



REVIEW

Intelligent Identification over Power Big Data: Opportunities, Solutions, and Challenges

Liang Luo¹, Xingmei Li¹, Kaijiang Yang¹, Mengyang Wei¹, Jiong Chen¹, Junqian Yang¹ and Liang Yao^{2,*}

¹Dali Power Supply Bureau of Yunnan Power Grid Co., Ltd., Dali, 671000, China

²School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China

*Corresponding Author: Liang Yao. Email: yaoliang@nuist.edu.cn

Received: 31 December 2021 Accepted: 03 May 2022

ABSTRACT

The emergence of power dispatching automation systems has greatly improved the efficiency of power industry operations and promoted the rapid development of the power industry. However, with the convergence and increase in power data flow, the data dispatching network and the main station dispatching automation system have encountered substantial pressure. Therefore, the method of online data resolution and rapid problem identification of dispatching automation systems has been widely investigated. In this paper, we perform a comprehensive review of automated dispatching of massive dispatching data from the perspective of intelligent identification, discuss unresolved research issues and outline future directions in this area. In particular, we divide intelligent identification over power big data into data acquisition and storage processes, anomaly detection and fault discrimination processes, and fault tracing for dispatching operations during communication. A detailed survey of the solutions to the challenges in intelligent identification over power big data is then presented. Moreover, opportunities and future directions are outlined.

KEYWORDS

Data acquisition; data storage; anomaly detection; service fault-tolerant scheduling

1 Introduction

Recently, with the rapid development of the social economy and technology, the demand for a safe and stable power supply has generated higher requirements [1]. The relevant power sector has also been constantly improving its own ability to meet the increasing needs of people, gradually recognizing the importance of safety in power production [2,3].

The emergence of power dispatching automation systems has greatly improved the efficiency of power industry operations, solved many problems faced by the power system, and greatly promoted the rapid development of the power industry [4,5]. Driven by the development level of the social economy and science and technology, the development of power dispatching automation has strong scientific and economic support, and the dispatching level has gradually improved [6]. With the continuous development of teleoperation technology and improvement in the computer network



level, the development of power grid dispatching automation systems has undergone qualitative changes, from a single-function teleoperation telemetry device to the current multifunctional intelligent microcomputer teleoperation system, and the popularity and stability of products, accuracy and integrity of information have been greatly improved [7,8]. Currently, substations above 110 kV, most 35 kV substations and various types of power plants are equipped with corresponding dispatching automation systems [9,10].

As the speed of power grid construction accelerates and its scale expands, various dispatching automation systems are continuously built, and the business interaction applications between local dispatching systems and county dispatching systems become more frequent, so the power dispatching data network emerges [11–13]. The power dispatching data network is a special data network serving power dispatching and production, and its safe, stable and reliable operation is a basic guarantee for the safe production of the whole power grid [14–16]. The network has an important role in coordinating the joint operation of power system components, such as generation, transmission, transformation, distribution and consumption, and in ensuring the safe, economic, stable and reliable operation of the power grid and strongly guarantees the communication needs of power production, power dispatch, reservoir dispatch, fuel dispatch, relay protection, safety automatic devices, remote operation and power grid dispatch automation [17,18].

With the construction of a power supply company's power dispatching data network, the main station has been consecutively connected to all county dispatching services, resulting in further convergence and an increase in information flow in the main station, which puts considerable pressure on the dispatching data network and the main station dispatching automation system [19,20]. Due to access to county transfer automation business, the operation analysis report of power dispatch automation shows that many hidden defects are generated in the information flow, which is very difficult to trace and analyze due to the large amount of data and fast disappearance of information flow [21,22]. The dispatching automation system sometimes encounters the problem of misreporting or omitting information due to misdirected data from communication channels or plant and station general control equipment [23–25]. In addition to regular functional defects, there are still tremendous abnormal phenomena for which it is difficult to determine specific causes, which is not conducive to the elimination of power dispatch automation system abnormalities and becomes a bottleneck for improving the operation level of power dispatch automation systems [26,27].

- 1) Data analysis is not sufficiently accurate, and hidden problems are difficult to discover. The timely detection of many hidden problems from the master control and dispatch automation system is difficult through the dispatch automation system. For example, after the integration of local and county consolidation and regulation and control, an increasing number of dispatching objects are consecutively connected to the master station, and the dispatching automation system is stuck and unresponsive, which is likely to be related to problems such as intermittent errors and frequency of messages.
- 2) Problem data capture and diagnosis are not immediate enough. Intermittent false online, unsuccessful remote control, and partial signals that are not uploaded when tremendous signals are uploaded occasionally occur. Due to the lack of timely and in-depth analysis, the problem cannot be immediately identified, which hinders remote regulation and control and increases work costs. For example, if historical information is checked without a breaker variation signal, the system cannot determine the accident tripping after successfully tripping the reclosing action. This situation affects the efficiency of power grid repair. On the other hand, the outage information cannot be instantly reported.

- 3) Lack of data support for work such as counterevent analysis. Due to the lack of a black box mechanism, data for many problems are not instantly saved, which causes problems for scheduling automation personnel in learning, fault analysis, and drills.

To effectively identify the safety hazards inherent in the dispatching automation system in real time, it is necessary to improve the automation operation and management level of power supply companies and to lay a solid foundation for the safe, stable, high-quality and economic operation of the power grid through online analysis of the data volume of the dispatching automation system and rapid problem identification research.

The remainder of this paper is organized as follows: The acquisition and structured storage methods of the data flow are surveyed in Section 2. Section 3 investigates the existing anomaly detection approaches. In Section 4, the service fault-tolerant scheduling methods are surveyed. Section 5 concludes this paper.

2 Acquisition and Structured Storage of Data Flow

Presently, the single storage function enables optimization analysis and fault tracing when retrieving data [28,29]. In addition, the current scheduling data network service information flow storage function mainly stores the information flow data generated for the remote communication between the station and the master station but lacks the storage function for the internal data flow of the master station system [30,31]. It is difficult to combine data network information flow with scheduling key services to carry out data flow fault anomaly discrimination based on business scenarios [32,33].

To achieve the integrity of big data acquisition, data flow acquisition equipment depends on the high performance and fast processing capacity of software and hardware, which can quickly and deeply analyze the complete collected data and upload the data information after pattern recognition to the data business analysis platform [34,35]. Therefore, we need to analyze large amounts of data collection and storage solutions. The framework of data acquisition and storage is illustrated in Fig. 1.

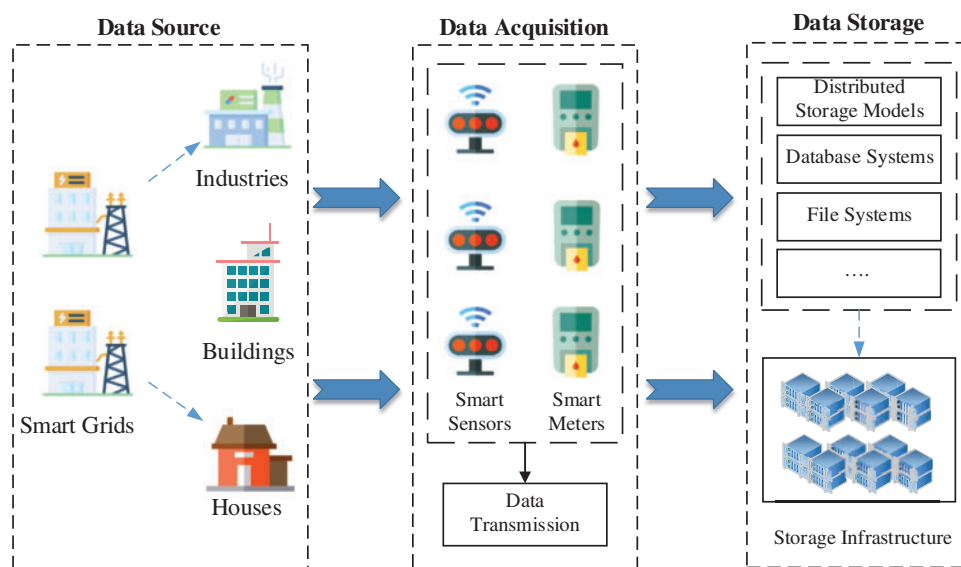


Figure 1: Framework of data acquisition and storage

2.1 Challenges in Data Acquisition and Storage

In existing works on the acquisition and structured storage of data flow, there are some open challenges, which are illustrated in Fig. 2.

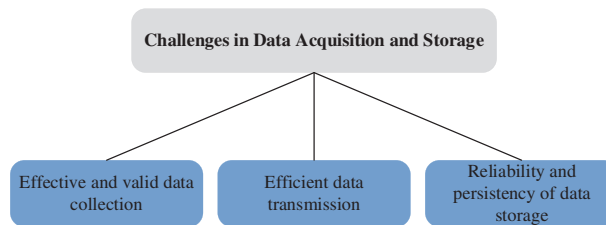


Figure 2: Challenges in data acquisition and storage

- 1) Effective and valid data collection. The process of obtaining raw data from an external system or network is referred to as data collection [36,37]. Effective and valid data collection is necessary because inefficient and improper data collection will negatively impact subsequent processing.
- 2) Efficient data transmission. Due to the high bandwidth consumption and the energy efficiency, transmitting numerous data to storage facilities becomes challenging [38,39].
- 3) Reliability and persistency of data storage. Considering the tremendous amount of data, it is challenging to achieve reliability and persistency of data storage while balancing the cost [40,41].

2.2 Current Solutions

2.2.1 Current Solutions to Data Acquisition

The data acquisition methods are listed in Table 1. Fig. 3 illustrates the main performance of the acquisition methods.

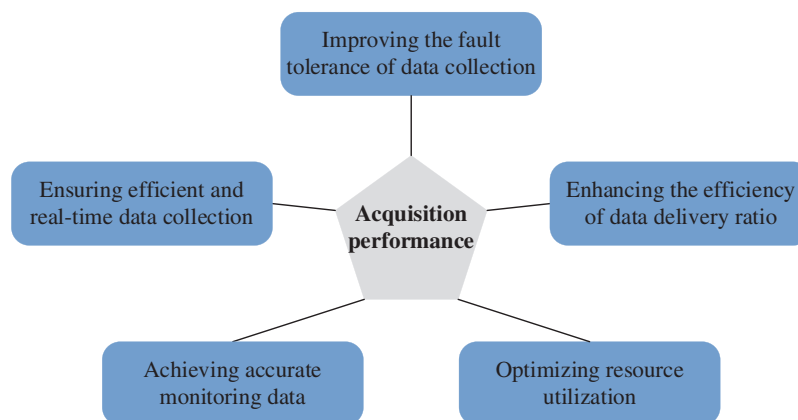
Table 1: Data acquisition methods

Target	Reference	Methods	Advantage
Acquisition performance	[42]	Introduced vehicular micro clouds to data collection	Improve the fault tolerance of data collection
	[43]	Data forwarding algorithm	Improves the efficiency of the data delivery ratio through restricting the broadcast messages
	[44]	Combine the long-term evolution network with vehicular hoc network to optimize the data volume and collection cycle	Optimize the data volume and collection cycle considering system robustness
	[45]	Data relay mule-based collection scheme	Improve the data acquisition performance

(Continued)

Table 1 (continued)

Target	Reference	Methods	Advantage
Optimizing resource utilization	[46]	Method of traffic data collection using vehicle-mounted monocular camera	Enhance the flexibility and coverage of traffic data collection
	[47]	Data acquisition scheme supported resource utilization optimization	Optimize the limitation of the wireless communication bandwidth
	[48]	On-demand vehicular sensing framework without infrastructure	Achieve accurate monitoring data and reduce the number of participating vehicles, energy consumption and communication costs
Security protection	[49]	Route optimization method	Improve the collection of multimedia data considering dynamic factors
	[50]	Data acquisition system based on blockchain	Utilize the UAV as a relay to gather data
	[51]	Security-preserving data collection and sharing scheme based on blockchain	Ensure efficient and real-time data collection with security protection

**Figure 3:** Main performance of the data acquisition methods

Hagenauer et al. [42] introduced vehicular micro clouds as edge servers into vehicular networks, which contributes to data collection. In [43], a data forwarding algorithm, which improves the efficiency of the data delivery ratio by restricting the broadcast messages, was proposed.

