Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Review:

Deep anomaly detection of temporal heterogeneous data in AIOps: a survey*

Jiayi GUI†§, Zhongnan MA†§, Hao ZHOU, Yan SU, Miaoru ZHANG, Ke YU†‡, Xiaofei WU

School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China

†E-mail: hypatia@bupt.edu.cn; zhongnanma@bupt.edu.cn; yuke@bupt.edu.cn

Received June 1, 2024; Revision accepted Nov. 18, 2024; Crosschecked Aug. 1, 2025; Published online Sept. 10, 2025

Abstract: The advancement of the fifth generation (5G) mobile communication and Internet of Things (IoT) has facilitated the development of intelligent applications, but has also rendered these networks increasingly complex and vulnerable to various targeted attacks. Numerous anomaly detection (AD) models, particularly those using deep learning technologies, have been proposed to monitor and identify network anomalous events. However, the implementation of these models poses challenges for network operators due to lacking expert knowledge of these black-box systems. In this study, we present a comprehensive review of current AD models and methods in the field of communication networks. We categorize these models into four methodological groups based on their underlying principles and structures, with particular emphasis on the role of recent promising large language models (LLMs) in the field of AD. Additionally, we provide a detailed discussion of the models in the following four application areas: network traffic monitoring, networking system log analysis, cloud and edge service provisioning, and IoT security. Based on these application requirements, we examine the current challenges and offer insights into future research directions, including robustness, explainability, and the integration of LLMs for AD.

Key words: Anomaly detection; AIOps; Large language models; Communication networks
https://doi.org/10.1631/FITEE.2400467 **CLC number:** TP3

1 Introduction

The advent of the fifth generation (5G) and the convergence of artificial intelligence have revolutionized Internet applications, enabling a symbiotic relationship between the physical and digital realms, and further ushering in an era of digital and intelligent transformation across various sectors (Zhang P et al., 2019; IMT-2030 (6G) Promotion Group, 2021). However, this transformation brings unprecedented challenges to network management due to the increasingly complex network structures, the vast volume of network traffic, and the multitude of connected devices required for these intelligent applications.

Recent incidents, such as the border gateway protocol (BGP) routing error by Abrams (2020), widespread outages by Heinle (2022) and KYODO NEWS (2022), and cellular phone outage across US by Montgomery (2024), indicate the susceptible vulnerabilities of current network management systems. Thus, there is an urgent demand for transition from traditional human-centric management to intelligent and automatic operational management technologies, which can ensure secure and efficient network performance.

In this context, the concept of artificial intelligence for information technology (IT) operations (AIOps) (Dang YN et al., 2019) has been proposed to manage the complexity and scale of modern Internet systems. AIOps aims to develop automated

‡ Corresponding author
§ These two authors contributed equally to this work
* Project supported by the National Natural Science Foundation of China (No. 62371057) and the 111 Project of China (No. B08004)
ORCID: Ke YU, https://orcid.org/0000-0002-1158-1483
© Zhejiang University Press 2025

anomaly detection (AD) models capable of processing complex, voluminous data streams, identifying emergent patterns of anomalies, and remaining resilient against the ever-changing landscape of network threats. The emergence of large language models (LLMs) has introduced novel solutions for AIOps development. Recent studies show that LLMs excel in pattern recognition and reasoning across complex token sequences. Beyond traditional natural language tasks, LLMs can integrate text and time-series data to effectively perform time-series forecasting and AD (Su et al., 2024). Time-LLM (Jin et al., 2024b) demonstrates remarkable performance in time-series forecasting tasks by reprogramming time-series data into textual formats suitable for LLMs and leveraging meticulously crafted text prompts. To address time-series AD, TS-BERT (Dang WX et al., 2021) introduces the pretraining and fine-tuning paradigm, effectively overcoming the challenge of modeling long-range dependencies. As demonstrated in Fig. 1, the regular procedure of AD in AIOps for secure communication system monitoring can be divided into data collection, data preparation and representation, data analysis and fusion, and pattern recognition and decision-making. Although several survey papers (Cook et al., 2020; Li G and Jung, 2023; Zhong et al., 2023; Su et al., 2024) have reviewed the implementation of AD models, they either lack comprehensive coverage of technologies, such as recent LLM-based methods, or do not describe application-specific details.

The scope of this study is to provide a comprehensive review of current AD models, especially deep learning models, including those based on LLMs, and to introduce their specific applications in the field of wireless communication networks. Specifically, this study covers network traffic monitoring, which involves sophisticated pattern recognition techniques to identify deviations from normative data flows, networking system log analysis, which parses and interprets vast logs to detect aberrant behaviors and system states, cloud and edge computing service provisioning, tailored to energy-limited devices to develop AD models, and Internet of Things (IoT) security, which monitors malicious sensor and actuator data in industrial networks.

The contributions of our study can be summarized as follows:

1. The survey presents the first comprehensive content on AD in AIOps within the communication domain, providing a holistic view of the field and its applications.

2. The survey delves deeply into critical AIOps scenarios within the communication domain, such as network traffic monitoring, networking system log analysis, cloud and edge service provisioning, and IoT security. We provide thorough analysis and detailed explanations for each scenario.

3. The survey establishes a comprehensive categorization framework that includes specific methods and detailed introductions to datasets, enabling

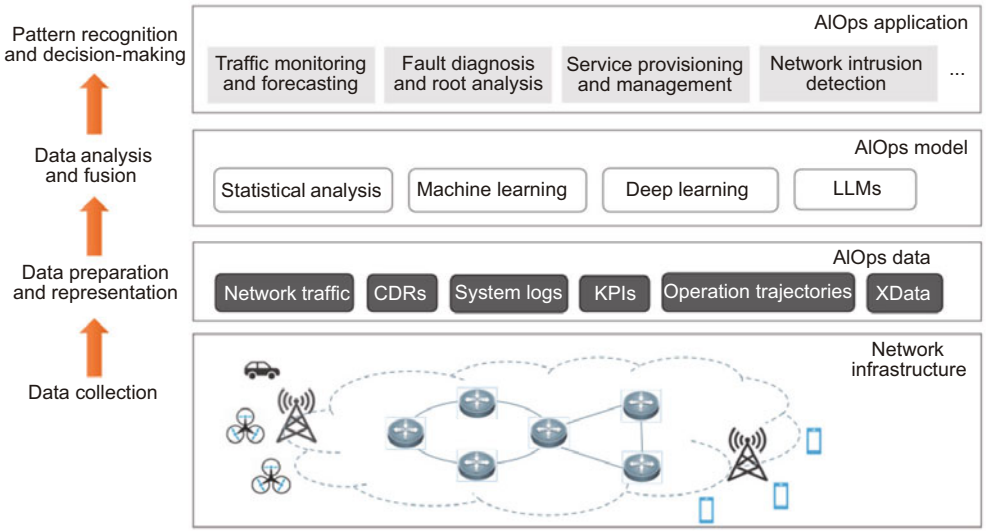

Fig. 1 AIOps framework with data processing for communication networks. CDRs: call detail records

researchers and practitioners to navigate the field more effectively.

4. The survey introduces and discusses the current approaches and future directions of leveraging LLMs for AD in AIOps within the communication domain, highlighting their potential to revolutionize the field. By identifying key challenges and opportunities in explainability, robustness, and the integration of LLMs, we provide valuable insights and guidance for future research efforts in AD for AIOps.

2 Overview

AIOps introduces innovative approaches, and presents new challenges for the evolution of communication networks, highlighting the need for a thorough understanding of the workflow of intelligent communication systems. As demonstrated in Fig. 2, the pipeline of AIOps in the communication network can be divided into the following five stages: data collection, data preprocessing, feature extracting, model training, and model deploying. Data collection gathers key metrics such as network key performance indicators (KPIs), traffic data, and logs. These datasets serve as the foundation for subsequent analysis, and are crucial for understanding network performance and detecting anomalies. Data preprocessing addresses the missing data imputation, normalization, and sliding windows to prepare the data for analysis. At the feature extracting stage, key features are extracted from the preprocessed data to capture underlying patterns in the network. Temporal features (e.g., time-based patterns in network traffic), spectral features (e.g., frequency domain characteristics), and spatial features (e.g., topology-related network behaviors) are identified. The model training stage uses the collected dataset to train a range of techniques aimed at identifying unusual or abnormal patterns in the network behavior. These models include statistical methods, machine learning methods, deep learning methods, and LLMs, which have recently been applied to analyze network logs and time-series data more effectively. Finally, the model deploying stage applies these models to real-world network tasks, such as traffic monitoring, log analysis, cloud and edge service provisioning, and IoT security.

The survey is structured as follows: Section 3 defines anomaly and AD, and compares the scope of our study with related works. Section 4 discusses

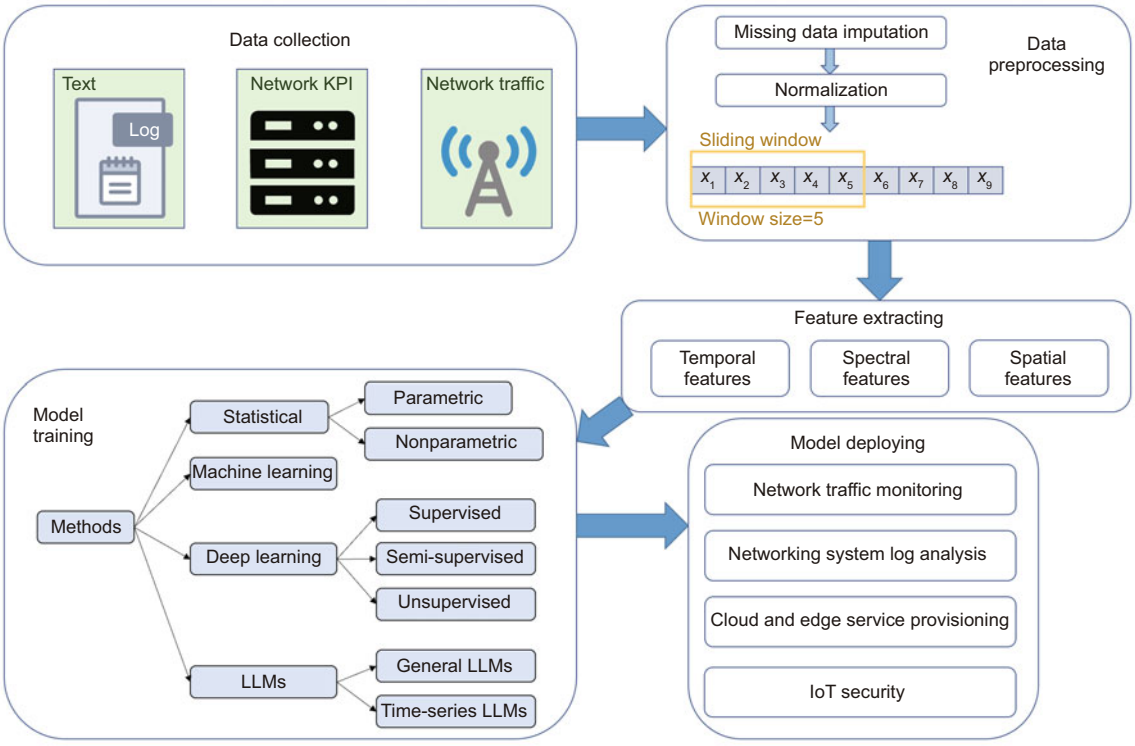

Fig. 2 Pipeline of AIOps in communication network

categorization frameworks, current methods, and datasets. Section 5 explores AD for network traffic monitoring with datasets and methods. Section 6 focuses on networking system log analysis. Section 7 examines AD for cloud and edge service provisioning. Section 8 discusses AD for IoT security. Section 9 projects challenges nowadays and future trends, emphasizing interpretability, robustness, and the use of LLMs. Section 10 concludes with a summary of insights.

3 Background

3.1 AD definition

AD is a prevalent and crucial topic across different domains and applications. Hawkins (1980) provided an early definition of anomaly, or outlier.

Definition 1 An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.

Definition 1 implicitly presupposes the existence of a norm-generating mechanism that produces normal, or nonanomalous, data points. Anomalies, in terms of features and data patterns, exhibit significant deviations from this norm. Therefore, the process of AD entails identifying patterns in data that diverge from the anticipated behaviors (Chandola et al., 2009).

In communication networks, AD primarily focuses on the analysis of KPIs that reflect various aspects of network performance. These KPIs, such as throughput, latency, packet loss, error rate, and quality of experience (QoE), serve as essential metrics for evaluating the efficiency, reliability, and overall performance of network infrastructure. By monitoring these indicators, network administrators can optimize operations, and ensure high-quality service. Since KPIs are predominantly time-series data, the following three main types of anomalies are relevant: point, contextual, and collective (Choi et al., 2021). Point anomalies are individual data points or sequences that deviate sharply from the normal range, often due to sensor errors or abnormal operations, and are detected by comparing values against control limits. Contextual anomalies involve a group of points that do not contain extreme values, but alter the signal's shape, making them harder to detect. Collective anomalies consist of data points that gradually form a divergent pattern over time, requiring long-term context analysis to identify the deviation from normal behavior. Other researchers have identified five key data-oriented dimensions to describe anomaly types and subtypes as follows: data type, relationship cardinality, anomaly level, data distribution, and data structure. These dimensions help classify and characterize anomalies in network data. The data type distinguishes among quantitative, qualitative, and mixed attributes. The relationship cardinality differentiates between the univariate and multivariate relationships. The anomaly level categorizes anomalies as atomic (individual cases) or aggregate (groups or collective patterns). The data distribution examines the collection and patterns of attribute values, further refining anomaly classification. Finally, the data structure considers formats such as log data and time series. Consequently, the definition of anomalies varies slightly depending on the specific application domain.

