Active Learning for Transformer-Based Fault Diagnosis in 5G and Beyond Mobile Networks

Seyed Soheil Johari*, Massimo Tornatore†, Nashid Shahriar‡, Raouf Boutaba*, Aladdin Saleh§
*Department of Computer Science, University of Waterloo, {ssjohari | rboutaba}@uwaterloo.ca
‡Politecnico di Milano, massimo.tornatore@polimi.it
†Department of Computer Science, University of Regina, nashid.shahriar@uregina.ca
§Rogers Communications Canada Inc., aladdin.saleh@rci.rogers.com

***Abstract*—As 5G and beyond mobile networks evolve, their increasing complexity necessitates advanced, automated, and data-driven fault diagnosis methods. While traditional data-driven methods falter with modern network complexities, Transformer models have proven highly effective for fault diagnosis through their efficient processing of sequential and time-series data. However, these Transformer-based methods demand substantial labeled data, which is costly to obtain. To address the lack of labeled data, we propose a novel active learning (AL) approach designed for Transformer-based fault diagnosis, tailored to the time-series nature of network data. AL reduces the need for extensive labeled datasets by iteratively selecting the most informative samples for labeling. Our AL method exploits the interpretability of Transformers, using their attention weights to create dependency graphs that represent processing patterns of data points. By formulating a one-class novelty detection problem on these graphs, we identify whether an unlabeled sample is processed differently from labeled ones in the previous training cycle and designate novel samples for expert annotation. Extensive experiments on real-world datasets show that our AL method achieves higher F1-scores than state-of-the-art AL algorithms with 50% fewer labeled samples and surpasses existing methods by up to 150% in identifying samples related to unseen fault types.**

***Index Terms*—Fault Diagnosis, Active Learning, Transformers**

I. Introduction

Root cause analysis (RCA), also known as fault diagnosis, consists in the ability to quickly and accurately pinpoint the underlying root causes of a fault, so that network engineers can promptly take necessary actions to rectify the problems at their core [1]. While it represents a crucial aspect of network Operation and Maintenance (O&M) in mobile networks, realizing this task is not without challenges. As mobile networks continue to evolve towards the 5G technology and beyond, fault diagnosis becomes an increasingly vital aspect and new fault analysis methodologies need to be developed [2]. Current fault diagnosis techniques can leverage real-time data analytics to monitor network behavior, identify potential trouble spots, and isolate their root causes [1]. Machine Learning (ML) has become a valuable ally in this task, aiding in the automated analysis of vast amounts of data to identify patterns and root causes of anomalies [3]–[5].

Transformer models, among the ML approaches, have gained significant attention for network management tasks due to their exceptional performance in handling sequential data and time series analysis. Studies [6]–[8] show that lightweight Transformers can achieve state-of-the-art (SOTA) performance in time series representation learning. Consequently, these models have been effectively adopted for various network management tasks, including RCA [9]–[13]. Transformers excel in capturing long-range dependencies and relationships within sequential data, making them particularly suited for monitoring and analyzing network behaviors over time. Unlike traditional ML models, which often struggle with long-term dependencies or require complex feature engineering, Transformers use self-attention mechanisms to automatically learn these dependencies. This enables Transformers to more accurately model the temporal dynamics and intricate interactions in multivariate time series (MTS)-based network data, improving performance in network management tasks.

However, despite their advantages, Transformer-based RCA methods have a fundamental limitation: they require a large number of labeled fault instances. The data labeling process is time-consuming and costly as it necessitates domain experts to annotate extensive logs of failure scenarios [4], [5]. Labeling faulty data in mobile networks is widely recognized as a costly and resource-intensive process due to the requirement of expert knowledge and the complexity of network behavior. In [14], the authors emphasize that effective fault detection in cellular systems often involves dealing with imbalanced datasets and high annotation costs, particularly when domain expertise is required to accurately distinguish between subtle failure patterns. Similarly, in the context of next-generation networks, [15] underscores that the high cost of labeled data acquisition is a major bottleneck in deploying ML-based solutions at scale, motivating the use of AL to reduce annotation requirements while maintaining model performance. This is further reinforced by [16], which highlights how AL techniques can strategically select the most informative samples for labeling, thereby significantly reducing manual effort and operational overhead in real-world network management scenarios. Beyond the labeling effort itself, there are broader operational implications of not addressing this problem. According to the *Heavy Reading 2024 5G AIOps Operator Survey* [17], 29% of operators identified data labeling and quality costs as a major obstacle to AI deployment. Moreover, DriveNets [18] reports that properly trained AIOps solutions can reduce network downtime by up to 87% compared to traditional manual RCA methods. These findings highlight the real-world impact of effective training and, by extension, the critical need to reduce labeling cost without sacrificing model accuracy.

To address the lack of labeled faulty data, we design a

novel active learning (AL) approach that iteratively selects batches of informative unlabeled data samples to be labeled by domain experts, minimizing the total labeled data required for satisfactory performance. In AL [19], the informativeness of data points is determined by a query strategy. Our AL approach proposes a novel query strategy specialized for the case where RCA is accomplished by a Transformer model. We leverage the high interpretability of the Transformer model, using attention weights to create a dependency graph for each data point. These graphs are a high-level abstraction of how each data point is processed by the Transformer-based RCA model. We define a one-class novelty/outlier detection task for these graphs to determine if an unlabeled data point is processed by the model in a novel way (compared to labeled data points included in the previous training cycle of the model). We use a method based on Graph Neural Networks (GNNs) [20] that exploits random distillation, similar to [21], to detect abnormal dependency graphs and assign high novelty scores to them. These scores estimate the informativeness of each unlabeled sample. We expect that by labeling samples with high novelty scores and incorporating them into the subsequent training cycle, we can significantly enhance the model's performance. Our experiments show that these novelty scores are far more effective for selecting informative samples compared to other AL criteria. To avoid selecting similar or redundant samples, which leads to lack of diversity in each selected batch [19], [22], we use Determinantal Point Processes (DPP) [23] to select a batch of samples with high novelty scores that are also diverse and disparate in the input space.

Furthermore, our AL method should adapt to the dynamic nature of network faults and anticipate the emergence of new fault types not present in the initial labeled dataset. A critical aspect of an AL method for fault diagnosis is its ability to identify samples associated with these unseen fault types within the unlabeled dataset, enabling continuous model improvement. Our experimental results indicate that our proposed AL method significantly outperforms SOTA AL methods in identifying such unseen fault types.

We show the effectiveness of our proposed AL approach through comprehensive experimental evaluation on two public real-world datasets, one generated in a 5G core network [24] and the other produced in a Network Function Virtualization (NFV)-based test environment that simulates a 5G IP core network [25]. The following summarizes our contributions:

- We develop a novel AL approach to minimize the total labeled data needed for fault diagnosis. Our AL strategy, tailored for the case where fault diagnosis is accomplished by a Transformer model, uses the Transformer's interpretability to determine if the model processes an unlabeled data point in a novel way. This novelty measurement is the primary criterion for assessing the informativeness of each sample.
- Through comprehensive experiments on two real-world datasets, we show that our AL method achieves higher F1-scores than SOTA AL algorithms with 50% fewer labeled samples, significantly reducing labeling costs.
- In further experiments, we systematically exclude samples of a particular fault type from the labeled dataset to evaluate various AL methods on their ability to identify unseen fault types in the unlabeled dataset. Our results show that our AL method outperforms SOTA AL approaches by up to 150% in selecting samples associated with unseen fault types.

II. Background

A. Transformer Model for MTS Classification

As described earlier, we assumed fault diagnosis is performed by a Transformer model that considers fault diagnosis as an MTS classification problem. Recently, several works have proposed using Transformers for MTS classification [6], [7], achieving SOTA performance on numerous benchmarks. In this subsection, we describe the details of the architecture and training process of such a Transformer-based MTS classification model, following [6], [7]. These details are essential for grasping the methodology of our proposed AL method. It is important to note that our primary focus in this paper is on developing the AL method rather than optimizing the Transformer architecture for fault diagnosis. For this reason, we adopt the same design as [6], [7] for the Transformer model architecture, ensuring consistency with existing benchmarks. As described by [7], the Transformer MTS classification model comprises two parallel Transformer encoders [26]. The first encoder, called the temporal encoder, learns temporal relationships among different time steps. The second encoder, called the metrical encoder, captures the explicit relationships among different performance metrics. The output embeddings learned by these two encoders are then used to predict the classification target.

The temporal encoder's architecture is based on the well-known Transformer model described in [26], designed to capture temporal dependencies among time steps. The input to a Transformer encoder is a series of tokens. In the temporal encoder, for the input data $x_n = \{x_{1,n}, x_{2,n}, \ldots, x_{T,n}\}$, each $x_{t,n} \in \mathbb{R}^P$ is an input token, resulting in T tokens. The encoder computes a d_α-dimensional token embedding $h^{(0)}_{\{t,x_n\}} \in \mathbb{R}^{d_\alpha}$ from each input token $x_{t,n}$ via a linear projection layer (positional encodings are also added to each token embedding to provide positional information to the encoder about the ordering of the input tokens). The token embeddings are then updated through L_α attention layers, with the output of each layer l denoted as $h^{(l)}_{\{t,x_n\}}$. The final embeddings $h^{(L_\alpha)}_{\{t,x_n\}}$ are utilized for determining the classification target.

The metrical encoder, similar in architecture to the temporal encoder, is designed to learn relationships among different performance metrics. For the input data $x_n = transp([S^1_n, S^2_n, \ldots, S^P_n])$, each $S^p_n \in \mathbb{R}^T$ is an input token, resulting in P tokens. The encoder generates a d_β-dimensional token embedding $z^{(0)}_{\{p,x_n\}} \in \mathbb{R}^{d_\beta}$ from each input token S^p_n through a linear projection layer. These embeddings are refined through L_β layers, with the output of each layer l denoted as $z^{(l)}_{\{p,x_n\}}$. The final embeddings $z^{(L_\beta)}_{\{p,x_n\}}$ are used alongside the temporal encoder's final embeddings to predict the classification target. One option is to concatenate all the embeddings and feed them to a Multi-Layer Perceptron classifier. Another

option is to use a single-head attention layer decoder to predict the classification target from the embeddings. We chose the latter in our implementation as it yielded better performance.

Following [6] and [7], the Transformer model includes three attention layers ($L_\alpha = L_\beta = 3$) with encoder dimensions set to 128 ($d_\alpha = d_\beta = 128$), making it highly lightweight (with around 300,000 learnable parameters). To utilize unlabeled samples, the temporal and metrical encoders are pre-trained on the entire dataset using a self-supervised task as per [6]. After pre-training, both encoders and the decoder are jointly fine-tuned on the labeled dataset, training the Transformer in a semi-supervised manner. However, our results indicate that pre-training has an insignificant effect on performance due to the domain-specific patterns in our MTS data, which are not effectively captured through pre-training on unrelated tasks.

B. Pool-Based Active Learning

In pool-based AL [19], the learner is provided with a large pool of unlabeled samples $\mathcal{X}_U^{(0)} = X^{(U)}$ and a small initial labeled set $(\mathcal{X}_L^{(0)}, \mathcal{Y}_L^{(0)}) = (X^{(L)}, Y^{(L)})$. The learning process proceeds in cycles. At each iteration c, a classification model (e.g., our Transformer-based RCA model) is trained using the current labeled set and potentially the unlabeled data as part of a semi-supervised framework.