Considering the system robustness, Turcanu et al. [44] combined the long-term evolution network with a vehicular hoc network to optimize the data volume and collection cycle. Due to the lightweight signaling process, the repeated collection of large amounts of data is avoided, and the impact on system load is minimal.

Ren et al. [45] proposed a data relay mule-based collection scheme, named DRMCS, to accomplish efficient data acquisition without numerous increases in redundancy. Considering the delay and task completion rate with data acquisition, DRMCS utilized a micro mobile data center selection method based on a simulated annealing algorithm to improve the data acquisition performance and fault tolerance of data collection.

To address the limitation of flexibility and coverage of traffic data collection, a traffic data collection method that uses a vehicle-mounted monocular camera is proposed [46].

Nie et al. [47] focused on the potential of vehicular sensors in traffic data acquisition. The data are collected by sensors and then transmitted to road side units while moving along the road. However, wireless communication bandwidth is limited due to the numerous update data generated by vehicular network applications. A data acquisition scheme for supporting resource utilization optimization is proposed.

Rahman et al. [48] designed an on-demand, vehicular sensing framework without infrastructure, in which the users' phones serve as mobile collecting sensors. Through the proposed framework, accurate monitoring data can be obtained, and the number of participating vehicles, energy consumption and communication costs can be reduced. Considering the dynamic factors in multimedia data collection, Li et al. [49] proposed a route optimization method in which two rules of data and vehicle priority are applied to improve the collection of multimedia data in the Internet of Things.

Islam et al. [50] designed a data acquisition system based on blockchain, which utilized an unmanned aerial vehicle (UAV) as a relay to gather data in the Internet of Things environment. In the proposed system, the data are encrypted before the transmission process with the assistance of the UAV. The data are then integrally stored after the verification of the edge servers.

To address the problem of data security, Kong et al. [51] proposed a security-preserving, data collection and sharing scheme based on blockchain. During the data acquisition process, the disjunctive normal form cryptosystem and an identity-based signcryption scheme are integrated into the secure variance calculation of the collected data. In intelligent transportation systems (ITSs), efficient and real-time data collection is significant.

2.2.2 Current Solutions on Data Storage

The data storage methods are listed in Table 2. Fig. 4 illustrates the main performance of the acquisition methods.

Table 2: Data storage methods

Target	Reference	Method	Advantage
Storage performance	[52]	Distributed database based on Apache	Determine the validity of packets by detecting the payload
	[53]	Multilevel filtering method	Accomplish similarity join of the fuzzy string
	[54]	Reformative method in DPI based on regular expression	Utilize character intervals to describe multiple consecutive characters and improve transmission efficiency

(Continued)

Table 2 (continued)

Target	Reference	Method	Advantage
	[55]	Semistructured data storage and processing engine	Extract semantic values from a tremendous amount of patient data
	[56]	Semistructured data in the data flow, based on the DL semistructured tree miner algorithm	Effectively mined and stored data with the time attenuation model

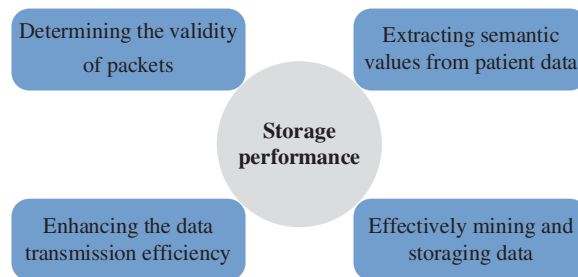


Figure 4: Main performance of the data storage methods

Power systems generate millions or even billions of status, debug, and error records every day. To ensure the security and sustainability of power systems, it is necessary to quickly process and analyze a large amount of power data to realize real-time decisions. Traditional solutions typically use relational databases to manage power data. However, when the amount of data substantially increases, the relational database cannot effectively process and analyze a large amount of power data.

Jin et al. [52] proposed a distributed database based on Apache. DPI is a packet-based, deep inspection technology that detects different network application layer payloads, such as HTTP and DNS, and that determines the validity of packets by detecting the payload.

A multilevel filtering method is proposed to accomplish similarity join of the fuzzy string. With the proposed method, Wang et al. [53] designed an elastic framework that transformed the problem of calculating fuzzy matching similarity into the weighted maximum matching problem at the element level, record level and similarity level.

Based on regular expression, a reformative method in DPI is proposed. In the face of increasingly complex attacks, accurate string recognition has difficulty accurately obtaining features. Regular expressions with flexibility and high efficiency are widely employed in feature fuzzy matching. In the matching process, Sun et al. [54] utilized the character interval to describe multiple consecutive characters, which improves the transmission efficiency.

With the intention of handling the problem of increasing medical expenses caused by the swift increase in the quantity and quality of medical data, Satti et al. [55] designed a semistructured data storage and processing engine, which can extract semantic values from numerous patient data generated by a variety of data sources at different rates and different levels of abstraction.

In [56], the authors concentrated on the mining of semistructured data in the data flow based on the deep learning (DL), semistructured tree miner algorithm. The data can be effectively mined and stored with the time attenuation model.

2.3 Opportunities in Data Acquisition and Storage

Although some of the previously described challenges in data acquisition and storage are addressed, opportunities remain, as illustrated in Fig. 5.

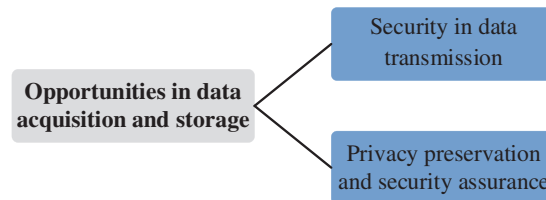


Figure 5: Opportunities in data acquisition and storage

- 1) Security in data transmission. Due to the limitation of network transmission conditions, data transmission is vulnerable to attack [57]. Yuan et al. [58] discussed the security of wireless data transmission. Therefore, the joint optimization of security protection and efficiency during data transmission is one of the future directions of data acquisition.
- 2) Privacy preservation and security assurance. Although data storage has received widespread attention in recent years, as summarized in Fig. 6, data storage security still faces severe opportunities. Bazai et al. [59] pointed out that MapReduce has potential risks of privacy disclosure. Hence, balancing privacy preservation and stability during the storage process is a promising direction for data storage.

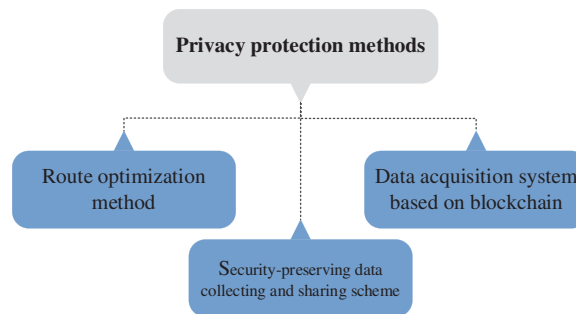


Figure 6: Privacy protection schemes for data acquisition and storage

3 Anomaly Detection of Data Flow

3.1 Challenges

Anomaly detection and fault discrimination have been investigated in many existing works [60,61]. Nevertheless, with the advent of the fifth generation (5G), the amount of data has significantly increased, which causes potential data anomalies and operating faults [62,63]. Specifically, challenges for anomaly detection and fault discrimination methods arise, as illustrated in Fig. 7.

- 1) Lack of training samples. Sufficient training samples are required to construct a model with high performance [64]. However, as data collection is challenging, existing samples are usually lacking. Therefore, training samples with the same characteristics should be generated.
- 2) Anomaly detection in time series data. Time-series data, such as weather data and power data, have high requirements for real time [65]. Therefore, to ensure the integrity of time series data, anomaly detection should be considered.

- 3) Privacy preservation in anomaly detection. To more efficiently detect anomalies, online, real-time anomaly detection methods are usually adopted [66]. However, the private information of users may be detected, resulting in the disclosure of users. Therefore, privacy preservation is one of the challenges in anomaly detection.

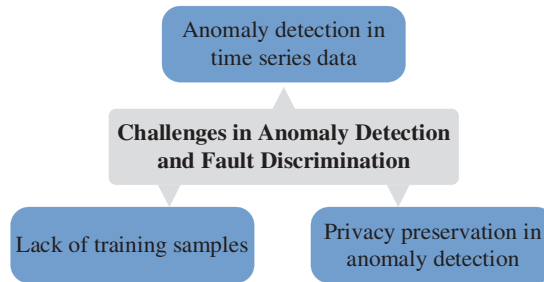


Figure 7: Challenges in anomaly detection of data flow

3.2 Goals on Anomaly Detection Methods

The reliability and real-time service scheduling flow directly influence the function [67]. Therefore, the service latency should be matched with the requirements of users while scheduling the service flow [68,69]. The investigation of the abnormal feature detection and fault discrimination of data flow is divided into two parts. First, with the feature analysis of the information flow, the formation mechanism and feature quantity should be examined. Second, the abnormal feature detection method of data flow is formed by monitoring the information flow. In the next section, we summarize existing works on anomaly detection methods regarding the two goals of improving the stability of data flow and enhancing the accuracy of data flow. The framework of anomaly detection is shown in Fig. 8. In addition, the main anomaly detection methods are illustrated in Fig. 9.

