2025 International Conference on Cyber Resilience and Endogenous Safety & Security (CRESS) | 979-8-3315-5949-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/CRESS68073.2025.11452567

A Two-Stage Anomaly Detection Framework for 5G Core Networks Based on Multi-Source Data

Xinming Wang[12], Xingxing Liao[*2], Jie Yang[*3], Runhan Feng[2], Jingchen Xu[3]
[1]*School of Cyber Science and Engineering, Southeast University, Nanjing, China*
[2]*Purple Mountain Laboratories, Nanjing, China*
[3]*Information Engineering University, Zhengzhou, China*
wang_xm@seu.edu.cn, liaoxingxing@pmlabs.com.cn, yj_csu@126.com

***Abstract*—With the cloud-native transformation of 5G core networks, the complexity of network operations and maintenance (O&M) has increased significantly, accompanied by a growing number of targeted attacks. Traditional anomaly detection methods based on single data sources or specific anomaly types are no longer sufficient to cope with the diversity of abnormal scenarios. To address this challenge, this paper proposes a two-stage anomaly detection framework that leverages multi-source data fusion to enable accurate anomaly identification and root cause analysis in 5G core networks. The framework first collects heterogeneous data sources—including core network function logs, network traffic, and pod monitoring metrics—and constructs comprehensive features through preprocessing and temporal alignment. In the first stage, an XGBoost-based model integrates these multi-source features to detect overall anomalous network states. In the second stage, a weighted fusion of three single-source models is employed to perform fine-grained root cause analysis, identifying 43 specific anomaly types. These anomalies result from combinations of different network functions and various anomaly categories, including both internal failures of network functions and anomalies induced by external attacks. Experimental results demonstrate that the fused multi-source anomaly detection model achieves an F1-score of 0.9812, significantly outperforming single-source models. Furthermore, the root cause analysis stage achieves an F1-score of 0.9723 after weighted fusion, representing a 5.6% improvement in precision over the best-performing single-source model. This study provides a robust and effective approach to anomaly detection and O&M enhancement for 5G core networks.**

***Index Terms*—5G Core Network, Operational Maintenance, XGBoost, Anomaly Detection, Multi-Source**

I. Introduction

With the rapid advancement of 5G technology, the cloud-native transformation of the core network has become an inevitable trend in network evolution [1]. Cloud-native 5G core networks enable capabilities such as on-demand scaling, gray level upgrades, and flexible deployment, while achieving decoupling from the underlying infrastructure [2]. This architectural shift significantly enhances the agility, robustness, and elasticity of the network. However, transitioning from a highly integrated monolithic architecture to a multi-layer decoupled cloud-native architecture greatly increases the complexity of network operations and maintenance (O&M) [3]. The heightened technical complexity, diversification of fault scenarios, and heterogeneity of data types pose significant challenges for fault detection and localization [4].

Against this backdrop, traditional manual fault identification methods are increasingly inadequate due to their high costs and proneness to errors, making them unsuitable for the operational demands of cloud-native 5G core networks [5]. Virtualization within the 5G core has drastically increased software complexity, leading to a wide array of anomalies, including network function (NF) failures, abnormal inter-component interactions, network latency, bandwidth limitations, CPU overloads, and pod-level malfunctions in network functions [6]. These anomalies not only disrupt normal network operations but can also severely degrade user experience [7].

In recent years, telecom network failure incidents have surged globally. For instance, on April 18, 2021, Canadian mobile operator Rogers experienced a massive service outage that paralyzed communication services nationwide [8]. The root cause was a software issue during an update in the core network. This incident underscores the operational fragility of software-based and virtualized 5G core networks, highlighting the urgent need for comprehensive and efficient anomaly detection and root cause analysis mechanisms.

Existing research on anomaly detection still faces notable limitations. Most approaches rely on a single type of data—for example, using traffic data to detect external anomalies, monitoring metrics to detect internal faults, or logs to uncover software issues [9]–[15]. Single-source detection methods often fail to identify complex anomalies accurately. Furthermore, current studies tend to focus on a limited set of specific attacks, such as DDoS attacks on the N4 interface [16] or HTTP/2 multiplexing-based attacks [17]. These attacks may result in increased CPU load, network latency, or constrained bandwidth. However, such narrow approaches are insufficient to address the diverse and evolving anomaly scenarios in cloud-native 5G core networks.

To address the aforementioned challenges, we propose a two-stage anomaly detection framework based on multi-source data fusion, which enables root cause analysis and facilitates rapid identification of specific anomalies by O&M personnel.

This work was supported by the National Science and Technology Major Project of China (No. 2025ZD1303100) and the National Key Research and Development Plan of China (No. 2020YFB1806607, No. 2022YFB2902205).
Corresponding author: Xingxing Liao and Jie Yang.