3.2 Problem formulation

The collected data can be denoted as $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]$, where $\boldsymbol{X} \in \mathbb{R}^{D\times N}, \boldsymbol{x}_i \in \mathbb{R}^D$, N is the number of samples, $\mathbb{R}$ is the set of real numbers, and D is the number of features. In the context of AD, the collected data can be in either temporal formats (e.g., univariate and multivariate time-series data) or non-temporal formats (e.g., tabular and graph data). The task of AD is to find a map function $f : \mathbb{R}^D \to \mathbb{R}$, such that $y_n = f(\boldsymbol{x}_n)$; the binary variable $y_n = 1$ $(n = 1, 2, ..., N)$ indicates that the sample is predicted to be an anomaly, and $y_n = 0$ indicates that the sample is normal.

3.3 Related works

Several surveys have addressed AD tasks, methods, and applications according to Table 1. Cook et al. (2020) examined IoT data's challenges and methods for detecting anomalies in univariate and multivariate time series. Our survey expands beyond IoT to include network traffic, logs, cloud and edge services, and so on, offering a broader application scope. Li G and Jung (2023) categorized three anomaly types in multivariate series, focusing on applications such as fraud and fault detection. In contrast, our study delves deeper into the AIOps domain

in communication systems, incorporating a wider range of methods, including traditional machine learning and statistical approaches, not just deep learning. While Zhong et al. (2023) extensively covered AIOps AD separately in univariate and multivariate data, they missed detailed classifications and introduction of application domains within AIOps. Su et al. (2024) summarized LLM methodologies, their effectiveness compared to traditional methods, and their challenges. Our survey, distinctively, focuses on cutting-edge research into LLM applications in varied domains, providing a comprehensive and specialized examination.

Our survey encompasses the following four application domains, as illustrated in Fig. 3: network traffic monitoring, networking system log analysis, cloud and edge service provisioning, and IoT security. These domains are built on foundational models, including statistical methods, machine learning methods, deep learning methods, and LLMs, which will be discussed later.

4 Methodology

In this section, we introduce practical methods for AD. We categorize these methods into three main categories as follows: statistical methods, machine learning methods, and deep learning methods. Given the recent trend of leveraging LLMs for AD, we also provide a separate discussion on LLM-based methods, including both general practical applications and time-series specific implementations. This study presents Table 2 to enhance the understanding of the advantages and disadvantages of different methods. To respect the original text, the letters of the formulas appearing are not otherwise modified, and the relevant meaning is given each time they appear.

Table 1 The scope of related works and our paper

Reference	Data type		Methodology				Application			
	Time series	Log	Statistical	Machine learning	Deep learning	LLMs	Traffic analysis	Log analysis	Cloud and edge services	IoT
Cook et al. (2020)	✓		✓	✓	✓					✓
Li G and Jung (2023)	✓				✓					✓
Zhong et al. (2023)	✓		✓	✓	✓					
Su et al. (2024)	✓				✓	✓				
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

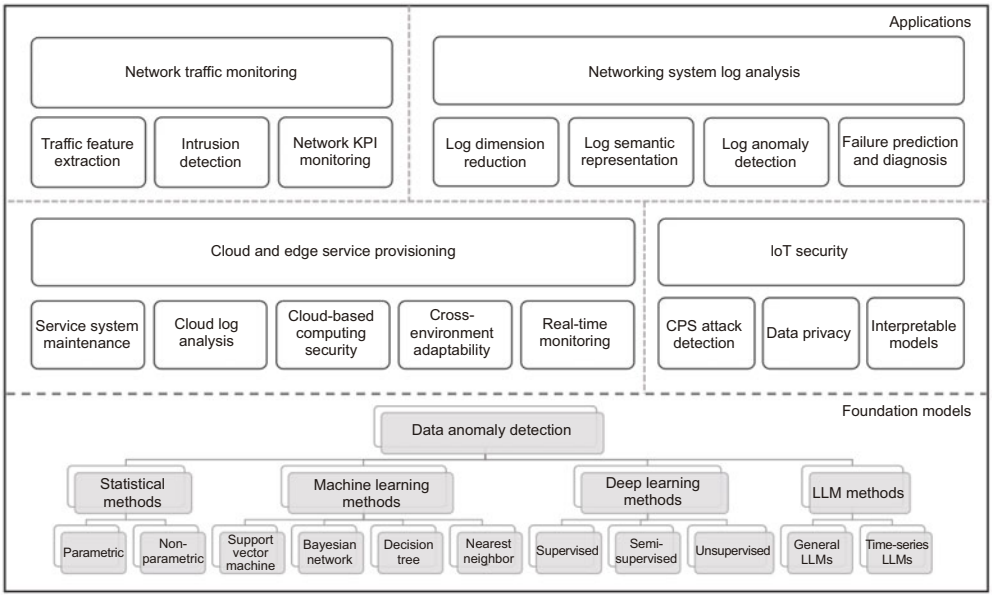

Fig. 3 Structure of the survey

Table 2 Summary of advantages and disadvantages of time-series anomaly detection (AD) methods

Method	Reference	Advantage	Disadvantage
Gaussian mixture model	Luo and Zhong, 2017	Captures multimodal distribution of data. Suitable for complex distribution AD	Complex parameter tuning and sensitive to initialization. High computational complexity for high-dimensional data
Histogram analysis	Bansod and Nandedkar, 2020	Simple and easy to implement. Good for detecting global distribution shifts	Limited in capturing complex patterns. Not suitable for detecting local anomalies in high-dimensional data
Markovian model	Lüdtke et al., 2020	Captures hidden structures and state transitions. Suitable for noisy sequences	Depends on the number of states and the initial parameter selection. Poor performance on long-term dependencies
Kernel density estimator	Aboubacar and El Machkouri, 2020	Nonparametric and flexible in capturing complex data distributions. Effective for small and continuous datasets	Sensitive to bandwidth selection. Poor performance with high-dimensional data
Isolation forest	Liu FT et al., 2008	Scales well to high-dimensional data. No need for data standardization	Limited flexibility in handling complex time series. Less effective for strongly correlated time series
One-class SVM	Amer et al., 2013; Song J et al., 2013; Chand et al., 2016	Handles nonlinear distributions. Effective for small sample sizes	Sensitive to parameter selection and high computational complexity. Long training time
Clustering	Song J et al., 2013	Identifies similar groups in data. Applicable for unsupervised AD	Depends on distance metrics and high complexity. Difficult to handle high-dimensional and dynamic data
Nearest neighbor	Pajouh et al., 2017	Intuitive and easy to implement. Nonparametric and does not require assumptions about data distribution	Highly sensitive to the choice of the distance metric. Suffers from the "curse of dimensionality"
Bayesian network	Smith et al., 2010; Mascaro et al., 2014	Models probabilistic relationships between variables. Can handle uncertainty and missing data well	Computationally intensive for large datasets. Requires expert knowledge for structure learning
LSTM	Hundman et al., 2018; Le et al., 2019; Gao S et al., 2020; Lee et al., 2020	Captures long and short-term dependencies. Effective on complex time series	Long training time and high computational cost. Complex model tuning and sensitive to hyperparameters
Active learning	Zhang X et al., 2019a; Ren PZ et al., 2021	Reduces labeling cost by selecting the most informative data points. Improves model efficiency	Relies on a good initial model to start with. May struggle in imbalanced datasets
Transfer learning	Zhang X et al., 2019a; Zhang SL et al., 2021 Zhuang et al., 2021;	Leverages pretrained models and reduces the need for large datasets. Can generalize well across tasks	May not perform well when the source and target domains are too different. Requires significant fine-tuning in some cases
Autoencoder	Chen ZM et al., 2018; Xu HW et al., 2018; Liu C et al., 2022	Captures nonlinear features. Effective in unsupervised tasks	Sensitive to overfitting. Complex architecture and hyperparameter tuning
GAN	Li D et al., 2019b; Audibert et al., 2020	Can generate realistic synthetic data for AD. Effective in modeling complex distributions	Sensitive to hyperparameters. High computational cost, and requires a large amount of training data
Transformer	Tuli et al., 2022	Captures long-range dependencies well. Efficient training using parallelization	Requires large datasets for effective training. Complexity in model architecture and tuning
CNN	Ren HS et al., 2019; He et al., 2020	Parallel processing of long sequences and fast training. Captures time dependencies effectively	Poor performance on nonstationary data. Sensitive to noise
General LLMs	Chen ZH et al., 2023; Gruver et al., 2024; Jin et al., 2024b; Xue and Salim, 2024	Effective at understanding patterns in sequential data. Suitable for structured, unstructured, and mixed data types	High computational resources required for training. Challenging to transfer time series into natural language
Time-series LLMs	Chang et al., 2024; Ekambaram et al., 2024; Gao SH et al., 2024	Tailored for time-series prediction and AD. Efficient processing with reduced computational costs compared to general LLMs	Requires clean data for training, unlike LLMs which are more flexible with raw data. Limited generalization beyond time-series tasks

4.1 Statistical methods

4.1.1 Parametric methods

Parametric models (Kourtis et al., 2016) assume that the data are sampled from a known distribution. During training, the model parameters are estimated using statistical techniques applied to the training data. These models are well-suited for large datasets since their complexity is independent of data size, enabling fast testing and evaluation. However, accurate distribution assumptions are crucial, requiring prior knowledge of the data. Gaussian distribution is commonly assumed (Luo and Zhong, 2017) for both univariate and multivariate continuous data, with mean and covariance estimated using the maximum likelihood estimation. For categorical data, multinomial distributions are often used (Reynolds, 2009), formulated by

$$P(x|\theta)=\sum_{k=1}^{K}\alpha_k\phi(x|\theta_k), \tag{1}$$

where K represents the number of sub-Gaussian models in the mixture model, α_k denotes the probability that the data belong to the k^{th} submodel, and $\phi(x|\theta_k)$ is the probability density function of the Gaussian distribution for the k^{th} submodel (Reynolds, 2009), specifically given by

$$\phi(x|\theta_k)=\frac{1}{\sigma_k\sqrt{2\pi}}\exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right). \tag{2}$$

Here, x is the one-dimensional data, μ_k is the mean value of the k^{th} Gaussian component, and σ_k^2 is the variance of the k^{th} Gaussian component. Sequential data can be modeled using Markovian models (Lüdtke et al., 2020). In real-world applications, datasets often exhibit random and diverse distributions, making a single model insufficient to capture the true underlying statistical characteristics. In such cases, a mixture of multiple models is commonly assumed (Ahmed M et al., 2016), and the parameters and model weight coefficients are jointly optimized.

4.1.2 Nonparametric methods

The nonparametric method (Kourtis et al., 2016) differs from parametric methods as it does not make assumptions about data distributions. These methods adapt their complexity to fit the size and complexity of the data. One representative approach is histogram analysis (Bansod and Nandedkar, 2020), which uses frequency of occurrences to capture normal behavior patterns. Histogram analysis efficiently handles low-dimensional univariate data, but faces challenges as dimensions increase. Kernel density estimators (KDEs) (Aboubacar and El Machkouri, 2020) employ a collection of distributed kernels to estimate the probability density function over the data space. To estimate an unknown density function f, the KDE is defined as follows (Aboubacar and El Machkouri, 2020):

$$\hat{f}_h(x)=\frac{1}{nh}\sum_{i=1}^{n}K\left(\frac{x-x_i}{h}\right), \tag{3}$$

where x_i $(i=1,2,...,n)$ denotes the independent samples drawn from a distribution with density f, K represents the kernel function, and $h>0$ is a smoothing parameter known as the bandwidth.

4.2 Machine learning methods

Machine learning is capable of effectively identifying patterns in data, specifically in the context of AD, where it distinguishes between the normal patterns and the significantly deviating abnormal patterns. In recent years, a lot of techniques based on isolation forest, support vector machine (SVM), Bayesian networks, clustering, and the nearest neighbor have been proposed for AD.

Isolation forest (Liu FT et al., 2008) is an AD algorithm that isolates observations by creating binary trees. It identifies outliers based on their shorter path lengths in the trees, making it effective for high-dimensional datasets. SVM has been used for AD (Ratsch et al., 2002) to learn a boundary region containing the training instances. To effectively address complex, nonlinear relationships in data, the Gaussian kernel function is widely employed. It is mathematically defined as follows (Ratsch et al., 2002):

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{|\boldsymbol{x}_i-\boldsymbol{x}_j|^2}{2\sigma^2}\right). \tag{4}$$

In this expression, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ represent the input data vectors, while σ is a parameter that governs the width of the kernel. The Gaussian kernel function facilitates the transformation of data into a higher-dimensional space, thereby enabling the effective separation of data points that are not linearly

separable in the original feature space. Extensions of the basic technique have been developed to enhance one-class SVMs (Amer et al., 2013). Bayesian networks are used for AD in the multi-class setting. Several variations of the basic technique have been proposed that capture the conditional dependencies between the different attributes using more complex Bayesian networks (Smith et al., 2010; Mascaro et al., 2014). Clustering and the nearest neighbor methods are commonly used approaches in AD. Clustering methods partition the data into K clusters by minimizing the intra-cluster variance. For cluster C_k, its centroid $\boldsymbol{\mu}_k$ is calculated as follows (Rokach and Maimon, 2005):

$$\boldsymbol{\mu}_k = \frac{1}{|C_k|} \sum_{\boldsymbol{x}_i \in C_k} \boldsymbol{x}_i, \tag{5}$$

where $|C_k|$ is the number of data points in the cluster, and $\boldsymbol{x}_i$ is the data in C_k. The objective of clustering is to minimize the sum of squared distances between each data point and its assigned centroid. The objective function is formulated as follows (Rokach and Maimon, 2005):

$$J = \sum_{k=1}^{K} \sum_{\boldsymbol{x}_i \in C_k} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2, \tag{6}$$

where K is the number of clusters, $\boldsymbol{\mu}_k$ is the centroid of cluster C_k, and $\|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2$ is the Euclidean distance between the point $\boldsymbol{x}_i$ and the centroid $\boldsymbol{\mu}_k$. The nearest neighbor method relies on a distance metric to identify the nearest neighbors. A common choice is the Euclidean distance $d(\boldsymbol{x}, \boldsymbol{x}_i)$ between a test point $\boldsymbol{x}$ and a training point $\boldsymbol{x}_i$ (Rokach and Maimon, 2005):

$$d(\boldsymbol{x}, \boldsymbol{x}_i) = \sqrt{\sum_{j=1}^{m} (\boldsymbol{x}_j - \boldsymbol{x}_{ij})^2}, \tag{7}$$

where m is the number of features. Other distance metrics, such as Manhattan distance or cosine similarity, can also be employed depending on the application.