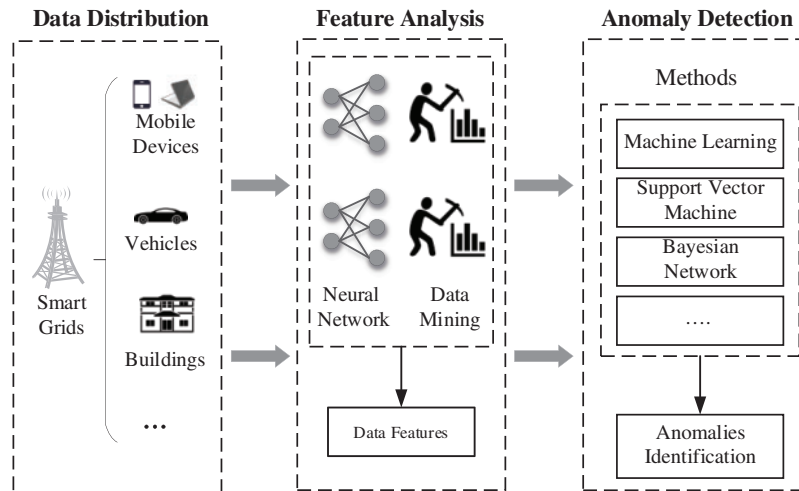


Figure 8: Framework of anomaly detection

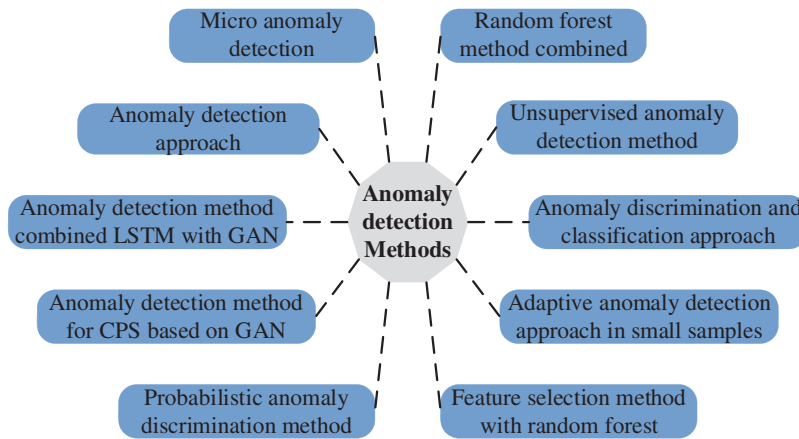


Figure 9: Main anomaly detection methods

3.2.1 Improving the Stability of Data Flow

The anomaly detection methods to improve the stability of data flow are listed in [Table 3](#).

Table 3: Anomaly detection to improve the stability of data flow

Target	Reference	Method	Advantage
Improving stability of data flow	[70]	Anomaly detection method that combines LSTM with a GAN	Improve the stability of time series data
	[71]	Survey of anomaly detection methods in computer network	Discuss the challenges and present the open problems
	[72]	Feature selection approach with random forest and support vector machine	Improve the stability of feature selection
	[73]	Adaptive online anomaly detection approach in small samples	Predict the unknown anomalies by classifying the known anomalies
	[74]	Random forest method combined with feature selection and DL classification	Perform better when facing financial data
	[75]	Anomaly detection method for CPS based on GAN	Identify the anomalies caused by various attacks
	[76]	Probabilistic anomaly discrimination method for wind turbine	Improve the stability of the wind turbine
	[77]	Micro anomaly detection in satellite telemetry data	Improve the stability of satellite telemetry data
[78]	Unsupervised anomaly detection method in the IoT	Effectively characterize the time series data in the IoT	

(Continued)

Table 3 (continued)

Target	Reference	Method	Advantage
	[79]	Anomaly detection approach with signal filtering discrimination	Enhance the security of the automated vehicle transportation
	[80]	Anomaly discrimination and classification approach	Improve the stability of the automotive industry

Zhu et al. [70] investigated the anomaly detection of data with time series. Conventional anomaly detection approaches can only detect abnormal data with time series at a shallow level because of the instability of time series data. The authors combined long short-term memory (LSTM) with a generative adversarial network (GAN) and designed a fusion model, named LSTM-GAN, to detect abnormal features in time series data.

The anomaly detection methods in computer networks were surveyed in [71]. First, the threat from attackers or crackers was introduced. Second, the deficiency of the traditional anomaly detection approach, which was based on the signature of threat, was analyzed. Last, the authors summarized existing anomaly detection systems and discussed the challenges and open problems.

Li et al. [72] explored network intrusion detection from the perspective of the stability of feature selection. Specifically, two feature selection approaches were evaluated, i.e., variables importance measure with a random forest (RF-VIM) and recursive feature elimination with a support vector machine (SVM-RFE). SVM-RFE could select the significant features but was susceptible to the imbalance rate. RF-VIM could provide stable subsets.

Existing anomaly detection methods performed poorly in the absence of training samples. Therefore, an adaptive online anomaly detection approach in small samples was proposed [73]. This method predicted the unknown anomalies by classifying the known anomalies.

To reduce the financial loss of financial statement fraud for investors, Yao et al. [74] investigated abnormal data detection in financial statements. First, the sources of abnormal data were analyzed. Second, the problem was modeled with DL. Last, a random forest method combined with feature selection and DL classification, which performed better when processing a large amount of financial data, was proposed.

Li et al. [75] proposed an anomaly detection approach based on a GAN for a cyber-physical system (CPS). By modeling the time sequence of sensors and actuators in the CPS, the potential anomaly was detected. Moreover, the proposed method effectively identified the anomalies caused by various attacks.

The stability of the wind turbine indicates its operating conditions. To develop the condition and anomaly detection for the wind turbine, Zhang et al. [76] presented a probabilistic anomaly discrimination method based on artificial intelligence, which was superior to conventional methods.

Sun et al. [77] analyzed the micro anomaly detection of the primary components in satellites. Due to the low discrimination accuracy in the conventional methods, an anomaly detection model based on the optimization sequence was constructed. By extracting the features of satellite telemetry data and segmenting the phases, anomaly discrimination for telemetry data was achieved.

Anomaly detection is necessary for the Internet of Things (IoT). However, the data are generally labeled to discriminate the anomalies. Guo et al. [78] investigated unsupervised anomaly detection in the IoT for time series data. The gated recurrent unit (GRU) was selected to represent the correlations among data, and Gaussian mixed priors were employed to characterize the data.

Currently, automated vehicle transportation, which is a novel MEC-based scenario, has been emerging. To enhance the security of automated vehicle transportation, Wang et al. [79] presented an anomaly detection approach combined with signal filtering discrimination. The anomalies were detected according to the trajectories of vehicles using the adaptive extended Kalman filter. Furthermore, the states of vehicles were realized by analyzing the states of surrounding traffic, which was more consistent with real conditions.

Numerous data were generated during the production and testing phases in the automotive industry. To evaluate the performance of vehicular systems, potential faults should be discriminated against. By analyzing the connections of vehicular systems, a robust anomaly discrimination and classification approach was presented [80].

3.2.2 Improving the Accuracy of Data Flow

The anomaly detection methods to improve the accuracy of data flow are listed in [Table 4](#).

Table 4: Anomaly detection to improve the accuracy of data flow

Target	Reference	Method	Advantage
Improving accuracy of data flow	[81]	Bias scoring mechanism for anomalies	Adaptively detect the anomaly
	[82]	Framework to detect the abnormal sequence	Improve the identification capacity of anomaly detection
	[83]	Anomaly detection in power systems with artificial neural network	Reduce the consumption of online resources
	[84]	Counting method for the telemetry data features	Extract the features and improve the accuracy of telemetry data
	[85]	Unsupervised anomaly detection approach	Distinguish between normal data and abnormal data
	[86]	Weather data analysis framework	Extract the features and modes in complicated weather data
	[87]	Anomaly detection model for satellite telemetry data with sequence features	Improve the accuracy of satellite telemetry data
	[88]	Intelligent anomaly detection method	Solve the class imbalance
	[89]	Survey of the anomaly detection methods with DL	Classify and evaluate the methods
	[90]	Anomaly detection method combined the autoencoder with LSTM	Improve the accuracy of anomaly detection

(Continued)

Table 4 (continued)

Target	Reference	Method	Advantage
	[91]	Generation and identification model with GAN	Generate samples to train the identification model
	[92]	Identification and correction method for drilling data	Correct the abnormal drilling data
	[93]	Extreme gradient boosting framework	Improve the classification accuracy
	[94]	Abnormal traffic discrimination model	Generate the substation samples

Anomaly detection in computer networks was investigated in [81]. First, the anomaly was classified according to the abnormal data. Second, the anomaly was exposed through a specific filter. Last, the bias scoring mechanism was utilized to adaptively detect the anomaly.

Song et al. [82] presented a framework for detecting abnormal sequences, which is available for intrusion identification and fault discrimination. First, the framework projected the data into the feature space based on the model. To improve the identification capacity of anomaly detection, the discriminative features were extracted from the generation model. Second, the anomalies were detected by the classifiers generated from the transformed data.

Anomaly detection is necessary to maintain the stability of power systems. The artificial neural network (ANN) could train offline data and reduce the consumption of online resources. Therefore, ANNs can be applied to power systems to detect faults. Yadav et al. [83] discussed the application of ANNs to power systems for anomaly detection, identification, and classification and compared the performance of the approaches.

On-orbit anomaly detection is a primary problem in satellite management [84]. To identify the anomalies for complex satellite telemetry data, a counting method for the telemetry data features, which detected the abnormal data by extracting the changing frequency and extent of data to illustrate the data features, was presented.

To identify and detect the anomalies in the process of chemical plants, an unsupervised approach that combines graph theory (GT) with generative topographic mapping (GTM) was presented in [85]. Specifically, GTM offered a policy for calculating the similarity between two samples, whereas GT distinguished normal items from abnormal items by clustering.

Weather data analysis can be implemented by the IoT and big data framework. To extract the features and modes in complicated weather data, a weather sensor anomaly detection algorithm using clustering was explored [86].

To improve the accuracy of satellite telemetry data, Du et al. [87] proposed an anomaly detection model for satellite telemetry data with sequence features to improve their accuracy. First, the telemetry data were steadily separated to obtain stable residual and data trends. Second, the anomaly detection model was designed by fusing the features.

The monitoring and acquisition data in the wind turbine system were imbalanced because of the large amount of data. Therefore, the abnormal data were difficult to accurately discriminate. With

the deep neural network (DNN), Chen et al. [88] presented an intelligent anomaly detection method, which solved the class imbalance by classifying the source monitoring data.

The anomaly detection methods with DL are investigated in detail [89]. According to the basic assumptions and research methods, anomaly detection methods are classified to distinguish normal behavior from abnormal behavior. After evaluating the performance, the advantages and deficiencies of these methods are also discussed.

Park et al. [90] investigated fault detection for time sequence data and presented an anomaly detection method combined with an autoencoder and LSTM. Normal offline data were employed to train the autoencoder to identify and classify the anomaly.

DL, which is an algorithm driven by neural networks, is rapidly developing. DL models with feature representation are applied to fault detection. However, the misclassification rate was increased when the fault data were limited. To improve the accuracy of anomaly detection, Zhou et al. [91] designed a generation and identification model with a GAN, where global optimization was performed to generate more fault samples and train the identification model.

To improve the quality of drilling data, Yang et al. [92] investigated anomaly detection and proposed an identification and correction method for drilling data. First, to detect all kinds of abnormal data, the local detection algorithm was designed to obtain the reasons for abnormal data. Second, the effective k nearest algorithm was utilized to correct the abnormal data.

To improve the classification accuracy of the scheme for protecting the power transformer, Raichura et al. [93] designed the extreme gradient boosting framework to distinguish the outside faults and inner anomalies. Moreover, a convolutional neural network (CNN) was employed to classify the faults. A comparison with other machine learning algorithms, such as SVMs, revealed that the proposed method performed better in classification accuracy.

