electronics

MDPI

Article

Learning to Diagnose: Meta-Learning for Efficient Adaptation in Few-Shot AIOps Scenarios

Yunfeng Duan [1], Haotong Bao [2], Guotao Bai [1], Yadong Wei [2], Kaiwen Xue [2], Zhangzheng You [2], Yuantian Zhang [2], Bin Liu [1], Jiaxing Chen [1], Shenhuan Wang [1] and Zhonghong Ou [2,*]

[1] China Mobile Communications Group Co., Ltd., Beijing 102206, China; duanyunfeng@chinamobile.com (Y.D.); baiguotao@chinamobile.com (G.B.); liubinit02@chinamobile.com (B.L.); chenjiaxing@chinamobile.com (J.C.); dvswang@ucdavis.edu (S.W.)
[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
* Correspondence: zhonghong.ou@bupt.edu.cn

Abstract: With the advancement of technologies like 5G, cloud computing, and microservices, the complexity of network management systems and the variety of technical components have greatly increased. This rise in complexity has rendered traditional operations and maintenance methods inadequate for current monitoring and maintenance demands. Consequently, artificial intelligence for IT operations (AIOps), which harnesses AI and big data technologies, has emerged as a solution. AIOps plays a crucial role in enhancing service quality and customer satisfaction, boosting engineering productivity, and reducing operational costs. This article delves into the primary tasks involved in AIOps, such as anomaly detection, and log fault analysis and classification. A significant challenge identified in many AIOps tasks is the scarcity of fault sample data, indicating a natural alignment of these tasks with few-shot learning. Inspired by model-agnostic meta-learning (MAML), we propose a new anomaly detector, MAML-KAD, for application in various AIOps tasks. Observations confirm that meta-learning algorithms effectively enhance AIOps tasks, showcasing the wide-ranging application prospects of meta-learning algorithms in the field of AIOps. Moreover, we introduced an AIOps platform that embeds meta-learning within its diagnostic core and features streamlined log collection, caching, and alerting to automate the AIOps workflow.

Keywords: AIOps; meta-learning; few-shot learning; artificial intelligence

check for updates

Citation: Duan, Y.; Bao, H.; Bai, G.; Wei, Y.; Xue, K.; You, Z.; Zhang, Y.; Liu, B.; Chen, J.; Wang, S.; Ou, Z. Learning to Diagnose: Meta-Learning for Efficient Adaptation in Few-Shot AIOps Scenarios. *Electronics* **2024**, *13*, 2102. https://doi.org/10.3390/electronics13112102

Academic Editor: Maciej Ławryńczuk

Received: 24 April 2024
Revised: 19 May 2024
Accepted: 22 May 2024
Published: 28 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Information technology (IT) infrastructures are vital for modern daily life and industrial production, meeting the diverse and growing needs of businesses. Consequently, IT systems have become larger and increasingly complex. The significance of IT in daily operations necessitates improved availability and reliability to address challenges posed by advancements like Internet of Things (IoT) [1], 5G, future 6G [2,3], autonomous driving, and smart cities. Modern applications demand scalable, resource-efficient, and decentralized IT systems, resulting in jamming attacks [4], network anomalies, and intrusions [5,6]. Additionally, system failures disrupt services and can quickly lead to customer dissatisfaction, and, particularly in data-sensitive industries like finance, substantial economic losses [7].

Traditional IT operations and maintenance (O&M) confront challenges such as limited scalability, manual dependencies, and reactive strategies. In a fast-evolving digital environment, these limitations compromise system effectiveness, escalate downtime, and risk business losses from delayed responses and human errors. Furthermore, the reluctance to adopt newer technologies and the continued prevalence of reactive O&M inflate operational costs and stifle innovation. Thus, it is imperative to transition to automated, intelligent, and proactive O&M approaches [8–10].

To compensate for the various shortcomings in traditional O&M, in recent years, many researchers have focused on developing intelligent software systems to effectively solve operational problems, which are termed as artificial intelligence (AIOps) for IT operations [11]. AIOps studies the application of artificial intelligence in IT service automation management. AIOps helps SRE, DevOps, and operations teams improve the quality and reliability of IT services by utilizing intelligent algorithms and monitoring infrastructure to provide a large amount of data [12,13]. AIOps relies on data-driven methods, fully utilizing technologies such as machine learning, big data, data mining, analysis, and visualization to observe the operational status of infrastructure [14], minimize the impact of daily failures, and actively manage the allocation of computer resources [15].

The advent of AIOps marks a significant advancement in the field of IT operations and maintenance. AIOps integrates artificial intelligence, machine learning, and big data analytics to transform traditional IT O&M processes into more efficient, automated, and intelligent systems. By leveraging AI algorithms, AIOps can proactively identify, diagnose, and even predict potential system issues before they escalate into major problems. This shift from a reactive to a proactive approach greatly reduces system downtimes and enhances overall operational efficiency [16]. The ability of AIOps to analyze vast amounts of data in real time enables it to deliver insights that were previously unattainable with traditional methods, facilitating more informed decision-making and strategic planning.

As an emerging interdisciplinary and comprehensive field, AIOps is still largely an unstructured research field, leaving many unexplored issues. In this article, we delve into some basic tasks in AIOps and find that they all conform to the characteristics of few-shot learning. Naturally, we applied some techniques of meta learning to these tasks and found good results, improving the indicators of each task. More importantly, the application of meta learning [17,18] enhances the ability of AIOps intelligent models to perform zero-shot learning, highlighting the significant role of meta learning in the generalization ability of AIOps models.

The main contributions of this article are highlighted as follows:

- Analyzed the similarity between AIOps task characteristics and few-shot learning and clarified the applicability of few-shot methods to AIOps challenges.
- Applied meta-learning algorithms to AIOps tasks on public datasets, leading to results validating their enhanced performance and generalization capabilities, underscoring the benefits of integrating meta-learning with AIOps.
- Introduced an AIOps platform that embeds meta-learning within its diagnostic core and features streamlined log collection, caching, and alerting to automate the AIOps workflow.

This article is arranged as follows: Section 2 provides an overview of concepts related to few-shot learning and meta-learning and explains the suitability of few-shot learning to AIOps settings. Section 3 details the design of the anomaly detection model named MAML-KAD. Section 4 illustrates the experiments on anomaly detection [19] and fault classification tasks and analyzes the experimental results, proving that meta-learning can be integrated into AIOps. Section 5 discusses the integration of meta-learning into an AIOps system. Section 6 describes conclusions and future work.

2. Preliminary

2.1. Few-Shot Learning

In the big data era, deep learning models [20] have excelled in areas such as image and text classification, with their success heavily reliant on ample training data. Yet, for certain categories with sparse or few labeled examples, data labeling becomes a time-consuming and laborious task. Few-shot learning was introduced to address this [21–23], empowering models to rapidly learn from minimal data akin to human learning, thereby aligning machine learning more with human cognitive abilities.