Clustering can be particularly useful when the underlying structure of normal data points is well-defined, and distinct clusters can be formed. On the contrary, the nearest neighbor methods can be more effective when anomalies are isolated and have a few neighbors. Many models enhance the performance of AD by combining various machine learning algorithms mentioned above. For example, Chand et al. (2016) designed a model based on SVM and random forest for intrusion detection. Song J et al. (2013) developed a practical unsupervised AD system based on SVM and clustering. Similarly, Pajouh et al. (2017) designed a hierarchical AD system based on Naive Bayes and the K-nearest neighbors.

4.3 Deep learning methods

4.3.1 Supervised methods

Supervised methods require labeled data points in the training dataset to learn the distinctions between the normal and anomalous instances using discriminative or generative algorithms (Esling and Agon, 2012). In terms of the overall accuracy, supervised AD is theoretically superior due to its clear understanding of normality and abnormality. Neural network models, particularly multi-layer perceptron (MLP) and recurrent neural networks (RNNs), are known for their robustness and accuracy.

MLP, a feedforward neural network with multiple hidden layers, can effectively capture complex time dependencies and correlations within the data. On the contrary, RNNs have gained attention in time-series forecasting due to their ability to retain memory states from previous time steps. The long short-term memory (LSTM) model, a variant of RNN, has introduced innovative solutions and demonstrated exceptional performance in forecasting time-series data with long-term temporal dependencies (Le et al., 2019; Gao S et al., 2020). Previous studies (Hundman et al., 2018; Lee et al., 2020) have shown that LSTM outperforms many other approaches and neural networks in terms of prediction accuracy.

Another significant category of supervised methods is the convolutional neural network (CNN). CNNs are particularly effective in capturing temporal features. Additionally, they possess unique characteristics such as parameter sharing and permutation invariance, which helps reduce complexity and enhances robust representation learning. The MTAD-TF (He et al., 2020) uses a CNN to capture temporal patterns and a graph attention network (GAT) to model spatial correlations, subsequently integrating spatiotemporal features through a gated RNN.

4.3.2 Semi-supervised methods

Semi-supervised methods offer a promising approach in academic research, as they allow researchers to harness the precision and effectiveness of supervised learning (SL) while alleviating the laborious task of manual labeling. These functions are achieved by integrating auxiliary techniques, such as active learning and transfer learning, into the learning process.

Active learning (Ren PZ et al., 2021) reduces labeling workload by selecting informative samples for annotation. The model is iteratively trained on labeled data and evaluated on unlabeled data to identify the most valuable samples. For instance, active transfer AD (ATAD) (Zhang X et al., 2019a) combines active learning and transfer learning techniques. It extracts general features from labeled and unlabeled time series, employing instance- and feature-based transfer learning methods to train a base detection model. Active learning recommends informative samples for manual labeling, improving performance by considering uncertainty and context diversity. ATAD targets high-precision AD with limited labeled data, such as cross-dataset tasks.

Transfer learning (Zhuang et al., 2021) leverages knowledge from pretrained models to enhance performance on related target tasks. In this context, maximum mean discrepancy (MMD) is employed to align the distributions of the source and target domains. Minimizing MMD helps mitigate domain shift, thereby improving model performance in the target domain. Mathematically, MMD is defined as follows (Zhuang et al., 2021):

$$\begin{aligned}&\mathrm{MMD}(P_{\mathrm{s}}, P_{\mathrm{t}})\\&=\left\|\mathbb{E}_{X_{\mathrm{s}}\sim P_{\mathrm{s}}}[\phi(X_{\mathrm{s}})]-\mathbb{E}_{X_{\mathrm{t}}\sim P_{\mathrm{t}}}[\phi(X_{\mathrm{t}})]\right\|^2,\end{aligned}\tag{8}$$

where P_{s} and P_{t} are the source and target distributions, respectively, and ϕ is a feature mapping function. In this study, $\mathbb{E}_{X_{\mathrm{s}}\sim P_{\mathrm{s}}}[\phi(X_{\mathrm{s}})]$ denotes the expectation of the feature mappings of samples X_{s} drawn from the source distribution P_{s}, while $\mathbb{E}_{X_{\mathrm{t}}\sim P_{\mathrm{t}}}[\phi(X_{\mathrm{t}})]$ denotes the expectation of the feature mappings of samples X_{t} drawn from the target distribution P_{t}. By fine-tuning the pretrained models on a small labeled dataset, transfer learning adapts to specific tasks and capitalizes on prior knowledge, effectively reducing the need for extensive labeled data and minimizing training time. For instance, PUAD (Zhang SL et al., 2021) addresses label update challenges using positive-unlabeled (PU) learning (Zhang JQ et al., 2019). It pretrains a linear model on the positive set, and employs active learning-based self-training with a random forest classifier. The classifier identifies negative and potential positive samples in the unlabeled set, verified by operators. This iterative process continues until sufficient labeled samples are obtained.

4.3.3 Unsupervised methods

Unsupervised techniques are typically employed in situations where no prior knowledge of the data labels is available. A typical example of this approach is AD based on autoencoders (AEs) (Chen ZM et al., 2018). An AE generally consists of the following two primary components: an encoder, which compresses the input data, and a decoder, which reconstructs the input by minimizing the difference between the original and reconstructed vectors. Variational autoencoders (VAEs) extend AEs by introducing a probabilistic model, allowing them to generate new samples resembling the input data by learning the data's distribution (Xu HW et al., 2018). Conditional variational autoencoders (C-VAEs) further enhance VAEs by incorporating external variables, such as timestamps, to better capture anomalies related to time-dependent information (Liu C et al., 2022). Another example is the various generative models, such as generative adversarial network (GAN) (Li D et al., 2019b; Audibert et al., 2020) and Transformer (Tuli et al., 2022). These generative models develop the normal data distribution, and instances that do not conform to this distribution are detected as anomalies.

Other methods, such as spectral residual CNN (SR-CNN) (Ren HS et al., 2019), use a CNN directly on the output of the SR model. The SR algorithm involves applying the Fourier transform to obtain the log amplitude spectrum, calculating the spectral residual, and then applying the inverse Fourier transform to return to the time domain. The Fourier transform can be formulated as follows (Bochner and Chandrasekharan, 1949):

$$X(f)=\mathcal{F}\{x(t)\}=\sum_{t=0}^{N-1}x(t)\mathrm{e}^{-\mathrm{j}2\pi ft/N},\tag{9}$$

where $x(t)$ represents the time-series data, $X(f)$ is the Fourier transform of the time series, and N is the

number of samples in the time series. By employing a CNN as its discriminative model architecture, SR-CNN simplifies the learning process compared with that using the original time series.

4.4 LLM methods

4.4.1 General LLMs

General LLMs encompass a diverse array of models, including open-source options such as LLaMA (Touvron et al., 2023a, 2023b), GLM (Du ZX et al., 2022; Zeng et al., 2023), QWEN (Bai et al., 2023), and Mistral (Jiang AQ et al., 2023). On the contrary, application programming interface (API)-based LLMs such as the generative pretrained Transformer (GPT) family, including models such as GPT-3 (Brown et al., 2020) and its successor (OpenAI, 2024), and Claude offer robust pretrained capabilities accessible via cloud services. General LLMs are characterized by their extensive training on vast corpora of multimodal data, encompassing text, images, and videos in both structured and unstructured formats. Nevertheless, the challenge remains in seamlessly integrating time-series data with natural language to maximize the potential of these tools. One approach is to use LLMs through prompt-based methods. For instance, PromptCast (Xue and Salim, 2024) reformulates numerical inputs and outputs into prompts, framing the forecasting process as a sentence-to-sentence conversion, allowing language models to handle forecasting tasks. Another example involves prompting ChatGPT with financial news, social media insights, and corporate reports to enhance predictions of stock market movements (Chen ZH et al., 2023). In addition to prompt-based methods, a notable approach is reprogramming (Jin et al., 2024b), which introduces the concept of modifying input data to leverage a model's latent knowledge without altering its parameters. Time-LLM (Jin et al., 2024b) transforms input time-series into textual prototype representations, making them more interpretable for the LLM, and enhancing context with declarative prompts to guide the LLM's reasoning. Similarly, LLMTIME finds way to align time-series data with API-based LLMs such as GPT-3 including effectively tokenizing time-series data and converting discrete distributions over tokens into highly flexible densities over continuous values (Gruver et al., 2024).

4.4.2 Time-series LLMs

In the realm of time series, a specialized subset of LLMs is gaining prominence, tailored specifically for temporal data processing. These models are adept at handling sequences, and are increasingly used for tasks such as forecasting, detecting anomalies, classifying, and performing imputation. By leveraging architectures that excel at capturing long-term dependencies, such as RNNs, LSTMs as mentioned above, and more recently Transformer-based models that possess parameters over 0.5 billion adapted for sequential data, these LLMs can discern complex temporal dynamics. UniTS uses sequence and variable attention along with a dynamic linear operator to build a unified model that can do classification, forecasting, imputation, and AD tasks at the same time (Gao SH et al., 2024). LLM4TS designs a two-stage fine-tuning strategy: time-series stage to align LLMs with the nuances of time-series data and forecasting fine-tuning stage for downstream time-series forecasting tasks. The second stage can possibly be manipulated to adapt to AD tasks (Chang et al., 2024). Tiny time mixers (TTMs) make their effort to develop fast and tiny general pretrained models possessing less than one million parameters, with effective transfer learning capabilities for forecasting based on the lightweight TSMixer architecture (Ekambaram et al., 2024).

Generalized LLMs have notably demonstrated the capability to process a wide array of diverse input formats, including structured data formats such as JSON and XML, beyond the scope of time-series LLMs. Although no existing studies of generalized LLMs have specifically focused on addressing specialized anomalies, these models exhibit potential in detecting a broader range of anomalies, such as routing errors, by leveraging their ability to understand and analyze complex patterns within the data after specialized training. As our study delves deeply into AD in temporal data, the subsequent discussion of LLM techniques will primarily focus on their application in the temporal domain.

4.5 Current datasets

In the realm of AIOps, particularly within the context of AD in communication networks, the availability and utilization of diverse datasets play a pivotal role. This study provides a detailed overview

of the most prominent datasets used in the field as shown in Table 3, each contributing uniquely to the development and evaluation of AD methods. The datasets span various application scenarios, including IoT services such as IoT security, network traffic monitoring, cloud and edge service provisioning, and networking system log analysis. They encompass a wide range of data types, from multivariate time-series to log files, and vary significantly in terms of size, time length, and dimensionality. For instance, datasets such as PUMP and secure water treatment (SWaT) focus on IoT applications with detailed time-series data, while others such as Canadian Institute for Cyber Security Intrusion Detection System 2017 (CIC-IDS2017) and University of New South Wales-Network Benchmark 2015 (UNSW-NB15) are geared towards invasion detection with extensive network traffic records. Additionally, datasets such as hadoop distributed file system (HDFS) and BlueGene/L (BGL) provide rich log data crucial for identifying anomalies in large-scale systems. By presenting these datasets in a structured manner, this study aims to equip researchers and practitioners with the necessary resources to advance their work in AD, facilitating a deeper understanding of the challenges and opportunities within this domain.

4.6 Current methods

In this study, we conclude various methods employed for AD within AIOps, particularly focusing on their applications in communication networks as shown in Table 4. The methods are categorized based on their underlying techniques, such as RNN-, CNN-, graph-, and AE-based approaches. Each method is designed to address specific applications in AD, leveraging advanced algorithms to enhance detection accuracy and efficiency. Table 4 offers a detailed overview of these methods, including their supervision types (STs) and the datasets used for evaluation. The detailed introduction can be found in Sections 5–8 for each application domain.

