastructure of data centers and the need to reduce fault localization time. On the one hand, modern systems accumulate huge arrays of data in centralized logs, which theoretically increases the completeness of the incident picture. On the other hand, the excessive amount and temporal-structural heterogeneity of these data complicate the search for the root cause at the moment when a decision must be made immediately. Therefore, the objective of this work is to develop a monitoring system architecture that reduces the amount of data subject to urgent processing without losing causally significant information.

Within this problem, three interrelated properties must be ensured. First, the local node must preserve sufficient telemetry context over a short time interval in order to reflect the dynamics of the incident up to the moment of its occurrence. Second, the central system must be able to determine exactly when these local snapshots need to be requested and combined. Third, the engineer must receive not a raw set of trap messages, but interpreted diagnostic support in the form of causes, possible consequences, and recommended actions. That is why the research task is formulated as the construction of an AI-oriented agent for localized monitoring and event correlation that combines buffering of node states, centralized correlation of clusters of critical events, and intelligent support for the fault tolerance of data center infrastructure.

4 ANALYSIS OF RELATED RESEARCH

The proposed architecture consists of five logically interconnected layers (Table 1). The first layer is formed by local monitoring agents deployed on network and server elements. Each agent collects key telemetry information and forms a temporary state snapshot file for the last N minutes. Such a snapshot includes event timestamps, types of trap messages, severity levels, the states of BGP, OSPF, LDP, and BFD sessions, interface statistics, CPU and memory parameters, as well as service control messages. For example, for a network node, a snapshot may contain a record of an LDP session loss on a specific interface, accompanied by increased CPU utilization and repeated restarts of an adjacent protocol process.

Table 1: Architecture of the AI-based data center monitoring system.

Layer	Components
Layer 1 - Network Infrastructure	Network devices (routers / switches)
Layer 2 - Local Monitoring Agents	Servers / Virtual nodes
Layer 3 - Temporary Telemetry Buffer	Local monitoring agent
Layer 4 - Central Correlation Node	Telemetry collection
Layer 5 - AI Assistant Layer	Trap monitoring

The second layer is the local temporary buffer. Its fundamental difference from the full system log is that it is not an archive but performs the function of an operational context. After a new snapshot is accumulated, the previous buffer may be overwritten or deleted, while the full history continues to be stored in the main logging file. Such an organization of the process reduces the overhead of urgent analysis, because in the event of an incident there is no need to urgently review the entire history of logs;

it is sufficient to collect the latest relevant context from several nodes.

The third layer consists of the data transmission activation mechanism. The system does not send local buffers to the central node continuously; transmission is initiated only when a cluster of critical events is detected. A cluster in this case is understood as the appearance of several correlated incidents within a short time in one or adjacent sections of the network. This logic makes it possible to significantly reduce the traffic of diagnostic data while at the same time preserving the ability to respond quickly.

The fourth layer is the centralized correlation node. Here, local N-minute snapshots from different elements are compared by time, event types, logical and topological relationships. The task of this node is not only to detect the fact of a failure, but also to reduce uncertainty about its root cause. Given the findings of RCA research, correlation should rely not on simple counting of messages, but on the search for causally connected chains of events [4], [5], [7], [13].

The fifth layer is the engineer's AI assistant. It uses a structured knowledge base of typical failures, historical event patterns, and machine learning mechanisms to generate supporting instructions. In the simplest case, the AI assistant correlates the current incident with known patterns such as "LDP Session Down + increased CPU load + interface flapping", and in a more complex case performs contextual generalization, offering the engineer a probable cause, a list of initial checks, and the recommended sequence of actions. If the engineer is working in planned maintenance mode and expects an impact on the network, AI analysis may be disabled or switched to passive mode in order to avoid false interpretations of events.

5 MATHEMATICAL FORMULATIONS

The network infrastructure described by the set of nodes $N = \{n_1, n_2, \ldots, n_M\}$, where each element corresponds to a router, switch, server, or virtual node. For node n_i at discrete time instants t, a state vector is formed that contains only the parameters required for local incident assessment and subsequent centralized correlation. The state vector includes the event timestamp, trap type, severity level, aggregated protocol state, normalized resource-load indicators, interface statistics, and service messages. This description makes it possible to transform raw telemetry data into a formalized node model.

