Real-Time Transmission Line Fault Monitoring Using SVRG-DDPG and Lightweight Edge Computing

MengHao Lin*, Yang Ding, JinDa Lu, WeiYi Chen, TianLe Wang
State Grid Xinjiang Company Limited Electric Power Research Institute, Urumqi 830000, China
E-mail: MenghaoLin1996@outlook.com
*Corresponding author

Keywords: SVRG-DDPG, edge computing, deep deterministic policy gradient algorithm, transmission lines, fault monitoring

Received: January 13, 2026

In power system intelligent fault detection, real-time monitoring is critical due to grid complexity. To address the challenges posed by the complexity of continuous action selection and gradient estimation errors in transmission line fault monitoring, we developed a Deep Deterministic Policy Gradient (DDPG) algorithm enhanced by the Stochastic Variance Reduction Gradient (SVRG) method, termed SVRG-DDPG. This algorithm leverages the SVRG technique to mitigate the gradient estimation errors typically encountered in the DDPG algorithm. Utilizing the SVRG-DDPG framework, we further developed a transmission line fault monitoring model that directly employs real-time voltage sensor data from actual transmission line environments, encompassing a range of state information from normal operation to fault conditions. To achieve real-time monitoring and optimize system performance, we also propose a transmission line fault monitoring system based on a lightweight edge computing architecture. Using real-time voltage sensor data, the SVRG-DDPG-based fault monitoring model achieves a residual error within 30 kV accuracy. To enable real-time fault diagnosis in resource-constrained edge environments, we propose a lightweight edge-cloud collaborative architecture that dynamically allocates computational resources based on fault severity and sensor data volume. The framework is validated using high-fidelity simulation data from a power grid (covering 8 fault types under 12 operational conditions, e.g., humidity >90%, load fluctuations ±40%), which aligns with the dynamic resource demands of edge devices in practical systems. Finally, our method achieves 92% accuracy in fault diagnosis with 42% lower latency compared to baselines, leveraging SVRG-enhanced DDPG for adaptive edge-cloud synchronization. Experimental results on real-world 5G-V2X data validate its suitability for low-latency transmission line monitoring.

Povzetek: Da bi omogočili diagnosticiranje napak v realnem času v robnih okoljih z omejenimi viri, predlagamo lahkotno sodelovalno arhitekturo robnega omrežja v oblaku, ki dinamično dodeljuje računalniške vire glede na resnost napake in količino podatkov senzorjev.

1 Introduction

As the cornerstone of national energy and economic stability, the power system plays a crucial role in the sustainable development of the national economy. With the continuous expansion of grid infrastructure and the increasing demand on power loads, transmission lines are now exposed to more complex and variable operating environments, leading to a heightened probability of failure [1]. Ensuring the safe and reliable operation of the power grid has therefore become an urgent challenge. Traditional methods for investigating transmission line faults typically rely on manual inspections and the analysis of fault indicators, such as monitoring current, voltage, and other parameters based on changes in physical quantities and signal processing techniques. One common approach involves using differential protection devices to detect short-circuit faults in the grid by comparing current values at different points along the line. Alternatively, over- and under-frequency protection devices are employed to monitor frequency anomalies within the grid to identify potential faults. While these classical methods depend heavily on accurate monitoring and parameterization of the grid's operational conditions, they are constrained by their ability to precisely identify fault characteristics. This approach is not only inefficient but also makes it difficult to detect and address faults promptly. In regions with complex terrain and harsh climates, manual inspections become even more challenging and costly.