A query strategy is then applied to select a batch $b^{(c)} \subset \mathcal{X}_U^{(c)}$ of m informative samples to be labeled. The newly labeled samples $(b^{(c)}, \mathcal{Y}_b^{(c)})$ are added to the labeled set, and removed from the unlabeled pool. The updated sets are given by:

$$\mathcal{X}_L^{(c+1)} = \mathcal{X}_L^{(c)} \cup b^{(c)}, \quad \mathcal{Y}_L^{(c+1)} = \mathcal{Y}_L^{(c)} \cup \mathcal{Y}_b^{(c)}, \\ \mathrm{X}_U^{(c+1)} = \mathcal{X}_U^{(c)} \setminus b^{(c)}. \tag{1}$$

If the initial labeled set contains N_L samples and the unlabeled pool contains N_U, then after c iterations the labeled and unlabeled sets will contain $N_L^c = N_L + c \times m$ and $N_U^c = N_U - c \times m$ samples, respectively. This iterative procedure continues until a performance threshold is met or the labeling budget is exhausted.

III. Related Works

Data-driven Network Fault Diagnosis: Data-driven approaches are widely used for network fault diagnosis to tackle modern network complexities. Some methods create abstraction models to capture relationships between network metrics and events for effective fault cause analysis [27], [28]. However, these techniques often fail to generalize to complex, virtualized beyond-5G networks with dynamic topologies [5]. Some data-driven fault localization methods label fault instances based on their root cause, framing it as a multi-classification problem [3], [5]. For example, [3] performs RCA for wireless network failures as a time-series classification problem using temporal, directional, attributional, and interactional features. However, these methods often require abundant labeled faulty samples to achieve a good performance. More recently, few-shot learning has been utilized for fault diagnosis to mitigate the scarcity of labeled data [29], [30]. However, we assume the availability of ample unlabeled data samples in our problem, making AL and semi-supervised learning more suitable than few-shot learning. Nonetheless, to achieve significant performance improvements in semi-supervised learning, an adequate number of labeled samples is still required because the specific patterns and dependencies in our MTS network data for fault diagnosis are often highly domain-specific, and pre-training on unrelated tasks may not capture these nuances effectively. This highlights the importance of AL in efficiently leveraging both labeled and unlabeled data.

Transformer Models in Network Management: The application of Transformer models in network management has been explored extensively in recent literature, showcasing significant advancements. For instance, [10] proposes Simba, a framework combining GNNs and Transformers for anomaly detection and RCA in 5G networks. The work in [9] develops a dual attention-based federated learning model for wireless traffic prediction. [11] deploys a Transformer model to enhance data inference and long-term prediction in sparse mobile crowdsensing. Authors in [13] introduce the Multi-Task Transformer for simultaneous traffic characterization and application identification. [12] presents FlowTransformer, a flexible framework for Transformer-based Network intrusion detection systems. These studies demonstrate the significant impact of Transformers on network management.

Deep Active Learning: Different query strategies in AL literature incorporate various criteria for selecting the most informative samples for annotation. Most strategies are uncertainty-based (e.g., [31], [32]), choosing samples where the classification model is least confident in predicting their labels. However, deep learning models are often overly confident, making uncertainty-based measures like softmax probabilities or entropy unreliable for sample selection [19]. On the other hand, diversity-based strategies (e.g., [22]) focus on selecting samples that maximize diversity, ensuring that selected samples represent a wide range of variations or different regions in the input space. However, they are model-agnostic (task-agnostic), i.e., they do not explicitly consider the relevance or usefulness of the selected samples for the specific task at hand and might ignore domain-specific characteristics of our fault diagnosis problem [33]. Hybrid strategies combine diversity-based AL with other criteria but still primarily focus on uncertainty measurements, which are unreliable as explained earlier. More advanced deep AL methods like VAAL [34] and TA-VAAL [33] use adversarial training to separate labeled and unlabeled distributions in a learned latent space. While effective in vision tasks, they assume that the latent space is stable and discriminative — an assumption that breaks down in MTS data from (e.g., mobile) network deployments, where faults may manifest in subtle, noisy, or overlapping ways. Moreover, the effectiveness of such strategies often deteriorates when the latent space does not align with the semantic structure of the task.

Moreover, the potential of using AL to reduce reliance on labeled data in network management tasks remains largely unexplored. Few studies have applied AL in this area, and those that do (e.g., [16], [35]) mostly rely on simplistic uncertainty-based approaches, which are often ineffective and unsuitable for the MTS nature of network data. Crucially, existing deep

AL methods treat the model as a black box and do not exploit architectural features such as the attention mechanism in Transformers, which provide rich interpretability signals about how the model processes each input. To address the limitations of existing AL approaches, we propose a novel AL method for Transformer-based fault diagnosis tailored to MTS network data that is both task-oriented and diversity-aware: it leverages the interpretability of the Transformer to incorporate the unique characteristics of the fault diagnosis task in selecting relevant samples and uses DDP to ensure diversity in the selected batches for annotation. This represents a key research gap. Our proposed AL approach addresses this by constructing attention-based dependency graphs that capture both temporal and metrical relationships as seen by the Transformer model. We then perform graph-level novelty detection to identify structurally novel inputs and combine this with DPP-based batch selection to ensure diversity. This results in a task-aware, model-informed, and diversity-conscious AL strategy that is well-suited for RCA tasks in mobile networks.

IV. Problem Statement

In this work, we perform fault diagnosis based on the values of multiple performance metrics collected from the network over a time window around the fault occurrence. Our training data consists of N fault events, and for each event we have P metrics periodically monitored over a time window of size T. Examples of these metrics include general measurements like CPU utilization and packet rates of different network functions (NFs), or specific metrics related to a NF functionality, such as call success ratio. The time window should encompass time steps before and after a critical Key Performance Indicator exceeds (or falls below) a threshold, or when the anomaly detector in the network detects an anomaly [1].

For each fault event $n \in \{1, 2, \ldots, N\}$, we have the data sample $x_n = [x_{1,n}, x_{2,n}, \ldots, x_{T,n}]$, where each $x_{t,n} \in \mathbb{R}^P$ is a vector of size P containing the values of all P monitored performance metrics at time step $t \in \{1, 2, \ldots, T\}$. Thus, the training data X is the collection of these N samples: $X = \{x_1, x_2, \ldots, x_N\}$. Let $S_n^p = [S_{1,n}^p, S_{2,n}^p, \ldots, S_{T,n}^p]$ be a time series representing the values of metric $p \in \{1, 2, \ldots, P\}$ during the time frame of the n-th fault event ($S_{t,n}^p$ is the value of metric p at time step t). We can also represent the data sample x_n as the collection of these time series: $x_n = transp([S_n^1, S_n^2, \ldots, S_n^P])$, where $transp$ denotes the transpose operation. The root cause of each fault event n is represented as y_n, belonging to the set $R = \{r_1, r_2, \ldots, r_\kappa\}$, which contains the κ possible root causes of different fault scenarios. Examples of possible root causes for different fault scenarios include the failure of a specific node, packet loss due to congestion in a particular link, or inadequate resource allocation in a specific NF [24].

The training dataset X is divided into the labeled dataset $X^{(L)} = \{x_1^{(L)}, x_2^{(L)}, \ldots, x_{N_L}^{(L)}\}$ with N_L samples, and the unlabeled dataset $X^{(U)} = \{x_1^{(U)}, x_2^{(U)}, \ldots, x_{N_U}^{(U)}\}$ with N_U samples. For the labeled dataset $X^{(L)}$, we know the root causes of every data sample, i.e., we have the labels $Y^{(L)} = \{y_1^{(L)}, y_2^{(L)}, \ldots, y_{N_L}^{(L)}\}$. We assume fault diagnosis is performed by a semi-supervised Transformer classification model that learns the mapping from each data sample x_n to its label y_n utilizing the training dataset (basically, the Transformer model performs fault diagnosis as an MTS classification problem as each data sample is defined as the collection of multiple time series over a specific time window). In practice, most training samples are initially unlabeled, with only a small amount labeled [16], i.e., $N_L \ll N_U$ or even $N_L \approx 0$. However, our classification model requires an adequate number of labeled samples to effectively learn to map the monitored metrics to the fault root causes. Therefore, we need to select some unlabeled samples and acquire their labels with the help of domain experts. To minimize labeling costs and resources, our goal is to develop an AL method to iteratively select the most informative samples for labeling.

In our pool-based AL setting [19], at each cycle (iteration) c, we have the unlabeled dataset (pool) $\mathcal{X}_U^{(c)}$ and the labeled dataset $\mathcal{X}_L^{(c)}$ with corresponding labels $\mathcal{Y}_L^{(c)}$. The goal is to choose a batch of informative samples from $\mathcal{X}_U^{(c)}$, acquire their labels, and add them to $\mathcal{X}_L^{(c)}$ and $\mathcal{Y}_L^{(c)}$. Initially, $\mathcal{X}_U^{(0)} = X^{(U)}$, $\mathcal{X}_L^{(0)} = X^{(L)}$, and $\mathcal{Y}_L^{(0)} = Y^{(L)}$. At each iteration c, we train our (semi-supervised) Transformer-based classification model on $\mathcal{X}_U^{(c)}$, $\mathcal{X}_L^{(c)}$, and $\mathcal{Y}_L^{(c)}$. We then use a **query strategy** to select a batch $b^{(c)} \subset \mathcal{X}_U^{(c)}$ with m samples ($|b^{(c)}| = m$). We query the labels of $b^{(c)}$ with the help of domain experts (denoted by $\mathcal{Y}_b^{(c)}$). The newly labeled samples are added to $\mathcal{X}_L^{(c)}$ and removed from $\mathcal{X}_U^{(c)}$: $\mathcal{X}_L^{(c+1)} = \{\mathcal{X}_L^{(c)}, b^{(c)}\}$, $\mathcal{Y}_L^{(c+1)} = \{\mathcal{Y}_L^{(c)}, \mathcal{Y}_b^{(c)}\}$, $\mathcal{X}_U^{(c+1)} = \mathcal{X}_U^{(c)} \setminus b^{(c)}$. As $\mathcal{X}_L^{(0)}$ has N_L samples, the dataset $\mathcal{X}_L^{(c)}$ at iteration $c > 0$ will have $N_L^c = N_L + c \times m$ labeled samples ($\mathcal{X}_L^{(c)} = \{\mathtt{x}_1^{(L)}, \mathtt{x}_2^{(L)}, \ldots, \mathtt{x}_{N_L^c}^{(L)}\}$), and $\mathcal{X}_U^{(c)}$ will have $N_U^c = N_U - c \times m$ samples ($\mathcal{X}_U^{(c)} = \{\mathtt{x}_1^{(U)}, \mathtt{x}_2^{(U)}, \ldots, \mathtt{x}_{N_U^c}^{(U)}\}$). This iterative process repeats until the model reaches a desired performance or we exhaust our labeling budget. So, our main objective is to design a robust query strategy within this AL setting that effectively selects the most informative samples from the unlabeled dataset for labeling.