Since the system uses a short-term local context, each node maintains a sliding buffer of duration Tb minutes. If Δt denotes the telemetry sampling interval, then the number of samples K in the buffer is defined by:

$$K = \frac{T_b}{\Delta t}. \quad (1)$$

The local buffer of node n_i is represented as the sequence of the latest K observations. Such a model preserves only the most relevant short-term context and avoids continuous processing of the full event log. The formalized state vector of node ni is defined in (2). For clarity, the interpretation of each component included in the vector is presented in Table 2.

$$x_{i(t)} = (\tau_i, \psi_i, s_i, p_i, c_i, m_i, q_i, u_i). \quad (2)$$

Table 2: Explanation of variables in formula (2).

Component	Explanation
τ_i	Time-related characteristic of node i, for example a timestamped temporal feature or a delay-related indicator.
ψ_i	Contextual or event-related descriptor associated with node i.
s_i	General state or status indicator of node i.
p_i	Protocol- or process-related feature of node i.
c_i	Control or computational characteristic of node i.
m_i	Memory-related characteristic of node i.
q_i	Queue-related metric, such as queue occupancy, backlog, or buffering state.
u_i	Utilization or load indicator of node i.

Temporal buffer or historical slice associated with node i at time t:

$$B_{i(t)} = \{x_{i(t-k\Delta t)}\}_{k=0}^{K-1}. \quad (3)$$

For primary local event filtering, the integrated criticality score $C_i(t)$ is introduced. It combines the normalized trap severity, protocol degradation, resource load, and interface-error intensity. The weights α, β, γ, and δ are selected so that their sum is equal to one.

$$C_{i(t)} = \alpha \cdot sev_{i(t)} + \beta \cdot p_{i(t)} + \gamma \cdot r_{i(t)} + \delta \cdot e_{i(t)}. \quad (4)$$

The protocol component pi(t) evaluates the loss of the key BGP, OSPF, LDP, and BFD sessions by means of indicator functions.

$$p_{i(t)} = \frac{I_{BGP_{i(t)}} + I_{OSPF_{i(t)}} + I_{LDP_{i(t)}} + I_{BFD_{i(t)}}}{4}. \quad (5)$$

The resource component $r_i(t)$ aggregates normalized CPU, memory, and traffic utilization, where $ci(t)=CPU_i(t)/100$, $m_i(t)=MEM_i(t)/100$, and $tri(t)=TR_i(t)/TR_{imax}$.

$$r_{i(t)} = \frac{c_{i(t)}+m_{i(t)}+tr_{i(t)}}{3}. \quad (6)$$

An event at node n_i is considered critical if the integrated score exceeds the threshold θ_c. In this case, the local agent marks the corresponding snapshot as relevant for transmission to the centralized node.

$$C_{i(t)} \geq \theta_c. \quad (7)$$

At the centralized level, the system processes not individual events but event sets observed in the interval $[t - T_w, t]$. The set of nodes with critical events defines the incident zone, while the incident cluster is formed only if the thresholds on both the number of events and the number of affected nodes are satisfied.

$$N_{c(t)} = \{n_i \in \mathcal{N} : \exists \tau \in [t - T_w, t], C_{i(\tau)} \geq \theta_c\}. \quad (8)$$

$$N_{evt(t)} \geq \theta_e \wedge |N_{c(t)}| \geq \theta_n. \quad (9)$$

For correlation, an integrated association measure between two events is used; it accounts for temporal proximity, topological adjacency, and semantic similarity of symptoms. This allows events belonging to the same cascade pattern to be grouped within a single incident context.

$$\kappa(e_a, e_b) = \lambda_1 C_{time} + \lambda_2 C_{topo} + \lambda_3 C_{sem}. \quad (10)$$

The root-cause search problem is formulated as the selection of the most probable cause from the set of possible causes R on the basis of the correlated symptom set E_t. In the general case, this formulation is expressed through posterior-probability maximization and Bayes' rule.

$$r_{opt} = argmax_{(r \in R) P(r|E_t)}. \quad (11)$$

$$P(r|E_t) = \frac{P(E_t|r)P(r)}{P(E_t)}. \quad (12)$$

Failure risk is estimated using logistic regression, where z(t) is the feature vector of the current state. The feature set includes protocol-state indicators, resource-load variables, interface-error counters, and the integrated event criticality score. The resulting failure probability is converted into a percentage risk score, which is used to signal increased network vulnerability even before a full incident cluster is formed.

