RAID: Root cause Anomaly Identification and Diagnosis

Joël Roman Ky[1,2](✉), Bertrand Mathieu[3], Abdelkader Lahmadi[1], Minqi Wang[3], Nicolas Marrot[3], and Raouf Boutaba[4]

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
abdelkader.lahmadi@loria.fr
[2] University of Luxembourg, Luxembourg joel.ky@uni.lu
[3] Orange Innovation, Lannion, France {bertrand2.mathieu, minqi.wang, nicolas.marrot}@orange.com
[4] David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada rboutaba@uwaterloo.ca

Abstract. Wi-Fi networks are widely used for modern connectivity but remain vulnerable to impairments such as bandwidth fluctuations, interference, packet loss and latency spikes. These challenges make it difficult to support latency-sensitive applications like Cloud Virtual Reality (Cloud VR), which offloads intensive computation to remote servers to reduce local hardware requirements but demands high throughput and ultra-low latency. Consequently, Wi-Fi network degradations can severely impact the Quality of Experience (QoE) of such applications. Traditional Root Cause Diagnosis (RCD) approaches rely on expert-defined rules or supervised ML (Machine Learning) models that require extensive labeled datasets. This dependence on manual labeling makes them costly, time-consuming, and impractical for real-world Wi-Fi diagnostics.

To overcome these limitations, we introduce RAID (Root cause Anomaly Identification and Diagnosis), a two-stage ML framework that diagnoses Wi-Fi performance issues using time series KPIs collected directly from the Wi-Fi access point, with Cloud VR serving as a use case. RAID combines contrastive learning-based anomaly detection with a lightweight classifier to categorize network impairments. We evaluate RAID, with a real-world Cloud VR use case, in a testbed using NVIDIA CloudXR and a Meta Quest 2, collecting Wi-Fi performance metrics on the access point, under controlled conditions. Results demonstrate that RAID outperforms existing RCD methods, achieving high accuracy even with minimal labeled data. Compared to conventional supervised and self-supervised time series models, RAID offers a scalable, real-time solution with a good trade-off between training efficiency and inference speed, making it well-suited for practical deployment in dynamic Wi-Fi network environments.

Keywords: Wi-Fi Networks · Root Cause Diagnosis · Cloud VR · Anomaly Detection · Contrastive Learning · Time Series Classification.

1 Introduction

Wi-Fi has become the dominant access technology for modern networks, offering flexibility and convenience. However, unlike wired connections, Wi-Fi is inherently unreliable due to environmental factors, interference from coexisting devices, bandwidth fluctuations, latency spikes, and packet loss. These impairments make it challenging to support the new generation of latency-sensitive applications, which demand both high throughput and ultra-low latency to maintain seamless performance. These emerging applications such as cloud gaming, telemedicine, cloud robotics, and Cloud Virtual Reality (Cloud VR) particularly suffer from these Wi-Fi limitations. Cloud VR, for instance, offloads intensive computation to remote servers, allowing for lightweight and cost-effective VR headsets. However, delivering high-resolution (4K-8K) immersive experiences requires substantial bandwidth ($\geq$ 80 Mbps) and ultra-low latency ($\leq$ 20 ms), making reliable performance over Wi-Fi networks a critical challenge. Network degradations in this context lead to lag, visual artifacts, and even cybersickness, ultimately disrupting immersive VR interactions.

Root Cause Diagnosis (RCD) plays a crucial role in identifying, predicting, and mitigating Wi-Fi-related network issues. Traditional RCD approaches rely on expert-defined heuristics to analyze Key Performance Indicators (KPIs). While useful in simple scenarios, these methods are manual, time-consuming, and struggle to scale in modern dynamic wireless environments. Recent advancements in Machine Learning (ML) and Time Series Classification (TSC) have enabled automated analysis of KPIs data, capturing temporal dependencies for improved anomaly detection. However, supervised TSC methods require large labeled datasets, which are costly and time-intensive to annotate, limiting their real-world applicability. As a result, there is a growing need for data-driven RCD approaches that reduce dependency on labeled data while maintaining high accuracy.

To address these challenges, we propose Root cause Anomaly Identification and Diagnosis (RAID), a two-stage ML framework for diagnosing Wi-Fi performance degradation, with Cloud VR used as a representative use case. In Stage one, a contrastive learning-based anomaly detection model differentiates normal from anomalous KPI patterns without requiring labeled samples. In Stage two, once anomalies are detected, a lightweight supervised classifier categorizes them into specific Wi-Fi impairments. We evaluate RAID using time series KPIs collected from a real-world Cloud VR testbed that emulates realistic network degradations under controlled conditions, using off-the-shelf devices and equipment, with user traffic generated by the Cloud VR game Beat Saber. This setup ensures that all collected data are real and representative of operational deployments. Although our experiments focus on Cloud VR, the RAID framework itself is domain-agnostic and can be readily applied to other root cause diagnosis scenarios by adapting the input KPIs. Our results demonstrate that RAID outperforms existing methods, even with limited labeled data, offering a scalable, efficient, and real-time root-cause diagnosis solution for Wi-Fi networks

supporting latency-sensitive applications. Specifically, the key contributions of this paper are as follows:

- We set up a controlled Wi-Fi testbed that faithfully replicates an operational network setup using the same commercial hardware. This environment can replicate real-world network impairments, enabling reproducible and realistic evaluation of Cloud VR performance under degraded conditions.
- We introduce a novel two-stage framework that combines contrastive learning-based anomaly detection with supervised classification to effectively detect and diagnose Wi-Fi impairments.
- We perform extensive empirical evaluations using time series KPI datasets collected from our testbed, comparing our proposed solution with state-of-the-art time series classification models.
- Our solution demonstrates strong performance even in low-label scenarios, highlighting its ability to generalize with minimal supervision. Additionally, it offers a balanced trade-off between moderate training time and low inference latency, making it well-suited for real-time deployment in practical Wi-Fi diagnostic applications.