We adopt the standard pool-based active learning framework described in Section II-B. At each iteration c, the data is partitioned into: (i) a labeled set $(\mathcal{X}_L^{(c)}, \mathcal{Y}_L^{(c)})$, and (ii) an unlabeled pool $\mathcal{X}_U^{(c)}$. The goal is to iteratively select a batch of m informative samples $b^{(c)} \subset \mathcal{X}_U^{(c)}$ via a query strategy and obtain their labels $\mathcal{Y}_b^{(c)}$ from domain experts.

The selected samples and their labels are added to the labeled set:

$$\mathcal{X}_L^{(c+1)} = \mathcal{X}_L^{(c)} \cup b^{(c)}, \quad \mathcal{Y}_L^{(c+1)} = \mathcal{Y}_L^{(c)} \cup \mathcal{Y}_b^{(c)}, \\ \mathrm{X}_U^{(c+1)} = \mathcal{X}_U^{(c)} \setminus b^{(c)}. \tag{2}$$

This process repeats until a stopping criterion (e.g., target accuracy or budget exhaustion) is met.

V. Proposed AL Approach for Fault Diagnosis

In this section, we introduce our AL method for selecting informative and diverse samples for annotation.

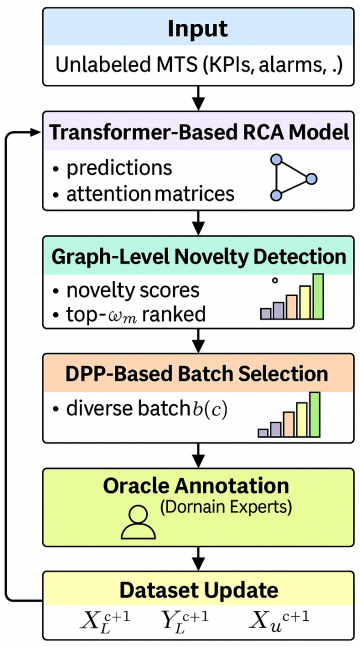

Fig. 1: Our AL approach: Global Dependency Graphs (GDGs) are created for all samples using Transformer's attention weights. A novelty detection problem on the GDGs calculates novelty scores for unlabeled samples. DDP then selects a diverse batch with high novelty scores for annotation.

A. Our proposed AL Query Strategy

To directly address the research gaps identified in Section III, we propose a novel AL query strategy specifically designed for Transformer-based RCA on MTS data from mobile networks. As discussed in the Related Works section, existing deep AL methods—such as uncertainty-based, diversity-based, and adversarial approaches—typically assume access to stable latent representations and treat the model as a black box. These assumptions do not hold in our setting, where the high-dimensional, noisy, and overlapping nature of MTS network data undermines embedding consistency, and the complexity of fault behaviors demands model-specific insight. Our approach addresses these limitations by leveraging the internal structure of the Transformer model: we extract and interpret the attention weights to construct dependency graphs that capture how the model internally reasons about each input sample. By performing graph-level novelty detection on these representations, we identify unlabeled samples that are not just different in input space but are processed in fundamentally novel ways by the model. This model-aware and task-specific strategy fills the gap left by prior work, which does not exploit the interpretability of attention mechanisms for guiding query selection. To further ensure that the selected batch is non-redundant and structurally diverse, we integrate DPP into our sampling process. The following subsections describe each step of our AL strategy in detail.

1) Constructing Dependency Graphs for Each Data Sample: The Transformer-based fault diagnosis model predicts the root cause of each input sample by applying multiple attention layers in its temporal and metrical encoders. The attention scores calculated by the different attention layers provide insights into the model's focus and the relevance of different input tokens, enabling analysis of dependencies and reasoning during prediction. To abstract this information, we construct a graph for each sample that describes the temporal and metrical dependencies among the different time steps and performance metrics from the attention scores in the encoders. We refer to this graph as the *global dependency graph* (GDG).

For example, consider the l-th attention layer in the temporal encoder. For the input sample x_n, this layer updates the output token embeddings of the previous layer $h^{(l-1)}_{\{t,x_n\}}, t \in \{1,2,...,T\}$ into the new token embeddings $h^{(l)}_{\{t,x_n\}}$ by calculating the attention weights $a^{(l)}_{\{t_1,t_2,x_n\}}$ per the query, key, and value vectors calculation of the attention mechanism [26] :

$$q^{(l)}_{\{t,x_n\}} = W^{(l)}_{\{q,\alpha\}} h^{(l-1)}_{\{t,x_n\}} \tag{3}$$

$$k^{(l)}_{\{t,x_n\}} = W^{(l)}_{\{k,\alpha\}} h^{(l-1)}_{\{t,x_n\}} \qquad v^{(l)}_{\{t,x_n\}} = W^{(l)}_{\{v,\alpha\}} h^{(l-1)}_{\{t,x_n\}} \tag{4}$$

$$u^{(l)}_{\{t_1,t_2,x_n\}} = \frac{\sqrt{M}}{\sqrt{d_\alpha}} transp(q^{(l)}_{t_1,x_n}) k^{(l)}_{t_2,x_n} \tag{5}$$

$$a^{(l)}_{\{t_1,t_2,x_n\}} = \frac{exp(u^{(l)}_{\{t_1,t_2,x_n\}})}{\sum_{t=1}^{T} exp(u^{(l)}_{\{t_1,t,x_n\}})} \tag{6}$$

In the above equations, M is the number of heads in the multi-head attention mechanism, and $W^{(l)}_{\{q,\alpha\}}$, $W^{(l)}_{\{k,\alpha\}}$, and $W^{(l)}_{\{v,\alpha\}}$ are $d_q \times d_\alpha$ parameter matrices ($d_q = \frac{d_\alpha}{M}$). The function $exp(.)$ denotes the exponential function. For each $t_1 \in \{1,2,\ldots,T\}$ and $t_2 \in \{1,2,\ldots,T\}$, the attention weight $a^{(l)}_{\{t_1,t_2,x_n\}} \in [0,1]$ determines the relationship between time steps t_1 and t_2 in layer l of the temporal encoder for input sample x_n. Note that in a multi-head attention layer, these attention weights are calculated M times in parallel and averaged across all M heads. From these attention weights, we construct the weighted attributed dependency graph $G^{(l)}_{\{\alpha,x_n\}} = \{\mathcal{V}^{(l)}_{\{\alpha,x_n\}}, \mathcal{E}^{(l)}_{\{\alpha,x_n\}}, \mathcal{F}^{(l)}_{\{\alpha,x_n\}}\}$. $\mathcal{V}^{(l)}_{\{\alpha,x_n\}}$ is the set of nodes, each corresponding to one time step in the time window of input sample x_n ($|\mathcal{V}^{(l)}_{\{\alpha,x_n\}}| = T$). $\mathcal{E}^{(l)}_{\{\alpha,x_n\}}$ is the set of edges, where node $v_i \in \mathcal{V}^{(l)}_{\{\alpha,x_n\}}$ is connected to node $v_j \in \mathcal{V}^{(l)}_{\{\alpha,x_n\}}$ by the directed edge $e_{\{i,j\}} \in \mathcal{E}^{(l)}_{\{\alpha,x_n\}}$ only if the attention weight $a^{(l)}_{\{i,j,x_n\}}$ is larger than a predefined threshold thr_α:

$$e_{\{i,j\}} = \begin{cases} a^{(l)}_{\{i,j,x_n\}}, & \text{if } a^{(l)}_{\{i,j,x_n\}} > thr_\alpha \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

The value of thr_α should be chosen to ensure the graph $G^{(l)}_{\{\alpha,x_n\}}$ retains information about strong relationships among time steps while removing noisy, insignificant attention weights. In our experiments, we set thr_α to maintain around 20% of all possible edges. $\mathcal{F}^{(l)}_{\{\alpha,x_n\}}$ is the set of node features in the graph $G^{(l)}_{\{\alpha,x_n\}}$. Let $f_i \in \mathcal{F}^{(l)}_{\{\alpha,x_n\}}$ be the node features of node $v_i \in \mathcal{V}^{(l)}_{\{\alpha,x_n\}}$. f_i describes that node v_i belongs to the i-th time step of the l-th layer of the temporal decoder. We encode

these three categorical information (which encoder, which time step, and which attention layer) using one-hot encoding and consider the result as the node features for each node v_i.

Similarly in the metrical encoder, for each input sample x_n and attention layer $l \in \{1, 2, ..., L_\beta\}$, we construct the dependency graph $G^{(l)}_{\{\beta,x_n\}} = \{\mathcal{V}^{(l)}_{\{\beta,x_n\}}, \mathcal{E}^{(l)}_{\{\beta,x_n\}}, \mathcal{F}^{(l)}_{\{\beta,x_n\}}\}$. Here, each node $v_i \in \mathcal{V}^{(l)}_{\{\beta,x_n\}}$ corresponds to one of the P performance metrics in the metrical encoder. The set of edges and the set of node features are defined similarly to those of the temporal encoder. For creating the graph connections, the threshold thr_β is used to keep only the main connections among the performance metrics.

For an input sample x_n, we have obtained L_α dependency graphs from the temporal encoder and L_β dependency graphs from the metrical encoder. We define the GDG graph $G_{x_n} = \{\mathcal{V}_{x_n}, \mathcal{E}_{x_n}, \mathcal{F}_{x_n}\}$ for the data sample x_n as the disjoint union of all the constructed dependency graphs, containing the overall information regarding the temporal and metrical dependencies among the different time steps and performance metrics:

$$G_{x_n} = G^{(1)}_{\{\alpha,x_n\}} \oplus ... \oplus G^{(L_\alpha)}_{\{\alpha,x_n\}} \oplus G^{(1)}_{\{\beta,x_n\}} \oplus ... \oplus G^{(L_\beta)}_{\{\beta,x_n\}} \tag{8}$$

Generating the GDGs is not computationally expensive, as the attention matrices are produced in a single feed-forward pass of the Transformer. Creating the dependency graphs is simply a matter of applying a threshold to these matrices.

2) One-class Novelty Detection for the Dependency Graphs: The GDG graphs can be considered as a high-level information about how different data samples (fault events) are processed by the Transformer model to determine the target class (the root cause of the fault). For cycle c of our AL procedure, let us define $GDG^{(c)}_L = \{G_{\mathbf{x}^{(L)}_1}, G_{\mathbf{x}^{(L)}_2}, \ldots, G_{\mathbf{x}^{(L)}_{N^c_L}}\}$ as the dataset that contains the GDG graphs of all the data samples in our labeled training dataset $\mathcal{X}^{(c)}_L$. ($G_{\mathbf{x}^{(L)}_i}$ is the GDG of sample $\mathbf{x}^{(L)}_i \in \mathcal{X}^{(c)}_L$.) Similarly, $GDG^{(c)}_U = \{G_{\mathbf{x}^{(U)}_1}, G_{\mathbf{x}^{(U)}_2}, \ldots, G_{\mathbf{x}^{(U)}_{N^c_U}}\}$ would be the dataset of the GDGs of all the data samples in our unlabeled training dataset $\mathcal{X}^{(c)}_U$. In our proposed query strategy, we aim to find unlabeled samples whose GDG graph has a different structure compared to GDG graphs of the labeled samples. Such unlabeled samples are processed in a novel way by the fault diagnosis model (compared to the standard patterns observed during training), and we expect that, by acquiring their labels and including them in the next training cycle of the model, we can significantly improve the model's performance.