With the development of digitalization, the flow of substation communication networks is increasing. Moreover, abnormal traffic discrimination has been the key to maintaining network security. Yang et al. [94] constructed an abnormal traffic discrimination model for a substation communication network based on the ResNet model, solving the problem of insufficient substation samples.

3.3 Opportunities for Anomaly Detection to Improve Detection Performance

Although some of the previously described challenges in data acquisition and storage are addressed, opportunities to improve the performance of anomaly detection are discussed in existing works and are illustrated in Fig. 10.

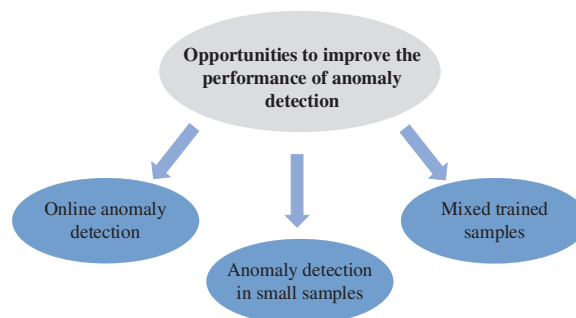


Figure 10: Opportunities in anomaly detection of data flow

- 1) Anomaly detection in small samples [73]. Ideally, the training process of anomaly detection and fault discrimination models should cover as many kinds of anomalies to be effective. However, the practical samples fail to reach the ideal situations, which reduces the prediction performance for the rare anomaly of the anomaly detection method [95,96]. Therefore, anomaly detection in small samples is a future research direction of anomalies.
- 2) Online anomaly detection [80]. Anomalies should be detected in time after occurring and eliminated with relevant techniques. Therefore, to improve the detection accuracy, an online anomaly detection approach is necessary [97,98]. Although online anomaly detection methods have been investigated in certain existing works, the detection accuracy and efficiency are too low to meet the requirements of practical application scenarios. Therefore, online anomaly detection is also a future direction of anomaly detection to improve the detection accuracy and efficiency.
- 3) Mixed trained samples [82]. Existing anomaly detection and fault discrimination methods are mostly inherently trained with only normal data. However, it is not possible to obtain samples that include only normal data in practical scenarios. Therefore, the anomaly detection and fault discrimination model should be trained in mixed samples, including normal and anomalous data [99]. In addition, the trained samples need to contain labeled and unlabeled data to improve the performance of the model [100,101]. Hence, detection under mixed trained samples is also a direction for future anomaly detection.

4 Service Fault-Tolerant Scheduling Based on Communication

4.1 Challenges

Most existing works focus on service fault-tolerant scheduling. However, with the development of a heterogeneous system and an increase in data, certain challenges about service fault-tolerant scheduling algorithms are surveyed, as illustrated in Fig. 11.

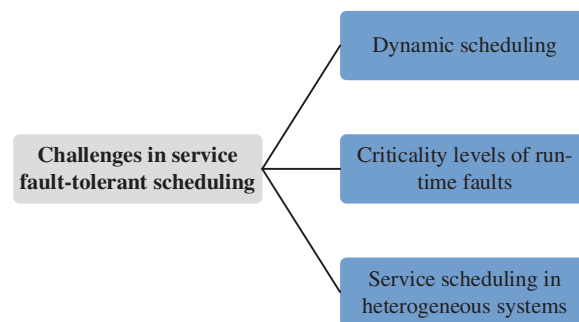


Figure 11: Challenges in service fault-tolerant scheduling

- 1) Dynamic scheduling. The dynamic fault-tolerant scheduling of services effectively reduces the delay and energy consumption caused by resource redistribution and improves resource utilization [102].
- 2) Criticality levels of run-time faults. The criticality levels of run-time faults represent the priority to handle. To enhance the efficiency of service fault-tolerant scheduling, run-time faults with high criticality levels must be addressed first [103].

- 3) Service scheduling in heterogeneous systems. Heterogeneous systems bring convenience to service scheduling and increase the complexity of the systems. Many complex faults occur in heterogeneous systems, which provides new challenges for fault-tolerant scheduling algorithms [104].

4.2 Goals on Service Fault-Tolerant Scheduling

With the development of big data and 5G, numerous data have been generated according to the requirements of users [105,106]. The service schedule, which is a technique that improves resource utilization by scheduling the execution of service, renders the data storage and analysis convenient [107,108]. However, potential faults, e.g., missing data, may occur in service scheduling. Therefore, service fault-tolerant scheduling has a significant role in service scheduling and directly affects the reliability of the network system [109,110]. Fig. 12 demonstrates the service fault-tolerant scheduling framework. The data center includes multiple hosts, denoted by $H = \{h_1, h_2, \dots, h_n\}$. In addition, each host provides many virtual machines (VMs). Let $v_{i,j}$ denote the j -th VM on h_i . As shown, first, the users provide the service requirements, which include a task queue. Second, to satisfy the requirements, the system scheduler is used to schedule the tasks. Due to possible faults, such as the loss of data and anomaly of the scheduler, the fault-tolerant mechanism is added to the system scheduler to eliminate the potential faults. Last, the scheduled tasks are processed in the VMs of the data center.

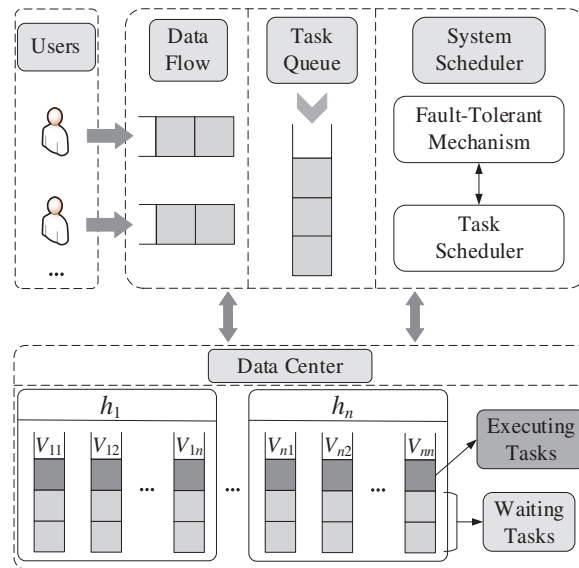


Figure 12: Framework of service fault-tolerant scheduling

Next, we will summarize existing works on service fault-tolerant scheduling from four optimization goals, i.e., reducing energy consumption, decreasing service response latency, improving resource utilization and enhancing the reliability of systems, which are shown in Fig. 13.

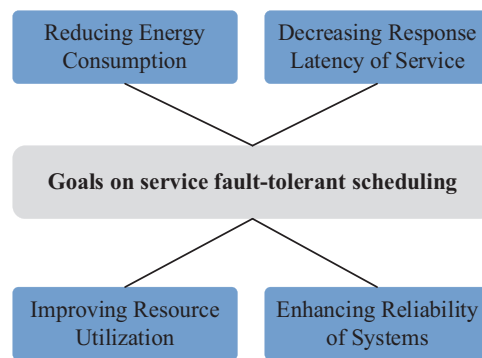


Figure 13: Goals of service fault-tolerant scheduling methods

4.2.1 Reducing Energy Consumption

The service fault-tolerant scheduling methods for decreasing energy consumption are listed in Table 5.

Table 5: Service fault-tolerant scheduling to reduce energy consumption

Target	Reference	Method	Advantage
Reducing the energy consumption	[111]	Heuristic algorithm based on the earliest finish time of service clusters	Reduce the resources consumption
	[112]	Survey of fault-tolerant service scheduling in CC	Evaluate the performance of existing methods
	[113]	Energy-aware, efficient heuristic scheme	Solve fault-tolerant scheduling for real-time tasks in heterogeneous systems
	[114]	Fault-tolerance scheduling method by developing three mode redundancy	Decrease the energy consumption in systems
	[115]	Fault-tolerant scheduling method with checkpoints	Address the potential faults and decrease the energy consumption in systems
	[116]	Byzantine fault detection algorithm	Decrease the fault-tolerant overhead

Currently, executing clustering services will increase the efficiency of scientific workflows (SWf) in cloud servers. Vinay et al. [111] presented a heuristic algorithm based on the earliest finish time of clusters to develop fault-tolerant scheduling in cloud computing (CC). When the service in clusters is unsuccessfully executed, the proposed algorithm will execute again using idle time, decreasing the resource consumption.

The scheduling algorithms in CC focus on high-performance computation and computing costs. However, because of the incomplete scheduling strategies, the execution efficiency of computing tasks is hard to guarantee. Therefore, to build a foundation for constructing an efficient fault-tolerant

framework, Pandita et al. [112] evaluated the fault-tolerance performance of the existing scheduling algorithm in CC.

Nair et al. [113] proposed an energy-aware, efficient heuristic scheme to solve fault-tolerant scheduling for real-time tasks in heterogeneous systems. The authors designed a standby-sparing method, where the efficient core was utilized to process the critical tasks and the high-performance core was utilized to process the tasks affected by faults.

Three mode redundancy (TMR) is used to eliminate faults in homogeneous systems with high energy consumption. Yu et al. [114] presented a fault-tolerance scheduling strategy in heterogeneous systems by developing TMR. Specifically, the services without the requirements of fault tolerance were still executed with the traditional TMR. Otherwise, the services were executed with the proposed method, which decreased the energy consumption in systems.

In mobile cloud computing (MCC), mobile devices are usually resource limited. The scheduling strategy must be updated when scheduling resources change. Lee et al. [115] proposed a fault-tolerant scheduling method with checkpoints and a replication mechanism to handle potential faults, which reduced energy consumption.

Chinnathambi et al. [116] presented a Byzantine fault detection algorithm and designed a scheduling and checkpoint optimization algorithm to tolerate and eliminate the Byzantine fault. In addition, the proposed algorithm exponentially decreased the fault-tolerant overhead and effectively allocated the virtual resources.

4.2.2 Decreasing Response Latency of Service

The service fault-tolerant scheduling approaches for reducing the response latency of service are listed in Table 6.

Table 6: Service fault-tolerant scheduling to reduce the response latency of service

Target	Reference	Method	Advantage
Decreasing response latency of service	[117]	Fault-tolerant service scheduling scheme with checkpoints in CC	Reduce the response latency of service
	[118]	Fault-tolerance scheduling algorithm based on resubmitting and duplication	Maximize the idle time during service scheduling
	[119]	Dynamic clustered scheduling method	Decrease the execution delay of tasks
	[120]	Heuristic service, fault-tolerant scheduling algorithm based on cross-entropy	Balance the running time of tasks and the lifetime of systems
	[121]	Multipath service scheduling algorithm	Optimize the reliability and fault tolerance of the IIoT

With the development of intelligent computing techniques in CC, fault tolerance has become increasingly significant [117]. Therefore, a fault-tolerant-oriented service scheduling scheme was presented. The checkpoint strategy was employed to migrate the execution-failed services, which efficiently reduced the service response delay.

Yao et al. [118] presented a fault-tolerance scheduling algorithm based on resubmitting and duplication. First, as the workflow was divided into several subtasks, the deadline was also divided. Afterward, the fault-tolerance strategy with resubmission and duplication of each subtask was allocated. To maximize the idle time, an online adjustment method for the scheduling strategy of unexecuted tasks was also designed.