2 Related work

ML-based Network Root Cause Diagnosis: Root Cause Diagnosis aims to identify the sources of network anomalies such as degraded performance or failures. The rise of ML has led to supervised and unsupervised approaches for network diagnosis. Supervised methods [10,21,11] have been used to troubleshoot Wi-Fi impairments [21] and classify home network issues using transformers [11]. Unsupervised approaches such as the two-stage VAE-MLP framework by Fida et al. [13] detect bottlenecks in cloudified 5G networks. Our work extends these efforts by incorporating contrastive learning for anomaly detection, reducing reliance on labeled data while enhancing root cause classification for low-latency Wi-Fi environments.

Time Series Classification: Time Series Classification (TSC) plays a key role in network diagnostics. Traditional approaches include distance-based [2], interval-based [9], shapelet-based [4], and ensemble-based [3] methods. More recently, deep learning architectures [31,30] have improved classification performance but require extensive labeled data. Self-Supervised Learning (SSL) has emerged as a scalable alternative, with frameworks like T-Loss [14], TNC [22], TS-TCC [12], TF-C [29] or TS2Vec [28] to further improve feature extraction by learning meaningful representations from unlabeled data. Our method leverages a two-stage TSC approach, where anomaly detection precedes classification, ensuring more accurate impairment diagnosis.

Anomaly Detection Techniques: Anomaly Detection (AD) methods span statistical models and deep learning approaches. Statistical methods include parametric (ARIMA [26], Gaussian-based [18]) and non-parametric (KDE [5]). Distance-based [6] and spectral-based [19] methods analyze distributional patterns, while isolation-based models [17] identify anomalies based on recursive

partitioning. Deep learning-based AD captures complex temporal dependencies using autoencoders (AEs, VAEs) [25], gaussian models [32], RNN-based methods [20], and transformer-based solutions [23]. Contrastive learning further enhances AD, with models like COCA [24], ContrastAD [16], and CARLA [8] improving representation learning, while DCdetector [27] refines spatial-temporal feature extraction. Our approach integrates contrastive learning in a two-stage RCD framework, effectively detecting and classifying impairments for real-time Cloud VR over Wi-Fi.

3 Proposed Method

We propose RAID, a two-stage root cause diagnosis framework for Cloud VR over Wi-Fi, formulated as a time series classification problem. Given a dataset $\mathcal{D}_{train} = \{(w_1, y_1), \ldots, (w_T, y_T)\}$ of multivariate time series KPIs, RAID consists of **1) an Anomaly Detection stage:** that identifies deviations from normal network behavior using contrastive learning and **2) a Root Cause Classification stage** that classifies detected anomalies into specific impairment types.

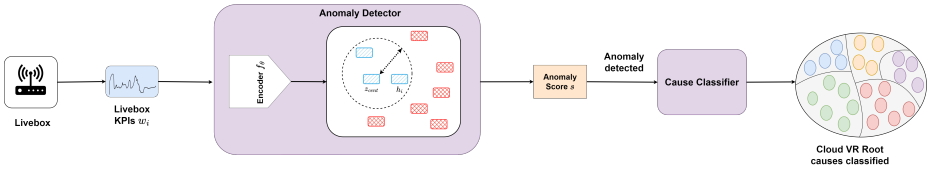


Fig. 1. RAID framework

3.1 Anomaly Detection Stage

The first stage identifies whether a time series is anomalous using a self-supervised anomaly detection approach based on contrastive learning. This approach eliminates the need for labeled data by leveraging only normal data collected during Cloud VR sessions without Wi-Fi impairments. Our anomaly detection module builds upon CATS (Contrastive learning for Anomaly detection in Time Series), a framework introduced in our previous work [15] that has demonstrated superior performance in time series anomaly detection. CATS enhances anomaly detection through synthetic anomaly generation and contrastive loss formulations, leveraging temporal dependencies to improve representation learning. Specifically, the anomaly detection model is trained using a combined global and temporal contrastive loss $\mathcal{L} = \frac{1}{2}(\mathcal{L}_{TCL} + \mathcal{L}_{GCL})$, ensuring robust detection of anomalous patterns.

We briefly summarize the components of the anomaly detector below, and refer the reader to [15] for a detailed analysis of its design choices and experimental validation.

- **Data augmentation:** From an input window w_i, it generates a set of time series views through positive data augmentations such as jittering and scaling $\{w_i^+, w_{i+N}^+\}$, and introduces synthetic anomalies via masking or trend perturbations w_i^-.
- **Encoder:** Maps the augmented time series into a low-dimensional latent space $h_i = f_\theta(w_i)$. The encoder architecture is model-agnostic, supporting convolutional, recurrent, or transformer-based models.
- **Projection head:** A non-linear MLP that refines the latent representations for contrastive learning $z_i = g_\theta(h_i)$.
- **Temporal contrastive loss:** Utilizes a differentiable variant of DTW (Soft-DTW) to learn temporal similarities in representations using a triplet of views.

$$\mathcal{L}_{TCL} = \frac{1}{N}\sum_{i=1}^{N} \max\left(D^\gamma(h_i^+ - h_{i+N}^+) - D^\gamma(h_i^+ - h_i^-) + m, 0\right) \tag{1}$$

  where $D^\gamma(.)$ is the Soft-DTW divergence measure, and m is the margin parameter (the minimum distance between positive and negative samples).
- **Global contrastive loss:** Uses a Normalized Temperature-Scaled Cross-Entropy Loss (NT-Xent) to learn global feature similarities with an extended set of negative pairs.