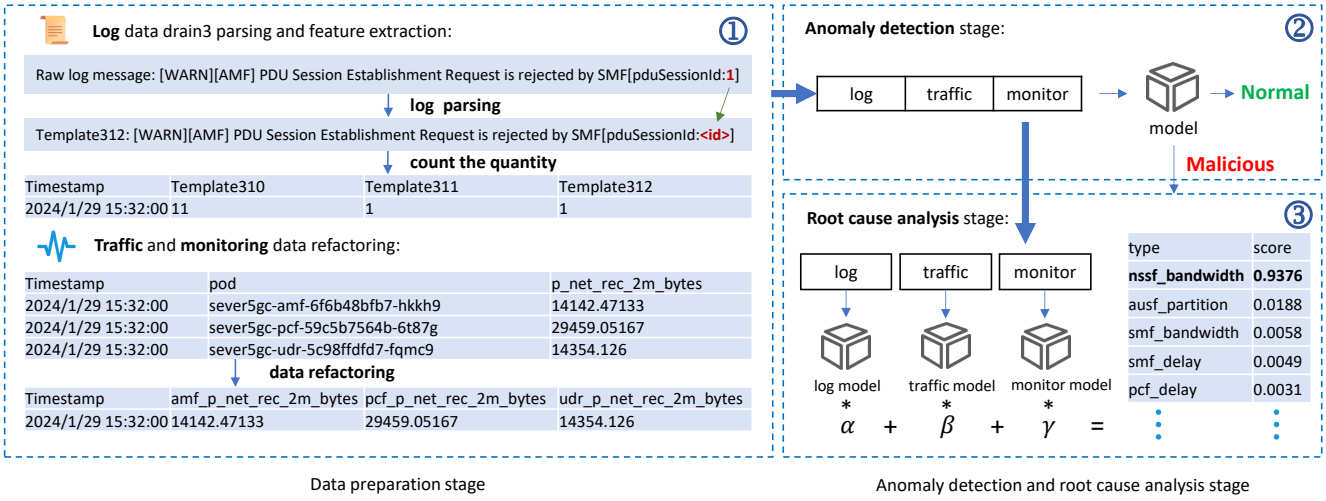

Fig. 1. Architecture of the two-stage anomaly detection framework based on multi-source data

The main contributions of this paper are summarized as follows:

- Comprehensive Anomaly Dataset: We utilize a diverse and representative anomaly dataset covering a wide range of scenarios. The anomalies span both internal failures of network functions and externally induced issues such as network latency caused by attacks. Each anomaly instance is accurately mapped to a specific network function and anomaly type, providing a rich data foundation for fault analysis in 5G core networks.
- Multi-Source Data Collection and Fusion: We collect heterogeneous data sources from the 5G core network, including network function logs, network traffic data, and pod-level monitoring metrics. This work introduces, for the first time in the context of 5G core networks, an anomaly detection approach based on multi-source feature fusion, overcoming the inherent limitations of single-source detection methods.
- Two-Stage Anomaly Detection Framework:In the first stage, features from different data sources are fused to train a model capable of detecting the overall anomalous state of the network.In the second stage, for samples identified as anomalous, the outputs of three single-source models—each classifying 43 distinct anomaly types, defined by combinations of network functions and anomaly categories—are combined using a weighted fusion strategy. The final classification is determined by selecting the result with the highest weighted score, thereby enabling accurate root cause localization.

The remainder of this paper is organized as follows. Section II describes the proposed two-stage anomaly detection framework based on multi-source data fusion. Section III presents the experimental results for anomaly detection and root cause analysis. Finally, Section IV concludes the paper.

II. Related Work

With the cloud-native transformation of 5G core networks, anomaly detection has become a critical technology to ensure stable network operations. In recent years, extensive research has been conducted in this area, primarily focusing on anomaly detection methods based on single data sources or targeting specific types of anomalies.

Traffic-based detection methods primarily focus on identifying external anomalies. For instance, several studies detect DDoS and other external attacks by analyzing network traffic features [9]–[12]. These methods typically utilize statistical characteristics of traffic—such as packet size, frequency, and the distribution of source and destination addresses—to train classification models for anomaly identification. However, such approaches often fail to detect internal functional failures or interaction anomalies within the network, as these internal issues may not significantly alter external traffic patterns.

Monitoring-metric-based methods target internal anomalies within the network. Researchers collect resource consumption metrics such as CPU and memory usage to detect issues like code injection attacks targeting 5G cloud-native network functions (NFs) [13]. While effective at identifying anomalies related to resource overload or performance degradation, these methods struggle to detect logical or interaction faults—such as signaling failures between NFs—that may not immediately manifest in resource metrics.

Log-based anomaly detection methods rely on analyzing network function logs to uncover abnormal behavior. These logs capture a wide range of events across network functions, such as heartbeat messages between the UPF and SMF, or session establishment and release in the 5G core. Researchers apply natural language processing or deep learning models (e.g., LSTM) to parse and process logs for anomaly detection [14], [15]. However, logs are often highly unstructured, making

parsing and feature extraction challenging. Additionally, solely relying on logs risks overlooking anomalies not captured in log entries.

Regarding specific anomaly types, existing studies tend to focus on detecting a few well-known attacks. For example, detection algorithms have been developed specifically for DDoS attacks on the N4 interface [16] or HTTP/2 stream multiplexing attacks in the 5G core [17]. While effective for these particular threats, such methods are inherently limited in scope and are ill-suited to address the constantly emerging and complex anomaly scenarios in cloud-native 5G networks.