Considering the failure of computation tasks in CC, Abd Latiff et al. [119] presented a dynamic clustered scheduling method based on the League Championship Algorithm (LCA). This algorithm could monitor the available resources and prevent premature failure of tasks, which decreased their execution latency.

Cao et al. [120] examined how to prolong the lifetime of fault-tolerant, mixed-criticality embedded systems. Since the mixed-integer linear programming was time-consuming when employed in large-scale systems, a heuristic algorithm based on the cross-entropy was presented, balancing the running time of tasks and lifetime of systems.

Applications in the Industrial Internet of Things (IIoT) usually require high reliability and low access latency. Ahrar et al. [121] explored the service schedule in IIoT and proposed a multipath scheduling algorithm that considered the potential faults in the paths. This algorithm optimized the reliability and fault tolerance of IIoT by analysis of the experiments in the simulated heterogeneous scenarios.

4.2.3 Improving Resource Utilization

The service fault-tolerant scheduling methods to improve resource utilization are listed in Table 7.

Table 7: Service fault-tolerant scheduling to improve resource utilization

Target	Reference	Method	Advantage
Improving resource utilization	[122]	Dynamic resource allocation method with a fault-tolerant mechanism	Improve the resource utilization of systems
	[123]	Service allocation and communication approach with real-time fault-tolerance	Improve the computation resources utilization
	[124]	Offline elastic scheduling algorithm	Dynamically regulate resource allocation and increase resources utilization
	[125]	Dynamic elastic fault-tolerant scheduling method	Improve resource utilization in the cloud
	[126]	Dynamic task allocation and scheduling scheme	Optimize the energy efficiency
	[127]	Fault-tolerant framework with criticality levels of run-time faults	Avoid the overallocation of computation resources

Fault tolerance is widely employed in CC. Soniya et al. [122] investigated fault-tolerant service scheduling in CC. First, a dynamic resource allocation method with a fault-tolerant mechanism, which

enhanced resource utilization, was presented. Second, combined with the virtual machine scheduling approach, a dynamic scheduling scheme with a fault-tolerant mechanism for real-time services in CC was proposed.

Zhu et al. [123] examined the fault-tolerant mechanism in a real-time workflow. First, based on the conducted workflow model, which tolerates real-time faults, the service allocation and communication approach is presented, improving the computation resource utilization by fully using idle resources.

In cloud systems, fault tolerance has been the primary requirement for the execution of computation tasks. Therefore, Ding et al. [124] worked on the fault tolerance of the task workflow and presented an offline elastic scheduling algorithm with fault tolerance to dynamically regulate resource allocation and increase resource utilization in cloud systems.

Yan et al. [125] presented a dynamic elastic fault-tolerant scheduling method for cloud services, realizing fault tolerance and increasing resource utilization. First, a fault-tolerant task allocation method was designed. Considering the uncertainty, two task scheduling models with fault tolerance were interchangeably employed. Second, the overlapping mechanism was adopted to improve resource utilization in the cloud.

Marahatta et al. [126] proposed a dynamic task allocation and scheduling scheme based on fault tolerance to coordinately optimize energy efficiency and resource utilization. Specifically, the upcoming tasks were classified and then allocated to the appropriate virtual machine for execution. In addition, a flexible resource supply mechanism was also developed to optimize energy efficiency.

Chen et al. [127] designed a fault-tolerant framework to address faults according to the criticality levels of run-time faults. To avoid the overallocation of computational resources, an overrun handling protocol was also proposed, which ensured fault recovery. In addition, an offline scheduling analysis technique was adopted to evaluate the proposed approach.

4.2.4 Enhancing the Reliability of Systems

The service fault-tolerant scheduling methods to improve the reliability of systems are listed in Table 8.

Table 8: Service fault-tolerant scheduling methods to improve the reliability of systems

Target	Reference	Method	Advantage
	[128]	Fault-tolerant scheduling method in power management system	Operate the deceleration mechanism according to the actual workload
	[129]	Fault-tolerant task scheduling algorithm in grid computing	Improve the reliability of systems
	[130]	Fault-tolerance method in criticality-mixed systems	Improve the security of tasks with different critical levels
	[131]	Fault trace approach for the power grid	Take full advantage of the monitoring data and infer the fault reasons

(Continued)

Table 8 (continued)

Target	Reference	Method	Advantage
Enhancing reliability of systems	[132]	Heuristic fault-tolerant task scheduling algorithm	Improve the reliability of systems by the number of tolerant permanent faults
	[133]	Adaptive fault-tolerant scheduling algorithm	Select the most appropriate fault-tolerant technique to address the faults
	[134]	Service fault-tolerant scheduling method in scientific workflows	Improve the reliability of systems
	[135]	Task clustering algorithm with fault tolerance	Synthetically consider the execution latency and cost of workflows

Zhang et al. [128] investigated fault tolerance in a power management system and designed a fault-tolerant scheduling method. Specifically, the online method calculates the running rate of the system and operates the deceleration mechanism according to the actual workload.

Grid computing serves computation-sensitive and long-operating applications. To guarantee the quality of service (QoS), these applications need to tolerate potential faults. Based on the ant colony algorithm, Idris et al. [129] presented a fault-tolerant task scheduling algorithm in grid computing, ensuring that the tasks can be normally executed when faults occur.

In real-time systems and embedded systems, criticality-mixed task scheduling is usually considered. Zhou et al. [130] designed a fault-tolerance method in criticality-mixed systems to improve the security of tasks with different critical levels.

Wang et al. [131] proposed a fault trace approach for the power grid in a big data platform. First, Spark was used to handle the faults. Second, the fault was analyzed by data mining. Last, according to the decision tree, the reasons for faults were inferred. In addition, this method could take full advantage of the monitoring data and infer the fault reasons.

The complexity of heterogeneous systems increases the possibility of faults in the systems, resulting in the growing significance of efficient task scheduling strategies with fault tolerance. Hence, Liu et al. [132] proposed a heuristic, fault-tolerant task scheduling algorithm to improve the reliability of systems by dynamically computing the number of tolerating permanent faults.

Alarifi et al. [133] presented a fault-tolerant scheduling algorithm to allocate service requests to the appropriate devices in the IoT. When the service execution failed, the proposed algorithm selected the most appropriate fault-tolerant technique from replication, resubmission, and checkpoint techniques, increasing the reliability of the IoT.

Fault tolerance is the primary technique in CC. When the workflow experiences faults, the applications provide a protection mechanism to ensure the safety of the systems. Talwani et al. [134] compared the techniques that optimized the fault tolerance in scientific workflows according to real-world workflows.

Task clustering can enhance the computational granularity of the scientific workflow to execute tasks with distributed computing resources. Khaldi et al. [135] proposed a task clustering algorithm with fault tolerance. The algorithm considered the constraints of workflow execution time and execution cost to develop the performance of the workflow.

4.3 Opportunities for Service Fault-Tolerant Scheduling to Enhance Scheduling Efficiency

Although some of the previously mentioned challenges in service fault-tolerant scheduling are addressed, opportunities to enhance the efficiency of service scheduling remain, as illustrated in Fig. 14.

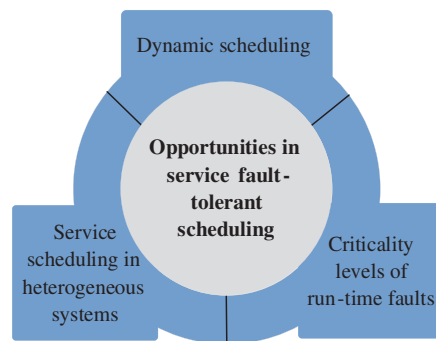


Figure 14: Opportunities in service fault-tolerant scheduling

- 1) Elastic resource scheduling. The rich resources in data centers are used to process a large amount of data, which consumes resources and power and negatively impacts the natural environment [136]. Therefore, the green processing of data becomes increasingly significant to protecting the environment. For each service, the resource requirements of services should be analyzed in detail. Based on the consideration of fault tolerance, the resources should be elastically allocated for services in the data center to save energy [137]. Therefore, elastic resource scheduling is a future direction of service fault-tolerant scheduling.
- 2) Prediction for errors in tasks. Frequent task migration causes additional energy consumption and influences the system performance [138]. Therefore, potential faults can be predicted based on the characteristics of tasks executed in the long term. Resources should be provided to address the faults, reducing the energy consumption [139]. Thereby, prediction for errors in tasks is also a future research direction of service fault-tolerant scheduling.
- 3) Security in service fault-tolerant scheduling. Service fault-tolerant scheduling requires the detection of multiple attributes of services, possibly including the privacy of the user, which leads to the significance of security [140]. A service fault-tolerant scheduling method with high performance should address the relationship of security and fault tolerance [141]. Hence, security in service fault-tolerant scheduling is also a future research direction.

5 Conclusion

In this work, a comprehensive and detailed survey of intelligent identification based on power big data is presented. First, the data acquisition and storage process are investigated. Second, the anomaly characteristics and fault discrimination techniques of a massive amount of data are analyzed. Furthermore, the problem of fault tracing for dispatching operations during communication is

discussed. This survey is presented to promote the further progress of intelligent identification based on power big data. However, numerous research issues in this area are still open and need further efforts, including optimizing the distributed intelligent identification process and balancing security protection and performance.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Liu, M., Guo, Q., Zhang, G., Su, J. (2019). Application of emergency control of central air conditioning in safe and stable operation of large power grid. *The Journal of Engineering*, 2019(16), 1539–1543. DOI 10.1049/joe.2018.8618.
2. Worigli, I., Maach, A., Hafid, A., Hegazy, O., Van Mierlo, J. (2019). Integrating renewable energy in smart grid system: Architecture, virtualization and analysis. *Sustainable Energy, Grids and Networks*, 18, 100226. DOI 10.1016/j.segan.2019.100226.
3. Bagdadee, A. H., Aurangzeb, M., Ali, S., Zhang, L. (2020). Energy management for the industrial sector in smart grid system. *Energy Reports*, 6, 1432–1442. DOI 10.1016/j.egy.2020.11.005.
4. Hou, H., Xu, Q., Pang, Y., Li, L., Wang, J. et al. (2017). Efficient storing energy harvested by triboelectric nanogenerators using a safe and durable all-solid-state sodium-ion battery. *Advanced Science*, 4(8), 1700072. DOI 10.1002/advs.201700072.
5. Xu, B., Chen, D., Venkateshkumar, M., Xiao, Y., Yue, Y. et al. (2019). Modeling a pumped storage hydropower integrated to a hybrid power system with solar-wind power and its stability analysis. *Applied Energy*, 248, 446–462. DOI 10.1016/j.apenergy.2019.04.125.
6. Yang, D., Wei, H., Zhu, Y., Li, P., Tan, J. C. (2018). Virtual private cloud based power-dispatching automation system—Architecture and application. *IEEE Transactions on Industrial Informatics*, 15(3), 1756–1766. DOI 10.1109/TII.2018.2849005.
7. Yan, F., Wang, C., Dou, J., Liu, Y., Yang, X. (2018). Application of speech recognition technology in power grid dispatching automation. *IOP Conference Series: Materials Science and Engineering*, 394(4), 42111. DOI 10.1088/1757-899X/394/4/042111.
8. Kulkarni, N., Lalitha, S. V. N. L., Deokar, S. A. (2019). Real time control and monitoring of grid power systems using cloud computing. *International Journal of Electrical & Computer Engineering*, 9(2), 2088–8708. DOI 10.11591/ijece.v9i2.pp941-949.
9. Zhao, B., Ma, L., Xie, H., Wu, K., Wang, X. et al. (2021). Self-adaptive multiblock-copolymer-based hybrid solid-state electrolyte for safe and stable lithium-metal battery. *Electrochimica Acta*, 371, 137702. DOI 10.1016/j.electacta.2020.137702.
10. Li, J., Peng, H., Cao, Y., Dou, Y., Zhang, H. et al. (2021). Higher-order attribute-enhancing heterogeneous graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 1. DOI 10.1109/TKDE.2021.3074654.
11. Liu, Y., Song, Z., Xu, X., Rafique, W., Zhang, X. et al. (2021). Bidirectional GRU networks-based next POI category prediction for healthcare. *International Journal of Intelligent Systems*, 37, 1–21. DOI 10.1002/int.22710.
12. Xu, X., Huang, Q., Zhang, Y., Li, S., Qi, L. et al. (2021). An LSH-based offloading method for IoMT services in integrated cloud-edge environment. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 16(3s), 1–19. DOI 10.1145/3408319.