For identifying such unlabeled samples, we design a one-class graph-level novelty detection algorithm for the GDGs. In this novelty-detection problem, the dataset $GDG^{(c)}_L$ is considered as the normal dataset, and our goal is to learn a scalar novelty (abnormality) score $s_{\mathbf{x}^{(U)}_i}$ for each GDG graph $G_{\mathbf{x}^{(U)}_i} \in GDG^{(c)}_U$ from the unlabeled dataset. A higher novelty score indicates the data point is more likely to be a novelty/anomaly compared to the majority of GDGs in the labeled dataset $GDG^{(c)}_L$. GNNs [20] have achieved SOTA performance in many graph data analysis tasks. Recently, authors of [21] proposed a GNN-based graph novelty detection approach based on random distillation of graphs' node representations. Motivated by the SOTA performance of the work in [21] on many graph anomaly detection benchmarks, we developed a modified version of their algorithm to effectively solve our novelty detection problem.

GNN-based Graph Novelty Detection Utilizing Random Distillation: Consider Q to be a 1-hop message passing GNN model with $\mathcal{L}$ aggregation layers. Let the GDG graph $G = \{V, E, F\}$ with node features $f_i \in F$ for each node $v_i \in V$ be the input of the GNN model Q. Denote $\phi^\ell_{v_i}$ as the output node representation of node v_i at layer ℓ of the GNN, with $\phi^0_{v_i} = f_i$. The node representations $\phi^\ell_{v_i}$ at layers $\ell > 0$ are d_{gnn}-dimensional vectors calculated in the GNN model as follows:

$$\delta^\ell_{v_i} = MES^\ell(\{(\phi^{\ell-1}_u, e_{u,v_i}) | u \in Ne(v_i, G)\}) \tag{9}$$

$$\phi^\ell_{v_i} = UPD^\ell(\delta^\ell_{v_i}, \phi^{\ell-1}_{v_i}) \tag{10}$$

Where $Ne(v_i, G)$ represents the immediate neighbors of node v_i in graph G, e_{u,v_i} is the weight of the edge between node v_i and its neighbor node u. $\delta^\ell_{v_i}$ is the message to node v_i at layer ℓ, MES^ℓ and UPD^ℓ are message and update functions at layer ℓ. These functions define how information is propagated and updated across the nodes and edges in the GNN model and are typically implemented by neural networks. Different choices of these functions lead to various GNN architectures. For our problem, we use the Graph Convolutional Network (GCN) architecture [20] as the GNN model. In GCN, the message function aggregates information from neighboring nodes by computing a weighted sum of their features and the update function then combines the aggregated information with the current node's features to generate an updated representation. After $\mathcal{L}$ layers of message passing, $\phi^{\mathcal{L}}_{v_i}$ is used as the final representation of node v_i.

In our distillation-based novelty-detection algorithm, we create a fixed, randomly-initialized target GNN model Q_{ta} with $\mathcal{L}_{ta}$ aggregation layers and a predictor GNN model Q_{pr} with $\mathcal{L}_{pr}$ aggregation layers. Both models have d_{gnn}-dimensional node representations, but $\mathcal{L}_{ta}$ is larger than $\mathcal{L}_{pr}$ (the predictor model has fewer layers and parameters than the target model). The parameters of the target GNN are randomly initialized and kept fixed/frozen. Denote the final node representations produced by the target GNN as $\phi^{\mathcal{L}_{ta}}_{\{v_i,ta\}}$ and those by the predictor GNN as $\phi^{\mathcal{L}_{pr}}_{\{v_i,pr\}}$ for each node v_i. We train the predictor model to generate final node representations similar to those produced by the random target network. For an input GDG graph $G = \{V, E, F\}$, the loss function for training the predictor GNN is defined as:

$$loss(G, Q_{ta}, Q_{pr}) = \frac{1}{|V|} \sum_{v_i \in V} ||\phi^{\mathcal{L}_{ta}}_{\{v_i,ta\}} - \phi^{\mathcal{L}_{pr}}_{\{v_i,pr\}}||_2 \tag{11}$$

The Euclidean distance $||\phi^{\mathcal{L}_{ta}}_{\{v_i,ta\}} - \phi^{\mathcal{L}_{pr}}_{\{v_i,pr\}}||_2$ is used as the distillation function to measure the difference between the two node representations. We train the predictor model on the $GDG^{(c)}_L$ dataset using the described loss function $loss(G, Q_{ta}, Q_{pr})$. We denote the trained/optimized predictor model as Q^*_{pr}. After training, for each unlabeled GDG graph $G_{\mathbf{x}^{(U)}_i} \in GDG^{(c)}_U$, its novelty score $s_{\mathbf{x}^{(U)}_i}$ is calculated based on the loss value of the trained predictor model:

$$s_{\mathbf{x}_i^{(U)}} = loss(G_{\mathbf{x}_i^{(U)}}, Q_{ta}, Q_{pr}^{*}) \tag{12}$$

The key intuition is that our training forces the predictor model's node representations to closely match the corresponding node representations of the fixed random target model for GDG graphs in the $GDG_L^{(c)}$ dataset. Consequently, graph patterns well-represented in $GDG_L^{(c)}$ yield a low loss value (prediction error). For an unlabeled GDG graph $G_{\mathbf{x}_i^{(U)}} \in GDG_U^{(c)}$, a high novelty score $s_{\mathbf{x}_i^{(U)}}$ indicates that it does not conform to the regularity information that exist in $GDG_L^{(c)}$. This novelty score is the main criterion for the informativeness of unlabeled data samples in each cycle.

Our distillation-based novelty detection algorithm differs from [21] in that their target and predictor GNN models share the same architecture and parameter count, which contradicts distillation learning principles. In distillation learning [36], a smaller student model is trained to mimic a larger teacher model's behavior to transfer knowledge in a compressed form (in our problem, the target and predictor GNNs are the teacher and the student models, respectively). By incorporating more aggregation layers in the target network ($L_{ta} > L_{pr}$), we address this limitation and effectively learn the regularity information of the labeled dataset. In our implementation, we set $d_{gnn} = 64$, $L_{ta} = 3$, and $L_{pr} = 2$ for both datasets.

3) Selecting a Diverse Batch of Samples with High Novelty Scores Utilizing DPP: One way of choosing the batch $b^{(c)}$ would be to greedily select the top-m unlabeled samples with the highest novelty scores. However, such a greedy approach may result in the selection of similar or redundant samples, leading to a lack of diversity in the selected batch [19]. To address this issue, we first create a candidate dataset $\mathcal{X}^{(can)} = \{\mathbf{x}_1^{(can)}, \mathbf{x}_2^{(can)}, ..., \mathbf{x}_{w\times m}^{(can)}\}$ that consists of $w\times m$ samples from $\mathcal{X}_U^{(c)}$ such that their GDG graphs have the highest novelty scores ($\mathcal{X}^{(can)} \subset \mathcal{X}_U^{(c)}$, $w > 1$). We then select a diverse batch $b^{(c)} \subset \mathcal{X}^{(can)}$ using DPP [23]. The parameter $w > 1$ balances the diversity of the samples and of their novelty scores.

DPP has recently been applied to select subsets with specific properties from larger nominee datasets in various ML applications, such as documentation abstraction, anomaly detection [37], and AL [38]. In our problem, the nominee dataset is $\mathcal{X}^{(can)}$ (with $w \times m$ samples), and we aim to choose a subset $b^{(c)} \subset \mathcal{X}^{(can)}$ with $|b^{(c)}| = m$ samples, prioritizing diversity. To create the desired subset, DPP constructs a $(w\times m)\times(w\times m)$ kernel matrix Δ for the $\mathcal{X}^{(can)}$ dataset. The subset $b^{(c)}$ is chosen to maximize $det(\Delta_b)$, where $det(\Delta_b)$ is the determinant of the principal minor Δ_b (Δ_b is a submatrix of Δ that only includes entries related to the samples in $b^{(c)}$).

Since we are interested in a diverse batch of samples with large novelty scores, we construct the kernel matrix Δ by defining its entries $\Delta_{\{i,j\}}$ as follows:

$$\Delta_{\{i,j\}} = \bar{s}_{\mathbf{x}_i^{(can)}} \times \bar{s}_{\mathbf{x}_j^{(can)}} \times Sim(\mathbf{x}_i^{(can)}, \mathbf{x}_j^{(can)}) \tag{13}$$

Where $\bar{s}_{\mathbf{x}_i^{(can)}}$ and $\bar{s}_{\mathbf{x}_j^{(can)}}$ are the normalized values of the novelty scores $s_{\mathbf{x}_i^{(can)}}$ and $s_{\mathbf{x}_j^{(can)}}$, respectively ($s_{\mathbf{x}_i^{(can)}}$ is the novelty score of the GDG graph of the sample $\mathbf{x}_i^{(can)}$, and normalization is done based on the largest novelty score in the candidate dataset). The function $Sim(\mathbf{x}_i^{(can)}, \mathbf{x}_j^{(can)})$ calculates the similarity between samples $\mathbf{x}_i^{(can)}$ and $\mathbf{x}_j^{(can)}$. For our multivariate time series dataset, we use the Extended Frobenius Norm (Eros), a PCA-based similarity measure [39] ranging from 0 to 1, with 1 being the most similar. The pairwise similarities $Sim(\mathbf{x}_i^{(can)}, \mathbf{x}_j^{(can)})$ make DPP prefer dissimilar samples [23], while the novelty scores $\bar{s}_{\mathbf{x}_i^{(can)}}$ and $\bar{s}_{\mathbf{x}_j^{(can)}}$ encourage DPP to select samples with high novelty scores by boosting their diagonal entries [37]. If the kernel matrix entries are only similarity measurements ($Sim(\mathbf{x}_i^{(can)}, \mathbf{x}_j^{(can)})$, the determinant of a submatrix correlates with sample diversity. Scaling up entries with novelty scores makes a submatrix's determinant larger if its samples have higher novelty scores. Thus, the submatrix with the largest determinant will have a diverse set of samples with high novelty scores [37]. We use the greedy algorithm in [23] to find this submatrix.

B. Computational Complexity

Let N denote the number of unlabeled samples, H the number of attention heads in the Transformer encoder, and T the sequence length (i.e., time steps per sample). Our active–learning query cycle comprises three stages. *(i) Attention-graph construction:* extracting attention weights and building a dependency graph per sample requires $O(N \cdot H \cdot T^2)$ time, because each head computes pairwise attention scores over the T tokens. *(ii) Graph-level novelty detection:* we compare every unlabeled graph with the labeled pool. If M is the size of the labeled set and G_d the cost of a single graph–distance computation, this step costs $O(N \cdot M \cdot G_d)$. *(iii) DPP batch selection:* we use a greedy algorithm for approximate DPP-based selection [23], which selects a batch of size k from a similarity kernel over the top-n novel graphs in $O(n \cdot k^2)$ time. Hence, the overall per-round complexity is:

$$O\big(N \cdot H \cdot T^2 \;+\; N \cdot M \cdot G_d \;+\; n \cdot k^2\big)\,.$$

In practice, stage (i) piggy-backs on attention already computed during inference, stage (ii) can be parallelised across CPU/GPU cores, and the AL routine is executed intermittently on a candidate subset, keeping runtime manageable for production deployments.