In summary, current research exhibits clear limitations in both data source utilization and anomaly coverage. Most existing approaches rely on a single data source, which makes it difficult to capture the multidimensional characteristics of complex network anomalies. Furthermore, many of these methods focus solely on binary anomaly detection, offering little support for fine-grained root cause analysis. Additionally, most models are designed for specific attack scenarios, leading to poor generalization and limited adaptability to diverse and evolving anomaly types in modern networks.

To address these challenges, some studies have begun exploring multi-source data fusion. For instance, Lin et al. [18] proposed a machine learning-based intrusion detection framework that combines system logs, packet flows, and host statistics. Their ensemble of XGBoost models demonstrates the advantage of leveraging heterogeneous data sources, achieving high detection accuracy with an F1-score of up to 0.99. However, their approach is still limited to binary classification of benign versus malicious behavior and does not provide fine-grained anomaly categorization or root cause analysis. Moreover, it is worth noting that their study was conducted in a general IT network context rather than within the 5G core network, which poses distinct architectural and operational complexities. As such, the applicability of their method to 5G environments remains uncertain.

To overcome these limitations, this paper presents a two-stage anomaly detection framework based on multi-source data fusion. By integrating complementary information from logs, traffic data, and monitoring metrics, the proposed framework enables accurate anomaly detection in the first stage and fine-grained root cause classification in the second. This design significantly improves both detection accuracy and anomaly interpretability, offering enhanced generalization and practical value for anomaly detection in cloud-native 5G core networks.

III. Two-Stage Anomaly Detection Framework Based on Multi-Source Data

To improve the accuracy of anomaly detection and enable precise root cause localization in 5G core networks, we propose a two-stage anomaly detection framework that integrates heterogeneous data sources. The overall workflow includes data preparation, model training, anomaly detection, and root cause analysis. The system architecture is illustrated in Fig. 1. This flowchart in Fig. 2 provides a clear visual breakdown of the main components, their interactions, and the data flow within the system.

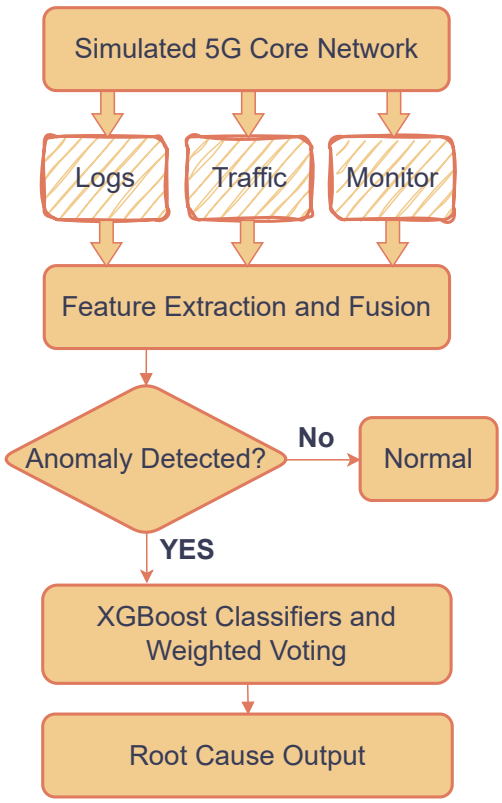

Fig. 2. Flow Chart of the two-stage anomaly detection framework

A. Overview

1) Data Preparation: A simulated 5G core network was deployed within a generic service environment constructed using Kubernetes (K8s) for container orchestration and OpenStack for virtual infrastructure management. This environment enables fine-grained visibility into control-plane and data-plane behaviors of 5G network functions. From this deployment, three heterogeneous types of data were collected: **Network Function Logs**, **Pod-Level Monitoring Metrics**, and **Network Traffic Data**. Each data source captures distinct aspects of system behavior and contributes complementary information for anomaly detection and root cause analysis. All data streams are aligned using a uniform 30-second time granularity, facilitating multi-source fusion during feature modeling.

- **Network Function Logs**: NF logs consist of timestamped system and application-level records emitted by 5G network functions, such as the AMF (Access and Mobility Management Function), SMF (Session Management Function), and UPF (User Plane Function). These logs capture control-plane events including user registration, PDU session establishment, handover signaling, authentication requests, and error reports. They provide rich

semantic context for identifying anomalies in protocol state transitions and operational workflows.

- **Pod-Level Monitoring Metrics**: Monitoring metrics are collected at the container (pod) level through Prometheus and related monitoring tools. These include resource utilization indicators such as CPU usage (e.g., CPU load, throttling rate), memory usage (e.g., working set size, memory limit), and network I/O statistics (e.g., transmitted and received bytes per second). These metrics reflect the internal performance status and resource consumption patterns of individual NFs, which are essential for detecting internal malfunctions such as resource leaks or overloads.
- **Network Traffic Data**: Traffic data is captured at virtual network interfaces associated with each NF, using tools such as tcpdump or veth-level monitors. It includes packet-level and flow-level features such as incoming/outgoing packet rates, packet size distributions, transport-layer protocol ratios (e.g., TCP vs. UDP), and connection churn. These metrics provide a low-level view of network behavior and are useful for identifying data-plane anomalies such as congestion, packet loss, or potential denial-of-service attacks.