The reason why few-shot learning is difficult lies in the essence of machine learning [24]. The goal of machine learning is to learn a function within a given sample space,

mapping input data to output data, and enabling accurate predictions for unknown input. Typically, in a hypothesis space $\mathcal{H}$, inputs x and outputs y follow an unknown joint probability distribution $F(x, y)$. A machine learning training set consists of n independent and identically distributed observational samples $D = \{(x_1, y_1)...(x_n, y_n)\}$. The objective is to find an optimal function $f^*(x)$ from a set of functions $\{f(x)\}$ that maps x to y, minimizing the expected risk: $R(f) = \int L(y, f(x))dF(x, y)$. Here, L represents the loss function used to estimate F with f. Since F is unknown, it is impossible to calculate the expected risk directly in real-world scenarios. Therefore, machine learning utilizes the empirical risk minimization (ERM) criterion [25,26], defining empirical risk as Equation (1):

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) \tag{1}$$

The empirical risk is used as an estimation of the expected risk, and algorithms are designed to minimize it. Assuming:

- $\hat{f}$ minimizes R.
- f^* minimizes R under the constraint f belongs to $\mathcal{H}$.
- $f^{'}$ minimizes R_{emp} under the constraint f belongs to $\mathcal{H}$.

Since the optimal function f might not exist, the error in the learning process can be decomposed into two parts, as in Equation (2):

$$\mathbb{E}(f^{'} - \hat{f}) = \mathbb{E}(f^* - \hat{f}) + \mathbb{E}(f^{'} - f^*) \tag{2}$$

The first part $\mathcal{E}_{app} = \mathbb{E}(f^* - \hat{f})$ is the approximation error, measuring how closely the mappings in hypothesis space $\mathcal{H}$ approximate the optimal mapping $\hat{f}$. The second part $\mathcal{E}_{est} = \mathbb{E}(f^{'} - f^*)$ is the generalization error, indicating the gap between minimizing empirical risk and expected risk given $\mathcal{H}$ and D.

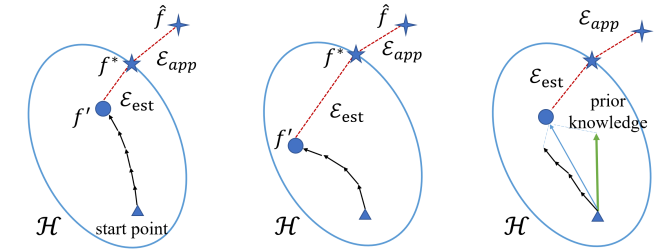
To reduce approximation error $\mathcal{E}_{app}$, a hypothesis space $\mathcal{H}$ with strong representational capacity is used. However, a more complex $\mathcal{H}$ increases the difficulty of generalization, leading to a larger generalization error $\mathcal{E}_{est}$.

Generalization error $\mathcal{E}_{est}$ is the fundamental reason for the difficulty in few-shot learning. In statistical machine learning theory for bounding generalization error, for any $0 < \delta < 1$, the empirical risk $R_{emp}(f)$ and the expected risk R satisfy at least $1 - \delta$ probability, as Equation (3):

$$R \leq R_{emp} + \Phi(\frac{CM}{n}) \tag{3}$$

Here, n is the size of the observational samples (training set) and CM represents capacity measurement, estimating the representational capacity of the hypothesis space, typically reflected in model complexity or function set learning capacity, with measures like VC dimension [27] and Rademacher complexity [28]. The function Φ is non-negative in both domain and range and varies in expression across different problems.

In Equation (3), as n approaches infinity, R approaches R_{emp}, and the generalization error becomes zero. Hence, a large number of observational samples can reduce the generalization error, bringing $f^{'}$ closer to f^* [29,30]. In few-shot learning, the scarcity of samples hinders the empirical risk from closely approximating the expected risk, making it inherently more challenging compared to traditional machine learning approaches, as illustrated in Figure 1.



(a) Sufficient samples (b) few-shot (c) meta-learning

Figure 1. Illustration of the difficulty in few-shot learning; the $\mathcal{E}_{est}$ error is greater in few-shot situations than in situations with sufficient samples.

2.2. Meta-Learning

Meta-learning, also known as "learn to learn", involves utilizing previously acquired knowledge or past experiences in a systematic, data-driven manner to enable artificial intelligence to autonomously and rapidly learn new tasks. The most prominent feature of meta learning is that it can enhance the model's generalization ability on new tasks. As meta-learning progresses with each historical task learned and experience accumulated, it becomes easier to learn new tasks, requiring fewer training samples while still maintaining algorithmic accuracy. Initially conceived to address the challenges of few-shot learning, meta-learning aims to learn how to learn, differentiating itself from the mapping-focused approach of deep learning and the trial-and-error method of reinforcement learning.

The term "meta-learning" encompasses any type of learning that benefits from prior experiences with other tasks. Therefore, any learning paradigm that leverages experiences from similar historical tasks to better accomplish future new tasks falls under the domain of meta-learning. The knowledge learned from previous tasks can be referred to as 'prior knowledge' or 'meta knowledge', as shown by the green arrow in Figure 1c. The greater the similarity of historical tasks, the richer the types of meta-data that can be utilized. Defining and measuring the similarity between tasks is also a key challenge. When new tasks have little similarity to historical tasks or contain significant random noise, the effectiveness of prior experiences may diminish. However, in real-world tasks, there are numerous opportunities to learn from past experiences.

In practice, meta-learning involves techniques such as quickly adapting a pre-trained model to new tasks with minimal data (transfer learning) or developing algorithms that can generalize from one task to another. This approach is particularly useful in environments where data are scarce or tasks change frequently, enabling AI systems to become more versatile and responsive to new challenges. Figure 2 contrasts traditional machine learning with meta-learning, illustrating the different approaches to model training and optimization.

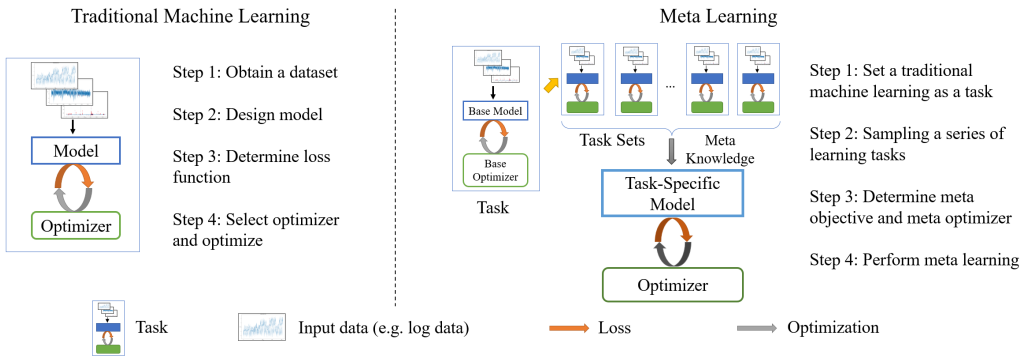


Figure 2. Contrast between traditional machine learning and meta-learning.