$$P_{fail(t)} = \frac{1}{1+exp\left(-\left(w^T z(t)+b\right)\right)}. \quad (13)$$

$$Risk(t) = 100 \cdot P_{fail(t)}. \quad (14)$$

The mathematical description presented above formalizes the key components of the system: node-state representation, local telemetry buffering, criticality estimation, incident-cluster detection, centralized correlation, root-cause evaluation, and failure-risk prediction. This provides a rigorous formal basis for the software implementation of the intelligent monitoring system for data center network infrastructure.

6 SYSTEM OPERATION ALGORITHM

The algorithm of system operation begins with local telemetry collection on each element (Fig. 1). Unlike the continuous streaming of all data to the center, the agent selects only those parameters that directly reflect the functional state of the node. This makes it possible to preserve the local context in a compact form. After that, a temporary N-minute snapshot is formed, which is a copy of the most important parameters of the latest time interval. When a new snapshot appears, the old one is deleted or archived locally, while the main log file continues to accumulate the complete history.

At the next stage, the local agent performs a primary assessment of event criticality. If an event or a set of events belongs to the critical, alert, or error levels and appears on interrelated elements, the system initiates the transmission of local snapshots to the centralized node. Here, the advantage of the proposed approach lies in the fact that not the entire log is transmitted, but only a short relevant context sufficient for RCA. Thus, the time required to prepare the material for analysis is minimized already at the data collection stage.

After the data arrive at the centralized node, correlation analysis is performed. Its content consists in aligning time sequences and identifying logically related changes in protocol states, interface overload, growth of system resource consumption, and service trap messages. If, for example, packet loss, degradation of BFD status, and disappearance of BGP sessions are simultaneously observed on several nodes, the system should interpret this not as a set of independent symptoms, but as a common incident with a high probability of damage to the transport segment or degradation of adjacent active equipment.

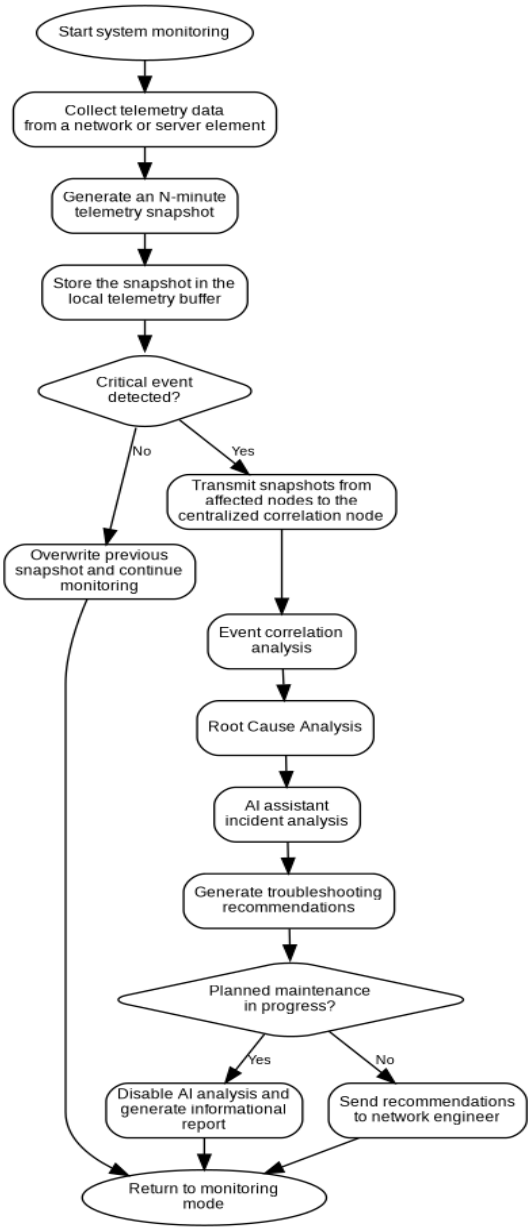

Figure 1: Operating algorithm of the incident analysis system based on telemetry.