2) Anomaly Detection and Root Cause Analysis: The preprocessed features from all sources are fused and used to train an XGBoost machine learning model. This model performs binary anomaly classification on incoming data. When an anomaly is detected, three additional single-source models—each trained on traffic, logs, or metrics—classify the anomaly into one of 43 specific types defined by combinations of network functions and anomaly categories. Their outputs are combined via a weighted voting mechanism to determine the final root cause.

B. Data Preparation

Different preprocessing strategies are applied based on the characteristics of each data source:

Log Data: We use Drain3, a tree-based log parser, to extract consistent templates from raw logs [19]. Dynamic parameters in logs—such as IP addresses, IMSI, UUID, GUTI, and SUCI—are normalized using regular expressions and replaced with generic identifiers to reduce template fragmentation. A total of 582 distinct templates were extracted. We then count the occurrences of non-debug log templates within each time window and use this count as a log-derived feature.

Monitoring and Traffic Data: To address the limitation that each entry in the raw dataset represents a single timestamp for a single NE, we restructured the dataset by fusing features from different NEs at the same timestamp. This includes components such as NSSF, SMF, AUSF, PCF, UDM, NRF, UDR, and AMF. This method provides a more holistic view of network operations and the interdependence between components.

Since the three data sources have varying collection frequencies, we synchronize them using a fixed time window of 30 seconds. Within each 30s window, features from logs, metrics, and traffic are aggregated into a unified feature vector. This alignment ensures temporal consistency for model training and anomaly detection.

C. Anomaly Detection

In this stage, we evaluate the detection performance using seven data source combinations: single-source (traffic, logs, or metrics), dual-source (any two), and tri-source (all three). Hyperparameters of the XGBoost model—such as tree depth, learning rate, and regularization terms—are carefully tuned to address class imbalance and enhance model generalization.

To implement our detection model, we adopt **XGBoost (eXtreme Gradient Boosting)**, a scalable and efficient implementation of gradient boosting machines (GBMs) [20]. XGBoost builds an ensemble of decision trees in a sequential manner, where each new tree is trained to correct the residual errors from the ensemble of previous trees.

Formally, the prediction after t iterations is given by:

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i), \quad f_k \in \mathcal{F} \tag{1}$$

where $\mathcal{F}$ is the space of regression trees (i.e., CARTs), f_k is the k-th tree, and x_i is the feature vector for instance i.

The training objective at iteration t is defined as:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{2}$$

where $l(y_i, \hat{y}_i)$ is the loss function (e.g., logistic loss), and $\Omega(f_t)$ is a regularization term defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{3}$$

Here, T is the number of leaves in the tree, and w_j is the score on leaf j. XGBoost uses a second-order Taylor approximation of the loss function to optimize this objective:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2 \right] + \Omega(f_t) \tag{4}$$

where $g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ is the first-order gradient, and $h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ is the second-order gradient (Hessian). This formulation allows efficient tree construction through greedy splitting.

Experimental results show that the tri-source fusion achieves the highest F1-score among all combinations. Therefore, the final anomaly detection model uses the fused features from all three data sources as input and is trained using the XGBoost algorithm. This model captures traffic behaviors, system metrics, and log events comprehensively, enabling accurate anomaly detection.

D. Root Cause Analysis

Ensemble learning is a widely used machine learning technique that combines multiple models to make decisions [21]. Rather than relying on a single model, ensemble methods aim to reduce bias, variance, or improve generalization by leveraging the collective strengths of different models. Common ensemble strategies include bagging (e.g., Random Forest), boosting (e.g., AdaBoost, XGBoost), and stacking. Each technique has its strengths: bagging primarily reduces variance, boosting focuses on minimizing bias, and stacking combines multiple models' predictions through a meta-model to enhance overall performance.

In this work, we focus on two ensemble approaches: boosting and stacking. Boosting builds a sequence of weak learners, assigning different weights to each, and combines them iteratively to form a strong model. XGBoost, a highly efficient and accurate implementation of boosting, is used in our model construction phase to minimize prediction error and improve accuracy. On the other hand, stacking offers a more intuitive strategy by collecting predictions from multiple models and combining them to make the final decision.

We use boosting to train our models and adopt a stacking-based decision strategy for final predictions. Specifically, we train three independent models, each representing a distinct type of system or network behavior—network traffic features, NF log features, and monitoring metrics—to detect and classify anomalies.

Each model outputs a set of confidence scores corresponding to predefined anomaly categories. These outputs are then integrated using a weighted fusion strategy, where weights are assigned based on each model's historical accuracy on validation data. This ensures that models with higher reliability contribute more significantly to the final root cause determination.

By leveraging the complementary strengths of the three data sources, this ensemble-based approach enhances the precision of root cause localization. It provides network operators with actionable insights and facilitates efficient troubleshooting in complex 5G core network environments.