Table 3 Overview of current datasets in different application domains

Dataset	Domain	Statistics (training; test)	Time length	Data type	Dim	Overall anomaly ratio
PUMP (DataSetsAI, 2020)	IoT	76 901; 143 401	5 months	Mul TS	44	10.05
SWaT (Mathur and Tippenhauer, 2016)	IoT	99 360; 89 984	11 days	Mul TS	25	11.99
WADI (Ahmed CM et al., 2017)	IoT	241 921; 15 701	16 days	Mul TS	67	7.09
N-BaIoT (Meidan et al., 2018)	IoT	7 062 606		Mul TS	115	92.13
BoT-IoT (Koroniotis et al., 2019)	IoT	73 million		Mul TS	29	0.013
5% BoT-IoT subset (Koroniotis et al., 2019)	IoT	3 668 522		Mul TS	43	0.013
NSL-KDD (Tavallaee et al., 2009)	IoT/Network traffic	125 973; 22 544		Mul TS	41	36.98
UNSW-NB15 (Moustafa and Slay, 2015)	Network traffic	175 341; 82 332	2 days	Mul TS	49	63.9
CIC-IDS2017 (Sharafaldin et al., 2018)	Network traffic	125 973; 22 533		Mul TS	83	47.1
CIC-IDS2018 (Sharafaldin et al., 2018)	Network traffic	16 232 943		Mul TS	80	16.9
CIC-DDoS2019 (Sharafaldin et al., 2019)	Network traffic	11 687 590		Mul TS	87	99.84
Edge-IIoTset (Ferrag et al., 2022)	IIoT/Network traffic	324 928; 38 813		Mul TS	61	29.3
USTC-TFC2016 (Wang W et al., 2017)	Network traffic	245 437; 27 271		Mul TS		95.59
KDD Cup 1999 (Cup, 2007)	Cloud service	5 000 000; 2 000 000	9 weeks	Mul TS	41	
ISOT CID phase1 (Aldribi et al., 2018)	Cloud service	24 519 987	5 days	Mul TS	5	37.58
ISOT CID phase2 (Aldribi et al., 2018)	Cloud service	12 418 998	6 days	Mul TS	5	21.32
TON_IoT (Alsaedi et al., 2020)	Cloud service	276 625; 184 428		Mul TS	46	3.6
X-IIoTID (Al-Hawawreh et al., 2022)	Cloud service	820 834	114 days	Mul TS	67	48.66
CICIoT2023 (Neto et al., 2023)	Cloud service	–		Mul TS	47	
HDFS (Xu W et al., 2009)	Log	11 175 629		Log		2.93
BGL (Liang Y et al., 2005)	Log	4 747 963		Log		7.34
Thunderbird (Oliner and Stearley, 2007)	Log	10 000 000		Log		0.49
Spirit (Oliner and Stearley, 2007)	Log	5 000 000		Log		15.29

Mul TS means multivariate time series; "–" means that the statistics is inaccessible; the blank means that the data do not exist. Dim: dimension; IIoT: Industrial Internet of Things

Table 4 Overview of current methods in different application domains

Domain	Method	Technique	ST	Dataset
Network traffic monitoring	SGmVRNN (Dai et al., 2022)	RNN-based	SL	CDN multivariate KPI dataset and SMD
	D-PACK (Hwang et al., 2020)	CNN-based	UL	USTC-TFC 2016
	MSRC (Duan et al., 2023)	AE-based	UL	KDDCUP99, NSL-KDD, UNSW-NB15, and CIC-IDS2018
	LSTM-GAN-G (Huang et al., 2022)	RNN-based	UL	CellPAD
	RENOIR (Andresini et al., 2021)	AE-based	SL	KDDCUP99, AAGM17, and CIC-IDS2017
	CMAE (Lu et al., 2022)	AE-based	UL	KDDCUP99
	ARCADE (Lunardi et al., 2023)	AE-based	UL	ISCX-IDS, USTC-TFC, and MIRAI-RGU
	APAE (Basati and Faghih, 2023)	AE-based	UL	UNSW-NB15, CIC-IDS2017, and KDDCUP99
	HCRNNIDS (Khan, 2021)	RNN-based	SL	CSE-CIC-IDS2018
	CNN-LSTM (Halbouni et al., 2022)	RNN-based	SL	CIC-IDS2017, UNSW-NB15, and WSN-DS
	DCNNBiLSTM (Hnamte and Hussain, 2023)	RNN-based	UL	CIC-IDS2018 and Edge_IIoT
	CANET (Ren KY et al., 2023)	CNN-based	SL	UNSW-NB1, NSL-KDD, CIC-IDS2017, and CICDDoS2019
	CBR-CNN (Chouhan et al., 2019)	CNN-based	UL	NSL-KDD
Networking system log analysis	LogPrompt (Liu YL et al., 2024)	LLM-based	UL	HDFS, Hadoop, Zookeeper, BGL, HPC, Linux, Proxifier, Android, and Spirit
	DeepLog (Du M et al., 2017)	RNN-based	SL	HDFS and OpenStack
	LogAnomaly (Meng et al., 2019)	RNN-based	UL	BGL and HDFS
	PLELog (Yang L et al., 2021)	RNN-based	SSL	BGL
	LogRobust (Zhang X et al., 2019b)	RNN-based	SL	HDFS and Microsoft's data
IoT security	NSIBF (Feng and Tian, 2021)	CNN-based+BF	SL	PUMP, WADI, and SWaT
	GAN-AD (Li D et al., 2019a)	RNN-based+ GAN-based	UL	SWaT
	MAE and MDAE (Vu et al., 2022)	AE-based	SL	N-BaIoT
	N-BaIoT (Meidan et al., 2018)	AE-based	SL	N-BaIoT
	FDL (Popoola et al., 2022)	DNN-based+FL	SL	BoT-IoT and N-BaIoT
	FeCo (Wang N et al., 2022)	FL+CL	SL	NSL-KDD
	FedPG (Nguyen et al., 2023)	FL	SL	NSL-KDD
Cloud and edge service provisioning	LogNL (Zhu et al., 2020)	RNN-based	SL	HDFS and OpenStack
	GLLD (Khalaf et al., 2022)	Graph-based	SL	BoT IoT and code red worm
	CGNN-MHSA-AR (Song YJ et al., 2023)	GRU+self-attention	SL	Server machine

BF: Bayes filter; SSL: semi-supervised learning; UL: unsupervised learning

5 AD for network traffic monitoring

In the 5G communication environment, the increasing volume of traffic data is vulnerable to malicious activities which aim at stealing private user information or disrupting communication links. AD models or intrusion detection systems are widely employed to identify such threats, such as spikes in traffic, unusual access patterns, and potential distributed denial of service (DDoS) attacks or other

intrusion attacks. This section introduces the applications of AD models for malicious network traffic monitoring.

5.1 Application description

The backbone network comprises hundreds of interconnected nodes, and various illegitimate activities such as DDoS attacks, port scans, and worms can cause traffic anomalies. These anomalies often manifest as fluctuations in the original-destination (OD) traffic volume between nodes. However, monitoring traffic at OD level is resource-intensive and difficult to analyze due to the scale of the network. On the contrary, sudden increases or decreases in OD traffic can directly impact link traffic volume. Since link-level traffic monitoring is more scalable and manageable, it becomes more feasible as the indicator for AD. The task of AD for network traffic focuses on identifying malicious traffic patterns at the link level to enhance network security and performance.

5.2 Traffic feature extraction

Traffic volume is crucial for analyzing network operational status. Various feature extraction methods are used to obtain informative representations of traffic volume, and can be applied to downstream tasks, such as AD. In this context, deep neural networks, such as RNNs and CNNs, are used. HCRNNIDS (Khan, 2021) exemplifies this approach by merging CNN and RNN, where the CNN component extracts spatial features and the RNN component captures temporal dependencies. This synergy enhances the ability of intrusion detection system to classify and predict cyber attacks by leveraging the temporal AD capabilities of RNNs, making it highly effective in detecting time-series anomalies in network traffic. Similarly, DCNNBiLSTM (Hnamte and Hussain, 2023) combines CNN for spatial feature extraction, bidirectional LSTM (Bi-LSTM) for sequence prediction, and deep neural networks (DNN) for error optimization, leveraging the strengths of each component. CNN-LSTM (Halbouni et al., 2022) uses CNN to extract spatial features and LSTM to capture temporal dependencies, creating a robust hybrid model for intrusion detection. The inclusion of batch normalization and dropout layers further enhances its performance and robustness, effectively detecting temporal anomalies within the network traffic. CBR-CNN (Chouhan et al., 2019) exemplifies this approach by combining channel boosting and residual learning to enhance feature representation. It employs stacked AEs to model normal traffic and uses a multipath residual learning-based CNN to detect intrusions at various levels of granularity. CANET (Ren KY et al., 2023) integrates CNN with attention mechanisms in hierarchical certificate authority (CA) blocks, focusing on spatiotemporal feature extraction. By addressing class imbalance using equalization loss v2 (a gradient-guided reweighting loss), the extracted features are particularly suitable for large-scale intrusion detection downstream tasks.

5.3 Intrusion detection techniques

A variety of intrusion detection systems, leveraging diverse deep learning technologies, are deployed for network security monitoring. RENOIR (Andresini et al., 2021) combines AEs with triplet networks to create embeddings that effectively distinguish between the normal and attack traffic by training AEs on historical data, and using a triplet network to ensure accurate classification of network flows based on their reconstructions. Similarly, cognitive memory-guided autoencoder (CMAE) (Lu et al., 2022) enhances traditional AEs with a memory module to store normal feature patterns, making it robust to imbalanced samples, and leveraging reconstruction error, feature reconstruction loss, and feature sparsity loss to improve detection capabilities, particularly for unknown attacks. Adversarially regularized convolutional autoencoder for unsupervised network anomaly detection (ARCADE) (Lunardi et al., 2023) uses a convolutional AE to profile normal network traffic, and employs adversarial training to reduce the AE's ability to reconstruct out-of-distribution flows, thereby enhancing its AD capabilities. Multi-scale residual classifier (MSRC) (Duan et al., 2023) uses wavelet transforms for multi-scale analysis and a stacked AE for learning data distributions, effectively capturing anomalies at various scales and complexities. Meanwhile, asymmetric parallel AE (APAE) (Basati and Faghih, 2023) features a dual-encoder design that captures both local and long-range features using convolutional layers and attention mechanisms. Its lightweight architecture and robust decoder make it suitable for real-time intrusion detection and generalization even with

limited training data. D-PACK (Hwang et al., 2020) addresses the vulnerabilities exploited by large-scale cyber attacks. This system integrates CNN with an unsupervised deep learning model, to efficiently profile traffic patterns and filter out abnormalities. What sets D-PACK apart is its method of inspecting just the first few bytes of the initial packets in each flow, enabling early detection of malicious activities. This strategic focus allows for prompt intervention, potentially stopping attacks before they escalate. Training LLM models for intrusion detection leverages sophisticated approaches to optimize performance and resource efficiency. Specifically, models such as bidirectional encoder representations from Transformers (BERT) are adapted for identifying malicious network flows. The training process begins with fine-tuning the pretrained BERT model on specialized datasets of network traffic. Enhancements such as temporal feature extraction and the integration of flow- and packet-level data within heterogeneous graph structures further refine the model's ability to provide detailed, contextually rich insights (Farrukh et al., 2024).

5.4 Network KPI analysis

Monitoring KPIs, such as hit ratio and delay in content delivery networks (CDNs) (Dai et al., 2022), is crucial as performance degradation can be reflected in these KPIs. The LSTM-GAN-G model (Huang et al., 2022) integrates GAN with LSTM cells, enabling it to effectively capture the temporal dynamics of cellular KPI time series. For AD, the model uses the trained generator to reconstruct testing samples, and identifies anomalies by comparing these with the actual data. The SGmVRNN model (Dai et al., 2022) uses a variational RNN architecture that combines a mixture of Gaussian distributions for latent variables, capturing the intricate time-series data of KPIs in CDN effectively. It also introduces a switching mechanism to better represent the variable structures and dynamics within the KPIs.

6 AD for networking system log analysis

The log data encompass a substantial volume, extending across diverse system layers and formats. Therefore, to adeptly identify log anomalies, it is crucial to use efficient methodologies grounded in machine learning and deep learning. AD methods in system logs can be effectively categorized based on the specific-applied scenarios.

6.1 Application description

System logs record critical states and changes within a system, enabling the analysis of anomalous events. In log AD, the process begins with the collection of system logs, each comprising a standardized template and system-specific variable. Subsequently, logs are parsed to differentiate between the static components (e.g., predefined text or templates) and the dynamic components (e.g., thread names or job ID). The logs are then partitioned into several groups based on timestamps and log identifiers. Feature extraction techniques are applied to capture relevant characteristics of the data. Finally, these features are used by anomaly detectors to identify instances of anomalous logs.

6.2 Log dimension reduction

In scenarios dealing with high-dimensional data where capturing essential variance is important, methods such as principal component analysis (PCA) (Xu W et al., 2009) come into play. PCA reduces dimensionality to identify anomalous patterns in log vectors, which are useful in recognizing anomalies within complex log structures. For real-time monitoring and quick AD for large scale log data, clustering approaches are highly effective. These methods evaluate the proximity of new logs to cluster centers (Lin QW et al., 2016), enabling rapid identification of anomalies. This is particularly beneficial for systems that require constant monitoring and immediate response to unusual activities.

6.3 Log semantic representation

In applications requiring sequence prediction and detailed log pattern analysis, DeepLog (Du M et al., 2017) transforms raw log data into structured sequences for training an LSTM neural network. This enables the model to learn normal patterns of log sequences, and predict subsequent logs, detecting anomalies by assessing deviations from expected templates. LogAnomaly (Meng et al., 2019) incorporates semantic analysis with Template2Vec, converting log templates into semantic vectors processed

by a Bi-LSTM with an attention mechanism. This method is particularly effective in scenarios where understanding the context and semantics of log data is critical for accurate AD. Transformer-based models (Balasubramanian et al., 2023) use pre-trained embedding to represent log entries in a high-dimensional space, capturing semantic information beyond simple token-level representations.

6.4 Log AD

For robust AD in noisy environments, approaches such as PLELog (Yang L et al., 2021) and LogRobust (Zhang X et al., 2019b) are highly effective. PLELog uses a gated recurrent unit (GRU) with an attention mechanism to classify log sequences, focusing on parts of the log data that indicate anomalies. LogRobust, designed to handle noisy or unstable log data, starts by converting log data into semantic vectors, and uses an LSTM model for AD. This method ensures reliable detection even under less ideal conditions, making it suitable for applications with significant log data variability and noise. LLMs have demonstrated a profound capability for understanding and processing complex data sequences. By leveraging self-attention mechanisms, these models can understand log content, conduct anomaly classification among parallel file system logs (Egersdoerfer et al., 2023), and localize configuration errors (Shan et al., 2024).

By aligning each method with its optimal application scenario, both machine learning and deep learning techniques enhance the effectiveness and accuracy of detecting anomalies in system logs, addressing diverse challenges with tailored approaches.

6.5 Failure prediction and diagnosis

For failure prediction and diagnosis, methods such as the decision tree approach (Chen M et al., 2004), which categorizes logs based on predefined attributes and branching decisions, are particularly useful. This approach can highlight whether logs are normal or abnormal, aiding in predicting and diagnosing system failures. Similarly, the SVM (Liang YL et al., 2007) approach constructs a hyperplane to separate normal and abnormal activities, making it ideal for scenarios where precise failure prediction is critical.