13. Xu, X., Yao, L., Bilal, M., Wan, S., Dai, F. et al. (2021). Service migration across edge devices in 6G-enabled internet of vehicles networks. *IEEE Internet of Things Journal*, 9(3), 1930–1937. DOI 10.1109/JIOT.2021.3089204.
14. Lasseter, R. H., Chen, Z., Pattabiraman, D. (2019). Grid-forming inverters: A critical asset for the power grid. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 8(2), 925–935. DOI 10.1109/JESTPE.6245517.
15. Xu, X., Zhang, X., Liu, X., Jiang, J., Qi, L. et al. (2020). Adaptive computation offloading with edge for 5G-envisioned internet of connected vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(8), 5213–5222. DOI 10.1109/TITS.2020.2982186.
16. Llopis-Albert, C., Rubio, F., Valero, F. (2021). Impact of digital transformation on the automotive industry. *Technological Forecasting and Social Change*, 162, 120343. DOI 10.1016/j.techfore.2020.120343.
17. Liu, J., Yao, W., Wen, J., Fang, J., Jiang, L. et al. (2019). Impact of power grid strength and PLL parameters on stability of grid-connected DFIG wind farm. *IEEE Transactions on Sustainable Energy*, 11(1), 545–557. DOI 10.1109/TSTE.5165391.
18. Qi, L., Song, H., Zhang, X., Srivastava, G., Xu, X. et al. (2021). Compatibility-aware web API recommendation for mashup creation via textual description mining. *ACM Transactions on Multimedia Computing Communications and Applications*, 17(1s), 1–19. DOI 10.1145/3417293.
19. Xu, X., Jiang, Q., Zhang, P., Cao, X., Khosravi, M. R. et al. (2022). Game theory for distributed IoV task offloading with fuzzy neural network in edge computing. *IEEE Transactions on Fuzzy Systems*. DOI 10.1109/TFUZZ.2022.3158000.
20. Liu, Z., Xu, X. (2022). Latency-aware service migration with decision theory for internet of vehicles in mobile edge computing. *Wireless Networks*, 1–13. DOI 10.1007/s11276-022-02978-y.
21. Ilić, M. D., Jaddivada, R., Miao, X. (2018). Scalable electric power system simulator. *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp. 1–6. Sarajevo.
22. Xin, Y., Zhang, B., Zhai, M., Li, Q., Zhou, H. (2018). A smarter grid operation: New energy management systems in China. *IEEE Power and Energy Magazine*, 16(2), 36–45. DOI 10.1109/MPE.2017.2779551.
23. Xu, X., Huang, Q., Yin, X., Abbasi, M., Khosravi, M. R. et al. (2020). Intelligent offloading for collaborative smart city services in edge computing. *IEEE Internet of Things Journal*, 7(9), 7919–7927. DOI 10.1109/JIOT.6488907.
24. Xu, X., Huang, Q., Zhu, H., Sharma, S., Zhang, X. et al. (2020). Secure service offloading for internet of vehicles in SDN-enabled mobile edge computing. *IEEE Transactions on Intelligent Transportation Systems*, 22(6), 3720–3729. DOI 10.1109/TITS.2020.3034197.
25. Guo, M., Li, Q., Peng, Z., Liu, X., Cui, D. (2022). Energy harvesting computation offloading game towards minimizing delay for mobile edge computing. *Computer Networks*, 204, 108678. DOI 10.1016/j.comnet.2021.108678.
26. Popov, M. G., Petrushin, D. E., Efimov, N. S. (2019). Increasing the emergency control systems efficiency in Kola's and Karelia's power systems. *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EICoN Rus)*, pp. 1040–1043. Saint Petersburg and Moscow.
27. Xu, X., Tian, H., Zhang, X., Qi, L., He, Q. et al. (2022). DisCOV: Distributed COVID-19 detection on X-ray images with edge-cloud collaboration. *IEEE Transactions on Services Computing*. DOI 10.1109/TSC.2022.3142265.
28. Mattoo, A., Meltzer, J. P. (2018). International data flows and privacy: The conflict and its resolution. *Journal of International Economic Law*, 21(4), 769–789. DOI 10.1093/jiel/jgy044.
29. Chong, H. Y., Diamantopoulos, A. (2020). Integrating advanced technologies to uphold security of payment: Data flow diagram. *Automation in Construction*, 114, 103158. DOI 10.1016/j.autcon.2020.103158.
30. Dai, H. N., Wong, R. C. W., Wang, H., Zheng, Z., Vasilakos, A. V. (2019). Big data analytics for large-scale wireless networks: Challenges and opportunities. *ACM Computing Surveys*, 52(5), 1–36. DOI 10.1145/3337065.

31. Ranjan, R., Rana, O., Nepal, S., Yousif, M., James, P. et al. (2018). The next grand challenges: Integrating the Internet of Things and data science. *IEEE Cloud Computing*, 5(3), 12–26. DOI 10.1109/MCC.2018.032591612.
32. Macedo, R., Paulo, J., Pereira, J., Bessani, A. (2020). A survey and classification of software-defined storage systems. *ACM Computing Surveys*, 53(3), 1–38. DOI 10.1145/3385896.
33. Ezhilarasi, T. P., Sudheer Kumar, N., Latchoumi, T. P., Balayesu, N. (2021). A secure data sharing using IDSS CP-ABE in cloud storage. *Advances in Industrial Automation and Smart Manufacturing*, pp. 1073–1085. Springer, Singapore.
34. Liu, B., Zhang, Y., Zhang, G., Zheng, P. (2019). Edge-cloud orchestration driven industrial smart product-service systems solution design based on CPS and IIoT. *Advanced Engineering Informatics*, 42, 100984. DOI 10.1016/j.aei.2019.100984.
35. Yuan, L., He, Q., Chen, F., Zhang, J., Qi, L. et al. (2021). CSEdge: Enabling collaborative edge storage for multi-access edge computing based on blockchain. *IEEE Transactions on Parallel and Distributed Systems*, 33(8), 1873–1887. DOI 10.1109/TPDS.2021.3131680.
36. Wang, Y., Cheng, S., Zhang, X., Leng, J., Liu, J. (2021). Block storage optimization and parallel data processing and analysis of product big data based on the hadoop platform. *Mathematical Problems in Engineering*, 2021. DOI 10.1155/2021/3839800.
37. Xu, X., Fang, Z., Zhang, J., He, Q., Yu, D. et al. (2021). Edge content caching with deep spatiotemporal residual network for IoV in smart city. *ACM Transactions on Sensor Networks*, 17(3), 1–33. DOI 10.1145/3447032.
38. Pasetti, M., Ferrari, P., Bellagente, P., Sisinni, E., de Sá, A. O. et al. (2021). Artificial neural network-based stealth attack on battery energy storage systems. *IEEE Transactions on Smart Grid*, 12(6), 5310–5321. DOI 10.1109/TSG.2021.3102833.
39. Xu, X., Fang, Z., Qi, L., Zhang, X., He, Q. et al. (2021). Tripres: Traffic flow prediction driven resource reservation for multimedia IoV with edge computing. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 17(2), 1–21. DOI 10.1145/3401979.
40. Apostol, G. C., Borcea, L., Dobre, C., Mavromoustakis, C. X., Mastorakis, G. (2021). A survey on privacy enhancements for massively scalable storage systems in public cloud environments. *Big Data Platforms and Applications*, pp. 207–223. Springer, Cham.
41. Yuan, Q., Zhou, H., Li, J., Liu, Z., Yang, F. et al. (2018). Toward efficient content delivery for automated driving services: An edge computing solution. *IEEE Network*, 32(1), 80–86. DOI 10.1109/MNET.2018.1700105.
42. Hagenauer, F., Sommer, C., Higuchi, T., Altintas, O., Dressler, F. (2017). Vehicular micro clouds as virtual edge servers for efficient data collection. *Proceedings of the 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services*, pp. 31–35. Snowbird.
43. Khan, M. A., Sargento, S., Luis, M. (2017). Data collection from smart-city sensors through large-scale urban vehicular networks. *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pp. 1–6. Sydney.
44. Turcanu, I., Klingler, F., Sommer, C., Baiocchi, A., Dressler, F. (2018). Duplicate suppression for efficient floating car data collection in heterogeneous LTE-DSRC vehicular networks. *Computer Communications*, 123, 54–64. DOI 10.1016/j.comcom.2018.03.015.
45. Ren, Y., Wang, T., Zhang, S., Zhang, J. (2020). An intelligent big data collection technology based on micro mobile data centers for crowdsensing vehicular sensor network. *Personal and Ubiquitous Computing*, 1–17. DOI 10.1007/s00779-020-01440-0.
46. Zhuang, Y., Ke, R., Wang, Y. (2020). Edge-based traffic flow data collection method using onboard monocular camera. *Journal of Transportation Engineering, Part A: Systems*, 146(9), 4020096. DOI 10.1061/JTEPBS.0000416.