Meta-learning is designed to generalize across multiple tasks [31], not just optimize for one. The first step is to redefine traditional machine learning tasks into a series of

learning tasks. Then, these tasks are sampled, and for each one, a base model is trained. A meta-objective and a meta-optimizer are determined to guide how the model should learn from each task, aiming to develop meta-knowledge that encapsulates the essence of all tasks. The final step is to perform meta-learning, where the model is optimized not just for individual task performance but for its ability to adapt quickly and efficiently to new tasks using the meta-knowledge acquired. The outcome is a task-specific model that benefits from this meta-learning process, achieving better performance across a range of tasks with potentially fewer data points required for each new task it encounters.

Currently, the mainstream meta-learning methods can be broadly categorized into optimization-based and metric-based approaches.

Optimization-based: Optimization-based methods [32–34] seek parameters that are sensitive to specific tasks through gradient iteration. The trained model can reach high accuracy with only a few gradient updates. The model-agnostic meta-learning (MAML) [34] algorithm is a prominent representative of this category. MAML is designed to prepare models for rapid adaptation to new tasks with minimal data, using a two-level training process that optimizes for a model's ability to learn quickly. ANIL (almost no inner loop) [35] is a variant of MAML, which diverges from MAML in its parameter update strategy.

Metric-based: Metric-based methods [36–38] learn new tasks by comparing new inputs with input examples, where a higher similarity suggests a greater probability of having the same label. Such methods do not alter the network but rely on comparison, representing a form of non-parametric learning. The information about new tasks is not absorbed into the network parameters but is used to directly predict the weighted combination of labels in the support set. A typical example of a metric-based method is the prototypical network [38], which predicts based on the categories in the support set.

These meta-learning approaches enable quick adaptation to new tasks with minimal data, addressing the challenges of few-shot learning by either optimizing the model for rapid learning or using similarity metrics to classify new instances based on learned examples. Both methods play a crucial role in advancing the capabilities of machine learning models in scenarios with limited data availability.

2.3. AIOps Needs Few-Shot Learning

AIOps is thriving in the challenging environments created by 5G and 6G technologies, while the integration of ML and operations is more common in traditional industrial environments. Despite the transformative impact of AIOps in IT operations and maintenance, its application in certain stages of the ML lifecycle, such as deployment, remains underrepresented in the academic literature. In addition, AIOps faces challenges such as data scarcity [39]. This issue primarily arises in the contexts listed below.

Lack of High-Quality, Relevant Data AIOps requires extensive, varied data to effectively train its machine learning models. Nevertheless, acquiring enough appropriate data can pose a challenge in certain IT spheres, particularly specialized or niche sectors. Such data scarcity can compromise the learning efficiency, accuracy, and performance of AIOps systems.

Rarity of Anomaly or Failure Data AIOps systems aim to foresee and prevent problems from worsening. However, the target events—system failures or anomalies—are inherently infrequent. Insufficient historical data on these rare events complicate AIOps' ability to precisely recognize and forecast them, posing challenges in developing models that can identify such occurrences with minimal false alerts.

Data Diversity and Complexity Despite the availability of data, their diversity and complexity present obstacles. AIOps must handle data from multiple sources—such as logs, metrics, and performance data—which can be structured or unstructured and vary in format. This necessitates advanced preprocessing and feature engineering to render the data suitable for AI models, further complicating the task.

High Cost of Labeling Data The expense of labeling data for AIOps stems from factors such as human expertise labor costs, infrastructure and tool expenditures for management

and security, quality control measures, the need for specialized skills due to data complexity, large datasets, and adherence to regulatory requirements [40]. These costs can inhibit deployment, degrade model accuracy, and decelerate development, underscoring the importance of alternatives such as semi-supervised, active, and transfer learning, alongside crowdsourcing and automated labeling approaches.

AIOps is poised to transform IT operations, yet the rarity and intricacy of required data present substantial obstacles. Given that AIOps frequently faces few-shot learning situations, leveraging few-shot learning techniques like meta-learning becomes a natural approach to navigate these issues.

As mentioned earlier, meta learning's primary advantage lies in enhancing a model's ability to generalize to new tasks, which is crucial for addressing the diverse demands of AIOps tasks. Given the dynamic and evolving nature of IT environments, AIOps solutions must adapt quickly to new conditions, a challenge few-shot learning is particularly suited for with its proficiency in learning from limited data.

In summary, meta-learning equips AIOps with the ability to swiftly assimilate limited data and leverage past experiences. This is essential for managing infrequent occurrences, unprecedented challenges, and tailoring approaches to unique settings where conventional models struggle with limited data and the urgency of adaptation. Consequently, meta-learning is a pivotal strategy in AIOps, fostering agility, efficiency, and nuanced responses amidst the dynamic landscape of IT management.

3. Anomaly Detection via MAML

In the following two sections, we will implement meta-learning methods to solve two AIOps fundamental tasks—fault classification and anomaly detection—to demonstrate the various ways in which meta learning algorithms can be combined with AIOps tasks.

Anomaly detection in AIOps is a critical task that involves identifying unusual patterns or behaviors within IT systems and could indicate problems such as system failures and security breaches [13]. It utilizes various data sources, including system logs, performance metrics, and network traffic data, to monitor the IT environment and ensure the reliability of IT infrastructure.

The few-shot problem in anomaly detection arises due to the rarity and variability of anomalies in IT systems. Typically, there are only a few examples of specific types of anomalies, making it challenging to train conventional detection models that require large datasets. This scarcity necessitates models that can effectively identify and categorize anomalies with limited data.

So, few-shot learning techniques, such as meta-learning, are particularly valuable for anomaly detection in IT system, as they generalize from a small number of training examples. This approach enables AIOps systems to recognize and react to unusual patterns or behaviors, even when such events have been infrequently observed. In our experiment, we leveraged the MAML algorithm to address a well-designed zero-shot learning anomaly detection scenario.

3.1. Model-Agnostic Meta-Learning

Model-agnostic meta-learning (MAML) is an influential approach in the field of meta-learning. Developed with the versatility to be applied across a wide range of learning models and tasks, MAML's primary goal is to find a model parameter initialization that is particularly suited for rapid adaptation.

The basic optimization process of MAML is shown in Figure 3, which can be divided into two phases: the meta-learning phase and the adaptation phase.

Meta-Learning Phase: This is the initial training phase during which the model's parameters θ are adjusted across a variety of tasks (task a, task b, task c). The goal here is to find a set of initial parameters θ^* that can serve as a good starting point for further task-specific adaptation. The model learns these parameters such that, when they are

fine-tuned for a few iterations on a new task, the model achieves significant improvement on that task.