The final stage is the work of the AI assistant. Based on the results of correlation and the knowledge base, it generates a short but technically meaningful conclusion. The conclusion must answer at least three questions: what is the most probable root cause, which checks should be performed first, and what consequences may arise if the incident is not isolated. Thus, the AI assistant transforms raw telemetry data into a recommendation for troubleshooting.

7 ADVANTAGES OF THE PROPOSED APPROACH

The main advantage is the reduction in incident analysis time due to shifting the focus from the complete historical log to a localized buffer of the latest state. In practice, this means that at the moment of an incident the engineer or analytical module works not with gigabytes of data, but with compact telemetry snapshots only from those nodes that are directly involved in the incident. This reduces both computational costs and the workload on personnel.

The second advantage lies in increasing the accuracy of event correlation. When the system receives a limited but temporally coherent context over the last N time intervals from several affected nodes, the correlation mechanism obtains a much cleaner basis for building cause-and-effect relationships than in the case of chaotic searching through the entire event log over a long period Ths is particularly important for cascading failure scenarios, where secondary symptoms quickly mask the root cause [4], [7], [12].

The third advantage is related to support for the engineer. In many modern approaches, the operator is effectively overloaded either with raw logs or with an overly abstract indicator that “an anomaly has been detected”. By contrast, the proposed AI assistant is oriented toward generating explainable supporting instructions tied to specific types of trap messages, protocol states, and resource symptoms. Such an organization is consistent with the modern trend toward explainable AIOps and increases the real operational value of the system [8], [11].

Finally, an important advantage is a higher degree of automation of the fault diagnosis process in the data center environment. Unlike approaches that require specialized hardware additions or substantial changes to network topology, the proposed system can be deployed as a software layer on top of the existing monitoring, logging, and network control infrastructure. This makes it particularly attractive for multicloud environments, where changes to the hardware platform are expensive or organizationally difficult.

8 RESULTS

The developed concept and its software prototype show that the localization of diagnostic information in the form of short-term buffers creates a fundamentally different mode of incident response.

If, in traditional practice, incident investigation begins with a centralized log of large volume, then in the proposed approach the system operates with a reduced but contextually relevant set of data. This means that the first step of diagnostics is performed faster, and the probability of losing a causally important fragment of information is lower than in the case of selective manual review of logs.

In terms of operational efficiency, this directly affects MTTR (Mean Time To Repair / Mean Time To Restore), that is, the average time required for recovery after a failure. The reduction of MTTR is achieved not only through faster data processing, but also through better support for decision-making. The engineer does not spend additional time manually collecting symptoms from different nodes, because the system already aggregates local snapshots and forms a preliminary interpretation of the incident. In cases of typical failures, the AI assistant effectively shortens the path from "identify what happened" to"check these specific points", which has a direct practical effect for NOC/SRE teams.

At the same time, the results should be considered critically. First, the effectiveness of correlation depends on the quality of the rules and the knowledge base model. Second, in complex multilayer failures some symptoms may fall outside the N-minute window, and therefore the choice of buffer duration requires further optimization for different types of network infrastructure. Third, in order to fully move to a proactive mode, the system should be supplemented with models for load prediction and degradation probability, combining short-term local diagnostics with longer time horizons of analysis [3], [10], [11].

At the same time, the results should be considered critically. First, the effectiveness of correlation depends on the quality of the rules and the knowledge base model. Second, in complex multilayer failures some symptoms may fall outside the N-minute window, and therefore the choice of buffer duration requires further optimization for different types of network infrastructure. Third, in order to fully move to a proactive mode, the system should be supplemented with models for load prediction and degradation probability, combining short-term local diagnostics with longer time horizons of analysis [3], [10],[11].

For multicloud environments, the proposed architecture is particularly promising because it combines decentralized collection of node states with centralized analytics. Local agents can be deployed in different management domains, and the central correlation node can operate as an overlay on top of existing logging and monitoring systems. In such a configuration, the autonomy of individual infrastructure segments is preserved, while at the same time a unified logic of incident response is formed. This corresponds to current trends in the development of intelligent network management systems, where local sensitivity to events is combined with global coordination of decisions [1], [5], [11].

9 CONCLUSIONS

The article proposes a software-oriented approach to intelligent monitoring of the network infrastructure of data centers, based on the combination of local N-minute telemetry buffers, centralized correlation of critical events, and an AI assistant for engineer support. Unlike the traditional heavily log-oriented approach, the proposed architecture operates with a localized incident context, which makes it possible to reduce analysis time, decrease the load on log-processing subsystems, and improve the quality of root-cause identification.