47. Nie, W., Lee, V. C., Niyato, D., Duan, Y., Liu, K. et al. (2018). A quality-oriented data collection scheme in vehicular sensor networks. *IEEE Transactions on Vehicular Technology*, 67(7), 5570–5584. DOI 10.1109/TVT.25.
48. Rahman, S. A., Mourad, A., El Barachi, M., Al Orabi, W. (2018). A novel on-demand vehicular sensing framework for traffic condition monitoring. *Vehicular Communications*, 12, 165–178. DOI 10.1016/j.vehcom.2018.03.001.
49. Li, T., Tian, S., Liu, A., Liu, H., Pei, T. (2018). DDSV: Optimizing delay and delivery ratio for multimedia big data collection in mobile sensing vehicles. *IEEE Internet of Things Journal*, 5(5), 3474–3486. DOI 10.1109/JIoT.6488907.
50. Islam, A., Shin, S. Y. (2019). BUAV: A blockchain based secure UAV-assisted data acquisition scheme in Internet of Things. *Journal of Communications and Networks*, 21(5), 491–502. DOI 10.1109/JCN.5449605.
51. Kong, Q., Su, L., Ma, M. (2020). Achieving privacy-preserving and verifiable data sharing in vehicular fog with blockchain. *IEEE Transactions on Intelligent Transportation Systems*, 22(8), 4889–4898. DOI 10.1109/TITS.2020.2983466.
52. Jin, J., Song, A., Gong, H., Xue, Y., Du, M. et al. (2018). Distributed storage system for electric power data based on hbase. *Big Data Mining and Analytics*, 1(4), 324–334. DOI 10.26599/BDMA.2018.9020026.
53. Wang, J., Lin, C., Zaniolo, C. (2019). MF-join: Efficient fuzzy string similarity join with multi-level filtering. *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 386–397. Macao.
54. Sun, R., Shi, L., Yin, C., Wang, J. (2019). An improved method in deep packet inspection based on regular expression. *The Journal of Supercomputing*, 75(6), 3317–3333. DOI 10.1007/s11227-018-2517-0.
55. Satti, F. A., Khan, W. A., Lee, G., Khattak, A. M., Lee, S. (2019). Resolving data interoperability in ubiquitous health profile using semi-structured storage and processing. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 762–770. Limassol.
56. Suseendran, G., Balaganesh, D., Akila, D., Pal, S. (2021). Deep learning frequent pattern mining on static semi structured data streams for improving fast speed and complex data streams. *2021 7th International Conference on Optimization and Applications (ICOA)*, pp. 1–8. Wolfenbüttel.
57. Li, J. (2022). Influence of big data technology on the diversity of college physical education teaching methods. *Innovative Computing*, pp. 629–637. Springer, Singapore.
58. Yuan, L., He, Q., Tan, S., Li, B., Yu, J. et al. (2021). Coopedge: A decentralized blockchain-based platform for cooperative edge computing. *Proceedings of the Web Conference 2021*, pp. 2245–2257. Ljubljana.
59. Bazai, S. U., Jang-Jaccard, J., Zhang, X. (2017). A privacy preserving platform for MapReduce. *International Conference on Applications and Techniques in Information Security*, pp. 88–99. Singapore.
60. Zhang, X., Mu, J., Zhang, X., Liu, H., Zong, L. et al. (2022). Deep anomaly detection with self-supervised learning and adversarial training. *Pattern Recognition*, 121, 108234. DOI 10.1016/j.patcog.2021.108234.
61. Yan, W. J., Chronopoulos, D., Yuen, K. V., Zhu, Y. C. (2022). Structural anomaly detection based on probabilistic distance measures of transmissibility function and statistical threshold selection scheme. *Mechanical Systems and Signal Processing*, 162, 108009. DOI 10.1016/j.ymssp.2021.108009.
62. Feng, Y., Cai, W., Yue, H., Xu, J., Lin, Y. et al. (2022). An improved X-means and isolation forest based methodology for network traffic anomaly detection. *PLoS One*, 17(1), e0263423. DOI 10.1371/journal.pone.0263423.
63. Xiang, L., Yang, X., Hu, A., Su, H., Wang, P. (2022). Condition monitoring and anomaly detection of wind turbine based on cascaded and bidirectional deep learning networks. *Applied Energy*, 305, 117925. DOI 10.1016/j.apenergy.2021.117925.
64. Soldani, J., Brogi, A. (2022). Anomaly detection and failure root cause analysis in (micro)service-based cloud applications: A survey. *ACM Computing Surveys*, 55(3), 1–39.

65. Rashid, A. B., Ahmed, M., Sikos, L. F., Haskell-Dowland, P. (2022). Anomaly detection in cybersecurity datasets via cooperative co-evolution-based feature selection. *ACM Transactions on Management Information Systems*, 13(3), 1–39. DOI 10.1145/3495165.
66. Kou, H., Liu, H., Duan, Y., Gong, W., Xu, Y. et al. (2021). Building trust/distrust relationships on signed social service network through privacy-aware link prediction process. *Applied Soft Computing*, 100, 106942. DOI 10.1016/j.asoc.2020.106942.
67. Qi, L., He, Q., Chen, F., Dou, W., Wan, S. et al. (2019). Finding all you need: Web APIs recommendation in web of things through keywords search. *IEEE Transactions on Computational Social Systems*, 6(5), 1063–1072. DOI 10.1109/TCSS.6570650.
68. Li, R., Li, Y., He, W., Chen, L., Luo, J. (2021). Multi-layer reconstruction errors autoencoding and density estimate for network anomaly detection. *Computer Modeling in Engineering & Sciences*, 128(1), 381–397. DOI 10.32604/cmcs.2021.016264.
69. Huo, Y., Cao, Y., Wang, Z., Yan, Y., Ge, Z. et al. (2021). Traffic anomaly detection method based on improved GRU and EFMS-Kmeans clustering. *Computer Modeling in Engineering & Sciences*, 126(3), 1053–1091. DOI 10.32604/cmcs.2021.013045.
70. Zhu, G., Zhao, H., Liu, H., Sun, H. (2019). A novel LSTM-GAN algorithm for time series anomaly detection. *2019 Prognostics and System Health Management Conference*, pp. 1–6. Qingdao.
71. Patcha, A., Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12), 3448–3470. DOI 10.1016/j.comnet.2007.02.001.
72. Li, F., Mi, H., Yang, F. (2011). Exploring the stability of feature selection for imbalanced intrusion detection data. *2011 9th IEEE International Conference on Control and Automation (ICCA)*, pp. 750–754. Santiago.
73. Dong, L. I., Liu, S. L., Zhang, H. (2017). A method of anomaly detection and fault diagnosis with online adaptive learning under small training samples. *Pattern Recognition*, 64, 374–385. DOI 10.1016/j.patcog.2016.11.026.
74. Yao, J., Zhang, J., Wang, L. (2018). A financial statement fraud detection model based on hybrid data mining methods. *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 57–61. Chengdu, China.
75. Li, D., Chen, D., Goh, J., Ng, S. K. (2018). Anomaly detection with generative adversarial networks for multivariate time series. arXiv preprint arXiv:1809.04758.
76. Zhang, Y., Li, M., Dong, Z. Y., Meng, K. (2019). Probabilistic anomaly detection approach for data-driven wind turbine condition monitoring. *CSEE Journal of Power and Energy Systems*, 5(2), 149–158. DOI 10.17775/CSEEJPES.
77. Sun, C., Chen, S., E, M. Z., Du, Y., Ruan, C. (2020). Satellite micro anomaly detection based on telemetry data. *2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 140–144. Liuzhou, China.
78. Guo, Y., Ji, T., Wang, Q., Yu, L., Min, G. et al. (2020). Unsupervised anomaly detection in IoT systems for smart cities. *IEEE Transactions on Network Science and Engineering*, 7(4), 2231–2242. DOI 10.1109/TNSE.6488902.
79. Wang, Y., Masoud, N., Khojandi, A. (2020). Real-time sensor anomaly detection and recovery in connected automated vehicle sensors. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1411–1421. DOI 10.1109/TITS.6979.
80. Theissler, A. (2017). Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection. *Knowledge-Based Systems*, 123, 163–173. DOI 10.1016/j.knosys.2017.02.023.
81. Barford, P., Kline, J., Plonka, D., Ron, A. (2002). A signal analysis of network traffic anomalies. *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurement*, pp. 71–82. Marseille.

82. Song, Y., Cao, L., Yin, J., Wang, C. (2013). Extracting discriminative features for identifying abnormal sequences in one-class mode. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. Dallas.
83. Yadav, A., Dash, Y. (2014). An overview of transmission line protection by artificial neural network: Fault detection, fault classification, fault location, and fault direction discrimination. *Advances in Artificial Neural Systems, 2014*. DOI 10.1155/2014/230382.
84. Zheng, L., Guang, J., Tang, S. H. (2016). Fluctuation feature extraction of satellite telemetry data and on-orbit anomaly detection. *2016 Prognostics and System Health Management Conference*, pp. 1–5. Chengdu, China.
85. Escobar, M. S., Kaneko, H., Funatsu, K. (2017). On generative topographic mapping and graph theory combined approach for unsupervised non-linear data visualization and fault identification. *Computers & Chemical Engineering, 98*, 113–127. DOI 10.1016/j.compchemeng.2016.12.009.
86. Onal, A. C., Sezer, O. B., Ozbayoglu, M., Dogdu, E. (2017). Weather data analysis and sensor fault detection using an extended IoT framework with semantics, big data, and machine learning. *2017 IEEE International Conference on Big Data*, pp. 2037–2046. Boston.
87. Du, Y., Liang, X., Wang, F., Sun, C., Hua, X. (2018). Anomaly detection of satellite telemetry in orbit based on sequence and point feature combination. *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 872–878. Enshi.
88. Chen, L., Xu, G., Zhang, Q., Zhang, X. (2019). Learning deep representation of imbalanced SCADA data for fault detection of wind turbines. *Measurement, 139*, 370–379. DOI 10.1016/j.measurement.2019.03.029.
89. Chalapathy, R., Chawla, S. (2019). Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407.
90. Park, P., Marco, P. D., Shin, H., Bang, J. (2019). Fault detection and diagnosis using combined autoencoder and long short-term memory network. *Sensors, 19(21)*, 4612. DOI 10.3390/s19214612.
91. Zhou, F., Yang, S., Fujita, H., Chen, D., Wen, C. (2020). Deep learning fault diagnosis method based on global optimization GAN for unbalanced data. *Knowledge-Based Systems, 187*, 104837. DOI 10.1016/j.knosys.2019.07.008.
92. Yang, A., Wu, M., Hu, J., Chen, L., Lu, C. et al. (2021). Discrimination and correction of abnormal data for condition monitoring of drilling process. *Neurocomputing, 433*, 275–286. DOI 10.1016/j.neucom.2020.11.064.
93. Raichura, M., Chothani, N., Patel, D. (2021). Efficient CNN-XGBoost technique for classification of power transformer internal faults against various abnormal conditions. *IET Generation, Transmission & Distribution, (5)*, 972–985. DOI 10.1049/gtd2.12073.
94. Yang, T., Hou, Y., Liu, Y., Zhai, F., Niu, R. (2021). WPD-Resnest: Substation station level network anomaly traffic detection based on deep transfer learning. *CSEE Journal of Power and Energy Systems*, 1–12.
95. Astillo, P. V., Duguma, D. G., Park, H., Kim, J., Kim, B. et al. (2022). Federated intelligence of anomaly detection agent in IoTMD-enabled diabetes management control system. *Future Generation Computer Systems, 128*, 395–405. DOI 10.1016/j.future.2021.10.023.
96. Wang, Z., Tian, J., Fang, H., Chen, L., Qin, J. (2022). LightLog: A lightweight temporal convolutional network for log anomaly detection on the edge. *Computer Networks, 203*, 108616. DOI 10.1016/j.comnet.2021.108616.
97. Sunny, J. S., Patro, C. P. K., Karnani, K., Pingle, S. C., Lin, F. et al. (2022). Anomaly detection framework for wearables data: A perspective review on data concepts, data analysis algorithms and prospects. *Sensors, 22(3)*, 756.
98. Saqaeyan, S., Javadi, H. H. S., Amirkhani, H. (2019). A novel probabilistic hybrid model to detect anomaly in smart homes. *Computer Modeling in Engineering & Sciences, 121(3)*, 815–834. DOI 10.32604/cmcs.2019.07848.