Task-Specific Adaptation Phase: After the meta-learning phase, the well-initialized parameters θ^* undergo a few gradient updates (adaptations) specific to a new task. This phase utilizes a small amount of task-specific data to fine-tune the model parameters θ_T to achieve good performance on this new task. The model does not need extensive retraining; instead, it quickly adapts, leveraging the foundation laid during the meta-learning phase.

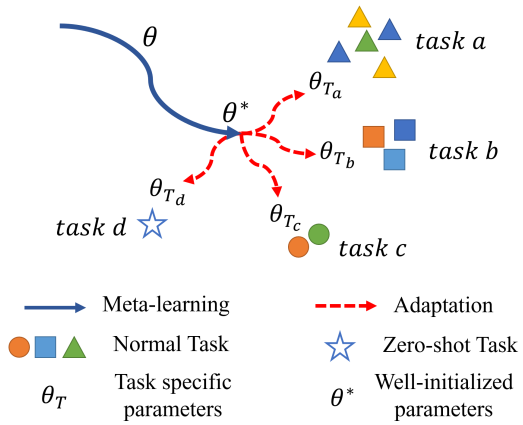


Figure 3. Illustration of MAML.

3.2. Scenario Dataset

We use the KPI anomaly detection dataset constructed in [41]. This dataset is sourced from five prominent Internet corporations, comprising a comprehensive collection of 27 key performance indicators (KPIs) which has been annotated for anomalies by skilled engineers. The proportion of anomalies to normal data in this dataset varies and has different time spans ranging from two to seven months.

3.3. MAML-KAD

Building on the foundational principles of MAML, we have designed a specialized zero-shot detection model named the MAML-KPI anomaly detector (MAML-KAD). This innovative model extends MAML's capability to zero-shot learning scenarios, where it is tasked with identifying anomalies without having been exposed to specific examples in training. MAML-KAD leverages the generalization power of MAML to quickly adapt to new and unforeseen KPIs, making it highly effective for applications in AIOps. This is in line with MAML's proven track record in various domains.

The basic workflow of the MAML-KAD algorithm is shown in Algorithm 1. To highlight the capabilities of the MAML-KAD in zero-shot anomaly detection, we organized our dataset into two primary sections: the meta-train and the meta-test sets. The meta-train set included 16 KPIs in which we employed the model-agnostic meta-learning (MAML) approach to fine-tune our baseline KPI anomaly detection model, achieving a robust set of initial parameters. Meanwhile, the meta-test set consisted of 10 KPIs used to evaluate the model's performance without further training. This innovative zero-shot learning setup was crafted to explore whether MAML-KAD could effectively generalize across diverse historical datasets by extracting universal feature insights, thereby enhancing the overall anomaly detection capability of the model.

Algorithm 1: MAML-KAD

Data: AIOps Task distribution $\mathcal{T}$ with KPIs
Result: Optimized anomaly detection parameters θ^*
Meta-Learning Phase: Randomly initialize model parameters θ;
foreach *KPI task* T_i *sampled from* $\mathcal{T}$ **do**
    Sample mini-batch of KPI data D_i from T_i;
    Compute KPI-specific loss $L_{T_i}(f_\theta)$ on D_i using current parameters θ;
    Compute gradients $\nabla_\theta L_{T_i}$ on anomaly detection tasks;
    Update parameters $\theta' = \theta - \alpha \nabla_\theta L_{T_i}$ with K gradient steps;
end
Set $\theta^* = \theta$;
Zero-Shot Anomaly Detection Phase:
foreach *New KPI task T in meta-test set* **do**
    Directly use θ^* to detect anomalies in T;
    Assess detection effectiveness without further training;
end

Modeling: To address the imbalance between positive and negative samples during MAML training, we implemented a widely used oversampling technique. Our anomaly detection model is a straightforward yet robust KPI detector consisting of three dense layers activated by ReLU functions and followed by a sigmoid function to generate a probabilistic score for anomalies. Furthermore, we employed a cutting-edge evaluation method as proposed in [42].

Comparatives: We evaluated the efficacy of our approach against conventional training techniques that do not utilize MAML and which were implemented on the same meta-train set.We evaluated the efficacy of our approach against conventional training techniques that do not utilize MAML and which were implemented on the same meta-train set and forgo the nuanced bi-level optimization process inherent in MAML. Our comparative analysis aimed to highlight the advantages of meta-learning in enhancing model generalizability and performance in anomaly detection tasks.

Implementation We implemented our method with the open-source machine learning framework PyTorch (v1.7.1). We trained and evaluated our model with two NVIDIA 2080Ti GPUs on a Linux server (64-bit Ubuntu 20.04, Linux kernel 5.4.0) with an Intel Xeon® CPU E5-2620 CPU and 128GB RAM. The subsequent scenario was conducted in the same environment and thus will not be mentioned again in the following text.

3.4. Main Results

Figure 4 presents a comparison of the accuracies of different models across various key performance indicators (KPIs) labeled from A to J. Accuracy in machine learning measures the proportion of correct predictions made by the model out of all predictions made.

Basestands for the baseline model without any specialized training or meta-learning optimization. *MAML* stands for a model that has been optimized using MAML.

Across the board, for KPIs A through J, the MAML-optimized model tended to catch up with the base models, as indicated by the generally even accuracy. The most surprising result was that the model initialized by MAML could be so successful with zero-shot learning.

However, the performance gain varied across different KPIs, indicating that MAML's approach to learning is particularly well-suited to the types of anomalies or patterns present in these KPIs. On the other hand, in KPIs G, F and J, the differences among the models' accuracies were less pronounced, suggesting that the challenges presented by these KPIs might be less sensitive to the advantages of MAML. Another limitation is that some KPIs share the same anomaly pattern, while other KPIs differ significantly, resulting in unstable performance. However, these phenomenons can be mitigated by simple adaptation on specific data.

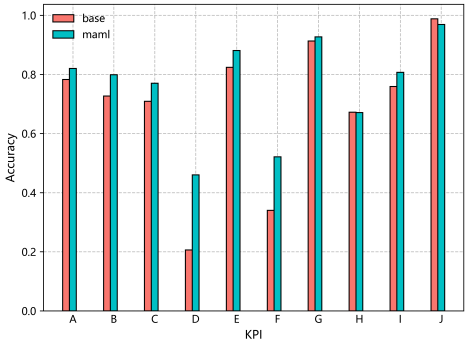


Figure 4. KPI detection results.

Overall, the results indicate the potential benefits of incorporating MAML into AIOps platforms for anomaly detection tasks, highlighting its superiority in learning from limited data and its adaptability across various types of operational data.

4. Fault Classification with Meta-Learning

In this section, we will carry out other common tasks in AIOps practice, namely, log data fault classification. Log files, generated by various components of an IT infrastructure, are rich sources of information, chronicling a detailed record of events, operations, and errors.