7 AD for cloud and edge service provisioning

In this section, we examine current studies regarding the detection of anomalies in cloud and edge service provisioning, using a variety of approaches, including statistical analysis, machine learning, deep learning, and hybrid methods.

7.1 Application description

The cloud system includes various connected resources through the distributed network, and can dynamically allocate resources based on user demand. However, cloud environment is susceptible to various attacks, including those targeting virtual machines, unauthorized access to interfaces, and vulnerabilities in APIs or Internet protocols. To detect attacks within cloud infrastructure, data packets and system logs are collected and analyzed for detecting anomalies. On the contrary, KPIs can be gathered from local devices for edge-level AD, which enhances efficiency and enables the swift identification of anomalous events with minimal delay.

7.2 Service system maintenance

Statistical analysis empowers cloud providers to detect anomalies within their cloud environments, facilitating prompt interventions and minimizing disruptions. For instance, SCHEDA, introduced by Guigou et al. (2019), integrates three algorithms to compute Euclidean anomaly scores for detecting anomalies in cloud data. This system excels in real-time anomaly scoring without the need for data buffering, significantly enhancing the security and efficiency of cloud services. Similarly, the approach proposed by Khatibzadeh et al. (2019) leverages the catastrophe theory and entropy to identify anomalies in cloud traffic. These statistical methods are particularly useful for real-time monitoring and ensuring system stability.

7.3 Cloud log analysis

Deep learning algorithms significantly enhance the detection of anomalies in cloud computing by capturing intricate patterns in log data. Zhu et al. (2020) introduced the LogNL technique, which uses LSTM and natural language processing (NLP) techniques for log AD. This approach is particularly

effective for analyzing cloud platform logs, and has demonstrated high accuracy and F1 score in evaluations. Similarly, Khalaf et al. (2022) developed a graph-based layer-driven learning (GLLD) method for AD in cloud networks. This method excels in detecting anomalies in cloud traffic, though it faces challenges in identifying various types of anomalies.

7.4 Cloud-based computing security

Machine learning algorithms offer sophisticated solutions for enhancing the security of cloud-based computing. Parameswarappa et al. (2023) applied machine learning methods to propose a novel firewall strategy aimed at enhancing cloud security. Their approach combines historical and current decisions to form the "most frequent decision" technique, achieving high accuracy and F1 score on the UNSW-NB15 dataset, despite high computational costs. Similarly, Jiang J et al. (2023) proposed the adaptive ensemble random fuzzy (AERF) algorithm for cloud network AD. AERF effectively addresses disruptions from atypical data distributions, showcasing its utility in maintaining robust cloud security.

7.5 Cross-environment adaptability

Zhang X et al. (2019a) developed the ATAD method, which integrates active and transfer learning techniques to identify abnormalities in an unlabeled dataset. This method demonstrates high F1 score in cross-dataset evaluations. Yu et al. (2020) introduced an AD framework based on the behavioral characteristics of time-series networks, using the DBN-BiGRU algorithm model. Lalotra et al. (2022) presented the intelligent real-time AD system (iReTADS), which employs a data summarization strategy to reduce network traffic and enhance security in real time. These hybrid methods are particularly beneficial for applications requiring adaptability and robust AD across diverse datasets and environments.

7.6 Real-time monitoring

For real-time monitoring and quick AD, methods such as those proposed by Girish and Rao (2023) and Song YJ et al. (2023) are highly effective. Girish and Rao (2023) used stacked and bidirectional LSTM models to detect cloud abnormalities from time-series textual records, achieving high detection accuracy in the OpenStack environment. Song YJ et al. (2023) employed parallel graph neural networks (GNNs) and multi-head self-attention mechanisms to enhance detection accuracy in multivariate time-series, making it particularly suitable for real-time AD.

7.7 Distributed hierarchical edge computing (HEC)

In distributed HEC systems, AD is crucial for reducing latency and ensuring high accuracy across different computational layers, including edge servers and cloud platforms. Schneible and Lu (2017) presented an AD method using AEs on distributed edge devices, reducing the need for data transmission to a central server and enabling more localized processing. Peng et al. (2019) presented a multi-source AD system based on HEC, focusing on industrial applications such as mining. It improves detection accuracy and reduces latency. Ngo et al. (2020) demonstrated that using their proposed method in distributed HEC systems significantly reduces detection delay by approximately 71% while maintaining accuracy, making it highly suitable for real-time applications in edge computing environments. Ngo et al. (2021) introduced an adaptive model selection scheme for AD in IoT environments with HEC. It optimizes both accuracy and delay through a reinforcement learning-based policy network.

7.8 Intelligent video surveillance

Intelligent video surveillance systems use edge and cloud computing to enable real-time AD. In these systems, visual sensors collect raw data, which are then preprocessed and subjected to feature extraction (Georgiou et al., 2020). The processed data are analyzed by machine learning models to detect abnormal behavior. While cloud computing is traditionally used for data processing and storage, it often suffers from longer response time due to bandwidth limitations and network latency. For delay-sensitive applications such as video surveillance, edge computing provides a better solution (Srivastava and Singh, 2016). By performing preprocessing and initial analysis locally on edge devices, edge computing reduces data transmission to the cloud and minimizes latency. This hybrid approach improves

real-time AD and response, making it suitable for public safety, industrial security, and critical infrastructure monitoring (Shi WB et al., 2016).

Recent advances, such as language-based video anomaly detection (LAVAD) (Zanella et al., 2024) and AnomalyRuler (Yang YC et al., 2024), propose innovative training-free paradigms leveraging pretrained LLMs and vision language models (VLMs) for intelligent video surveillance. LAVAD uses VLM-based caption for frame-level descriptions, applying prompts for temporal aggregation and anomaly scoring, thus eliminating extensive training. AnomalyRuler employs few-shot learning to induce rules from normal patterns, using deductive reasoning for AD. These methodologies harness the inherent reasoning and contextual capabilities of LLMs, providing robust, scalable, and rapidly adaptive solutions essential for real-time surveillance in dynamic environments.

These categorized approaches highlight the versatility and robustness of various methods in addressing different challenges in cloud and edge service AD. By aligning each method with its optimal application scenario, these techniques enhance the effectiveness and accuracy of detecting anomalies in diverse cloud environments.

8 AD for IoT security

In the realm of IoT security, cyber-physical systems (CPSs), such as water treatment systems and intelligent power control systems, expose critical devices to network vulnerabilities. These systems can be compromised, and even the slightest deviation from expected sensor readings or control commands can indicate critical malfunctions, potentially leading to cascading failures. Therefore, AD models are crucial in IoT security, enabling swift corrective actions when alarms are raised to protect these vulnerable systems.

8.1 Application description

CPSs comprise physical systems, control systems, and supervisory control and data acquisition (SCADA) systems. Various sensors are deployed to monitor the states of physical infrastructures, such as power plants and water treatment facilities. Based on sensor feedback, controllers issue commands to manage critical devices within these physical systems. The collected data are then transmitted via communication networks to SCADA for monitoring and system control. However, sensors and actuators may fail due to attacks or insufficient maintenance, and attackers can manipulate controllers, communication networks, and SCADA systems. Therefore, the deployment of AD mechanisms is essential for safeguarding CPS against potential threats.

8.2 CPS attack detection

Various AD models leverage the rich representational capacity of deep learning architectures to capture informative features from time-series data in CPSs. For instance, Li D et al. (2019a) proposed a novel approach employing a GAN coupled with LSTM-RNNs within a composite framework. This framework is designed to model the normal distribution of CPS measurements, with the discriminator tasked to efficiently identify anomalies. In addition to LSTM, a feedforward network (Feng and Tian, 2021) is employed to characterize practical CPSs. Subsequently, the identified system is simulated to predict sensor readings, with deviations between observed and predicted values flagged as anomalies.

An AE-based method for detecting IoT-based attacks was introduced, involving the construction of Mirai and BASHLITE botnets to target IoT networks in a laboratory setting (Meidan et al., 2018). The method included launching several types of attacks such as Scan, ACK, SYN, and UDP flooding. An AE trained on benign network traffic was used, with an alert being triggered when the reconstruction error exceeded a predefined threshold. Furthermore, a regularized AE was presented to segregate known attacks and normal data into separate regions within the latent space (Vu et al., 2022). It was posited that the latent space representations of unknown attacks would be close to those of known attacks, and this characteristic was used to develop a classifier for detection of IoT-based attacks.

Recent methods enhance CPS security by leveraging LLMs with embedding models (Diaf et al., 2024) and retrieval-augmented generation (RAG) to map attack patterns (Webb et al., 2024). Training involves encapsulating attack data into vector formats and using RAG for structured mappings that are evaluated against standard models for accuracy. Another approach integrates fine-tuned GPTs and BERT with LSTM networks. GPT predicts

network traffic, BERT evaluates predictions, and LSTM identifies malicious packets using datasets such as CICIoT2023. A feedback loop refines LLM accuracy, ensuring proactive and effective detection in dynamic IoT environments.

8.3 Data privacy

Data privacy is increasingly becoming a focal point in the field of CPS security, as the sensor and actuator data transmitted through industrial networks contain the operating status and control instructions of critical facilities. If these data are intercepted and used during global model training, it could result in significant security breaches. Federated learning (FL) has been employed to synchronize local models across gateways in the CPS or IoT environment, with a focus on maintaining the data privacy of individual gateways. The approach proposed by Wang N et al. (2022) uses contrastive learning (CL) techniques to improve the traffic data representation and has been evaluated using the NSK-KDD dataset. Additionally, a privacy-preserving federated PCA framework has been devised, combining federated learning with PCA to extract traffic features from various IoT devices without compromising data exchange (Nguyen et al., 2023). Furthermore, a method has been proposed that uses deep neural networks for each edge IoT device within a federated framework to aggregate local model parameters. A global DNN is also designed within this framework to detect zero-day attacks in IoT devices (Popoola et al., 2022).

8.4 Interpretable models

Although deep models provide informative and accurate results, they are often considered "black-box" models, which means that these models are unreliable for network operators. Despite advancements in model accuracy, building trust with network analysts remains crucial. Interpretative models usually generate human-understandable rules to explain anomalies, or apply existing interpretation strategies in deep-learning domain to enhance the understandability for the AD models. To this end, several studies have been conducted to provide insights into the interpretability of AD methods. For instances, Li RY et al. (2024) proposed a rule extraction strategy for unsupervised AD models. They developed an interior clustering tree to decompose data into sub-distribution based on model predictions and generate the decomposition distribution rules. Subsequently, they formulated a compositional boundary exploration algorithm to infer the decision boundaries and derive boundary inference rules. The decomposition distribution rules and boundary inference rules were further used to facilitate the interpretation of predictions from unsupervised AD models. Han et al. (2021) devised an interpreter and distiller to explain anomalies for analysts. They framed an optimization problem to generate "reference" for interpreting anomalies across various data formats: tabular data in the Kitsune system (Mirsky et al., 2018), time-series data in the DeepLog system (Du M et al., 2017), and graph data in the graph learning with global view (GLGV) system (Bowman et al., 2020). Additionally, they employed two finite-state machines to create the distiller, which can enable security-domain analysts to interact with detected anomalies effectively. Training LLMs for interpretable IoT security involves leveraging pretrained models such as GPT-4 for their vast knowledge base. Houssel et al. (2024) fine-tuned pretrained model GPT-4 with domain-specific network data. By using RAG, the model aligns real-world threat patterns, facilitating the generation of human-readable explanations. This process ensures that LLMs can contextualize threats and provide detailed interpretive insights.

9 Challenges and future directions

9.1 Challenges

Although various practical methods in the field of pattern regression are adapted for AD tasks, there are specific challenges in AD that must be considered when designing frameworks using methods from other machine learning domains such as computer vision (CV) and NLP. These challenges include the following:

1. High dimensionality. Communication systems generate high-dimensional data from multiple sources, such as network logs, performance metrics, and user activity records. Analyzing these complex data to detect anomalies requires techniques that can effectively handle the increased dimensionality while maintaining computational efficiency.

2. Nonstationary. Communication systems often exhibit nonstationary, such as seasonal patterns and long-term trends, in user behavior, network traffic, and resource utilization. Distinguishing between the genuine anomalies and the expected patterns becomes challenging due to the presence of these seasonal and trend components in the data.

3. Class imbalance. Anomalies in communication systems, such as network failures or network attacks, are relatively rare compared to normal system behavior. This class imbalance can bias AD algorithms toward the majority class, reducing their sensitivity in detecting actual anomalies.

4. Label sparsity. Obtaining labeled data for AD in communication systems is often difficult or impractical. Labeling anomalies requires domain expertise and can be time-consuming, especially given the vast amount of data generated by these systems. This scarcity of labeled data poses challenges for training supervised AD models.

5. Concept drift. Internet-based services in communication systems often operate in dynamic environments where the underlying patterns and behaviors change over time. This concept drift necessitates the adaptation of AD models to account for evolving data patterns and ensure detection accuracy.

9.2 Future directions

9.2.1 Robustness in AD

1. Enhancing robustness to address concept drift and class imbalance in communication systems

Future research should focus on developing AD models with enhanced cross-domain robustness and adopting robust learning strategies to tackle concept drift and class imbalance in communication systems. Transfer learning techniques, such as adversarial domain adaptation (Shi YJ et al., 2022; Ozyurt et al., 2023) and MMD minimization (Tzeng et al., 2014; Venkateswara et al., 2017; Wang YX et al., 2022), can align feature distributions across different communication environments, enabling models to learn transferable and domain-invariant representations. GNNs (Dhadhania et al., 2024; Jin et al., 2024a) can capture complex relationships and dependencies in communication networks, facilitating the generalization of AD models across various network topologies and configurations. Oversampling techniques, such as SMOTE or ADASYN, and ensemble learning methods, such as bagging or boosting, can help balance the distribution of anomalous and normal samples and improve the overall performance and stability of the AD system. Additionally, unsupervised AD approaches, such as isolation forests or AEs, can reduce the dependency on labeled data by learning the inherent patterns and structures within the communication data.