VI. Evaluation

In this section, we start by describing the two datasets used in our experiments. We then detail the SOTA AL approaches that serve as benchmarks for comparison against our proposed AL method. Following this, we evaluate the performance of different AL approaches when applied to the Transformer-based fault diagnosis model. Next, we conduct an ablation study to highlight the importance of creating the GDG graphs in our AL method for identifying the most novel and informative unlabeled samples. Moreover, we compare the AL approaches in their ability to identify and select samples from unseen fault types. Finally, we compare the Transformer-based MTS classification model with alternative MTS classification methods to emphasize the importance of Transformer-based fault diagnosis and the effectiveness of our proposed AL method designed for this purpose.

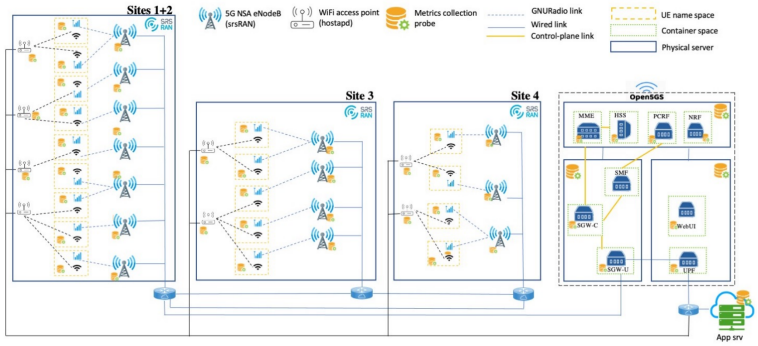

Fig. 2: The 5G3E dataset's system [24].

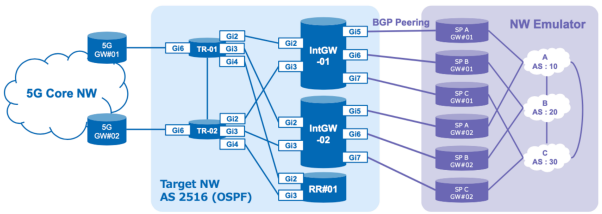

Fig. 3: Network topology of the ITU dataset [25].

A. Datasets

Our focus in this work is on automating fault diagnosis in 5G and beyond mobile networks, with an emphasis on the 5G core network, where timely and accurate RCA is critical for service continuity. To evaluate our method, we use two publicly available datasets that offer complementary coverage of 5G fault diagnosis scenarios. The ITU dataset is designed specifically for the 5G core and is collected from an NFV-based testbed that emulates a 5G IP core network. It includes real-world core-related faults such as node failures, interface down events, and packet-level impairments across individual VNFs. In contrast, the 5G3E dataset captures faults in an end-to-end virtualized 5G network, including CPU overload, congestion-induced packet loss, and link failures, all of which directly impact 5G core components and services. Both datasets provide multivariate time-series measurements and annotated fault root causes, making them well-suited for evaluating our active learning strategy and Transformer-based RCA model. Their fault diversity, time-series nature, and coverage of both core-specific and broader network scenarios enable robust performance benchmarking and generalizability analysis.

***5G3E* Dataset:** *5G3E* is a public dataset for beyond-5G network automation experiments [24]. Generated in a 5G end-to-end system using NS3 and a realistic virtualized 5G network software stack, it employs user traffic data from a mobile network provider to replicate real-time user data transmitted to the network. Figure 2 illustrates the physical model of the system. Three powerful servers in a triangular configuration replicate four sites, while a cluster of four servers forms the 5G core. Data on network components is collected periodically during data transfers. The dataset includes numerous time-series from monitoring 5G network operations, such as radio, computing, and network components, and covers features like radio front-end metrics, server OS data, and network function metrics. Fault scenarios (CPU overload, packet loss, link failures, and bandwidth limitations) are injected at various severity levels to generate faulty instances. Fault diagnosis for this dataset is an MTS classification problem with 22 target classes, distinguishing different faults and their severity levels (eight severity levels for CPU overload and packet loss, four severity levels for bandwidth limitation, and two different scenarios of link failures, totaling 22 target classes). Each data sample includes values of the collected metrics over 20 time steps during a specific fault injection. The training data has 10,000 samples, and the test data include 5,000 samples. The training and test datasets are completely balanced, containing an equal number of samples for each class.

***ITU* Dataset:** Our second dataset is from the "ITU AI/ML in 5G" challenge [25], generated in an NFV-based testbed simulating a 5G IP core network. The target topology of this NFV testbed is shown in Figure 3. It includes five Virtual Network Functions (VNFs): two IP core nodes, two internet gateway routers, and a router reflector, each implemented on a separate VM. The dataset includes diverse performance metrics per VNF per minute, such as CPU utilization and network packet rates. The following fault scenarios are periodically injected into the VNFs to generate faulty samples: 1) node failures, 2) interface failures, 3) packet loss, and 4) packet delay. The fault diagnosis method aims to distinguish these faults as an MTS classification problem with 4 target classes. Each data sample includes values of the metrics over 10 time steps. The training data contains 1,812 faulty samples (54 node failure, 233 interface failure, 762 packet loss, and 763 packet delay samples), while the test data has 1,000 samples and is fairly balanced.

B. Compared Approaches

We compare our proposed AL method with a comprehensive set of SOTA AL algorithms, encompassing uncertainty-based, diversity-based, and hybrid approaches:

Greedy-AT (ours): greedily selects the top-m unlabeled samples with the most abnormal GDG graphs.

Diverse-AT (ours): selects a diverse batch of unlabeled samples with abnormal GDG graphs using DDP (Section V-A3).

***Ent-GN* [32]**: an uncertainty-based query strategy considering the model's gradient norm as the informativeness criterion.

***VAAL* [34]**: formulates AL as a mini-max game, similar to Generative Adversarial Networks (GANs).

***TA-VAAL* [33]**: a modified VAAL incorporating also label information and classifier loss in the mini-max game.

***Coreset* [22]**: a diversity-based strategy defining AL as a core-set selection problem.

***MC-dropout* [16], [31], [35]**: an uncertainty-based method using dropout as a Bayesian approximation of model uncertainty.

***Badge* [40]**: a hybrid strategy incorporating both predictive uncertainty and sample diversity.

Rand: selects a batch of unlabeled samples at random.

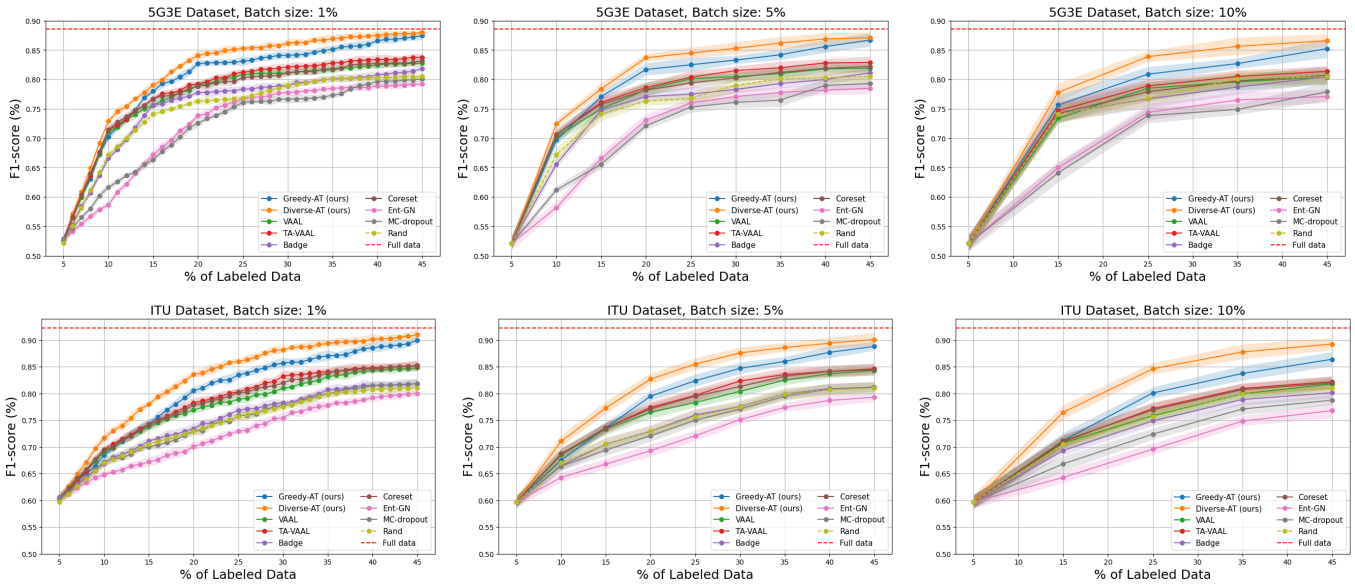

Fig. 4: Comparison of AL query strategies on the 5G3E and ITU datasets, depicting mean F1-scores with shaded regions representing standard deviations.

C. Experimental Results

For evaluating the AL methods on the two datasets, we start with a training dataset in which only 5% of the samples are labeled, chosen at random. In each cycle of the AL procedure, a batch of unlabeled samples is chosen for annotation according to the different AL query strategies. We consider three different batch sizes: 1%, 5%, and 10% of the entire training dataset. In each AL cycle, we report the F1-score of the semi-supervised Transformer classification model trained on the updated dataset.

Figure 4 shows the obtained F1-scores of different query strategies for the two datasets with batch sizes of 1%, 5%, and 10% of the training data (the obtained F1-scores are averaged over 10 trial runs). We can see that our proposed *Greedy-AT* query strategy has outperformed the considered SOTA AL approaches on both datasets. Moreover, by comparing the performance of *Greedy-AT* and *Diverse-AT*, we can see that accounting for the diversity of the samples in the chosen batches can further improve the performance of our AL method, especially for larger batch sizes. The results show that, by using our AL strategy, we can significantly reduce the number of required labeled samples. For example, in the 5G3E dataset when the batch size is 5% of the training data, *Diverse-AT* achieves an F1-score of 83.7% with only 20% of the training samples labeled, while *TA-VAAL* (the best performing approach among the existing methods) has a slightly lower F1-score (82.9) with 45% of the training data labeled. Similarly, in the ITU dataset, *Diverse-AT* achieves a higher F1-score with only 25% of the training data labeled compared to what *TA-VAAL* obtains with 45% of the training data labeled. So, the results indicate that our AL method *Diverse-AT* can accomplish a higher F1-score with even 50% less number of labeled samples compared to the best performing SOTA AL method on average in the two datasets.

Moreover, Figure 4 shows that most uncertainty-based methods performed worse even than random selection (*Rand*). This is because uncertainty is not a reliable criterion for sample informativeness, and these methods do not consider sample correlation, leading to similar and irrelevant sample selections [22]. Conversely, diversity-based and hybrid methods *Badge*, *Coreset*, *VAAL*, and *TA-VAAL* performed better than random selection, with *TA-VAAL* being the best among them. However, *TA-VAAL* still has a significantly lower F1-score compared to our *Greedy-AT* and *Diverse-AT* query strategies.