$$\mathcal{L}_{GCL} = \frac{1}{2N}\sum_{i\in\mathcal{B}^+} \log \frac{\exp(sim(z_i, z_{i+N})/\tau)}{\sum_{j\in\mathcal{B}, j\neq i}\exp(sim(z_i, z_j)/\tau)} \tag{2}$$

  where $\mathcal{B}$ is the set of all views, N is the batch size, τ is the temperature hyperparameter, and $sim(.)$ is the cosine similarity.
- **Anomaly scoring:** After training, anomalies are detected by calculating the distance between the latent representation of an unseen time series and the centroid of normal representations. If the score exceeds a predefined threshold, the instance is classified as anomalous.

$$s(\tilde{w}_t) = \mathcal{D}(f_\theta(\tilde{w}_t), z_{cent}) = \mathcal{D}(\tilde{z}_t, \frac{1}{N_{train}}\sum_{i=1}^{N_{train}} z_i) \tag{3}$$

  where $\mathcal{D}$ is the Euclidean distance.

3.2 Root cause classification

Once an anomaly is detected, the next step is to determine its underlying root cause. This stage is framed as a supervised classification problem, where the objective is to map each detected anomaly to a predefined class of root causes

$\{cause_1, cause_2, \ldots, cause_{K-1}\}$. To ensure efficiency and simplicity, we use a shallow classifier such as logistic Regression or SVM despite the various techniques for supervised TSC that were proposed in the literature. This approach, as shown in the following sections, achieves high accuracy with minimal computational overhead, making it suitable for real-time deployment.

4 Testbed

In this section, we introduce our testbed, designed for controlled experiments to assess Cloud VR performance over Wi-Fi while systematically injecting real-world network impairments.

4.1 Wi-Fi Testbed for Controlled Experiments

To systematically evaluate Wi-Fi's impact on Cloud VR performance, we developed a controlled Wi-Fi testbed (Fig. 2), designed to replicate real-world network impairments while maintaining precise experimental control. The testbed consists of two primary layers:

- **Infrastructure layer:** The infrastructure layer provides the core hardware setup for Cloud VR streaming and controlled Wi-Fi experimentation thanks to four Faraday cages used to isolate equipment from external electromagnetic interference. It comprises a Cloud VR system based on a CloudXR streaming setup, where a high-performance server (equipped with Intel Xeon W2235 CPU @ 3.8GHz, 32GB RAM, and NVIDIA RTX 3090 Ti) renders OpenVR applications using GPU acceleration and streams VR content wirelessly to a Meta Quest 2 headset (in cage 1). It also includes a Wi-Fi network environment that consists of two Wi-Fi APs: AP1 (in cage 2) that serves as the primary network for Cloud VR streaming and is used for both normal and coverage experiments and AP2 (in cage 4) that introduces network interference thanks to a traffic generator generating competing UDP to a station (in cage 3). The Faraday cages are interconnected using coaxial cables to transmit Wi-Fi signals, and Radio Frequency (RF) attenuators are employed to simulate variations in signal strength during experiments.
- **Control and Automation layer:** This layer ensures reproducibility and facilitates real-time monitoring, automation, and data collection. It includes a VR PC controller connected to the VR headset via USB, responsible for managing headset settings and collecting performance metrics, such as KPIs from OVR Metrics tools[5] or quality-of-service (QoS) statistics from CloudXR. The controller also automates game sessions using the Meta Quest Autodriver[6]. Additionally, this layer features an attenuator controller that configures and manages RF attenuators via APIs, enabling automated signal attenuation adjustments through FastAPI. Furthermore, the Livebox controller manages

[5] `https://developers.meta.com/horizon/downloads/package/ovr-metrics-tool/`
[6] `https://developers.meta.com/horizon/documentation/unity/ts-autodriver`

the Livebox via Telnet to collect Wi-Fi KPIs every 3 seconds. A local ELK Stack database aggregates data from all controllers for post-experiment analysis.

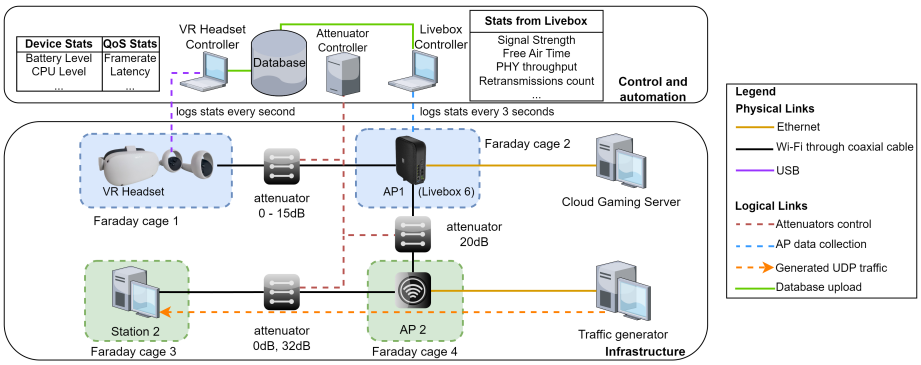


Fig. 2. Wi-Fi testbed for Cloud VR scenarios

4.2 Experimental Scenarios

Cloud VR experiments were conducted using the Beat Saber VR game as a benchmark across 2.4 GHz and 5 GHz Wi-Fi bands. Three experimental scenarios were evaluated:

- **Normal**: VR sessions under optimal conditions (RSSI: -45 dB (2.4 GHz), -65 dB (5 GHz), txops = 100%), with 5x 300s sessions per band;
- **Coverage**: Signal attenuation simulated via RF attenuators at different RSSI levels (-55 to -65 dB for 2.4 GHz and -80 to -90 dB for 5 GHz). Further degradation was limited by system constraints: VR disconnects below -65 dB for 2.4 GHz, and -65 dB was the highest achievable for 5 GHz.;
- **Interference**: Interference was introduced by the station connected to AP2, occupying 9% to 15% of the transmission opportunities (txops) available on AP1 once the game started.