2. Developing AD systems resistant to adversarial attacks

Future research on enhancing the resilience of AD systems against adversarial attacks may focus on integrating robust adversarial training regimes and ensemble learning methods (Nawaz et al., 2024). GANs (Xia et al., 2022; Lim et al., 2024; Lüer and Bohm, 2024) can be used to create robust AD models by training them on simulated adversarial scenarios, reducing false negatives and false positives.

3. Robust AD in dynamic and evolving communication networks

To ensure robust AD in dynamic communication networks, future research may concentrate on creating self-adaptive systems using online learning algorithms (Segerholm, 2023) and continual learning approaches such as elastic weight consolidation (EWC) (Ao and Fayek, 2023). Meta-learning algorithms, such as model-agnostic meta-learning (MAML) (Griffiths et al., 2019), can significantly enhance the adaptability of AD models in dynamic networks.

9.2.2 Explainability in AD

1. Enhancing explainability to address nonstationary and concept drift

Incorporating causal reasoning, temporal causal modeling, and domain knowledge into AD models can help address the challenges of nonstationary and concept drift in communication networks. By analyzing the relationships between the communication variables and their temporal dependencies, causal reasoning approaches can distinguish between the seasonal or trend-related changes and the true anomalies, providing a more comprehensive understanding of the system's behavior. Integrating domain knowledge through expert systems, graph-based representations, or modular neural networks can generate explanations that align with the unique challenges and terminology of communication networks. These techniques improve the interpretability

of AD models, and ensure that their predictions remain robust in the face of evolving network conditions and shifting anomaly patterns.

2. Standardization of explainability metrics

Future research could focus on developing a comprehensive framework for evaluating the explainability of AD models in communication systems. This includes defining quantitative measures that assess the clarity, conciseness, and relevance of explanations, considering the specific challenges and requirements of communication networks. Establishing standardized benchmarks and datasets will enable consistent evaluation and comparison of explainability across different models and techniques.

3. Tailoring explanations to different stakeholders

Future research should prioritize the development of user-centric explainability frameworks that adapt to diverse needs and expertise of various stakeholders. This involves designing explanation generation systems that consider user profiles, roles, and preferences to deliver personalized explanations for detected anomalies. The goal is to enhance user trust and decision-making by providing the most relevant and actionable insights for each stakeholder.

9.2.3 LLM applications in AD

1. Leveraging LLMs to address AD challenges in communication systems

LLMs, particularly time-series LLMs, offer promising solutions to address the challenges faced by AD in communication systems. By harnessing the instruction tuning paradigm and open domain knowledge of LLMs, researchers can generate new data and enable active learning of evolving communication patterns, tackling the issue of concept drift. Continual learning techniques (Gupta et al., 2021; Wu et al., 2024) and change point detection methods (Aminikhanghahi and Cook, 2017) can be employed to update LLMs incrementally and identify significant shifts in data patterns, ensuring that the models remain aligned with the latest trends. LLMs' few-shot learning capabilities and unsupervised representation learning can mitigate the impact of class imbalance and limited labeled data by effectively learning from a small number of anomalous samples and capturing temporal patterns within the data. Time-series LLMs, such as UniTS (Gao SH et al., 2024), excel in contextual comprehension due to their specialized pretraining, enabling them to differentiate between true anomalies and innocuous deviations caused by seasonal or trend-related factors. Furthermore, the powerful encoding capabilities of LLMs can be used for representation learning, extracting low-dimensional semantic features from high-dimensional communication time-series data, making them more tractable for AD tasks.

2. Training tailored LLMs for time series

To advance LLM performance in time-series AD, future research should focus on how to train time-series specific LLMs. There are three main approaches for creating such tailored LLMs: zero-shot, fine-tuning, and training from scratch. Zero-shot strategies use pretrained LLMs without task-specific data, leveraging the models' broad generalization abilities achieved through diverse pretraining. Prompt-based LLMs, such as those employing task-specific prompts for time-series tasks, exemplify this approach (Xue and Salim, 2024). Fine-tuning, a commonly used technique, involves adjusting a pretrained LLM on domain-specific data for enhanced task performance (Darban et al., 2024; Fang et al., 2024). Fine-tuning modifies a subset of parameters, retaining core knowledge while improving domain accuracy. An advanced fine-tuning method incorporates a multi-head attention mechanism to convert time-series data into text format while preserving the original model parameters (Jin et al., 2024b). Training from scratch involves developing a dedicated large model optimized for time-series analysis from the ground up (Gao SH et al., 2024). This method emphasizes learning temporal dependencies and sequence patterns, improving performance for time-series tasks, but requiring extensive computational power and a substantial dataset.

3. Addressing resource intensity in time-series LLMs for AD

The resource intensity of training and deploying time-series LLMs for AD is a significant challenge. When models such as UniTS (Gao SH et al., 2024) show superior performance, their computational costs may outweigh the benefits. Future directions should focus on developing efficient model architectures and applying compression techniques such as quantization (Lin Z et al., 2024), pruning (Lin Z et al., 2024), and knowledge distillation (Wang L and Yoon, 2022). Balancing performance with cost efficiency, scalability, and maintainability is crucial

for the practical value and long-term operational sustainability of LLMs in real-world AD applications.

10 Conclusions

This survey has comprehensively explored the landscape of AD in AIOps, covering applications in network traffic monitoring, networking system log analysis, cloud and edge service provisioning, and IoT security. By examining diverse datasets and methods across these domains, we have highlighted the challenges and opportunities in each area. As the field continues to evolve, the need for explainable, robust, and LLM-driven AD becomes increasingly apparent. To advance state-of-the-art technology, researchers should focus on developing interpretable models that can effectively handle complex, high-dimensional data while maintaining resilience against adversarial attacks. Furthermore, the integration of LLMs in AD pipelines holds promise for enhancing the accuracy and efficiency of these systems. By addressing these key challenges and leveraging the insights gained from this survey, the AIOps community can pave the way for more reliable, secure, and intelligent AD solutions.

Contributors

All the authors contributed to the outline of the review. Jiayi GUI, Zhongnan MA, Hao ZHOU, Yan SU, and Miaoru ZHANG drafted the paper. Jiayi GUI, Zhongnan MA, and Ke YU organized the paper. Jiayi GUI, Zhongnan MA, Hao ZHOU, Yan SU, and Miaoru ZHANG revised the paper, while Jiayi GUI, Zhongnan MA, Hao ZHOU, Ke YU, and Xiaofei WU reviewed and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

References

Aboubacar A, El Machkouri M, 2020. Recursive kernel density estimation for time series. *IEEE Trans Inform Theory*, 66(10):6378-6388. https://doi.org/10.1109/TIT.2020.3014797

Abrams L, 2020. CenturyLink routing issue led to outages on Hulu, Steam, Discord, more. https://www.bleepingcomputer.com/news/technology/centurylink-routing-issue-led-to-outages-on-hulu-steam-discord-more [Accessed on Apr. 1, 2024].

Ahmed CM, Palleti VR, Mathur AP, 2017. WADI: a water distribution testbed for research in the design of secure cyber physical systems. Proc 3rd Int Workshop on Cyber-Physical Systems for Smart Water Networks, p.25-28. https://doi.org/10.1145/3055366.3055375

Ahmed M, Mahmood AN, Hu JK, 2016. A survey of network anomaly detection techniques. *J Netw Comput Appl*, 60:19-31. https://doi.org/10.1016/j.jnca.2015.11.016

Aldribi A, Traore I, Moa B, 2018. Data sources and datasets for cloud intrusion detection modeling and evaluation. In: Mishra BSP, Das H, Dehuri S, et al. (Eds.), Cloud Computing for Optimization: Foundations, Applications, and Challenges. Springer, Cham, p.333-366. https://doi.org/10.1007/978-3-319-73676-1_13

Al-Hawawreh M, Sitnikova E, Aboutorab N, 2022. X-IIoTID: a connectivity-agnostic and device-agnostic intrusion data set for Industrial Internet of Things. *IEEE Int Things J*, 9(5):3962-3977. https://doi.org/10.1109/JIOT.2021.3102056

Alsaedi A, Moustafa N, Tari Z, et al., 2020. TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems. *IEEE Access*, 8:165130-165150. https://doi.org/10.1109/ACCESS.2020.3022862

Amer M, Goldstein M, Abdennadher S, 2013. Enhancing one-class support vector machines for unsupervised anomaly detection. Proc ACM SIGKDD Workshop on Outlier Detection and Description, p.8-15. https://doi.org/10.1145/2500853.2500857

Aminikhanghahi S, Cook DJ, 2017. A survey of methods for time series change point detection. *Knowl Inform Syst*, 51(2):339-367. https://doi.org/10.1007/s10115-016-0987-z

Andresini G, Appice A, Malerba D, 2021. Autoencoder-based deep metric learning for network intrusion detection. *Inform Sci*, 569:706-727. https://doi.org/10.1016/j.ins.2021.05.016

Ao SI, Fayek H, 2023. Continual deep learning for time series modeling. *Sensors*, 23(16):7167. https://doi.org/10.3390/s23167167

Audibert J, Michiardi P, Guyard F, et al., 2020. USAD: unsupervised anomaly detection on multivariate time series. Proc 26th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining, p.3395-3404. https://doi.org/10.1145/3394486.3403392

Bai JZ, Bai S, Chu YF, et al., 2023. QWEN technical report. https://doi.org/10.48550/arXiv.2309.16609

Balasubramanian P, Seby J, Kostakos P, 2023. Transformer-based LLMs in cybersecurity: an in-depth study on log anomaly detection and conversational defense mechanisms. IEEE Int Conf on Big Data, p.3590-3599. https://doi.org/10.1109/BigData59044.2023.10386976

Bansod SD, Nandedkar AV, 2020. Crowd anomaly detection and localization using histogram of magnitude and momentum. *Vis Comput*, 36(3):609-620. https://doi.org/10.1007/s00371-019-01647-0

Basati A, Faghih MM, 2023. APAE: an IoT intrusion detection system using asymmetric parallel auto-encoder. *Neur Comput Appl*, 35(7):4813-4833. https://doi.org/10.1007/s00521-021-06011-9

Bochner S, Chandrasekharan K, 1949. Fourier Transforms. Princeton University Press, Princeton, USA.

Bowman B, Laprade C, Ji YD, et al., 2020. Detecting lateral movement in enterprise computer networks with unsupervised graph AI. 23rd Int Symp on Research in Attacks, Intrusions and Defenses, p.257-268.

Brown TB, Mann B, Ryder N, et al., 2020. Language models are few-shot learners. Proc 34th Int Conf on Neural Information Processing Systems, p.1877-1901.

Chand N, Mishra P, Krishna CR, et al., 2016. A comparative analysis of SVM and its stacking with other classification algorithm for intrusion detection. Int Conf on Advances in Computing, Communication, & Automation, p.1-6. https://doi.org/10.1109/ICACCA.2016.7578859

Chandola V, Banerjee A, Kumar V, 2009. Anomaly detection: a survey. *ACM Comput Surv*, 41(3):15. https://doi.org/10.1145/1541880.1541882

Chang C, Peng WC, Chen TF, 2024. LLM4TS: two-stage fine-tuning for time-series forecasting with pre-trained LLMs. https://doi.org/10.48550/arXiv.2308.08469

Chen M, Zheng AX, Lloyd J, et al., 2004. Failure diagnosis using decision trees. Int Conf on Autonomic Computing, p.36-43. https://doi.org/10.1109/ICAC.2004.1301345

Chen ZH, Zheng LN, Lu C, et al., 2023. ChatGPT informed graph neural network for stock movement prediction. https://arxiv.org/abs/2306.03763

Chen ZM, Yeo CK, Lee BS, et al., 2018. Autoencoder-based network anomaly detection. Wireless Telecommunications Symp, p.1-5. https://doi.org/10.1109/WTS.2018.8363930

Choi K, Yi JH, Park C, et al., 2021. Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. *IEEE Access*, 9:120043-120065. https://doi.org/10.1109/ACCESS.2021.3107975

Chouhan N, Khan A, Khan HUR, 2019. Network anomaly detection using channel boosted and residual learning based deep convolutional neural network. *Appl Soft Comput*, 83:105612. https://doi.org/10.1016/j.asoc.2019.105612

Cook AA, Mook AA G, Fan Z, 2020. Anomaly detection for IoT time-series data: a survey. *IEEE Int Things J*, 7(7):6481-6494. https://doi.org/10.1109/jiot.2019.2958185

Cup K, 2007. KDD Cup 1999 Data. https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [Accessed on Apr. 1, 2024].

Dai L, Chen WC, Liu YW, et al., 2022. Switching Gaussian mixture variational RNN for anomaly detection of diverse CDN websites. IEEE Conf on Computer Communications, p.300-309. https://doi.org/10.1109/INFOCOM48880.2022.9796836

Dang WX, Zhou BY, Wei LW, et al., 2021. TS-Bert: time series anomaly detection via pre-training model Bert. 21st Int Conf on Computational Science, p.209-223. https://doi.org/10.1007/978-3-030-77964-1_17

Dang YN, Lin QW, Huang P, 2019. AIOps: real-world challenges and research innovations. IEEE/ACM 41st Int Conf on Software Engineering: Companion Proceedings, p.4-5. https://doi.org/10.1109/ICSE-Companion.2019.00023

Darban ZZ, Yang YY, Webb GI, et al., 2024. DACAD: domain adaptation contrastive learning for anomaly detection in multivariate time series. https://doi.org/10.48550/arXiv.2404.11269

DataSetsAI, 2020. Water Pumps. https://datasets.ai/datasets/water-pumps [Accessed on Aug. 17, 2025].