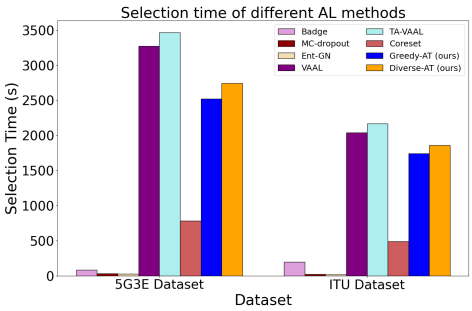

Fig. 5: Average running time of different AL methods for selecting a batch for annotation.

D. Running Time of Different AL Methods

In large-scale network data analysis, the paramount concern lies in mitigating the exorbitant expenses associated with manual labeling. The time taken for selecting samples for annotation within our AL approach is of negligible importance as our primary objective centers on the substantial reduction of labeling costs. However, it might still be noteworthy to evaluate the running time of the different AL methods in our experiments. We report the average running time for selecting a batch of unlabeled samples by different AL strategies in Figure 5. The figure illustrates that uncertainty-based approaches (*MC-dropout* and *Ent-GN*) have the lowest running time, but as discussed earlier they achieved lower F1-socre

than even random sampling. On the other hand, *TA-VAAL* and *VAAL* methods have the highest running time as they need to train a GAN architecture in each cycle of the AL procedure. The results show that our proposed AL methods have lower running time than *TA-VAAL* and *VAAL* while achieving significantly higher F1-scores.

It is crucial to highlight that in our AL approach, the bulk of the computational effort is dedicated to solving the one-class novelty detection problem using the GNN-based knowledge distillation method. Generating the global dependency graphs doesn't incur significant computational costs since the attention matrices can be computed through a single feed-forward pass of the Transformer model, and deriving the dependency graphs involves a straightforward process of applying a threshold to these attention matrices.

E. Memory Requirements

Our AL method operates in an offline setting, where the AL procedure, including data selection, annotation, and iterative model training, is performed prior to deployment. Once the AL process is complete, only the final trained Transformer-based fault diagnosis model is deployed in the 5G network for real-time inference. As a result, the memory and resource requirements of the deployed Transformer model are the key factors for evaluating practical deployment feasibility.

The Transformer model used in our approach is designed to be lightweight, with approximately 300,000 learnable parameters. This significantly reduces the memory footprint compared to more complex models commonly used in similar tasks. For example, the Transformer requires only around 1.2 MB of memory to store the model weights (assuming 32-bit floating-point representation), making it well-suited for resource-constrained 5G and beyond mobile networks.

F. Ablation Study of Our AL Method

To demonstrate the significance of creating the dependency graph for identifying informative unlabeled samples, we conducted an ablation study comparing our *Greedy-AT* method with three alternatives: 1) *Encoder-Novelty*: In this approach, the concatenation of the two encoders' outputs replaces the GDG graph for each data point in the anomaly/novelty detection problem compared to *Greedy-AT*. We employed two established anomaly detection techniques for Euclidean data, namely *iForest* [41] and *Deep SVDD* [42], to solve this modified anomaly detection problem. 2) *Matrix-Novelty*: Here, novelty detection is performed on the concatenation of all attention matrices, serving as the equivalent of GDG graphs, using *iForest* and *Deep SVDD*. 3) *GDG-GraphEmbed*: This method conducts novelty detection on GDG graphs (similar to *Greedy-AT*) using the graph kernel *PK* [43] to extract fixed-size features, followed by *iForest* and *Deep SVDD* for anomaly detection on these obtained features.

We report the performance of these alternative AL methods in Figure 6. For example, *Encoder-Novelty-iForest* and *Encoder-Novelty-SVDD* are two versions of the *Encoder-Novelty* algorithm, where anomaly detection is done by *iForest* and *Deep SVDD*, respectively. The results show that *Encoder-Novelty* performed worse than random sampling, highlighting the importance of the interpretability of the Transformer's information over its latent representation in finding informative samples. Additionally, *Matrix-Novelty* performed similarly to random sampling, meaning that to effectively distinguish unlabeled samples that are processed by the model in a novel way, it is crucial to construct the dependency graphs to obtain an adequate representation of the data points' processing patterns. Finally, we can observe that *GDG-GraphEmbed* achieved competitive F1-scores compared to *Greedy-AT* and significantly outperformed random sampling. However, *Greedy-AT* consistently achieved higher F1-scores than *GDG-GraphEmbed*, indicating that the GNN-based approach solves the graph-level novelty detection problem more accurately than the traditional graph kernel approaches.

G. Identifying Unseen Fault Types

A crucial aspect of an effective AL method for RCA is its ability to identify and select data samples related to fault types not present in the current labeled dataset. In a new set of experiments, we intentionally craft labeled datasets encompassing samples from various fault types, deliberately excluding a specific fault type. This allows us to evaluate AL methods by observing whether they can intelligently select samples from the omitted fault type.

Let $FT = \{ft_1, ft_2, \ldots, ft_{n_f}\}$ represent the set of n_f fault types in the network. In the 5G3E dataset, this includes CPU overload, link failure, packet loss, and bandwidth limitation: $FT_{5G3E} = \{cpu, link, packetL, bandwidth\}$. In the ITU dataset, the fault types are node down, interface down, packet loss, and packet delay: $FT_{ITU} = \{node, interface, packetL, packetD\}$. To evaluate the AL query strategies' ability to identify samples from the unseen fault type ft_i, we create the labeled dataset $X^{(L)}_{ft_i}$ with $\mathcal{N}_L$ samples that includes abundant samples from all fault types other than ft_i, but has no samples from the ft_i fault type. We also generate the unlabeled dataset $X^{(U)}_{ft_i}$ with $\mathcal{N}_U$ samples that contains abundant samples from all fault types other than ft_i, and only has a limited number ($\mathcal{N}_{ft}$) of samples from fault type ft_i ($\mathcal{N}_{ft} \ll \mathcal{N}_U$). This simulates a scenario where the unlabeled dataset contains samples from a fault type not present in the labeled dataset. We selected $\mathcal{N}_{ft}$ to be a small number deliberately making the task of selecting samples from the unseen fault type challenging.

Given the MTS classification model, $X^{(L)}_{ft_i}$ as the labeled dataset, and $X^{(U)}_{ft_i}$ as the unlabeled dataset, we want to select a batch b with m samples from the unlabeled dataset for annotation according to the different query strategies. Since the labeled dataset has no samples from fault type ft_i and has numerous samples from other fault types, we want the query strategy to choose as many samples as possible from fault type ft_i. Let n_{ft_i} be the number of selected samples from fault type ft_i. We define $r_{ft_i} = \frac{n_{ft_i}}{m}$ as the ratio of selected samples from unseen fault type ft_i to the total number of selected samples. Ideally, r_{ft_i} should be as close to 1 as possible. We conducted experiments on both datasets for each fault type being excluded from the labeled dataset. In the 5G3E

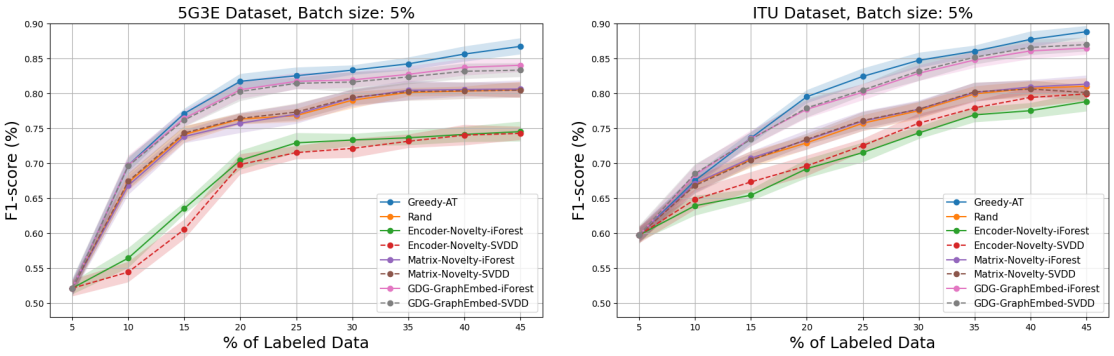

Fig. 6: Results of the ablation study of our AL approach.

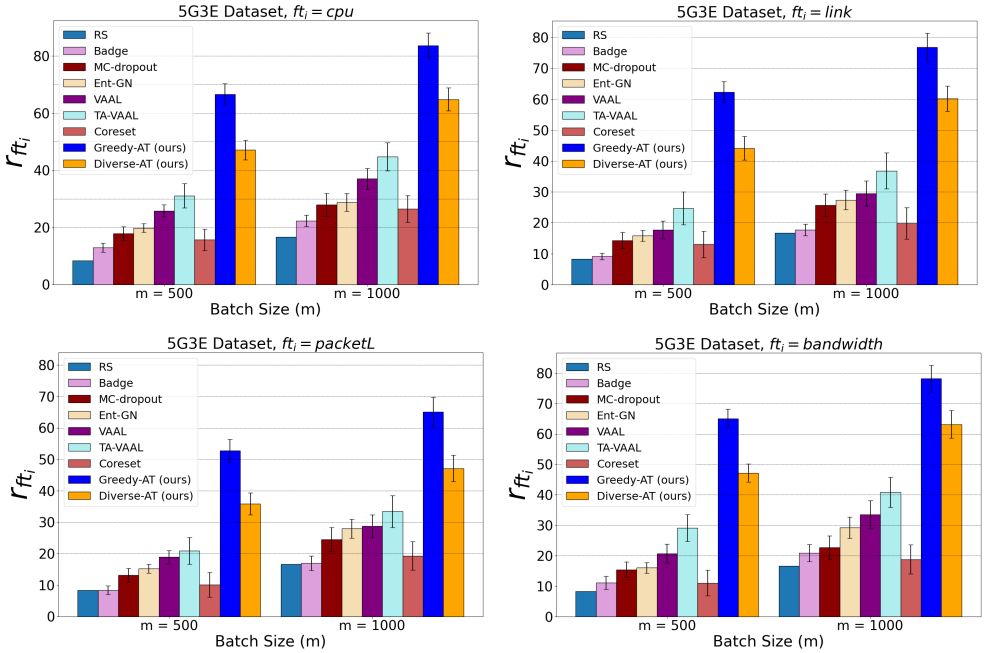

Fig. 7: Ratio of the selected samples from the unseen fault type ft_i for different AL methods and different unseen fault types in the 5G3E dataset (reported in percentage).

dataset, we chose $\mathcal{N}_L = 4000$, $\mathcal{N}_U = 6000$, $\mathcal{N}_{ft} = 300$, and batch sizes $m = 500$ and $m = 1000$. In the ITU dataset, we set $\mathcal{N}_L = \mathcal{N}_U = 1000$, $\mathcal{N}_{ft} = 50$, and batch sizes $m = 100$ and $m = 200$. These samples were chosen randomly from all available samples in the 5G3E and ITU datasets. We repeated the experiments 10 times for each dataset and report the average results over these trials.