4.3 Data Collection

With this testbed, three types of data can be collected: **i) Application-Level Metrics** that are extracted from the OVR Metrics Tool and CloudXR stack; **ii) Livebox-Level Metrics**: collected from the Wi-Fi AP, including RSSI, noise levels, airtime, MCS index, retry rates, and bitrate statistics; and **iii) Raw Traffic Captures**.

While the network impairments in our testbed are emulated in a controlled setting, the data used in this study is entirely real, collected directly from commercial hardware (Livebox, CloudXR stack, and Meta Quest 2) during representative Cloud VR sessions. The impairment scenarios are carefully designed to reproduce typical real-world conditions such as signal degradation, and interference. This setup enables reproducible experimentation while preserving the complexity and variability inherent to practical Wi-Fi deployments, thanks to the use of physical RF manipulation, real-time VR streaming, and traffic dynamics generated by actual application workloads, including interference created through competing traffic injected via a neighboring access point.

5 Evaluation setup

5.1 Dataset Description

To evaluate our proposed solution, we utilize the time series datasets collected from the experimental testbed. Although our setup gathers KPIs from both the VR headset and the CloudXR stack, which provide insights into QoS and QoE during VR sessions, this study focuses exclusively on data retrieved from the Livebox. This choice is motivated by the practical accessibility of these metrics for network operators, who own and manage the Livebox. Leveraging these metrics for RCD allows for the development of smarter APs and more intelligent network management solutions, aligning closely with the operational needs of network operators.

The dataset consists of 112 time series features extracted from the Livebox. These features include signal strength indicators (e.g., RSRP, RSSI), transmission performance metrics (e.g., txops), channel utilization measures (e.g., air time), among others, offering a comprehensive view of Wi-Fi performance in various conditions. Monitoring was performed at a frequency of one sample every three seconds. To facilitate analysis, the data is structured into overlapping time series windows, each spanning 10 time steps (30 seconds per window). In total, the dataset (presented in Table 1) contains 13,657 time series windows, which are partitioned into training and testing subsets using a 70:30 split ratio categorized into three classes, corresponding to distinct experimental scenarios during data collection.

Table 1. Dataset Summary

Class	**Train**	**Test**	**Features**	**Time Steps**
Normal	4718	1924	112	10
Coverage	2984	1270	112	10
Interference	1822	939	112	10
Total	9524	4133	112	10

5.2 Competing Solutions

To demonstrate the effectiveness of RAID, we compare it against several baseline including one-stage and two-stage TSC models.

One-Stage Models

- **1-NN-DTW:** A nearest-neighbor classifier with Dynamic Time Warping (DTW), a strong baseline for time series classification [2].

We also include SSL time series representation learning methods that undergo pretraining before classification with an SVM classifier with an RBF kernel following the protocol outlined in [14].

- **T-Loss [14]:** A SSL approach that uses triplet loss with time-based negative sampling for generalizable representations.
- **TS-TCC [12]:** A SSL framework that combines weak/strong augmentations with temporal/contextual contrastive learning.
- **TS2Vec [28]:** A framework that learns both instance-wise and temporal-wise representations via a hierarchical contrastive objective.

Two-Stage Models For the two-stage models, we replace RAID's anomaly detector with alternative unsupervised AD methods which are:

- **iForest [17]:** An isolation-based model that recursively partitions feature space to isolate anomalies.
- **USAD [1]:** Uses dual autoencoders in a min-max game with the first learns to reconstruct data and the second attempts to differentiate between true data and reconstructions.
- **SimCLR [7]:** A contrastive learning framework adapted for time series that learn representations from augmented views of data and can be used for AD.

5.3 Evaluation Metrics

We evaluate the performance of our root-cause diagnosis models using well-known multi-class classification metrics, including weighted Precision (P), weighted Recall (R), weighted F1-score (F1), Accuracy (Acc), and Normalized Accuracy (N-Acc). These metrics are defined as follows:

- **Precision:** The macro-weighted precision is the weighted average of precision values computed for each class, w_i being the proportion of class i:

$$P = \sum_{i=1}^{K} w_i \times P_i, \quad P_i = \frac{TP_i}{TP_i + FP_i} \tag{4}$$

– **Recall:** The macro-weighted recall is the weighted average of recall values computed for each class, w_i being the proportion of class i:

$$R = \sum_{i=1}^{K} w_i \times R_i, \quad R_i = \frac{TP_i}{TP_i + FN_i} \tag{5}$$

– **F1-Score:** The macro-weighted F1-score is the harmonic mean of macro-weighted Precision and Recall. This metric coupled P and R are suitable for imbalanced datasets.

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{6}$$

– **Accuracy:** The fraction of correctly classified samples over the total number of samples. This metric is widely used and easy to interpret.
– **Normalized Accuracy (N-Acc):** This metric adjusts the balanced accuracy ($bac = \frac{1}{K} \sum_{i=1}^{K} w_i \times R_i$) which is the weighted average of the recall of each class with respect to the accuracy of random guessing (bac_{RG}), ensuring that random predictions score 0 while perfect predictions score 1. It is more interpretable and suitable for imbalanced datasets.

$$\text{N-Acc} = \frac{bac - bac_{RG}}{1 - bac_{RG}} \tag{7}$$

5.4 Implementation Details

All datasets are normalized and split into training and testing sets. The architecture and hyperparameters of the anomaly detection stage in RAID are directly inherited from our prior work on CATS [15]. Specifically, we employ a dilated CNN with 10 residual blocks as encoder and a three-layer MLP with ReLU activations as projection head. As the anomaly detection module is reused without modification, we do not repeat the extensive evaluation conducted on CATS, which includes ablation studies on the loss components and augmentation strategies. This paper focuses instead on the integration of the detection module into a complete root cause diagnosis pipeline and its evaluation in a realistic Wi-Fi testbed setting. The model is trained for 100 epochs using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 512. Competing models are trained using their official implementations with consistent optimization settings. The classification stage is performed via an SVM with an RBF kernel, with hyperparameters tuned via grid search.