IV. Performance Evaluation

A. Dataset

The dataset used in this study was provided by China Mobile and focuses on fault detection and localization scenarios in cloud-native 5G networks [6]. It includes three core data types: log data, pod monitoring metrics, and network traffic data, all synchronized with a uniform 30-second time granularity. The complete feature compositions for pod metrics and network traffic are detailed in Tables VII and VIII respectively (see Appendix). The dataset covers multiple types of anomalies, as detailed in Table I introduces the different anomaly types, including delay, bandwidth, CPU, failure, partition, memory, and loss. Table II shows the full set of labels, which includes 43 anomaly types defined by combinations of network functions and anomaly categories, as well as the normal class. The dataset consists of 1,833 normal samples and 1,459 anomalous samples.These were split into training and testing sets with a 7:3 ratio for model development and evaluation.

TABLE I
Types of Anomalies in Network Functions

Anomaly Code	Anomaly Type
delay	Network Delay
bandwidth	Bandwidth Limitation
cpu	High CPU Load
failure	NF Fault
partition	Partition Fault
memory	Memory Anomaly
loss	Data Loss

TABLE II
Label Categories for Network Functions Anomalies

normal	ausf_bandwidth	ausf_delay	ausf_failure
ausf_loss	ausf_memory	ausf_partition	nrf_bandwidth
nrf_cpu	nrf_delay	nrf_loss	nrf_memory
nrf_partition	nssf_bandwidth	nssf_cpu	nssf_delay
nssf_failure	nssf_loss	nssf_memory	nssf_partition
pcf_cpu	pcf_delay	pcf_failure	pcf_loss
pcf_memory	pcf_partition	smf_bandwidth	smf_cpu
smf_delay	smf_failure	smf_loss	smf_memory
udm_bandwidth	udm_cpu	udm_delay	udm_loss
udm_memory	udm_partition	udr_bandwidth	udr_cpu
udr_delay	udr_loss	udr_memory	udr_partition

In the first stage of anomaly detection, all types of anomalous samples were uniformly labeled as "anomaly" to achieve holistic recognition of network abnormal states. In the subsequent root cause analysis stage, only the anomalous samples were used for training and testing the models, aiming at precise classification of specific anomaly categories, thus enabling an end-to-end process from anomaly detection to fault localization.The corresponding XGBoost hyperparameter settings for both stages are summarized in Table III.

TABLE III
Comparison of XGBoost Hyperparameters for Anomaly Detection and Root Cause Analysis

Task	Objective Function	epoches	Max Depth	Learning Rate
Anomaly Detection	binary:logistic	200	10	0.1
Root Cause Analysis	multi:softmax	200	6	0.1

B. Evaluation Setup

Model performance was assessed using precision, recall, and F1-score as core metrics. Precision measures the proportion of correctly detected anomalies among all detected anomalies, reflecting the accuracy of identification. Recall quantifies the proportion of true anomalies successfully detected, indicating the model's coverage capability. The F1-score, the harmonic mean of precision and recall, provides a balanced assessment of overall detection performance.

All evaluation experiments were conducted on a consistent hardware platform configured with an Intel i5-12500H proces-

sor, 16GB RAM, and a 1TB hard drive, ensuring stable and reproducible results.

C. Anomaly Detection Experiments (Binary Classification)

We first compared the performance of various models—Decision Tree, KNN, SVM, Random Forest, MLP, and XGBoost—on single data sources, as summarized in Table IV. Among the data sources, models trained on pod monitoring metrics consistently achieved better performance. The XGBoost model led with a precision of 0.9924, recall of 0.9155, and an F1-score of 0.9524. Models trained on network function (NF) logs performed moderately well, with XGBoost achieving an F1-score of 0.8624, thus outperforming other models. Although the MLP model achieved a precision of 0.9208 on logs, its recall was relatively low at 0.7352. For network traffic data, MLP attained the highest F1-score of 0.8728, while SVM showed high precision (0.9866) but low recall. XGBoost balanced precision (0.9494) and recall (0.7923) effectively. Overall, pod monitoring data best supports model performance, whereas NF logs and network traffic data yielded more variable results, especially in precision-recall trade-offs. Considering all three data sources, XGBoost was deemed the most effective model overall.

TABLE IV
PERFORMANCE METRICS OF DIFFERENT MODELS BY DATA SOURCE

Data Source	Model	*Precision*	*Recall*	*F1-Score*
Monitoring	Decision Tree	0.9520	0.9085	0.9297
	KNN	0.9773	0.9085	0.9416
	SVM	0.8926	**0.9366**	0.9141
	Random Forest	**0.9961**	0.8908	0.9405
	MLP	0.9667	0.9190	0.9422
	XGBoost	0.9924	0.9155	**0.9524**
Logs	Decision Tree	0.8434	0.8300	0.8367
	KNN	0.8734	0.7905	0.8299
	SVM	0.8415	0.8182	0.8297
	Random Forest	0.9062	0.8024	0.8512
	MLP	**0.9208**	0.7352	0.8176
	XGBoost	0.8974	**0.8300**	**0.8624**
Traffic	Decision Tree	0.8185	**0.8099**	0.8142
	KNN	0.9079	0.7641	0.8298
	SVM	**0.9866**	0.7782	0.8701
	Random Forest	0.9561	0.7676	0.8516
	MLP	0.9824	0.7852	**0.8728**
	XGBoost	0.9494	0.7923	0.8637