Figure 8 shows the ratio of selected samples from the unseen fault types for experiments on the 5G3E and ITU datasets. Due to the limited number of unseen fault type samples in the unlabeled dataset, random sampling results in a low r_{ft_i}. *Badge* and *Coreset* also select very few samples from the unseen fault types since they focus on diversity rather than than choosing samples from an unseen fault type. On the other hand, *MC-dropout* and *Ent-GN* achieved higher ratios than random sampling, suggesting that model uncertainty helps distinguish novel samples to some extent. *TA-VAAL* and *VAAL* have the highest ratios among SOTA query strategies on both datasets due to their adversarial learning. However, our results show that *TA-VAAL* and *VAAL* are significantly outperformed by our proposed query strategies (*Greedy-AT* and *Diverse-AT*) in selecting the most samples from the ft_i fault type. This indicates that selecting samples with the highest novelty scores in our query strategies results in a high number of samples from unseen fault types. Moreover, *Greedy-AT* obtained higher ratios compared to *Diverse-AT* because *Diverse-AT* increases batch diversity and selects only dissimilar samples from the unseen fault type, as opposed to *Greedy-AT*.

TA-VAAL chose the most samples from the unseen fault types among the SOTA query strategies on both datasets. Comparing our proposed method *Greedy-AT* with *TA-VAAL* on the 5G3E dataset, we see that for $m = 500$, *Greedy-AT* selects 2.1x, 2.5x, 2.5x, and 2.2x more samples from the unseen fault type when the unseen type is CPU overload, link failure, packet loss, and bandwidth limitation, respectively (similar performance gap is observed for $m = 1000$ in Figure 8). In the ITU dataset, for $m = 100$, *Greedy-AT* selects 1.8x, 2.2x, 2.1x, and 2x more samples from the unseen fault type compared to *TA-VAAL* when the unseen type is node down, interface down, packet loss, and packet delay, respectively.

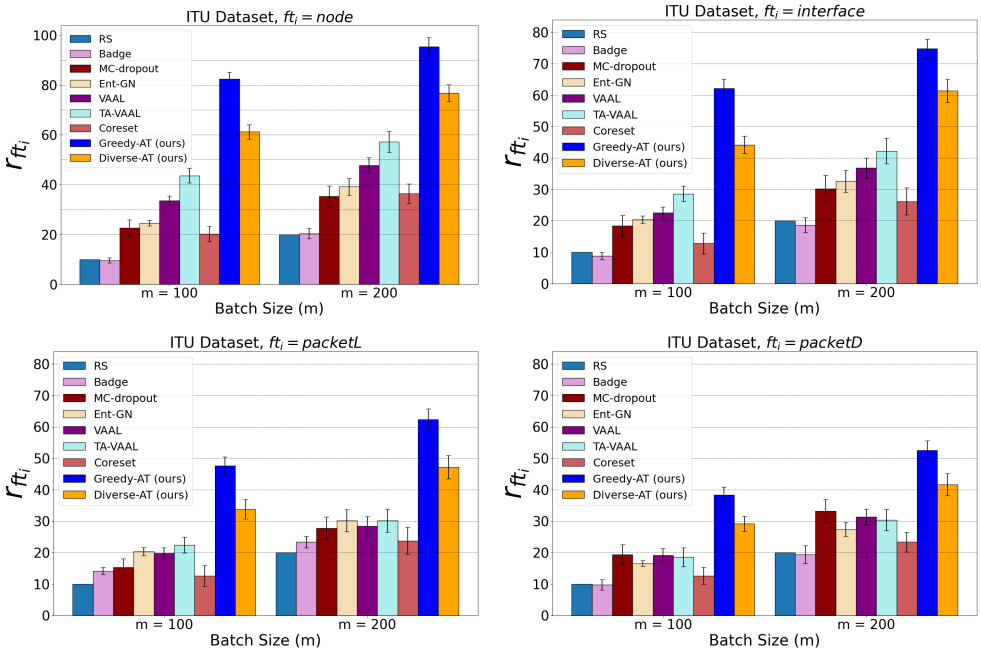

Fig. 8: Ratio of the selected samples from the unseen fault type ft_i for different AL methods and different unseen fault types in the ITU dataset (reported in percentage).

Therefore, the results indicate a substantial improvement of up to 150% in pinpointing and selecting samples related to unseen fault types compared to the SOTA AL strategies.

It is important to note that these experiments considered an extreme case where there are no samples from the newly emerged fault type in the labeled dataset. We also repeated the experiments with a new setting where a few samples from the new fault type are in the labeled dataset. The results showed that our AL methods significantly outperform the existing AL approaches in this new setting as well.

H. Transformers vs. other MTS Classification Models

Even though the primary focus of our paper is on reducing labeling costs by identifying the most informative unlabeled samples, rather than optimizing the classification model architecture for fault diagnosis, it is still valuable to compare the performance of the Transformer model with other state-of-the-art MTS classification and fault diagnosis methods since our AL method is designed for a Transformer model. For this comparison, we evaluate the Transformer MTS classification model against several alternatives, grouped as: 1) the network fault cause localization algorithm *NETRCA* [3]; 2) ensemble-based MTS classification methods *HIVE-COTE* [44], *XEM* [45], and *Rocket* [46]; and 3) deep learning-based MTS classification methods *LSTM-FCN* [47] and *CDC* [48]. We refer to the semi-supervised version of the Transformer model as *Semi-Trans* and the fully-supervised version (without pre-training) as *Sup-Trans*. Among all the approaches, only *Semi-Trans* and *CDC* are semi-supervised (incorporating unlabeled samples), while the other methods are fully supervised.

Table I shows the F1-scores of these methods on the 5G3E and ITU datasets, with varying labeled training data percentages (10% to 100%), averaged over 10 runs. The results show that The Transformer models (*Semi-Trans* and *Sup-Trans*) consistently outperformed other alternatives on both datasets, underscoring their superior performance for fault diagnosis. Moreover, comparing *Semi-Trans* and *Sup-Trans* shows that the pre-training scheme slightly enhances performance, especially with smaller labeled datasets.

I. Hyper-parameter Analysis

In this subsection, we analyze the impact of two new hyper-parameters in our AL method: the threshold values determining the number of edges in each dependency graph (thr_α and thr_β, Section V-A1) and the size of the candidate dataset for DDP-based diverse batch selection (w, Section V-A3).

To transform the attention matrix into a dependency graph, we retain an edge from node v_i to v_j if the attention weight exceeds a threshold (thr_α for the temporal encoder, thr_β for the metrical encoder). This preserves strong relationships while filtering noisy, insignificant attention weights. Previously, $thr_\alpha = thr_\beta = thr$ was set to retain 20% of possible edges. Here, we vary thr to retain 10%-90% of edges. Figure 9 shows the F1-scores of *Greedy-AT* and *Diverse-AT* for 5G3E and ITU datasets (batch size = 5% of training data) as thr changes. The figure shows that performance remains stable for 10%-40% edges but degrades significantly with higher edge percentages. Including too many edges makes the dependency graphs resemble complete graphs, complicating novelty detection and introducing noisy information, leading to random-like sampling performance.

Moreover, for the *Diverse-AT* method, we create a candidate dataset $X^{(can)}$ containing $w \times m$ samples (where m is the batch size) with the highest novelty scores and apply the DDP algorithm. Figure 10 reports F1-scores for w values ranging

Method	5G3E Dataset				ITU Dataset			
	100%	40%	20%	10%	100%	40%	20%	10%
Semi-Trans	**88.6±0.6**	**80.4±0.6**	**76.3±0.7**	**67.2±0.9**	**92.3±0.3**	**80.8±0.5**	**72.9±0.5**	**67.0±0.7**
Sup-Trans	88.5±0.5	79.6±0.6	75.2±0.9	65.7±0.9	92.3±0.4	79.4±0.4	71.1±0.7	65.3±0.8
HIVE-COTE	84.6±0.2	75.7±0.3	66.8±0.3	57.6±0.3	88.5±0.5	75.2±0.6	62.3±0.6	55.7±0.8
XEM	82.1±0.2	75.4±0.2	65.7±0.5	58.5±0.4	85.1±0.5	74.2±0.6	60.8±0.6	53.1±0.8
Rocket	81.6±1.2	73.2±1.4	66.6±1.7	59.8±1.5	85.3±0.9	73.8±1.1	61.0±1.2	53.7±1.2
NETRCA	80.3±0.8	70.1±1.3	62.3±1.2	57.2±1.7	84.6±0.5	74.0±0.6	58.7±0.9	50.8±1.1
LSTM-FCN	75.3±2.8	68.7±3.7	52.1±3.2	44.0±3.5	81.7±1.2	69.5±1.6	52.9±1.6	46.3±2.0
CDC	76.9±2.3	70.3±2.8	62.3±3.5	55.4±3.8	77.9±3.7	72.1±3.5	59.4±4.8	56.3±4.7

TABLE I: F1-score (mean ± standard deviation) of different MTS classification methods for fault diagnosis.

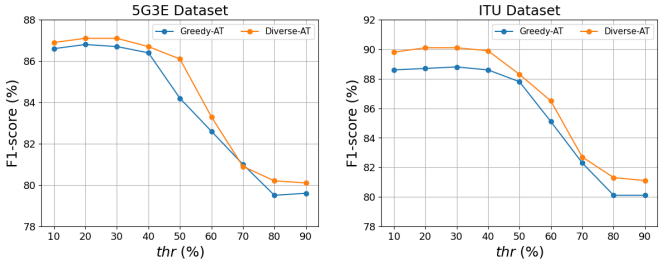

Fig. 9: The effect of the parameter thr on the performance of our Greedy-AT and Diverse-AT methods.

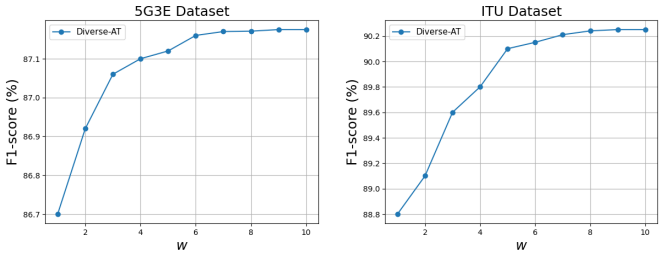

Fig. 10: The effect of the parameter w on the performance of our Diverse-AT method.

from 1 to 10. At $w = 1$, *Diverse-AT* is equivalent to *Greedy-AT*. We can observe that increasing w improves diversity and F1-scores up to $w = 5$, beyond which further increases yield negligible gains. This is because samples with very low novelty scores do not influence DDP's output, making larger $X^{(can)}$ sets unnecessary and only increasing execution time. This trade-off can be effectively managed in our AL method.

Practitioners can tune the threshold parameters thr_α and thr_β by analyzing the sparsity or density of their time-series attention patterns. For datasets with noisy attention maps or weaker temporal/metrical dependencies, retaining fewer edges (e.g., 10–20%) helps focus on the most relevant interactions. Conversely, denser graphs (e.g., 30–40% edge retention) may be more suitable for highly structured datasets with stable performance metrics across time. Similarly, the candidate multiplier w can be set in proportion to the dataset size and expected fault type diversity: larger datasets or those with more heterogeneous fault types benefit from larger w (e.g., 5–10) to ensure better batch diversity, whereas smaller or more homogeneous datasets may suffice with lower values (e.g., 2–3).