The scientific novelty of the work lies in the formalization of an architecture in which a short-term local state buffer is considered as the primary unit of analytical incident processing, rather than merely as an auxiliary cache for monitoring. An additional contribution lies in the integration of the AI assistant with a knowledge base of typical failures into the logic of network diagnostics and in taking into account the operational scenario of planned maintenance, when automatic interpretation of events must be adapted to the context of the engineer's actions.

Further research should be aimed at extending the mechanisms of causal analysis, automatically replenishing the knowledge base from accumulated incidents, integrating models for load forecasting and risk of failure, as well as experimentally validating the approach using real network logs of multicloud infrastructures. It is in this direction that the transition from reactive monitoring systems to full-fledged intelligent platforms for proactive management of next-generation telecommunication networks can be envisioned.

REFERENCES

[1] M. Nouioua, P. Fournier-Viger, G. He, F. Nouioua, and M. Zhou, "A survey of machine learning for network fault management," in Machine Learning and Data Mining for Emerging Trend in Cyber Dynamics, Springer, 2021, [Online]. Available: https://doi.org/10.1007/978-3-030-66288-2_1.

[2] M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, “Deep learning for anomaly detection in log data: A survey,” Machine Learning with Applications, vol. 12, art. no. 100470, 2023, [Online]. Available: https://doi.org/10.1016/j.mlwa.2023.100470.

[3] D. L. Vajda, T. V. Do, T. Bérczes, and K. Farkas, “Machine learning-based real-time anomaly detection using data pre-processing in the telemetry of server farms,” Scientific Reports, vol. 14, art. no. 23288, 2024, [Online]. Available: https://doi.org/10.1038/s41598-024-72982-z.

[4] T. Wittkopp, P. Wiesner, and O. Kao, “LogRCA: Log-based root cause analysis for distributed services,” in Euro-Par 2024: Parallel Processing, LNCS, vol. 14802, Springer, 2024, pp. 362-376, [Online]. Available: https://doi.org/10.1007/978-3-031-69766-1_25.

[5] T. Wang and G. Qi, “A comprehensive survey on root cause analysis in (micro) services: Methodologies, challenges, and trends,” arXiv preprint, 2024, [Online]. Available: https://arxiv.org/abs/2408.00803.

[6] W. Guan, J. Cao, S. Qian, J. Gao, and C. Ouyang, “LogLLM: Log-based anomaly detection using large language models,” arXiv preprint, 2024, [Online]. Available: https://arxiv.org/abs/2411.08561.

[7] Z. Yao, C. Pei, et al., “Chain-of-Event: Interpretable root cause analysis for microservices through automatically learning weighted event causal graph,” in Proc. ACM Found. Softw. Eng. (FSE), 2024, [Online]. Available: https://doi.org/10.1145/3663529.3663827.

[8] Y. Gebreyesus et al., “AI for automating data center operations: Model explainability in the data centre context using SHAP,” Electronics, vol. 13, no. 9, art. no. 1628, 2024.

[9] P. Himler, M. Landauer, F. Skopik, and M. Wurzenberger, “Anomaly detection in log-event sequences: A federated deep learning approach and open challenges,” Machine Learning with Applications, vol. 16, art. no. 100554, 2024, [Online]. Available: https://doi.org/10.1016/j.mlwa.2024.100554.

[10] L. Pham, H. Ha, and H. Zhang, “Root cause analysis for microservices based on causal inference: How far are we?,” in Proc. IEEE/ACM Int. Conf. Automated Software Engineering (ASE), 2024, pp. 706-718.

[11] E. Edozie et al., “Artificial intelligence advances in anomaly detection for telecom networks,” Artificial Intelligence Review, 2025, [Online]. Available: https://doi.org/10.1007/s10462-025-11108-x.

[12] I. Kotenko, D. Gaifulina, and I. Saenko, “Systematic literature review of security event correlation methods,” IEEE Access, vol. 10, pp. 43387-43420, 2022, [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3168976.

[13] H. Maosa et al., “A hierarchical security event correlation model for real-time detection,” Signals, vol. 4, no. 1, art. no. 4, 2024.