Dhadhania A, Bhatia J, Mehta R, et al., 2024. Unleashing the power of SDN and GNN for network anomaly detection: state-of-the-art, challenges, and future directions. *Secur Priv*, 7(1):e337. https://doi.org/10.1002/spy2.337

Diaf A, Korba AA, Karabadji NE, et al., 2024. Beyond detection: leveraging large language models for cyber attack prediction in IoT networks. 20th Int Conf on Distributed Computing in Smart Systems and the Internet of Things, p.117-123. https://doi.org/10.1109/DCOSS-IoT61029.2024.00026

Du M, Li FF, Zheng GN, et al., 2017. DeepLog: anomaly detection and diagnosis from system logs through deep learning. Proc ACM SIGSAC Conf on Computer and Communications Security, p.1285-1298. https://doi.org/10.1145/3133956.3134015

Du ZX, Qian YJ, Liu X, et al., 2022. GLM: general language model pretraining with autoregressive blank infilling. https://doi.org/10.48550/arXiv.2103.10360

Duan XY, Fu Y, Wang K, 2023. Network traffic anomaly detection method based on multi-scale residual classifier. *Comput Commun*, 198:206-216. https://doi.org/10.1016/j.comcom.2022.10.024

Egersdoerfer C, Zhang D, Dai D, 2023. Early exploration of using ChatGPT for log-based anomaly detection on parallel file systems logs. Proc 32nd Int Symp on High-Performance Parallel and Distributed Computing, p.315-316. https://doi.org/10.1145/3588195.3595943

Ekambaram V, Jati A, Dayama P, et al., 2024. Tiny time mixers (TTMs): fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. https://doi.org/10.48550/arXiv.2401.03955

Esling P, Agon C, 2012. Time-series data mining. *ACM Comput Surv*, 45(1):12. https://doi.org/10.1145/2379776.2379788

Fang YQ, Yap PT, Lin WL, et al., 2024. Source-free unsupervised domain adaptation: a survey. *Neur Netw*, 174:106230. https://doi.org/10.1016/j.neunet.2024.106230

Farrukh YA, Wali S, Khan I, et al., 2024. XG-NID: dual-modality network intrusion detection using a heterogeneous graph neural network and large language model. https://doi.org/10.48550/arXiv.2408.16021

Feng C, Tian PW, 2021. Time series anomaly detection for cyber-physical systems via neural system identification and Bayesian filtering. Proc 27th ACM SIGKDD Conf on Knowledge Discovery & Data Mining, p.2858-2867. https://doi.org/10.1145/3447548.3467137

Ferrag MA, Friha O, Hamouda D, et al., 2022. Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. *IEEE Access*, 10:40281-40306. https://doi.org/10.1109/ACCESS.2022.3165809

Gao S, Huang YF, Zhang S, et al., 2020. Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample

generation. *J Hydrol*, 589:125188. https://doi.org/10.1016/j.jhydrol.2020.125188

Gao SH, Koker T, Queen O, et al., 2024. UniTS: a unified multi-task time series model. https://doi.org/10.48550/arXiv.2403.00131

Georgiou T, Liu Y, Chen W, et al., 2020. A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision. *Int J Multimed Inform Retr*, 9(3):135-170. https://doi.org/10.1007/s13735-019-00183-w

Girish L, Rao SKN, 2023. Anomaly detection in cloud environment using artificial intelligence techniques. *Computing*, 105(3):675-688. https://doi.org/10.1007/s00607-021-00941-x

Griffiths TL, Callaway F, Chang MB, et al., 2019. Doing more with less: meta-reasoning and meta-learning in humans and machines. *Curr Opin Behav Sci*, 29:24-30. https://doi.org/10.1016/j.cobeha.2019.01.005

Gruver N, Finzi M, Qiu SK, et al., 2024. Large language models are zero-shot time series forecasters. Proc 37th Int Conf on Neural Information Processing Systems, p.19622-19635. https://dl.acm.org/doi/10.5555/3666122.3666983

Guigou F, Collet P, Parrend P, 2019. SCHEDA: lightweight Euclidean-like heuristics for anomaly detection in periodic time series. *Appl Soft Comput*, 82:105594. https://doi.org/10.1016/j.asoc.2019.105594

Gupta V, Narwariya J, Malhotra P, et al., 2021. Continual learning for multivariate time series tasks with variable input dimensions. IEEE Int Conf on Data Mining, p.161-170. https://doi.org/10.1109/ICDM51629.2021.00026

Halbouni A, Gunawan TS, Habaebi MH, et al., 2022. CNN-LSTM: hybrid deep neural network for network intrusion detection system. *IEEE Access*, 10:99837-99849. https://doi.org/10.1109/ACCESS.2022.3206425

Han DQ, Wang ZL, Chen WQ, et al., 2021. DeepAID: interpreting and improving deep learning-based anomaly detection in security applications. Proc ACM SIGSAC Conf on Computer and Communications Security, p.3197-3217. https://doi.org/10.1145/3460120.3484589

Hawkins DM, 1980. Identification of Outliers. Springer, Dordrecht, Netherlands. https://doi.org/10.1007/978-94-015-3994-4

He Q, Zheng YJ, Zhang CL, et al., 2020. MTAD-TF: multivariate time series anomaly detection using the combination of temporal pattern and feature pattern. *Complexity*, 2020:8846608. https://doi.org/10.1155/2020/8846608

Heinle A, 2022. The Canada wide Rogers outage on July 8, 2022: what exactly happened & how can it be prevented? https://www.coguard.io/post/canada-rogers-outage-root-cause-analysis [Accessed on Apr. 15, 2024].

Hnamte V, Hussain J, 2023. DCNNBiLSTM: an efficient hybrid deep learning-based intrusion detection system. *Telemat Inform Rep*, 10:100053. https://doi.org/10.1016/j.teler.2023.100053

Houssel PR, Singh P, Layeghy S, et al., 2024. Towards explainable network intrusion detection using large language models. https://doi.org/10.48550/arXiv.2408.04342

Huang JJ, Kurniawan E, Sun SM, 2022. Cellular KPI anomaly detection with GAN and time series decomposition. IEEE Int Conf on Communications, p.4074-4079. https://doi.org/10.1109/ICC45855.2022.9838810

Hundman K, Constantinou V, Laporte C, et al., 2018. Detecting spacecraft anomalies using LSTMS and nonparametric dynamic thresholding. Proc 24th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining, p.387-395. https://doi.org/10.1145/3219819.3219845

Hwang RH, Peng MC, Huang CW, et al., 2020. An unsupervised deep learning model for early network traffic anomaly detection. *IEEE Access*, 8:30387-30399. https://doi.org/10.1109/ACCESS.2020.2973023

IMT-2030 (6G) Promotion Group, 2021. White Paper on 6G Vision and Candidate Technologies. Technical Report.

Jiang AQ, Sablayrolles A, Mensch A, et al., 2023. Mistral 7B. https://doi.org/10.48550/arXiv.2310.06825

Jiang J, Liu FG, Ng WWY, et al., 2023. AERF: adaptive ensemble random fuzzy algorithm for anomaly detection in cloud computing. *Comput Commun*, 200:86-94. https://doi.org/10.1016/j.comcom.2023.01.004

Jin M, Koh HY, Wen QS, et al., 2024a. A survey on graph neural networks for time series: forecasting, classification, imputation, and anomaly detection. https://doi.org/10.48550/arXiv.2307.03759

Jin M, Wang SY, Ma LT, et al., 2024b. Time-LLM: time series forecasting by reprogramming large language models. https://doi.org/10.48550/arXiv.2310.01728

Khalaf OI, Ogudo KA, Sangeetha SKB, 2022. Design of graph-based layered learning-driven model for anomaly detection in distributed cloud IoT network. *Mob Inform Syst*, 2022:6750757. https://doi.org/10.1155/2022/6750757

Khan MA, 2021. HCRNNIDS: hybrid convolutional recurrent neural network-based network intrusion detection system. *Processes*, 9(5):834. https://doi.org/10.3390/pr9050834

Khatibzadeh L, Bornaee Z, Bafghi AG, 2019. Applying catastrophe theory for network anomaly detection in cloud computing traffic. *Secur Commun Netw*, 2019:5306395. https://doi.org/10.1155/2019/5306395

Koroniotis N, Moustafa N, Sitnikova E, et al., 2019. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: BoT-IoT dataset. *Future Gener Comput Syst*, 100:779-796. https://doi.org/10.1016/j.future.2019.05.041

Kourtis MA, Xilouris G, Gardikis G, et al., 2016. Statistical-based anomaly detection for NFV services. IEEE Conf on Network Function Virtualization and Software Defined Networks, p.161-166. https://doi.org/10.1109/NFV-SDN.2016.7919492

KYODO NEWS, 2022. KDDI network outage affects record 30.91 million users. https://english.kyodonews.net/news/2022/07/57bbb532c4d7-kddi-network-outage-affects-record-3091-million-users.html [Accessed on Apr. 15, 2024].

Lalotra GS, Kumar V, Bhatt A, et al., 2022. iReTADS: an intelligent real-time anomaly detection system for cloud communications using temporal data summarization and neural network. *Secur Commun Netw*, 2022:9149164. https://doi.org/10.1155/2022/9149164

Le XH, Ho HV, Lee G, et al., 2019. Application of long short-term memory (LSTM) neural network for flood forecasting. *Water*, 11(7):1387. https://doi.org/10.3390/w11071387

Lee MC, Lin JC, Gran EG, 2020. RePAD: real-time proactive anomaly detection for time series. Proc 34th Int Conf on Advanced Information Networking and Applications, p.1291-1302. https://doi.org/10.1007/978-3-030-44041-1_110

Li D, Chen DC, Goh J, et al., 2019a. Anomaly detection with generative adversarial networks for multivariate time series. https://doi.org/10.48550/arXiv.1809.04758

Li D, Chen DC, Jin BH, et al., 2019b. MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks. 28th Int Conf on Artificial Neural Networks, p.703-716. https://doi.org/10.1007/978-3-030-30490-4_56

Li G, Jung JJ, 2023. Deep learning for anomaly detection in multivariate time series: approaches, applications, and challenges. *Inform Fus*, 91:93-102. https://doi.org/10.1016/j.inffus.2022.10.008

Li RY, Li Q, Zhang Y, et al., 2024. Interpreting unsupervised anomaly detection in security via rule extraction. Proc 37th Int Conf on Neural Information Processing Systems, p.62224-62243.

Liang Y, Zhang Y, Sivasubramaniam A, et al., 2005. Filtering failure logs for a BlueGene/L prototype. Int Conf on Dependable Systems and Networks, p.476-485. https://doi.org/10.1109/DSN.2005.50

Liang YL, Zhang YY, Xiong H, et al., 2007. Failure prediction in IBM BlueGene/L event logs. 7th IEEE Int Conf on Data Mining, p.583-588. https://doi.org/10.1109/ICDM.2007.46

Lim W, Yong KSC, Lau BT, et al., 2024. Future of generative adversarial networks (GAN) for anomaly detection in network security: a review. *Comput Secur*, 139:103733. https://doi.org/10.1016/j.cose.2024.103733

Lin QW, Zhang HY, Lou JG, et al., 2016. Log clustering based problem identification for online service systems. Proc 38th Int Conf on Software Engineering Companion, p.102-111.

Lin Z, Qu GQ, Chen QY, et al., 2024. Pushing large language models to the 6G edge: vision, challenges, and opportunities. https://doi.org/10.48550/arXiv.2309.16739

Liu C, Antypenko R, Sushko I, et al., 2022. Intrusion detection system after data augmentation schemes based on the VAE and CVAE. *IEEE Trans Reliab*, 71(2):1000-1010. https://doi.org/10.1109/TR.2022.3164877

Liu FT, Ting KM, Zhou ZH, 2008. Isolation forest. 8th IEEE Int Conf on Data Mining, p.413-422. https://doi.org/10.1109/ICDM.2008.17

Liu YL, Tao SM, Meng WB, et al., 2024. LogPrompt: prompt engineering towards zero-shot and interpretable log analysis. 46th IEEE/ACM Int Conf on Software Engineering, p.364-365. https://doi.org/10.1145/3639478.3643108

Lu HM, Wang T, Xu X, et al., 2022. Cognitive memory-guided autoencoder for effective intrusion detection in Internet of Things. *IEEE Trans Industr Inform*, 18(5):3358-3366. https://doi.org/10.1109/TII.2021.3102637

Lüdtke O, Robitzsch A, West SG, 2020. Regression models involving nonlinear effects with missing data: a sequential modeling approach using Bayesian estimation. *Psychol Methods*, 25(2):157-181. https://doi.org/10.1037/met0000233

Lüer F, Bohm C, 2024. Anomaly detection using generative adversarial networks reviewing methodological progress and challenges. *ACM SIGKDD Explor Newsl*, 25(2):29-41. https://doi.org/10.1145/3655103.3655109

Lunardi WT, Lopez MA, Giacalone JP, 2023. ARCADE: adversarially regularized convolutional autoencoder for network anomaly detection. *IEEE Trans Netw Serv Manage*, 20(2):1305-1318. https://doi.org/10.1109/TNSM.2022.3229706

Luo H, Zhong SS, 2017. Gas turbine engine gas path anomaly detection using deep learning with Gaussian distribution. Prognostics and System Health Management Conf, p.1-6. https://doi.org/10.1109/PHM.2017.8079166

Mascaro S, Nicholso AE, Korb KB, 2014. Anomaly detection in vessel tracks using Bayesian networks. *Int J Approx Reason*, 55(1):84-98. https://doi.org/10.1016/j.ijar.2013.03.012

Mathur AP, Tippenhauer NO, 2016. SWaT: a water treatment testbed for research and training on ICS security. Int Workshop on Cyber-Physical Systems for Smart Water Networks, p.31-36. https://doi.org/10.1109/CySWater.2016.7469060

Meidan Y, Bohadana M, Mathov Y, et al., 2018. N-BaIoT—network-based detection of IoT botnet attacks using deep autoencoders. *IEEE Pervasive Comput*, 17(3):12-22. https://doi.org/10.1109/MPRV.2018.03367731

Meng WB, Liu Y, Zhu YC, et al., 2019. LogAnomaly: unsupervised detection of sequential and quantitative anomalies in unstructured logs. Proc 28th Int Joint Conf on Artificial Intelligence, p.4739-4745.