All experiments were conducted on an Ubuntu 22.04 with an AMD Ryzen 9 5900X CPU and an NVIDIA RTX 3090 Ti GPU (24GB), using PyTorch 2.2.0 and CUDA 12.1. The code and datasets to reproduce all experiments are publicly available.[7]

[7] `https://github.com/joelromanky/raid`

Table 2. Performance comparison on the datasets. Mean and standard deviation computed over five runs for Cloud VR datasets. Bold values indicate best results and underlined values the second best.

Models	Metrics	Accuracy	N-Accuracy	Precison	Recall	F1-score
One-stage	1-NN-DTW	51.54$_{(\pm 0.11)}$	26.36$_{(\pm 0.17)}$	56.74$_{(\pm 0.09)}$	51.54$_{(\pm 0.11)}$	52.96$_{(\pm 0.10)}$
	T-Loss	79.47$_{(\pm 4.39)}$	**75.22**$_{(\pm \mathbf{5.58})}$	**83.98**$_{(\pm \mathbf{4.74})}$	79.47$_{(\pm 4.39)}$	79.60$_{(\pm 4.53)}$
	TS2Vec	70.12$_{(\pm 5.28)}$	55.71$_{(\pm 6.53)}$	75.42$_{(\pm 3.22)}$	70.12$_{(\pm 5.28)}$	70.49$_{(\pm 4.88)}$
	TS-TCC	73.78$_{(\pm 6.38)}$	66.17$_{(\pm 8.12)}$	79.29$_{(\pm 6.07)}$	73.78$_{(\pm 6.38)}$	73.86$_{(\pm 6.68)}$
Two-stage	iForest	72.48$_{(\pm 2.69)}$	62.22$_{(\pm 4.69)}$	72.26$_{(\pm 3.14)}$	72.48$_{(\pm 2.69)}$	72.24$_{(\pm 2.99)}$
	USAD	72.22$_{(\pm 0.80)}$	63.39$_{(\pm 1.36)}$	72.72$_{(\pm 0.97)}$	72.22$_{(\pm 0.80)}$	72.38$_{(\pm 0.84)}$
	SimCLR	57.76$_{(\pm 3.25)}$	37.59$_{(\pm 4.84)}$	61.01$_{(\pm 2.60)}$	57.76$_{(\pm 3.25)}$	58.65$_{(\pm 3.06)}$
	RAID	**81.83**$_{(\pm \mathbf{2.96})}$	74.80$_{(\pm 4.19)}$	81.85$_{(\pm 3.02)}$	**81.83**$_{(\pm \mathbf{2.96})}$	**81.60**$_{(\pm \mathbf{3.05})}$

6 Results

6.1 Performance Evaluation

Table 2 summarizes the evaluation results of our solution compared to competing TSC methods using various performance metrics, including accuracy, normalized accuracy, precision, recall, and F1-score. The results highlight the superiority of our approach over both one-stage and two-stage methods.

Evaluation of One-Stage Models One-stage models, including 1-NN-DTW, T-Loss, TS2Vec, and TS-TCC, directly perform root cause classification without a preliminary anomaly detection step. Among these models, 1-NN-DTW exhibits the lowest overall performance, with an accuracy of 51.54% and an F1-score of 52.96%. Despite being a strong baseline for TSC, it struggles to handle the complex time series data encountered in CloudVR scenarios.

Contrastive learning-based SSL models outperform 1-NN-DTW. Among them, T-Loss emerges as the most effective technique, achieving the highest normalized accuracy (75.22%) and precision (83.98%) within this category. This demonstrates its capability to learn meaningful representations for RCD tasks. TS-TCC follows with an accuracy of 73.78% and an F1-score of 73.86%, while TS2Vec achieves an accuracy of 70.12% and an F1-score of 70.49%.

Evaluation of Two-Stage Models Two-stage models incorporate a preliminary anomaly detection step, enabling better focus on relevant patterns before root cause classification. iForest and USAD achieve comparable performance,

with accuracy scores of 72.48% and 72.22%, respectively. Both models demonstrate strong F1-scores around 72%, yet they fall short of advanced one-stage approaches like T-Loss. Meanwhile, SimCLR performs suboptimally with an accuracy of 57.76% and an F1-score of 58.65%.

Our custom solution significantly outperforms all competing methods across most metrics. It achieves the highest accuracy (81.83%), recall (81.83%), and F1-score (81.60%), demonstrating robustness and effectiveness for CloudVR RCD. While T-Loss marginally outperforms in normalized accuracy and precision, our custom model achieves the best balance across all metrics, establishing it as the most reliable approach in this evaluation.

The superior performance of our solution can be attributed to the efficiency of its anomaly detection stage. As shown in Fig. 3, our solution outperforms other two-stage techniques in detecting anomalies across various well-known metrics. RAID achieves the best overall anomaly detection performance, which directly contributes to its effectiveness in RCD tasks.