99. Nachman, B. (2022). Anomaly detection for physics analysis and less than supervised learning. *Artificial Intelligence for High Energy Physics*, 85–112. DOI 10.1142/12200.
100. Chen, A., Fu, Y., Zheng, X., Lu, G. (2022). An efficient network behavior anomaly detection using a hybrid DBN-LSTM network. *Computers & Security*, 114, 102600. DOI 10.1016/j.cose.2021.102600.
101. Ullah, I., Mahmoud, Q. H. (2022). An anomaly detection model for IoT networks based on flow and flag features using a feed-forward neural network. *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 363–368. Las Vegas.
102. Malik, M. K., Singh, A., Swaroop, A. (2022). A planned scheduling process of cloud computing by an effective job allocation and fault-tolerant mechanism. *Journal of Ambient Intelligence and Humanized Computing*, 13, 1153–1171. DOI 10.1007/s12652-021-03537-7.
103. Li, C., Liu, J., Wang, M., Luo, Y. (2022). Fault-tolerant scheduling and data placement for scientific workflow processing in geo-distributed clouds. *Journal of Systems and Software*, 187, 111227. DOI 10.1016/j.jss.2022.111227.
104. Shaukat, M., Alasmay, W., Alanazi, E., Shuja, J., Madani, S. A. et al. (2022). Balanced energy-aware and fault-tolerant data center scheduling. *Sensors*, 22(4), 1482. DOI 10.3390/s22041482.
105. Kumar T, S., M, H. S., Mustapha, S. M. F. D., Gupta, P., Tripathi, R. P. (2022). Intelligent fault-tolerant mechanism for data centers of cloud infrastructure. *Mathematical Problems in Engineering*, 2022. DOI 10.1155/2022/2379643.
106. Sahu, S., Silakari, S. (2022). Distributed multilevel K-coverage energy-efficient fault-tolerant scheduling for wireless sensor networks. *Wireless Personal Communications*, 1–30. DOI 10.1007/s11277-022-09495-3.
107. Feng, X., Ma, J., Liu, S., Miao, Y., Liu, X. (2022). Auto-scalable and fault-tolerant load balancing mechanism for cloud computing based on the proof-of-work election. *Science China Information Sciences*, 65(1), 1–16. DOI 10.1007/s11432-015-5506-4.
108. Li, L., Yuan, Y., Zhang, X., Wu, S., Zhang, T. (2022). Fault-tolerant control scheme for the sensor fault in the acceleration process of variable cycle engine. *Applied Sciences*, 12(4), 2085. DOI 10.3390/app12042085.
109. Fu, X., Zhao, Q. (2022). Fault-tolerant tracking control base on iterative learning for nonlinear NCSs with time-delay and data loss failure in dual-channel. *Journal of Control Engineering and Applied Informatics*, 24(1), 26–36.
110. Chen, J., Chen, B., Zeng, Z. (2022). Adaptive dynamic event-triggered fault-tolerant consensus for nonlinear multiagent systems with directed/undirected networks. *IEEE Transactions on Cybernetics*. DOI 10.1109/TCYB.2022.3151653.
111. Vinay, K., Kumar, S. D. (2017). Fault-tolerant scheduling for scientific workflows in cloud environments. *2017 IEEE 7th International Advance Computing Conference (IACC)*, pp. 150–155. Hyderabad.
112. Pandita, A., Upadhyay, P. K., Joshi, N. (2018). Fault tolerance based comparative analysis of scheduling algorithms in cloud computing. *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, pp. 1–6. Kottayam.
113. Nair, P. P., Devaraj, R., Sarkar, A. (2018). Fest: Fault-tolerant energy-aware scheduling on two-core heterogeneous platform. *2018 8th International Symposium on Embedded Computing and System Design (ISED)*, pp. 63–68. Cochin.
114. Yu, S., Tang, Z., Ye, X., Zhang, Z., Fan, D. et al. (2018). High-performance and energy-efficient fault tolerance scheduling algorithm based on improved TMR for heterogeneous system. *2018 IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pp. 207–214. Melbourne.
115. Lee, J., Gil, J. (2019). Adaptive fault-tolerant scheduling strategies for mobile cloud computing. *The Journal of Supercomputing*, 75(8), 4472–4488. DOI 10.1007/s11227-019-02745-5.

116. Chinnathambi, S., Santhanam, A., Rajarathinam, J., Senthilkumar, M. (2019). Scheduling and checkpointing optimization algorithm for byzantine fault tolerance in cloud clusters. *Cluster Computing*, 22(6), 14637–14650. DOI 10.1007/s10586-018-2375-9.
117. Abdulhamid, S. M. Latiff, M. S. A. (2017). A checkpointed league championship algorithm-based cloud scheduling scheme with secure fault tolerance responsiveness. *Applied Soft Computing*, 61, 670–680. DOI 10.1016/j.asoc.2017.08.048.
118. Yao, G., Ding, Y., Hao, K. (2017). Using imbalance characteristic for fault-tolerant workflow scheduling in cloud systems. *IEEE Transactions on Parallel and Distributed Systems*, 28(12), 3671–3683. DOI 10.1109/TPDS.2017.2687923.
119. Abd Latiff, M. S., Madni, S. H. H., Abdullahi, M. (2018). Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm. *Neural Computing and Applications*, 29(1), 279–293. DOI 10.1007/s00521-016-2448-8.
120. Cao, K., Xu, G., Zhou, J., Chen, M., Wei, T. et al. (2019). Lifetime-aware real-time task scheduling on fault-tolerant mixed-criticality embedded systems. *Future Generation Computer Systems*, 100, 165–175. DOI 10.1016/j.future.2019.05.022.
121. Ahrar, E. M., Nassiri, M., Theoleyre, F. (2019). Multipath aware scheduling for high reliability and fault tolerance in low power industrial networks. *Journal of Network and Computer Applications*, 142, 25–36. DOI 10.1016/j.jnca.2019.05.013.
122. Soniya, J., Sujana, J. A. J., Revathi, T. (2016). Dynamic fault tolerant scheduling mechanism for real time tasks in cloud computing. *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 124–129. Chennai.
123. Zhu, X., Wang, J., Guo, H., Zhu, D., Yang, L. T. et al. (2016). Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds. *IEEE Transactions on Parallel and Distributed Systems*, 27(12), 3501–3517. DOI 10.1109/TPDS.2016.2543731.
124. Ding, Y., Yao, G., Hao, K. (2017). Fault-tolerant elastic scheduling algorithm for workflow in cloud systems. *Information Sciences*, 393, 47–65. DOI 10.1016/j.ins.2017.01.035.
125. Yan, H., Zhu, X., Chen, H., Guo, H., Zhou, W. et al. (2019). DEFT: Dynamic fault-tolerant elastic scheduling for tasks with uncertain runtime in cloud. *Information Sciences*, 477, 30–46. DOI 10.1016/j.ins.2018.10.020.
126. Marahatta, A., Wang, Y., Zhang, F., Sangaiah, A. K., Tyagi, S. K. S. et al. (2019). Energy-aware fault-tolerant dynamic task scheduling scheme for virtualized cloud data centers. *Mobile Networks and Applications*, 24(3), 1063–1077. DOI 10.1007/s11036-018-1062-7.
127. Chen, G., Guan, N., Huang, K., Yi, W. (2020). Fault-tolerant real-time tasks scheduling with dynamic fault handling. *Journal of Systems Architecture*, 102, 101688. DOI 10.1016/j.sysarc.2019.101688.
128. Zhang, Y., Zhang, H. (2016). Low power fault tolerance scheduling scheme in DVS-enabled real-time systems. *International Journal of High Performance Systems Architecture*, 6(2), 110–117. DOI 10.1504/IJHPSA.2016.078820.
129. Idris, H., Ezugwu, A. E., Junaidu, S. B., Adewumi, A. O. (2017). An improved ant colony optimization algorithm with fault tolerance for job scheduling in grid computing systems. *PLoS One*, 12(5), e0177567. DOI 10.1371/journal.pone.0177567.
130. Zhou, J., Yin, M., Li, Z., Cao, K., Yan, J. et al. (2017). Fault-tolerant task scheduling for mixed-criticality real-time systems. *Journal of Circuits, Systems and Computers*, 26(1), 1750016. DOI 10.1142/S0218126617500165.
131. Wang, L., Shang, L., Ma, M., Ma, Z. (2018). Fault diagnosis and trace method of power system based on big data platform. *IOP Conference Series: Materials Science and Engineering*, 394(4), 42116. DOI 10.1088/1757-899X/394/4/042116.
132. Liu, J., Wei, M., Hu, W., Xu, X., Ouyang, A. (2018). Task scheduling with fault-tolerance in real-time heterogeneous systems. *Journal of Systems Architecture*, 90, 23–33. DOI 10.1016/j.sysarc.2018.08.007.

133. Alarifi, A., Abdelsamie, F., Amoon, M. (2019). A fault-tolerant aware scheduling method for fog-cloud environments. *PLoS one*, 14(10), e0223902. DOI 10.1371/journal.pone.0223902.
134. Talwani, S., Singla, J. (2019). Comparison of various fault tolerance techniques for scientific workflows in cloud computing. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 454–459. Faridabad.
135. Khaldi, M., Rebbah, M., Meftah, B., Smail, O. (2020). Fault tolerance for a scientific workflow system in a cloud computing environment. *International Journal of Computers and Applications*, 42(7), 705–714. DOI 10.1080/1206212X.2019.1647651.
136. Varma, P. S., Anand, V. (2022). Fault-tolerant indoor localization based on speed conscious recurrent neural network using Kullback–Leibler divergence. *Peer-to-Peer Networking and Applications*, 15(3), 1370–1384. DOI 10.1007/s12083-022-01301-y.
137. Yaser, R., Mirsaeid, H. S., Mehdi, G. (2022). Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure. *Complex & Intelligent Systems*, 8(1), 361–392. DOI 10.1007/s40747-021-00368-z.
138. Abdulrab, H., Hussin, F. A., Abd Aziz, A., Awang, A., Ismail, I. et al. (2022). Reliable fault tolerant-based multipath routing model for industrial wireless control systems. *Applied Sciences*, 12(2), 544. DOI 10.3390/app12020544.
139. Morell, J. Á., Alba, E. (2022). Dynamic and adaptive fault-tolerant asynchronous federated learning using volunteer edge devices. *Future Generation Computer Systems*, 133, 53–67. DOI 10.1016/j.future.2022.02.024.
140. Martinez, I., Hafid, A. S., Gendreau, M. (2022). Robust and fault-tolerant fog design & dimensioning for reliable operation. *IEEE Internet of Things Journal*. DOI 10.1109/JIOT.2022.3157557.
141. Aldağ, M., Kırsal, Y., Ülker, S. (2022). An analytical modelling and QoS evaluation of fault-tolerant load balancer and web servers in fog computing. *The Journal of Supercomputing*, 1–23. DOI 10.1007/s11227-022-04345-2.