The fault classification task involves analyzing log files to detect, categorize, and diagnose faults within the system. Models are trained to classify these faults into various types, such as hardware malfunctions, software bugs, network connectivity issues, or security threats. This process is crucial for timely interventions that prevent service disruptions and maintain system integrity. The accuracy of fault classification directly impacts the effectiveness of subsequent actions, such as alerting maintenance teams or triggering automated remediation processes.

The few-shot problem in fault classification arises from the challenge of identifying and classifying system faults with limited examples or data points. In many IT environments, certain types of faults occur infrequently, resulting in scarce relevant log data. This lack of data makes it difficult to train traditional machine learning models, which typically require large datasets for effective learning.

4.1. Few-Shot Setting

In our second experiment, we adopted the “N-way K-shot" configuration prevalent in few-shot learning. This approach challenges the algorithm to classify data into N distinct categories based on a sparse set of K examples from each category, testing the model’s ability to learn effectively from minimal data. Specifically, we designated two classes in the dataset as support classes, each with sufficient samples, while the remaining two were designated as query classes, characterized by their limited samples. This arrangement is consistent with typical few-shot learning scenarios.

We also employed the MAML method, but rather than applying it to the entire model, we focused its application on the model’s feed-forward network. This approach was driven by two key considerations: First, we regarded the transformer encoders as shared feature extractors that operated in a task-agnostic manner and did not possess an abundance of task-specific parameters. Second, the MAML process involved a bi-level optimization, which was associated with substantial computational and memory resource demands.

4.2. Dataset and Model

In our research, we utilized the Alibaba Tianchi AIOps competition dataset [43], encompassing system log data spanning a defined period and encompassing four unique types of faults. Our primary goal was to harness this limited dataset to meticulously

extract critical features, enabling precise identification of the fault types affecting servers. This endeavor underscores our commitment to achieving high diagnostic accuracy from constrained data resources.

To approach this task, we implemented a cutting-edge transformer-based model tailored to fault classification. The model employed two transformer encoder layers to deeply mine the essential features from the log data. This process resulted in a refined hidden representation, which was subsequently processed through a feed-forward network, culminating in a softmax layer that produced the final classification outcomes.

The experimental setup involved repeating each trial 100 times to ensure reliability, averaging the results to mitigate variance. This methodology was applied across different class divisions to achieve a comprehensive performance overview.

Regarding the model's hyperparameters, we opted for the AdamW optimizer with a learning rate of 1×10^{-3} and a batch size of 64, and we employed the CosineAnnealingLR scheduler to adjust the learning rate dynamically. This configuration was chosen in order to optimize the model's performance across the varied few-shot learning scenarios presented in our experiments.

4.3. Main Results

Table 1 showcases the performance outcomes for few-shot learning tasks, with the evaluation metric being the F1 score. These experimental data provide insight into the efficacy of diverse training methodologies under few-shot learning conditions, particularly highlighting their performance on both base and novel classes across one-shot, five-shot, and ten-shot scenarios. To maintain uniformity and facilitate an accurate comparison of the impact of various training strategies, all experiments were conducted using a simple transformer network model.

Table 1. Few-shot result.

	One-Shot		**Five-Shot**		**Ten-Shot**	
	Base (F1)	**Novel (F1)**	**Base (F1)**	**Novel (F1)**	**Base (F1)**	**Novel (F1)**
baseline	0.667	0.003	0.679	0.100	0.698	0.217
baseline+	0.667	0.004	0.680	0.105	0.704	0.267
baseline++	0.678	0.091	0.713	0.325	0.746	0.478
p-f	0.882	0.418	0.870	0.415	0.873	0.453
MAML	0.702	0.576	0.819	0.773	0.884	0.858

In the one-shot learning scenario, the "baseline" approach, which involved direct training without any specialized techniques, achieved modest accuracy in base classes at 0.667 but struggled significantly with novel classes at 0.003. The introduction of focal loss in "baseline+" marginally improved the performance on novel classes to 0.004, indicating a slight advantage of using focal loss to address class imbalance in extremely low-data regimes. However, the combination of focal loss and oversampling strategies in "baseline++" marked a substantial improvement, especially for novel classes, jumping to 0.091. This suggests that addressing class imbalance through both loss function adjustments and sampling techniques can notably enhance model performance in few-shot learning tasks.

The "p–f" strategy, which employs a pre-train–fine-tune paradigm, demonstrated a significant leap in performance across all scenarios. Notably, in the one-shot condition, it achieved 0.882 and 0.418 for base and novel classes, respectively. This underscores the effectiveness of leveraging pre-trained models followed by fine-tuning on a small subset of target data, offering a robust way to enhance few-shot learning capabilities.

Most impressively, the "MAML" approach outperformed all other strategies, particularly in the novel class domain. In the one-shot learning scenario, MAML achieved a remarkable accuracy of 0.576 for novel classes, which further increased to 0.773 and 0.858 in the five-shot and ten-shot scenarios, respectively. This demonstrates MAML's superior

ability to generalize from very few examples, thanks to its meta-learning framework that optimizes for quick adaptability to new tasks with minimal data.

Across all strategies, there is a clear trend of increasing performance with the number of shots, indicating that more examples per class consistently lead to better model performance. However, the rate of improvement varies significantly between approaches, with "MAML" showing the most substantial gains, highlighting its effectiveness in leveraging limited data for significant performance boosts.

In conclusion, the experimental results strongly advocate for the adoption of advanced training strategies, such as pre-train–fine-tune paradigms and meta-learning algorithms like MAML, in few-shot learning tasks. These methods significantly outperform traditional training approaches, especially in handling novel classes with extremely limited data, thereby offering promising avenues for enhancing the capabilities in few-shot learning scenarios.

5. Integrating Meta-Learning in an AIOps System

Through the successful execution of the aforementioned tasks, we have substantiated the practicality and applicability of meta-learning within AIOps functions. Moving forward, our objective is to architect a sophisticated AIOps platform that seamlessly incorporates meta-learning. This platform will automate monitoring, diagnostics, and response workflows, effectively managing IT operational issues within an IT infrastructure (see Figure 5). It aims to streamline anomaly detection and fault resolution while enhancing system adaptability and predictive maintenance capabilities through intelligent, data-driven models.