TABLE V
PERFORMANCE COMPARISON OF XGBOOST MODELS WITH DIFFERENT DATA SOURCE COMBINATIONS

Model Combination	*Precision*	*Recall*	*F1-Score*
Traffic + Logs + Monitoring	**0.9865**	**0.9759**	**0.9812**
Traffic + Monitoring	0.9863	0.9626	0.9743
Logs + Monitoring	0.9888	0.9465	0.9672
Monitoring	0.9886	0.9305	0.9587
Traffic + Logs	0.9563	0.8770	0.9149
Traffic	0.9164	0.8209	0.8660
Logs	0.8856	0.8075	0.8448

We also evaluated XGBoost performance on various data source combinations, as shown in Table V. Results indicate that multi-source data fusion significantly enhances model performance. Specifically, combining network traffic, NF logs, and pod monitoring metrics yielded the best results, with precision reaching 0.9865, recall 0.9759, and F1-score 0.9812—all surpassing other combinations. Two-source combinations (network traffic + pod metrics, NF logs + pod metrics) performed slightly worse, with the former achieving an F1-score of 0.9743 and the latter 0.9672. Among single data sources, pod monitoring metrics outperformed network traffic (F1-score 0.866) and NF logs (F1-score 0.8448). The fusion of network traffic and NF logs (F1-score 0.9149) fell between single-source and pod-including combinations. These results demonstrate that incorporating more data sources—particularly pod monitoring metrics—improves anomaly detection accuracy and completeness. This confirms that multi-source data fusion effectively integrates complementary information from diverse data types, thereby enhancing the reliability of 5G core network anomaly detection and validating the proposed approach. Consequently, models combining all three data sources were adopted in the first stage of the framework for anomaly detection.

D. Root Cause Analysis Experiments (Multi-class Classification)

In the root cause analysis stage, we conducted an in-depth examination of the identified anomalous dataset, which was further categorized into 43 distinct classes. Each class precisely specifies both the affected network function location and the corresponding anomaly type, enabling a fine-grained analysis process that spans from anomaly detection to root cause localization. The performance of XGBoost models under various data source configurations and weighted fusion strategies is summarized in Table VI, highlighting their effectiveness in accurately identifying these granular root causes.

TABLE VI
PERFORMANCE COMPARISON OF XGBOOST MODELS FOR ROOT CAUSE ANALYSIS ACROSS DIFFERENT DATA SOURCES AND WEIGHTED FUSION

Method	*Precision*	*Recall*	*F1-Score*
0.5*Monitoring + 0.25*Logs + 0.25*Traffic	**0.9755**	**0.9762**	**0.9723**
0.4*Monitoring + 0.3*Logs + 0.3*Traffic	0.9659	0.9658	0.9658
0.33*Monitoring + 0.33*Logs + 0.33*Traffic	0.9660	0.9658	0.9634
0.6*Monitoring + 0.2*Logs + 0.2*Traffic	0.9600	0.9555	0.9549
0.7*Monitoring + 0.15*Logs + 0.15*Traffic	0.9344	0.9384	0.9326
Monitoring	0.9203	0.9212	0.9157
Logs	0.7692	0.7397	0.7402
Traffic	0.6854	0.6678	0.6672

Comparisons of XGBoost models trained on individual data sources and their weighted fusion revealed the following: Among single-source models, the pod monitoring metrics model performed best, achieving precision, recall, and F1-score of 0.9203, 0.9212, and 0.9157, respectively. This is attributed to the detailed recording of container resource usage and operational status in pod monitoring data, providing direct insights for internal fault localization. In contrast, models trained on NF logs and network traffic data performed less

effectively, with F1-scores of 0.7402 and 0.6672 respectively, indicating limitations in capturing cross-NE correlated faults or complex chain anomalies from single data sources.

Applying ensemble learning with weighted fusion of the three data source models significantly improved root cause analysis performance. The optimal weighting scheme, consisting of 0.5 for monitoring data, 0.25 for logs, and 0.25 for traffic, achieved precision, recall, and F1-score values of 0.9755, 0.9762, and 0.9723, respectively, representing an approximate 5.6% improvement in F1-score over the best single-source model (pod monitoring). This result demonstrates that ensemble learning effectively integrates complementary information from different data sources—monitoring data reveals internal NF states, logs capture business process anomalies, and traffic data reflects network interaction issues. Through appropriate weighting, the model leverages these advantages to more accurately locate specific NF faults and anomaly types, validating the effectiveness of the proposed multi-source ensemble strategy in root cause analysis.