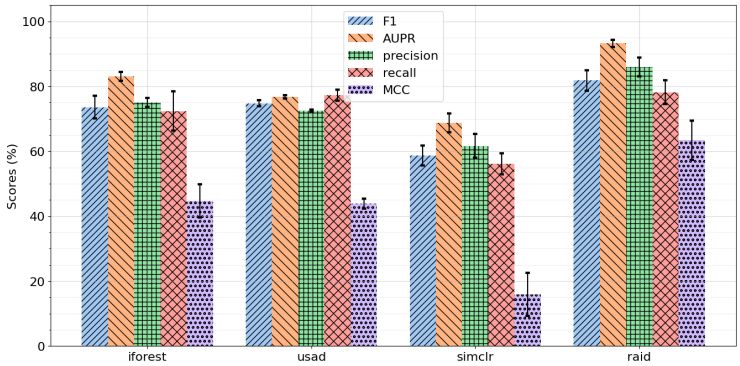
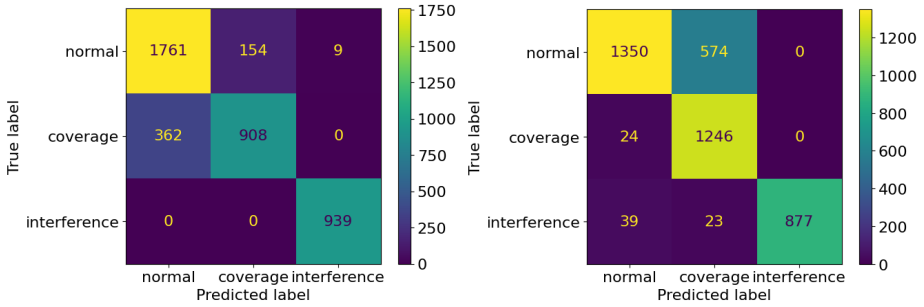


Fig. 3. Results of anomaly detectors of two-stage models.

Per-Class Performance Analysis Figures 4 and 5 provide a detailed comparison of the per-class performance metrics for T-Loss and our custom solution. Our approach demonstrates a significant advantage in efficiently distinguishing normal scenarios from both coverage and interference scenarios.

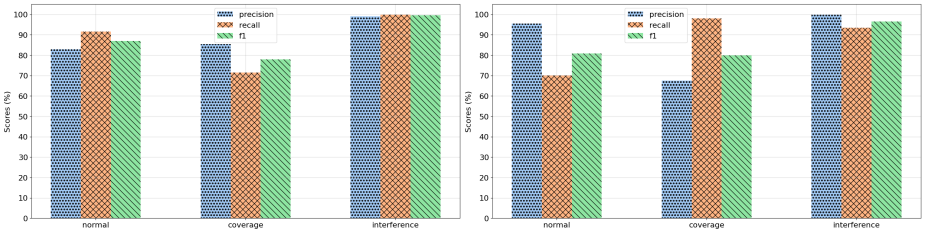
For normal scenarios, our solution achieves a notably lower misclassification rate compared to T-Loss, with 1,761 correctly classified normal samples versus 1,350 for T-Loss. This represents a substantial improvement in detecting normal behavior. Additionally, our solution attains perfect classification for interference scenarios, with a recall of 100%, highlighting its robustness in detecting distinct anomaly patterns such as interference.

However, Fig. 5 also reveals the limitations of our solution. It struggles to discriminate coverage scenarios, with a considerable number of coverage win-



(a) RAID **(b) T-Loss**

Fig. 4. Confusion matrix



(a) RAID **(b) T-Loss**

Fig. 5. Per-class precision, recall and F1-score.

dows misclassified as normal. This indicates challenges in capturing the subtle variations and transitional patterns between normal and coverage states. In contrast, T-Loss, while less accurate overall, shows a more balanced performance in handling coverage scenarios.

This trade-off underscores the strengths and weaknesses of our model: it is highly effective in detecting clear-cut anomalies but requires further refinement to enhance its sensitivity to nuanced variations between normal and coverage states. Future work could focus on addressing this limitation by incorporating advanced feature extraction techniques or domain-specific data augmentation strategies.

6.2 Efficiency with Few Labels

Fig. 6 illustrates the evolution of model performance as the percentage of labeled data increases. The left subplot depicts the accuracy scores across various label ratios, while the right subplot presents the corresponding F1-scores.

At the lowest label ratios (1%-5%), most models exhibit limited performance, reflecting the inherent difficulty of accurate RCD with minimal supervision. However, T-Loss and RAID stand out by achieving relatively higher accuracy and

F1 scores, showcasing their ability to generalize effectively even with sparse labeled data. T-Loss benefits significantly from its triplet-based pretraining strategy, which efficiently captures meaningful representations from the unlabeled dataset, thereby enhancing fine-tuning performance. Similarly, the pretraining stage of RAID contributes to its robustness in low-label scenarios by effectively leveraging the anomaly detection process to prioritize relevant patterns.

As the label ratio increases, all models demonstrate steady improvement in performance, highlighting the benefits of additional labeled data. Notably, RAID and T-Loss consistently lead in performance, with our solution exhibiting a steady performance boost. This consistency underscores the robustness of RAID across varying levels of supervision. While T-Loss initially competes closely, its performance shows a slight decline between the 5% and 20% label ratios, coupled with increased variability, indicating potential sensitivity to the quality or distribution of labeled data in these ranges.

The findings from Fig. 6 highlight the efficiency of RAID in leveraging limited labeled data, making it an ideal solution for real-world scenarios where labeling is both expensive and time-consuming. Its performance with sparse labels, along with its stable scalability as more labeled data becomes available, firmly establishes RAID as the most suitable model in this evaluation.