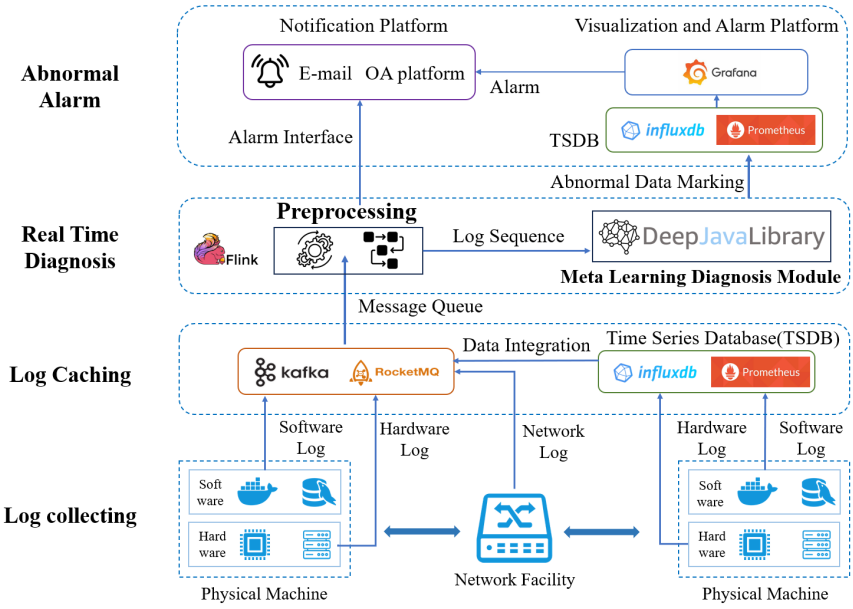


Figure 5. Illustration of platform.

The platform employs a layered approach to IT operations management, focusing on automation from fault detection to notification. Initially, the log-collecting layer aggregates data from software, hardware, and network systems, providing a foundation for comprehensive monitoring. The log-caching layer utilizes robust services like Kafka [44] and RocketMQ [45] to ensure efficient data throughput for subsequent analysis.

At the core is the real-time diagnosis layer, wherein tools such as Apache Flink [46] process log sequences and feed them into a meta-learning diagnosis module. This module, using advanced libraries like DeepJavaLibrary [47], applies meta-learning models to diagnose system states and identify anomalies with precision.

Detected anomalies are managed by the abnormal alarm layer, which triggers notifications via various channels, including email and organizational platforms, alerting system administrators and stakeholders. Integrated with visualization tools like Grafana [48] and

supported by time series databases such as InfluxDB [49] and Prometheus [50], the platform allows for real-time data visualization and abnormal data marking, ensuring prompt issue tracking and resolution.

This multi-tiered platform exemplifies a proactive, intelligent approach to IT operational management. It ensures high availability and performance by delivering a resilient and efficient solution for the early identification and rapid resolution of IT operational issues. The integration of meta-learning models enhances predictive capabilities, allowing the platform to learn from anomalies and continually refine its performance, thereby maintaining the highest standards of operational excellence.

5.1. Log Data Preprocessing

The data preprocessing for a log fault classification task involves several stages, as shown in Figure 6a:

Log Collecting and Caching: Data are gathered from various sources within the system, including servers, databases, and applications, resulting in a comprehensive collection of raw log files.

Log Parsing: These raw logs are processed to extract structured information, converting unstructured log data into a standardized format using predefined templates to identify specific events and parameters.

Log Grouping: The parsed logs are organized into meaningful groups based on time intervals or session IDs using techniques such as fixed and sliding windows. This segmentation creates coherent sequences representing specific activities or events.

Log Representation: Finally, the grouped log sequences are transformed into numerical representations suitable for classification models. This stage is crucial, as it captures the essential patterns and features necessary for fault identification and classification.

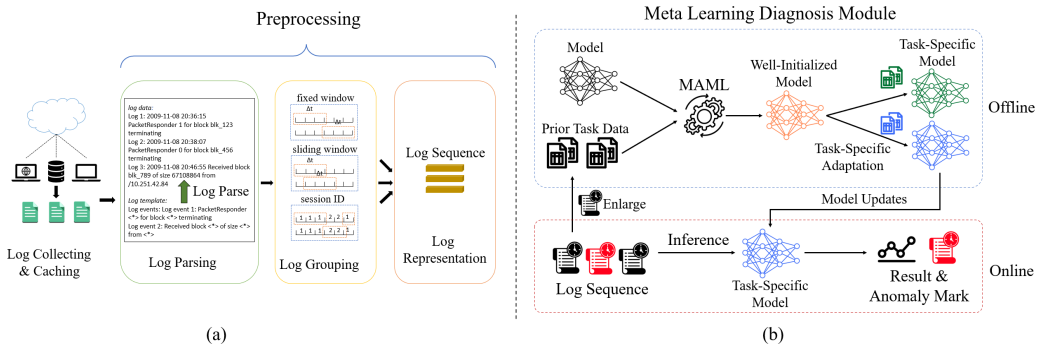


Figure 6. (**a**) Data preprocessing stage, (**b**) Offline training and online diagnosis of the meta-learning diagnosis module

5.2. Meta-Learning Diagnosis Module

Figure 6b illustrates the meta-learning diagnosis module within an AIOps system, showcasing the workflow from model training to real-time inference.

Initially, the model is trained offline using historical task data, which includes recorded outcomes serving as learning examples. The MAML algorithm helps the model find a well-initialized parameter setting, which is crucial for quick adaptation to future tasks.

After MAML, the model undergoes task-specific adaptation, fine-tuning with a small dataset of new task data and focusing on specific anomalies or operational behaviors. This refines the model, making it more precise for particular tasks.

In the online phase, the adapted model analyzes real-time log sequences to detect anomalies, marking any abnormal events. This can trigger alerts or automated responses, which are crucial for maintaining system performance and preventing downtime.

The interaction between online and offline components is dynamic. New data and anomalies encountered online are fed back into the offline training cycle, allowing the model to learn from the latest data. This continuous feedback loop ensures that the model

remains up-to-date and that its diagnostic capabilities improve over time, embodying the core concept of meta-learning.

6. Discussion

We have demonstrated the feasibility and effectiveness of using meta-learning in AIOps above, showcasing that meta-learning has many opportunities in AIOps practice. Nonetheless, there are still some open research issues related to meta-learning in AIOps which are deserving of further elaboration.

Complexity in Model Development Designing meta-learning models for AIOps is complex, requiring deep expertise in AI and an understanding of diverse IT operational contexts. This complexity extends to computational demands, as meta-learning algorithms typically require significant processing power and memory, which can be a barrier, especially for organizations with limited computational resources.

Data Quality and Relevance The success of meta-learning heavily relies on the quality and relevance of data. In AIOps, ensuring the quality of data, especially for rare or novel events, is challenging yet crucial for the model's performance. This issue is compounded by the need for large datasets to train robust models, which are not always available in IT operations.

Scalability and Real-World Implementation Implementing meta-learning models within existing IT ecosystems poses significant scalability and integration challenges. These models must be able to operate efficiently across different scales and configurations of IT infrastructure, often requiring custom adaptations to fit into existing monitoring tools and databases. Furthermore, the siloed nature of data in many organizations can hinder the effective deployment of meta-learning solutions.

Integration with Existing IT Infrastructure Beyond the technical challenges of integration, aligning meta-learning models with existing IT operations and workflows is crucial. This often involves overcoming resistance to change within organizations and ensuring that these advanced AI solutions complement rather than disrupt existing processes.