V. CONCLUSION

This work addresses anomaly detection and root cause analysis in the context of cloud-native 5G core networks, aiming to overcome the limitations of traditional single-source detection methods and enhance the accuracy and efficiency of network fault localization. The core contributions of this study include three aspects: First, a comprehensive anomaly dataset was constructed, covering a wide range of scenarios, including internal network function faults (e.g., increased CPU load, memory anomalies) and externally induced anomalies (e.g., network latency, bandwidth constraints), with precise mappings to specific network functions and anomaly types to support robust model training. Second, an innovative multi-source data fusion approach was proposed, integrating core network function logs, network traffic data, and pod monitoring metrics, thereby breaking through the bottleneck of limited information dimensions inherent in single-source data. Third, a two-stage anomaly detection framework was designed, where the first stage employs an XGBoost model to fuse multi-source features for holistic anomaly recognition, and the second stage uses weighted fusion of outputs from three single-source models to classify 43 specific anomaly categories, facilitating precise root cause localization.

Performance evaluation validates the effectiveness of the proposed approach: in the anomaly detection stage, the tri-source fusion model achieved an F1-score of 0.9812, significantly outperforming single- or dual-source combinations; in the root cause analysis stage, applying a weighting strategy of "0.5 * monitoring + 0.25 * logs + 0.25 * traffic" improved the model's F1-score to 0.9723, representing an approximate 5.6% increase over the best single-source model.

By leveraging multi-source data fusion and a two-stage framework, this study realizes a comprehensive, fine-grained analysis pipeline from detection to root cause localization in 5G core networks, providing network operators with an efficient tool for fault diagnosis and contributing to the stable operation of 5G core infrastructures. Future work will extend the range of anomaly types covered and explore more adaptive weighting strategies to further enhance the framework's generalization capability.

REFERENCES

[1] Aiello, Samuel. "5g cloud-native network functions security risks in public clouds." Available at SSRN 4182073 (2022).

[2] Zhang, Xue, Huanan Li, and Dong Wang. "A Novel 5G-advanced Core Network Intelligent Operation and Maintenance System." In Journal of Physics: Conference Series, vol. 2289, no. 1, p. 012008. IOP Publishing, 2022.

[3] Passas, Virgilios, Nikos Makris, Yue Wang, Apostolos Apostolaras, Asterios Mpatziakas, Anastasios Drosou, Thanasis Korakis, and Dimitrios Tzovaras. "Artificial Intelligence for network function autoscaling in a cloud-native 5G network." Computers and Electrical Engineering 103 (2022): 108327.

[4] Zhang, Xue, Huanan Li, and Dong Wang. "A Novel 5G-advanced Core Network Intelligent Operation and Maintenance System." In Journal of Physics: Conference Series, vol. 2289, no. 1, p. 012008. IOP Publishing, 2022.

[5] Zhou, Shiyu, Yong Wang, Xiqing Liu, Xinzhou Cheng, Zhenqiao Zhao, and Tian Xiao. "A method of 5G core network service fault diagnosis." In International Conference On Signal And Information Processing, Networking And Computers, pp. 902-910. Singapore: Springer Nature Singapore, 2022.

[6] China Mobile. Cloud-Native 5G Network Fault Detection and Localization. China Mobile, 2024. https://doi.org/10.12448/7yj0-c813.

[7] Settembre, Marina. "A 5g core network challenge: Combining flexibility and security." In 2021 AEIT International Annual Conference (AEIT), pp. 1-6. IEEE, 2021.

[8] Global News. "Rogers Outage: What We Know so Far about the Service Disruption across Canada." Global News, April 19, 2021. https://globalnews.ca/news/7768754/rogers-outage-canada-customers-internet-phone/.

[9] Park, Seongmin, Byungsun Cho, Dowon Kim, and Ilsun You. "Machine learning based signaling ddos detection system for 5g stand alone core network." Applied Sciences 12, no. 23 (2022): 12456.

[10] Haider, Usman, Muhammad Waqas, Muhammad Hanif, Hisham Alasmary, and Saeed Mian Qaisar. "Network load prediction and anomaly detection using ensemble learning in 5G cellular networks." Computer communications 197 (2023): 141-150.

[11] Sood, Keshav, Mohammad Reza Nosouhi, Dinh Duc Nha Nguyen, Frank Jiang, Morshed Chowdhury, and Robin Doss. "Intrusion detection scheme with dimensionality reduction in next generation networks." IEEE Transactions on Information Forensics and Security 18 (2023): 965-979.

[12] Kuadey, Noble Arden Elorm, Gerald Tietaa Maale, Thomas Kwantwi, Guolin Sun, and Guisong Liu. "DeepSecure: Detection of distributed denial of service attacks on 5G network slicing—Deep learning approach." IEEE Wireless Communications Letters 11, no. 3 (2021): 488-492.

[13] A. S. M. Asadujjaman, M. E. Kabir, H. Purohit, S. Majumdar, L. Wang, Y. Jarraya, and M. Pourzandi, "5gfiver: Functional integrity verification for 5G cloud-native network functions," in IEEE International Conference on Cloud Computing Technology and Science, 2022, pp. 162–169.

[14] Tan, Yawen, Jiadai Wang, Jiajia Liu, and Yuanhao Li. "Deep learning-based log anomaly detection for 5G core network." In 2023 IEEE/CIC International Conference on Communications in China (ICCC), pp. 1-6. IEEE, 2023.