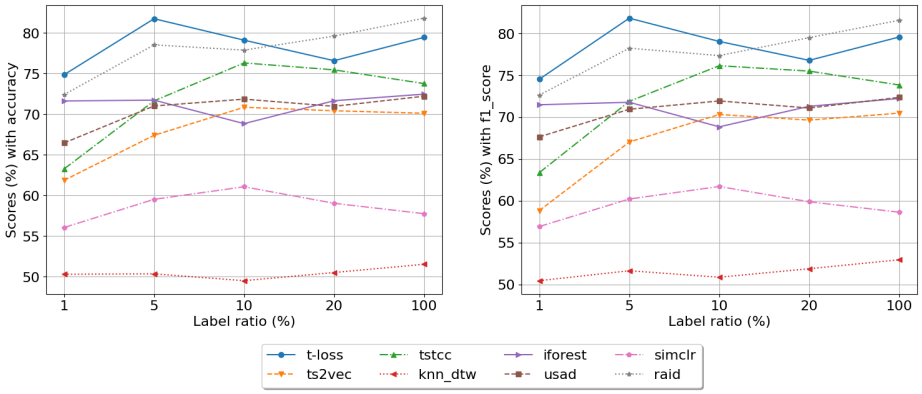


Fig. 6. Performance variation regarding the labels ratio.

6.3 Time complexity

Fig. 7 presents the training time (in seconds) and inference time per time series (in milliseconds) for each of the RCD models. The model with the longest training time is Triplet, which takes approximately 300 seconds, while the fastest training model is 1-NN-DTW, completing training in 500 milliseconds. In terms

of inference time, 1-NN-DTW significantly outpaces other models, with the highest inference time of 1800 milliseconds. In contrast, models such as Triplet or TS-TCC, achieve inference times as low as 0.5 milliseconds.

Our proposed solution, RAID, demonstrates a moderate training time of 200 seconds and an inference time of 3.5 milliseconds. While this inference time is the second highest among the models compared, it is still well-suited for real-time deployment, especially in our testbed where data is collected at frequent intervals (e.g., every 3 seconds). This makes RAID an excellent choice for RCD, as it balances moderate training overhead with sufficiently low inference latency, allowing for continuous monitoring and fast anomaly detection. Additionally, being a two-stage model, RAID offers a key advantage: when new causes or anomalies are detected, only the supervised classifier requires retraining. Most of the training time originates from the initial anomaly detection phase, unlike one-stage models that require complete retraining, including the pretraining phase. This makes RAID more efficient for scenarios requiring periodic updates or retraining, reducing overall downtime and resource consumption.

In summary, RAID strikes a practical balance between training efficiency and inference speed, making it highly effective for real-time RCD in dynamic and large-scale network environments.

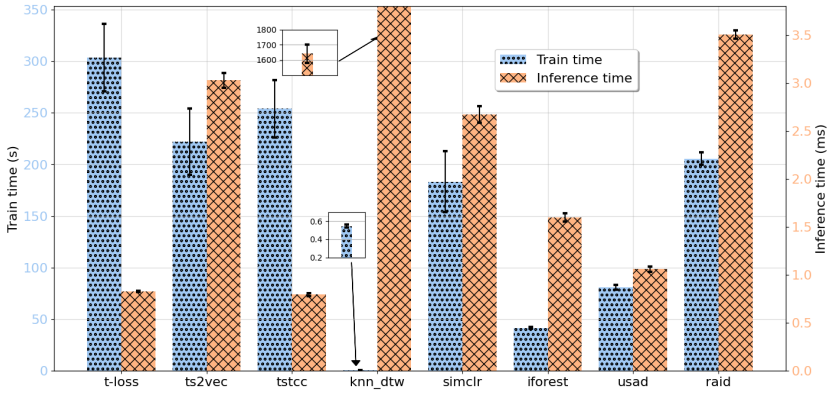


Fig. 7. Time complexity of each model.

7 Conclusion and Future work

This paper presents a root cause diagnosis approach for identifying network issues in Cloud VR sessions over Wi-Fi networks, utilizing time series KPIs collected from access points. By employing a two-stage ML framework, we demonstrated the effectiveness of our approach compared to traditional time series classification methods. Our proposed architecture, which integrates contrastive

learning into the anomaly detection process, has shown significant improvements in both identifying anomalies and diagnosing the root causes of Cloud VR performance issues. This provides a good foundation for future research in real-time diagnostics for cloud-based VR applications. One key strength of our approach is its ability to balance training time and inference speed, making it ideal for real-time diagnostics in dynamic network environments. Moreover, its two-stage design enhances efficiency by restricting retraining to the impairment classifier, thus avoiding full model retraining when new causes are introduced. Although our experimental evaluation focused on Cloud VR over Wi-Fi, the RAID framework is application-agnostic and can be seamlessly adapted to other root cause diagnosis scenarios. By replacing the time series inputs, RAID can be retrained without architectural changes.

Despite the promising results, there are several areas that warrant further exploration and improvement. First, while our model performs well in detecting clear anomaly patterns such as interference, its sensitivity to more nuanced variations—particularly in signal attenuation scenarios—needs to be enhanced. Future work will focus on improving the model's ability to detect subtle transitions between normal and degraded states. Second, scaling the solution to larger and more complex datasets is a priority. Our current framework has been tested in a controlled Cloud VR testbed with only two types of impairments. To improve its robustness, future research should include additional Wi-Fi impairments such as network congestion, hidden terminal issues, and non-Wi-Fi interference. Expanding the test environment to include more diverse real-world conditions, such as home networks where multiple sources of impairments may coexist, will also offer valuable insights. Finally, extending this two-stage model to a multi-modal diagnostic approach could provide a more comprehensive view of the root cause diagnosis process. By incorporating additional data sources, such as application-level performance metrics or raw network PCAP data, the system could offer even more accurate and proactive detection of network impairments. These enhancements will be crucial for addressing the growing demands of next-generation low-latency applications like Cloud VR.