Future research could explore the development of more computationally efficient meta-learning algorithms that require less data and are better suited for real-world applications. Additionally, strategies for incremental learning, through which models can be updated with minimal retraining, may help with adapting to evolving IT environments without the need for extensive computational resources. Collaborative efforts between AI researchers and IT professionals could also foster better understanding and solutions tailored to the practical realities of AIOps.

7. Conclusions

In summary, this paper has explored the potential of meta-learning as an integral component of AIOps to enhance the robustness and responsiveness of IT systems. Our study examined how meta-learning, particularly the MAML framework, can improve the adaptability and predictive accuracy of models in the context of AIOps. Rigorous experiments and evaluations demonstrated that meta-learning enables models to generalize from a limited number of examples and significantly outperform traditional machine learning approaches in few-shot learning scenarios.

We proposed an advanced AIOps platform incorporating meta-learning models at the core of its diagnostic module. This system features robust log collection, efficient caching, and dynamic anomaly detection, creating a fully automated AIOps workflow. These components work together seamlessly to ensure efficient data acquisition and anomaly detection, fostering a proactive and intelligent IT management environment.

In conclusion, integrating meta-learning into AIOps represents a significant advancement in IT operations management. Our findings advocate for broader adoption of meta-learning algorithms in future AIOps solutions, promising intelligent, self-adapting IT systems essential for supporting complex and evolving digital infrastructures.

Author Contributions: The conceptualization and design of the study were led by Y.D., G.B., J.C., and Z.O. Experiments were carried out by Y.D., H.B., Y.W., K.X., and Z.Y. Data were analyzed by Y.D., H.B., and G.B. The meta-learning algorithm (MAML-KAD) was developed by Y.D., H.B., and K.X. The AIOps platform was designed and implemented by Y.W., Z.Y., and B.L. The manuscript was primarily written by Y.D. and H.B. with substantial contributions from G.B., Y.Z., J.C., Z.O., B.L., and S.W., who provided critical feedback and helped shape the research, analysis, and manuscript. H.B., Y.W., and Z.Y. made significant contributions to the preparation, execution, and analysis of this study, which justifies a shared first-authorship. As Y.D. initiated the study (together with G.B., J.C., and Z.O.), he is listed first among the shared first authors. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant 62076035, and supported by the Joint Funds of the National Natural Science Foundation of China (Grant No.U21B2022) and CMCC and BUPT cooperative program (Grant No.A2022256).

Data Availability Statement: The research data of this paper are available by contracting the corresponding author upon reasonable requests.

Acknowledgments: The authors would like to thank the anonymous reviewers for their invaluable comments.

Conflicts of Interest: Authors Yunfeng Duan, Guotao Bai, Bin Liu, Jiaxing Chen and Shenhuan Wang were employed by the company China Mobile Communications Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IT	Information Technology
IoT	Internet of Things
O&M	Operations and Maintenance
AIOps	Artificial Intelligence for IT Operations
SRE	Site Reliability Engineering
DevOps	Development and Operations
ERM	Empirical Risk Minimization
MAML	Model-Agnostic Meta-Learning
MAML-KAD	MAML-KPI Anomaly Detector
ANIL	Almost No Inner Loop
CM	Capacity Measurement
KPI	Key Performance Indicator
MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of Open Access Journals
TLA	Three Letter Acronym
LD	Linear Dichroism

References

1. Čolaković, A.; Hadžialić, M. Internet of things (IOT): A review of Enabling Technologies, challenges, and open research issues. *Comput. Netw.* **2018**, *144*, 17–39. [CrossRef]
2. Mao, B.; Tang, F.; Kawamoto, Y.; Kato, N. AI Models for Green Communications Towards 6G. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 210–247. [CrossRef]
3. Mao, B.; Liu, J.; Wu, Y.; Kato, N. Security and Privacy on 6G Network Edge: A Survey. *Commun. Surveys Tuts.* **2023**, *25*, 1095–1127. [CrossRef]
4. Ali, A.S.; Baddeley, M.; Bariah, L.; Lopez, M.A.; Lunardi, W.T.; Giacalone, J.P.; Muhaidat, S. Performance analysis and evaluation of RF jamming in IoT networks. In Proceedings of the GLOBECOM 2022—2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 2745–2751. [CrossRef]
5. Lunardi, W.T.; Lopez, M.A.; Giacalone, J.P. ARCADE: Adversarially Regularized Convolutional Autoencoder for Network Anomaly Detection. *IEEE Trans. Netw. Serv. Manag.* **2023**, 20, 1305–1318. [CrossRef]
6. Northrop, L.; Feiler, P.; Gabriel, R.P.; Goodenough, J.; Linger, R.; Longstaff, T.; Kazman, R.; Klein, M.; Schmidt, D.; Sullivan, K.; et al. *Ultra-Large-Scale Systems: The Software Challenge of the Future*; Defense Technical Information Center: Fort Belvoir, VA, USA, 2006.