[15] Vinh, Tran Quang, Dinh Viet Quan, and Truong Thu Huong. "An Efficient Lightweight Anomaly Detection for 5G Core Network." In 2024 IEEE International Conference on Consumer Electronics (ICCE), pp. 1-6. IEEE, 2024.

[16] Radoglou-Grammatikis, Panagiotis, George Nakas, George Amponis, Sofia Giannakidou, Thomas Lagkas, Vasileios Argyriou, Sotirios Goudos, and Panagiotis Sarigiannidis. "5GCIDS: An Intrusion Detection System for 5G Core with AI and Explainability Mechanisms." In 2023 IEEE Globecom Workshops (GC Wkshps), pp. 353-358. IEEE, 2023.

[17] Wehbe, Nathalie, Hyame Assem Alameddine, Makan Pourzandi, and Chadi Assi. "5gshield: Http/2 anomaly detection in 5g service-based architecture." In 2023 IFIP Networking Conference (IFIP Networking), pp. 1-9. IEEE, 2023.
[18] Lin, Ying-Dar, Ze-Yu Wang, Po-Ching Lin, Van-Linh Nguyen, Ren-Hung Hwang, and Yuan-Cheng Lai. "Multi-datasource machine learning in intrusion detection: Packet flows, system logs and host statistics." Journal of information security and applications 68 (2022): 103248.
[19] He, Pinjia, Jieming Zhu, Zibin Zheng, and Michael R. Lyu. "Drain: An online log parsing approach with fixed depth tree." In 2017 IEEE international conference on web services (ICWS), pp. 33-40. IEEE, 2017.
[20] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785-794. 2016.
[21] Kabari, Ledisi G., and Ugochukwu C. Onwuka. "Comparison of bagging and voting ensemble machine learning algorithm as a classifier." International Journals of Advanced Research in Computer Science and Software Engineering 9, no. 3 (2019): 19-23.

APPENDIX

TABLE VII
POD MONITORING METRIC FEATURES

Feature	Description
p_cpu_avg_system_time	Average system CPU usage time
p_cpu_avg_throttle_pct	Percentage of throttled CPU time for the pod
p_cpu_avg_usage_pct	Pod CPU usage rate
p_cpu_avg_usage_time	Average container CPU usage duration
p_cpu_avg_user_time	Average user CPU usage time
p_file_desc_num	Number of open file descriptors in the container
p_fs_reads_2m_bytes	Increase in bytes read from disk/filesystem over 2 minutes
p_fs_reads_2m_num	Number of completed disk/filesystem reads over 2 minutes
p_fs_writes_2m_bytes	Total bytes written to disk/filesystem over 2 minutes
p_fs_writes_2m_num	Number of completed writes to disk/filesystem over 2 minutes
p_mem_avg_usage_5m_bytes	Average memory usage over the past 5 minutes
p_mem_avg_ws_5m_bytes	Average working set memory usage over the past 5 minutes
p_mem_failcnt_2m_times	Number of memory allocation failures over 2 minutes
p_mem_max_allocate_bytes	Maximum allocatable memory space
p_mem_max_usage_5m_bytes	Maximum memory usage over the past 5 minutes
p_mem_max_usage_bytes	Historical peak memory usage of the pod
p_mem_max_ws_5m_bytes	Maximum working set memory over the past 5 minutes
p_mem_min_usage_5m_bytes	Minimum memory usage over the past 5 minutes
p_mem_min_ws_5m_bytes	Minimum working set memory over the past 5 minutes
p_mem_rss_bytes	Actual physical memory used by the pod
p_mem_swap_bytes	Current virtual memory usage of the pod
p_mem_usage_bytes	Current memory usage of the pod
p_mem_usage_pct	Memory usage percentage
p_proc_num	Number of running processes in the container
p_sockets_num	Number of open sockets in the container
p_threads_max_num	Maximum allowed number of threads in the container
p_threads_num	Current number of threads running in the container

TABLE VIII
NETWORK TRAFFIC METRIC FEATURES

Feature	Description
p_net_rec_2m_bytes	Bytes received through the network interface in 2 minutes
p_net_rec_2m_pkt	Number of packets received in 2 minutes
p_net_rec_avg_2m_bytes	Average receiving rate (bytes) in 2 minutes
p_net_rec_avg_2m_pkt	Average receiving rate (packets) in 2 minutes
p_net_rec_drop_bytes	Number of dropped packets during receiving in 2 minutes
p_net_rec_err_bytes	Number of erroneous bytes received in 2 minutes
p_net_tran_2m_bytes	Bytes transmitted through the network interface in 2 minutes
p_net_tran_2m_pkt	Number of packets transmitted in 2 minutes
p_net_tran_avg_2m_bytes	Average transmitting rate (bytes) in 2 minutes
p_net_tran_avg_2m_pkt	Average transmitting rate (packets) in 2 minutes
p_net_tran_drop_bytes	Number of dropped packets during transmission in 2 minutes
p_net_tran_err_bytes	Number of erroneous bytes transmitted in 2 minutes