VII. Conclusion

In this paper, we presented a novel AL approach tailored for Transformer-based fault diagnosis in mobile networks. By leveraging the interpretability of Transformers, our method effectively identifies the most informative and diverse unlabeled samples for annotation. Extensive experiments on two real-world datasets demonstrate that our AL strategy significantly reduces the need for labeled data while achieving superior performance compared to state-of-the-art AL methods. Our method not only enhances fault diagnosis accuracy but also efficiently identifies samples from unseen fault types, addressing key challenges in network management.

References

[1] T. Zhang, Q. Chen, Y. Jiang, D. Miao, F. Yin, T. Quan, Q. Shi, and Z.-Q. Luo, "Icassp-spgc 2022: Root cause analysis for wireless network fault localization," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9301–9305.

[2] S. Cherrared, S. Imadali, E. Fabre, G. Gössler, and I. G. B. Yahia, "A survey of fault management in network virtualization environments: Challenges and solutions," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1537–1551, 2019.

[3] C. Zhang, Z. Zhou, Y. Zhang, L. Yang, K. He, Q. Wen, and L. Sun, "Netrca: an effective network fault cause localization algorithm," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 9316–9320.

[4] S. S. Johari, N. Shahriar, M. Tornatore, R. Boutaba, and A. Saleh, "Anomaly detection and localization in nfv systems: an unsupervised learning approach," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–9.

[5] A. Elmajed, A. Aghasaryan, and E. Fabre, "Machine learning approaches to early fault detection and identification in nfv architectures," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2020, pp. 200–208.

[6] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 2114–2124.

[7] M. Liu, S. Ren, S. Ma, J. Jiao, Y. Chen, Z. Wang, and W. Song, "Gated transformer networks for multivariate time series classification," *arXiv preprint arXiv:2103.14438*, 2021.

[8] H. Wu, X. Liu, H. Liu, J. Chen, M. Li, W. Xu, and P. S. Yu, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations (ICLR)*, 2023.

[9] C. Zhang, S. Dang, B. Shihada, and M.-S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *IEEE INFOCOM 2021-IEEE conference on computer communications*. IEEE, 2021, pp. 1–10.

[10] A. Hasan, C. Boeira, K. Papry, Y. Ju, Z. Zhu, and I. Haque, "Root cause analysis of anomalies in 5g ran using graph neural network and transformer," *arXiv preprint arXiv:2406.15638*, 2024.

[11] E. Wang, W. Liu, W. Liu, C. Xiang, B. Yang, and Y. Yang, "Spatiotemporal transformer for data inference and long prediction in sparse mobile crowdsensing," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[12] L. D. Manocchio, S. Layeghy, W. W. Lo, G. K. Kulatilleke, M. Sarhan, and M. Portmann, "Flowtransformer: A transformer framework for flow-based network intrusion detection systems," *Expert Systems with Applications*, vol. 241, p. 122564, 2024.

[13] W. Zheng, J. Zhong, Q. Zhang, and G. Zhao, "Mtt: an efficient model for encrypted network traffic classification using multi-task transformer," *Applied Intelligence*, vol. 52, no. 9, pp. 10 741–10 756, 2022.

[14] T. Zhang, K. Zhu, and E. Hossain, "Data-driven machine learning techniques for self-healing in cellular wireless networks: Challenges and solutions," *Intelligent Computing*, 2022.

[15] O. Alhussein, N. Zhang, S. Muhaidat, and W. Zhuang, "Active ml for 6g: Towards efficient data generation, acquisition, and annotation," *arXiv preprint arXiv:2406.03630*, 2024.

[16] M. Chen, K. Zhu, and B. Chen, "Root cause analysis for self-organizing cellular network: an active learning approach," *Mobile Networks and Applications*, vol. 25, no. 6, pp. 2506–2516, 2020.

[17] Heavy Reading, "2024 5g aiops operator survey," https://radcom.com/wp-content/uploads/2024/09/Heavy-Readings-2024-5G-AIOps-Operator-Survey-August-2024-1.pdf, Aug. 2024, [Online].

[18] DriveNets, "Aiops slashes network downtime by 87%," https://drivenets.com/blog/aiops-slashes-network-downtime-by-87/, May 2025, [Online]. DriveNets Blog.

[19] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[20] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[21] R. Ma, G. Pang, L. Chen, and A. van den Hengel, "Deep graph-level anomaly detection by glocal knowledge distillation," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 704–714.

[22] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," *arXiv preprint arXiv:1708.00489*, 2017.

[23] L. Chen, G. Zhang, and E. Zhou, "Fast greedy map inference for determinantal point process to improve recommendation diversity," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[24] C.-D. Phung, S. B. Ruba, S. Secci *et al.*, "An open dataset for beyond-5g data-driven network automation experiments," in *2022 1st International Conference on 6G Networking (6GNet)*. IEEE, 2022, pp. 1–4.

[25] ITU, "Itu-ai-ml-in-5g-challenge," https://www.ieice.org/ rising/AI-5G/, 2020.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[27] J. M. Sánchez, I. G. B. Yahia, and N. Crespi, "Self-modeling based diagnosis of services over programmable networks," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*. IEEE, 2016, pp. 277–285.

[28] Y. Zhang, Z. Guan, H. Qian, L. Xu, H. Liu, Q. Wen, L. Sun, J. Jiang, L. Fan, and M. Ke, "Cloudrca: a root cause analysis framework for cloud computing platforms," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4373–4382.

[29] Y. Song, T. Wang, P. Cai, S. K. Mondal, and J. P. Sahoo, "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–40, 2023.

[30] J. Li, K. Zhu, and Y. Zhang, "Knowledge-assisted few-shot fault diagnosis in cellular networks," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 1292–1297.

[31] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[32] T. Wang, X. Li, P. Yang, G. Hu, X. Zeng, S. Huang, C.-Z. Xu, and M. Xu, "Boosting active learning via improving test performance," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 8, 2022, pp. 8566–8574.

[33] K. Kim, D. Park, K. I. Kim, and S. Y. Chun, "Task-aware variational adversarial active learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8166–8175.

[34] S. Sinha, S. Ebrahimi, and T. Darrell, "Variational adversarial active learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5972–5981.

[35] A. Shahraki, M. Abbasi, A. Taherkordi, and A. D. Jurcut, "Active learning for network traffic classification: a technical study," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 1, pp. 422–439, 2021.

[36] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2021.

[37] X. Zhang, Y. Zhao, Z. Cui, L. Li, S. He, Q. Lin, Y. Dang, S. Rajmohan, and D. Zhang, "Towards lightweight, model-agnostic and diversity-aware active anomaly detection," in *The Eleventh International Conference on Learning Representations*, 2022.

[38] E. Bıyık, K. Wang, N. Anari, and D. Sadigh, "Batch active learning using determinantal point processes," *arXiv preprint arXiv:1906.07975*, 2019.

[39] K. Yang and C. Shahabi, "A pca-based similarity measure for multivariate time series," in *Proceedings of the 2nd ACM international workshop on Multimedia databases*, 2004, pp. 65–74.

[40] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, "Deep batch active learning by diverse, uncertain gradient lower bounds," *arXiv preprint arXiv:1906.03671*, 2019.

[41] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[42] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018, pp. 4393–4402.

[43] M. Neumann, R. Garnett, C. Bauckhage, and K. Kersting, "Propagation kernels: efficient graph kernels from propagated information," *Machine learning*, vol. 102, pp. 209–245, 2016.

[44] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, "Hive-cote 2.0: a new meta ensemble for time series classification," *Machine Learning*, vol. 110, no. 11-12, pp. 3211–3243, 2021.

[45] K. Fauvel, É. Fromont, V. Masson, P. Faverdin, and A. Termier, "Xem: An explainable-by-design ensemble method for multivariate time series classification," *Data Mining and Knowledge Discovery*, vol. 36, no. 3, pp. 917–957, 2022.

[46] A. Dempster, F. Petitjean, and G. I. Webb, "Rocket: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.

[47] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification," *Neural networks*, vol. 116, pp. 237–245, 2019.

[48] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," *Advances in neural information processing systems*, vol. 32, 2019.

Seyed Soheil Johari is a graduate researcher at the David R. Cheriton School of Computer Science, University of Waterloo. He received his B.Sc. in electrical engineering from Sharif University of Technology in 2020. He was a recipient of the best student paper award at IEEE/IFIP NOMS 2022. His research focuses on the application of machine learning techniques for data-driven management and orchestration of 5G network slices.

Massimo Tornatore is a Professor at Politecnico di Milano, Italy. He also held an appointment as Adjunct Professor at University of California, Davis, USA and as visiting professor at University of Waterloo, Canada. His research interests include performance evaluation and design of communication networks (with an emphasis on optical networking), and machine learning application for network management. He co-authored more than 400 conference and journal papers (with 19 best-paper awards) and of the recent Springer "Handbook of Optical Networks". He is member of the Editorial Board of IEEE Communication Surveys and Tutorials, IEEE Communication Letters, IEEE Transactions on Network and Service Management and IEEE/ACM Transactions on Networking.

Nashid Shahriar is an assistant professor in the Department of Computer Science at the University of Regina. He received his PhD from the School of Computer Science, University of Waterloo in 2020. He was a recipient of 2020 PhD Alumni Gold Medal, 2021 Mathematics Doctoral prize, Ontario Graduate Scholarship, President's Graduate Scholarship, and David R. Cheriton Graduate Scholarship with the University of Waterloo. His research received several recognitions, including the IEEE/IFIP NOMS 2022 Best Student Paper Award, IFIP/IEEE IM 2021 Best PhD Dissertation Award, the IEEE/ACM/IFIP CNSM 2019 Best Paper Award, IEEE NetSoft 2019 Best Student Paper Award, and the IEEE/ACM/IFIP CNSM 2017 Best Paper Award. His research interests include network virtualization, 5G network slicing, and network reliability.

Raouf Boutaba (Fellow, IEEE) M.Sc. and Ph.D. degrees in computer science from Sorbonne University in 1990 and 1994, respectively. He is currently a University Chair Professor and the Director of the David R. Cheriton School of Computer science at the University of Waterloo (Canada). He also holds an INRIA International Chair in France. He is the founding Editor-in-Chief of the IEEE Transactions on Network and Service Management (2007- 2010). He is a fellow of the IEEE, the Engineering Institute of Canada, the Canadian Academy of Engineering, and the Royal Society of Canada. His research interests include resource and service management in networks and distributed systems.

Aladdin Saleh received the Ph.D. degree in electrical and electronic engineering and the M.B.A. degree in international management from the University of London, U.K. He is currently an Adjunct Professor with the Cheriton School of Computer Science, University of Waterloo. He is currently priming research and innovation activities with Rogers Communications, among them the joint research partnership with the University of Waterloo on 5G and emerging technologies. He has over 20 years of industry experience in mobile telecom in Canada. He taught and conducted research on next-generation wireless networks at several universities as a full-time Professor, an Adjunct Professor, and a Visiting Researcher.