7. Summary of the Amazon S3 Service Disruption in the Northern Virginia (US-EAST-1) Region. 2021. Available online: https://aws.amazon.com/message/41926/ (accessed on 20 December 2023).
8. Xiang, Z.; Guo, D.; Li, Q. Detecting mobile advanced persistent threats based on large-scale DNS logs. *Comput. Secur.* **2020**, *96*, 101933. [CrossRef]
9. Houle, J.J.; Roseen, R.M.; Ballestero, T.P.; Puls, T.A.; Sherrard J.J. Comparison of Maintenance Cost, Labor Demands, and System Performance for LID and Conventional Stormwater Management. *J. Environ. Eng.* **2013**, *139*, 932–938. [CrossRef]
10. Cheng, Q.; Sahoo, D.; Saha, A.; Yang, W.; Liu, C.; Woo, G.; Singh, M.; Saverese, S.; Hoi, S.C. AI for IT Operations (AIOps) on Cloud Platforms: Reviews, Opportunities and Challenges. *arXiv* **2023**, arXiv:2304.04661.
11. Dang, Y.; Lin, Q.; Huang, P. AIOps: Real-world challenges and research innovations. In Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion) (ICSE'19), Montreal, QC, Canada, 25–31 May 2019; pp. 4–5. [CrossRef]
12. Lerner, A. AIOps Platforms—Gartner. 2017. Available online: https://blogs.gartner.com/andrew-lerner/2017/08/09/aiops-platforms/ (accessed on 20 December 2023).
13. Levin, A.; Garion, S.; Kolodner, E.K.; Lorenz, D.H.; Barabash, K.; Kugler, M.; McShane, N. AIOps for a cloud object storage service. In Proceedings of the 2019 IEEE International Congress on Big Data (BigDataCongress), Milan, Italy, 8–13 July 2019; pp. 165–169. [CrossRef]
14. Wang, H.; Zhang, H. AIOPS prediction for hard drive failures based on stacking ensemble model. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; pp. 0417–0423.
15. Rijal, L.; Colomo-Palacios, R.; Sánchez-Gordón, M. AIOps: A multivocal literature review. In *Artificial Intelligence for Cloud and Edge Computing*; Internet of Things; Misra, S., Tyagi, A.K., Piuri, V., Garg, L., Eds.; Springer: Cham, Switzerland, 2022._2. [CrossRef]
16. Notaro, P. Cardoso, J.; Gerndt, M. A Survey of AIOps Methods for Failure Management. *ACM Trans. Intell. Syst. Technol.* **2021**, *12*, 81. [CrossRef]
17. Hutter, F.; Kotthoff, L.; Vanschoren, J. *Automated Machine Learning*; The Springer Series on Challenges in Machine Learning; Springer: Berlin/Heidelberg, Germany, 2019; Available online: https://library.oapen.org/bitstream/handle/20.500.12657/23012/1/1007149.pdf#page=46 (accessed on 15 April 2024).
18. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-Learning in Neural Networks: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5149–5169. [CrossRef]
19. Pang, G.; Shen, C.; Cao, L.; Hengel, A.V.D. Deep Learning for Anomaly Detection. *ACM Comput. Surv.* **2021**, *54*, 38. [CrossRef]
20. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
21. Fei-Fei, L.; Fergus, R.; Perona, P. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 594–611. [CrossRef] [PubMed]
22. Fink, M. Object classification from a single example utilizing class relevance metrics. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2005; pp. 449–456.
23. Fe-Fei, L.; Fergus; Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In Proceedings of the Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; Volume 2, pp. 1134–1141. . [CrossRef]
24. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Comput. Surv.* **2021**, *53*, 63. [CrossRef]
25. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*; MIT Press: Cambridge, MA, USA, 2018.
26. Vaidyanathan, K.; Trivedi, K.S. A measurement-based model for estimation of resource exhaustion in operational software systems. In Proceedings of the Proceedings 10th International Symposium on Software Reliability Engineering (Cat. No.PR00443), Boca Raton, FL, USA, 1–4 November 1999; pp. 84–93. [CrossRef]
27. Vapnik, V.; Levin, E.; Le Cun, Y. Measuring the VC-Dimension of a Learning Machine. *Neural Comput.* **1994**, *6*, 851–876. [CrossRef]
28. Bartlett, P.L.; Bousquet, O.; Mendelson, S. Local Rademacher complexities. *Ann. Stat.* **2005**, *33*, 1497–1537. [CrossRef]
29. Bottou, L.; Bousquet, O. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2008; pp. 161–168.
30. Bottou, L.; Curtis, F.E.; Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Rev.* **2018**, 60, 223–311. [CrossRef]
31. Zhang, Y.; Wei, Y.; Yang, Q. Learning to multitask. In Proceedings of the Neural Information Processing Systems, Montréal, QC, Canada, 2–8 December 2018. Available online: https://proceedings.neurips.cc/paper/2018/hash/aeefb050911334869a7a5d9e4d0e1689-Abstract.html (accessed on 25 December 2023).
32. Li, Z.; Zhou, F.; Chen, F.; Li, H. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. *arXiv* **2017**, arXiv:1707.09835. [CrossRef]
33. Nichol, A.; Schulman, J. Reptile: A Scalable Meta-learning Algorithm. Available online:https://yobibyte.github.io/files/paper_notes/Reptile___a_Scalable_Metalearning_Algorithm__Alex_Nichol_and_John_Schulman__2018.pdf (accessed on 25 December 2023).

34. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the ICML'17: Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; Volume 70, pp. 1126–1135. Available online: http://proceedings.mlr.press/v70/finn17a.html (accessed on 25 December 2023).
35. Raghu, A.; Raghu, M.; Bengio, S.; Vinyals, O. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML. *arXiv* **2020**, arXiv:1909.09157.
36. Koch, G. Siamese Neural Networks for One-Shot Image Recognition. Available online: https://www.cs.utoronto.ca/~gkoch/files/msc-thesis.pdf (accessed on 25 December 2023).
37. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. In Proceedings of the Advances in Neural Information Processing Systems 29, Barcelona, Spain, 5–10 December 2016; Volume 29. Available online: https://proceedings.neurips.cc/paper/2016/hash/90e1357833654983612fb05e3ec9148c-Abstract.html (accessed on 25 December 2023).
38. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
39. Diaz-De-Arcaya, J.; Torre-Bastida, A.I.; Zarate, G.; Minon, R.; Almeida, A. A Joint Study of the Challenges, Opportunities, and Roadmap of MLOps and AIOps: A Systematic Survey. *ACM Comput. Surv.* **2023**, *56*, 84. [CrossRef]
40. Wu, R.; Keogh, E.J. Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 2421–2429. [CrossRef]
41. Li, Z.; Zhao, N.; Zhang, S.; Sun, Y.; Chen, P.; Wen, X.; Ma, M.; Pei, D. Constructing Large-Scale Real-World Benchmark Datasets for AIOps. *arXiv* **2022**, arXiv:2208.03938.
42. 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 (WWW '18), Lyon, France, 23–27 April 2018; pp. 187–196. [CrossRef]
43. Alibaba Tianchi's AIOps Competition. Available online: https://tianchi.aliyun.com/competition/entrance/531947/information (accessed on 24 December 2023).
44. Kreps, J.; Narkhede, N.; Rao, J. Kafka: A Distributed Messaging System for Log Processing. Available online: https://pages.cs.wisc.edu/~akella/CS744/F17/838-CloudPapers/Kafka.pdf (accessed on 30 December 2023).
45. Ke, Q.; He, X. Message middleware-based message timing for alerting systems. In Proceedings of the International Conference on Cryptography, Network Security, and Communication Technology (CNSCT 2023), Changsha, China, 6–8 January 2023. [CrossRef]
46. Carbone, P.; Katsifodimos, A.; Ewen, S.; Markl, V.; Haridi, S.; Tzoumas, K. Apache flink: Stream and batch processing in a single engine. *IEEE Database Eng. Bull.* **2015**, *36*, 28–38.
47. DJL—Deep Java Library. Available online: https://djl.ai (accessed on 12 January 2024).
48. Chakraborty, M.; Kundan, A.P. Grafana. In *Monitoring Cloud-Native Applications*; Apress eBooks: New York, NY, USA, 2021; pp. 187–240.. [CrossRef]
49. Naqvi, S.N.Z.; Yfantidou, S.; Zimányi, E. Université libre de Bruxelles Advanced Databases Time Series Databases and InfluxDB. 2017. Available online: https://www.devopsschool.com/blog/wp-content/uploads/2022/09/influxdb_2017.pdf (accessed on 14 January 2024).
50. Prometheus. Prometheus—Monitoring System & Time Series Database. Available online: https://prometheus.io (accessed on 12 January 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.