Mirsky Y, Doitshman T, Elovici Y, et al., 2018. Kitsune: an ensemble of autoencoders for online network intrusion detection. https://doi.org/10.48550/arXiv.1802.09089

Montgomery B, 2024. Large-scale cellular phone outage hits AT&T customers across US. https://www.theguardian.com/technology/2024/feb/22/phone-outage-us [Accessed on Apr. 15, 2025].

Moustafa N, Slay J, 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). Military Communications and Information Systems Conf, p.1-6. https://doi.org/10.1109/MilCIS.2015.7348942

Nawaz A, Khan SS, Ahmad A, 2024. Ensemble of autoencoders for anomaly detection in biomedical data: a narrative review. *IEEE Access*, 12:17273-17289. https://doi.org/10.1109/ACCESS.2024.3360691

Neto ECP, Dadkhah S, Ferreira R, et al., 2023. CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors*, 23(13):5941. https://doi.org/10.3390/s23135941

Ngo MV, Luo T, Chaouchi H, et al., 2020. Contextual-bandit anomaly detection for IoT data in distributed hierarchical edge computing. IEEE 40th Int Conf on Distributed Computing Systems, p.1227-1230. https://doi.org/10.1109/ICDCS47774.2020.00191

Ngo MV, Luo T, Quek TQS, 2021. Adaptive anomaly detection for Internet of Things in hierarchical edge computing: a contextual-bandit approach. *ACM Trans Int Things*, 3(1):4. https://doi.org/10.1145/3480172

Nguyen TA, He JY, Le LT, et al., 2023. Federated PCA on Grassmann manifold for anomaly detection in IoT networks. IEEE Conf on Computer Communications, p.1-10. https://doi.org/10.1109/infocom53939.2023.10229026

Oliner A, Stearley J, 2007. What supercomputers say: a study of five system logs. 37th Annual IEEE/IFIP Int Conf on Dependable Systems and Networks, p.575-584. https://doi.org/10.1109/DSN.2007.103

OpenAI, 2024. GPT-4 technical report. https://doi.org/10.48550/arXiv.2303.08774

Ozyurt Y, Feuerriegel S, Zhang C, 2023. Contrastive learning for unsupervised domain adaptation of time series. https://doi.org/10.48550/arXiv.2206.06243

Pajouh HH, Dastghaibyfard G, Hashemi S, 2017. Two-tier network anomaly detection model: a machine learning approach. *J Intell Inform Syst*, 48(1):61-74. https://doi.org/10.1007/s10844-015-0388-x

Parameswarappa P, Shah T, Lanke GR, 2023. A machine learning-based approach for anomaly detection for secure cloud computing environments. Int Conf on Intelligent Data Communication Technologies and Internet of Things, p.931-940. https://doi.org/10.1109/IDCIoT56793.2023.10053518

Peng YH, Tan AP, Wu JJ, et al., 2019. Hierarchical edge computing: a novel multi-source multi-dimensional data anomaly detection scheme for Industrial Internet of Things. *IEEE Access*, 7:111257-111270. https://doi.org/10.1109/ACCESS.2019.2930627

Popoola SI, Ande R, Adebisi B, et al., 2022. Federated deep learning for zero-day botnet attack detection in IoT-edge devices. *IEEE Int Things J*, 9(5):3930-3944. https://doi.org/10.1109/JIOT.2021.3100755

Ratsch G, Mika S, Scholkopf B, et al., 2002. Constructing boosting algorithms from SVMs: an application to one-class classification. *IEEE Trans Patt Anal Mach Intell*, 24(9):1184-1199. https://doi.org/10.1109/TPAMI.2002.1033211

Ren HS, Xu BX, Wang YJ, et al., 2019. Time-series anomaly detection service at Microsoft. Proc 25th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining, p.3009-3017. https://doi.org/10.1145/3292500.3330680

Ren KY, Yuan S, Zhang C, et al., 2023. CANET: a hierarchical CNN-attention model for network intrusion detection. *Comput Commun*, 205:170-181. https://doi.org/10.1016/j.comcom.2023.04.018

Ren PZ, Xiao Y, Chang XJ, et al., 2021. A survey of deep active learning. *ACM Comput Surv*, 54(9):180. https://doi.org/10.1145/3472291

Reynolds D. 2009. Gaussian mixture models. In: Li SZ, Jain A (Eds.), Encyclopedia of Biometrics. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-73003-5_196

Rokach L, Maimon O. 2005. Clustering methods. In: Maimon O, Rokach L (Eds.), Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA. https://doi.org/10.1007/0-387-25465-X_15

Schneible J, Lu A, 2017. Anomaly detection on the edge. IEEE Military Communications Conf, p.678-682. https://doi.org/10.1109/MILCOM.2017.8170817

Segerholm L, 2023. Unsupervised Online Anomaly Detection in Multivariate Time-Series. https://stsprogrammet.se/wp-content/uploads/2023/02/2312_Ludvig_Segerholm.pdf [Accessed on Apr. 1, 2024].

Shan SW, Huo YT, Su YX, et al., 2024. Face it yourselves: an LLM-based two-stage strategy to localize configuration errors via logs. Proc 33rd ACM SIGSOFT Int Symp on Software Testing and Analysis, p.13-25. https://doi.org/10.1145/3650212.3652106

Sharafaldin I, Lashkari AH, Ghorbani AA, 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proc 4th Int Conf on Information Systems Security and Privacy, p.108-116. https://doi.org/10.5220/0006639801080116

Sharafaldin I, Lashkari AH, Hakak S, et al., 2019. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. Int Carnahan Conf on Security Technology, p.1-8. https://doi.org/10.1109/CCST.2019.8888419

Shi WB, Cao J, Zhang Q, et al., 2016. Edge computing: vision and challenges. *IEEE Int Things J*, 3(5):637-646. https://doi.org/10.1109/JIOT.2016.2579198

Shi YJ, Ying XH, Yang JF, 2022. Deep unsupervised domain adaptation with time series sensor data: a survey. *Sensors*, 22(15):5507. https://doi.org/10.3390/s22155507

Smith D, Guan Q, Fu S, 2010. An anomaly detection framework for autonomic management of compute cloud systems. IEEE 34th Annual Computer Software and Applications Conf Workshops, p.376-381. https://doi.org/10.1109/COMPSACW.2010.72

Song J, Takakura H, Okabe Y, et al., 2013. Toward a more practical unsupervised anomaly detection system. *Inform Sci*, 231:4-14. https://doi.org/10.1016/j.ins.2011.08.011

Song YJ, Xin RY, Chen P, et al., 2023. Identifying performance anomalies in fluctuating cloud environments: a robust correlative-GNN-based explainable approach. *Future Gener Comput Syst*, 145:77-86. https://doi.org/10.1016/j.future.2023.03.020

Srivastava S, Singh SP, 2016. A survey on latency reduction approaches for performance optimization in cloud computing. 2nd Int Conf on Computational Intelligence & Communication Technology, p.111-115.

Su J, Jiang CF, Jin X, et al., 2024. Large language models for forecasting and anomaly detection: a systematic literature review. https://doi.org/10.48550/arXiv.2402.10350

Tavallaee M, Bagheri E, Lu W, et al., 2009. A detailed analysis of the KDD CUP 99 data set. IEEE Symp on Computational Intelligence for Security and Defense Applications, p.1-6. https://doi.org/10.1109/CISDA.2009.5356528

Touvron H, Lavril T, Izacard G, et al., 2023a. LLaMA: open and efficient foundation language models. https://doi.org/10.48550/arXiv.2302.13971

Touvron H, Martin L, Stone K, et al., 2023b. LLaMA 2: open foundation and fine-tuned chat models. https://doi.org/10.48550/arXiv.2307.09288

Tuli S, Casale G, Jennings NR, 2022. TranAD: deep transformer networks for anomaly detection in multivariate time series data. https://doi.org/10.48550/arXiv.2201.07284

Tzeng E, Hoffman J, Zhang N, et al., 2014. Deep domain confusion: maximizing for domain invariance. https://doi.org/10.48550/arXiv.1412.3474

Venkateswara H, Eusebio J, Chakraborty S, et al., 2017. Deep hashing network for unsupervised domain adaptation. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.5385-5394. https://doi.org/10.1109/CVPR.2017.572

Vu L, Cao VL, Nguyen QU, et al., 2022. Learning latent representation for IoT anomaly detection. *IEEE Trans Cybern*, 52(5):3769-3782. https://doi.org/10.1109/TCYB.2020.3013416

Wang L, Yoon KJ, 2022. Knowledge distillation and student-teacher learning for visual intelligence: a review and new outlooks. *IEEE Trans Patt Anal Mach Intell*, 44(6):3048-3068. https://doi.org/10.1109/TPAMI.2021.3055564

Wang N, Chen YM, Hu Y, et al., 2022. FeCo: boosting intrusion detection capability in IoT networks via contrastive learning. IEEE Conf on Computer Communications, p.1409-1418. https://doi.org/10.1109/INFOCOM48880.2022.9796926

Wang W, Zhu M, Zeng XW, et al., 2017. Malware traffic classification using convolutional neural network for representation learning. Int Conf on Information Networking, p.712-717. https://doi.org/10.1109/ICOIN.2017.7899588

Wang YX, Yan J, Ye XY, et al., 2022. Few-shot transfer learning with attention mechanism for high-voltage circuit breaker fault diagnosis. *IEEE Trans Ind Appl*, 58(3):3353-3360. https://doi.org/10.1109/TIA.2022.3159617

Webb BK, Purohit S, Meyur R, 2024. Cyber knowledge completion using large language models. https://doi.org/10.48550/arXiv.2409.16176

Wu TT, Luo LH, Li YF, et al., 2024. Continual learning for large language models: a survey. https://doi.org/10.48550/arXiv.2402.01364

Xia X, Pan XZ, Li N, et al., 2022. GAN-based anomaly detection: a review. *Neurocomputing*, 493:497-535. https://doi.org/10.1016/j.neucom.2021.12.093

Xu HW, Chen WX, Zhao NW, et al., 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. Proc World Wide Web Conf, p.187-196. https://doi.org/10.1145/3178876.3185996

Xu W, Huang L, Fox A, et al., 2009. Detecting large-scale system problems by mining console logs. Proc ACM SIGOPS 22nd Symp on Operating Systems Principles, p.117-132. https://doi.org/10.1145/1629575.1629587

Xue H, Salim FD, 2024. PromptCast: a new prompt-based learning paradigm for time series forecasting. *IEEE Trans Knowl Data Eng*, 36(11):6851-6864. https://doi.org/10.1109/TKDE.2023.3342137

Yang L, Chen JJ, Wang Z, et al., 2021. Semi-supervised log-based anomaly detection via probabilistic label estimation. IEEE/ACM 43rd Int Conf on Software Engineering, p.1448-1460. https://doi.org/10.1109/ICSE43902.2021.00130

Yang YC, Lee K, Dariush B, et al., 2024. Follow the rules: reasoning for video anomaly detection with large language models. https://doi.org/10.48550/arXiv.2407.10299

Yu XY, Li T, Hu AQ, 2020. Time-series network anomaly detection based on behaviour characteristics. IEEE 6th Int Conf on Computer and Communications, p.568-572. https://doi.org/10.1109/ICCC51575.2020.9345249

Zanella L, Menapace W, Mancini M, et al., 2024. Harnessing large language models for training-free video anomaly detection. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.18527-18536. https://doi.org/10.1109/CVPR52733.2024.01753

Zeng AH, Liu X, Du ZX, et al., 2023. GLM-130B: an open bilingual pre-trained model. https://doi.org/10.48550/arXiv.2210.02414

Zhang JQ, Wang ZZ, Meng JJ, et al., 2019. Boosting positive and unlabeled learning for anomaly detection with multi-features. *IEEE Trans Multimedia*, 21(5):1332-1344. https://doi.org/10.1109/TMM.2018.2871421

Zhang P, Niu K, Tian H, et al., 2019. Technology prospect of 6G mobile communications. *J Commun*, 40(1):141-148 (in Chinese).

Zhang SL, Zhao CY, Sui YC, et al., 2021. Robust KPI anomaly detection for large-scale software services with partial labels. IEEE 32nd Int Symp on Software Reliability Engineering, p.103-114. https://doi.org/10.1109/ISSRE52982.2021.00023

Zhang X, Lin QW, Xu Y, et al., 2019a. Cross-dataset time series anomaly detection for cloud systems. Proc USENIX Annual Technical Conf, p.1063-1076.

Zhang X, Xu Y, Lin QW, et al., 2019b. Robust log-based anomaly detection on unstable log data. Proc 27th ACM Joint Meeting on European Software Engineering Conf and Symp on the Foundations of Software Engineering, p.807-817. https://doi.org/10.1145/3338906.3338931

Zhong ZY, Fan QL, Zhang JC, et al., 2023. A survey of time series anomaly detection methods in the AIOps domain. https://doi.org/10.48550/arXiv.2308.00393

Zhu B, Li J, Gu RB, et al., 2020. An approach to cloud platform log anomaly detection based on natural language processing and LSTM. Proc 3rd Int Conf on Algorithms, Computing and Artificial Intelligence, Article 88. https://doi.org/10.1145/3446132.3446415

Zhuang FZ, Qi ZY, Duan KY, et al., 2021. A comprehensive survey on transfer learning. *Proc IEEE*, 109(1):43-76. https://doi.org/10.1109/JPROC.2020.3004555