Acknowledgments. This work is partially funded by a ANR - French government grant under the France 2030 program, project SPIREC of PEPR Cloud (ANR-23-PECL-0006) and the French National Research Agency (ANR) MOSAICO project, under grant No ANR-19-CE25-0012.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A.: Usad: Unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 3395–3404 (2020)

2. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data mining and knowledge discovery **31**, 606–660 (2017)
3. Bagnall, A., Lines, J., Hills, J., Bostrom, A.: Time-series classification with cote: the collective of transformation-based ensembles. IEEE Transactions on Knowledge and Data Engineering **27**(9), 2522–2535 (2015)
4. Bostrom, A., Bagnall, A.: Binary shapelet transform for multiclass time series classification. In: Big Data Analytics and Knowledge Discovery: 17th International Conference, DaWaK 2015, Valencia, Spain, September 1-4, 2015, Proceedings 17. pp. 257–269. Springer (2015)
5. Cao, V.L., Nicolau, M., McDermott, J.: One-class classification for anomaly detection with kernel density estimation and genetic programming. In: Genetic Programming: 19th European Conference, EuroGP 2016, Porto, Portugal, March 30-April 1, 2016, Proceedings 19. pp. 3–18. Springer (2016)
6. Chaovalitwongse, W.A., Fan, Y.J., Sachdeo, R.C.: On the time series k-nearest neighbor classification of abnormal brain activity. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **37**(6), 1005–1016 (2007)
7. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
8. Darban, Z.Z., Webb, G.I., Pan, S., Aggarwal, C.C., Salehi, M.: Carla: Self-supervised contrastive representation learning for time series anomaly detection. Pattern Recognition **157**, 110874 (2025)
9. Deng, H., Runger, G., Tuv, E., Vladimir, M.: A time series forest for classification and feature extraction. Information Sciences **239**, 142–153 (2013)
10. Dimopoulos, G., Leontiadis, I., Barlet-Ros, P., Papagiannaki, K., Steenkiste, P.: Identifying the root cause of video streaming issues on mobile devices. In: Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies. pp. 1–13 (2015)
11. Dötterl, J., Hemmati Fard, Z.: Classification of home network problems with transformers. In: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing. pp. 1081–1087 (2024)
12. Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C., Li, X., Guan, C.: Time-series representation learning via temporal and contextual contrasting. In: International Joint Conference on Artificial Intelligence (2021)
13. Fida, M.R., Ahmed, A.H., Dreibholz, T., Ocampo, A.F., Elmokashfi, A., Michelinakis, F.I.: Bottleneck identification in cloudified mobile networks based on distributed telemetry. IEEE Transactions on Mobile Computing (2023)
14. Franceschi, J.Y., Dieuleveut, A., Jaggi, M.: Unsupervised scalable representation learning for multivariate time series. Advances in neural information processing systems **32** (2019)
15. Ky, J.R., Mathieu, B., Lahmadi, A., Boutaba, R.: Cats: Contrastive learning for anomaly detection in time series. In: 2024 IEEE International Conference on Big Data (BigData). pp. 1352–1359. IEEE (2024)
16. Li, B., Müller, E.: Contrastive time series anomaly detection by temporal transformations. In: 2023 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2023)
17. Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 eighth ieee international conference on data mining. pp. 413–422. IEEE (2008)

18. Luo, H., Zhong, S.: Gas turbine engine gas path anomaly detection using deep learning with gaussian distribution. In: 2017 Prognostics and System Health Management Conference (PHM-Harbin). pp. 1–6. IEEE (2017)
19. Paffenroth, R., Kay, K., Servi, L.: Robust pca for anomaly detection in cyber networks. arXiv preprint arXiv:1801.01571 (2018)
20. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D.: Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2828–2837 (2019)
21. Syrigos, I., Sakellariou, N., Keranidis, S., Korakis, T.: On the employment of machine learning techniques for troubleshooting wifi networks. In: 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC). pp. 1–6. IEEE (2019)
22. Tonekaboni, S., Eytan, D., Goldenberg, A.: Unsupervised representation learning for time series with temporal neighborhood coding. arXiv preprint arXiv:2106.00750 (2021)
23. Tuli, S., Casale, G., Jennings, N.R.: Tranad: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284 (2022)
24. Wang, R., Liu, C., Mou, X., Gao, K., Guo, X., Liu, P., Wo, T., Liu, X.: Deep contrastive one-class time series anomaly detection. In: Proceedings of the 2023 SIAM International Conference on Data Mining (SDM). pp. 694–702. SIAM (2023)
25. Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., et al.: Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In: Proceedings of the 2018 world wide web conference. pp. 187–196 (2018)
26. Yaacob, A.H., Tan, I.K., Chien, S.F., Tan, H.K.: Arima based network anomaly detection. In: 2010 Second International Conference on Communication Software and Networks. pp. 205–209. IEEE (2010)
27. Yang, Y., Zhang, C., Zhou, T., Wen, Q., Sun, L.: Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 3033–3045 (2023)
28. Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., Xu, B.: Ts2vec: Towards universal representation of time series. In: AAAI Conference on Artificial Intelligence (2021)
29. Zhang, X., Zhao, Z., Tsiligkaridis, T., Zitnik, M.: Self-supervised contrastive pre-training for time series via time-frequency consistency. Advances in Neural Information Processing Systems **35**, 3988–4003 (2022)
30. Zhao, B., Lu, H., Chen, S., Liu, J., Wu, D.: Convolutional neural networks for time series classification. Journal of systems engineering and electronics **28**(1), 162–169 (2017)
31. Zheng, Y., Liu, Q., Chen, E., Ge, Y., Zhao, J.L.: Exploiting multi-channels deep convolutional neural networks for multivariate time series classification. Frontiers of Computer Science **10**, 96–112 (2016)
32. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International conference on learning representations (2018)