Smart grid construction represents a key direction for the advancement of the current power industry. By integrating various advanced technologies, smart grids enable comprehensive, all-weather monitoring and management of power systems. The development and application of real-time transmission line fault monitoring models, as a critical component of smart grids, are of great significance in enhancing the automation and intelligence of power grid operations. Presently, cloud computing technology can be applied to smart grid fault monitoring; however, it requires substantial computing resources. By integrating artificial intelligence with cloud computing, smart grid systems can leverage the benefits of cloud-based technologies [2]. Cloud computing offers enhanced computational power and broader data communication capabilities for fault monitoring, utilizing Internet connections or TCP/IP protocols for devices deployed on external platforms. Given the variety of service types and deployment models available in cloud computing, developers can augment the capabilities of traditional smart grid systems by selecting the most appropriate model. However, purely cloud-based architectures face challenges such as high data volumes and transmission latency. For instance, when monitoring the power consumption of smart grid devices to identify transmission line anomalies, vast amounts of data may need to be collected, which can overwhelm cloud-based solutions. Additionally, communication between the target device and the cloud server over the Internet introduces significant dynamic transmission delays, and the delay jitter cannot be precisely estimated, potentially causing the monitoring and transmission delay to exceed acceptable time thresholds.

Edge computing [3] has emerged in recent years as a promising approach to alleviate the burden on cloud computing and significantly reduce transmission and monitoring latency. Unlike cloud computing, edge computing enables real-time or near real-time data processing and analysis by performing these tasks closer to the data source. However, embedded edge devices are often limited in computing resources and may not be able to handle all data monitoring tasks. While cloud computing is typically associated with centralized data processing, edge computing refers to a geographically distributed model that is located near the network's end devices. With the rapid advancement of information technology, including the emergence of edge computing and adaptive reasoning, intelligent monitoring of power systems has become feasible. Edge computing reduces data transmission delays and network bandwidth consumption by processing and analyzing data on edge devices close to the source. Adaptive reasoning algorithms, such as reinforcement learning [4], genetic algorithms [5], and deep reinforcement learning [6], dynamically adjust model parameters based on real-time transmission line data, thereby enhancing the accuracy and efficiency of fault monitoring. However, in these adaptive algorithms, the selection of input features and the optimization strategies used during algorithm iterations can result in varying output outcomes. In edge computing, the use of high-performance, low-power edge computing chips ensures that real-time data processing and analysis occur efficiently at the edge, while also minimizing energy consumption, though this may increase the complexity of the models employed.

In transmission line fault detection, the diversity of fault modes and the nonlinear characteristics of signals make it difficult for traditional rule-based or model-based methods to achieve ideal detection results. Reinforcement learning, especially continuous-action reinforcement learning, can dynamically adapt to different fault conditions through its ability to learn from interactions with the environment, without the need to predefine all possible state and action models. The continuous action space further enhances the model's flexibility, enabling it to adjust detection parameters more precisely and thus achieve efficient and accurate fault detection in complex and variable fault environments.

Deep Reinforcement Learning (DRL) suffers training instability from gradient errors, especially in continuous action spaces. Edge AI offers real-time potential but is constrained by edge devices' limited compute and bandwidth, hindering centralized DRL deployment. This study innovatively merges stochastic variance-reduced gradients into DRL, proposing SVRG-DDPG for robust training, and designs a lightweight edge architecture to optimize resources. The goal is a high-precision, low-latency monitoring paradigm, addressing gaps in complex grid environments. The specific contributions of this paper are as follows:

(1) Development of the SVRG-DDPG (Stochastic Variance Reduced Gradient-Deep Deterministic Policy Gradient) algorithm: This study innovatively proposes the SVRG-DDPG algorithm, which effectively addresses the gradient estimation error issue in the DDPG algorithm by integrating the Stochastic Variance Reduced Gradient (SVRG) method. The SVRG technique utilizes mini-batch random training sample sets to more accurately estimate the gradient direction, thereby achieving faster convergence rates and higher stability during the continuous action selection process. Compared to the traditional DDPG algorithm, the SVRG-DDPG algorithm offers greater precision in gradient estimation, significantly enhancing the accuracy and robustness of the fault monitoring model.

(2)Construction of a transmission line fault monitoring model based on SVRG-DDPG: It directly utilizes real-time voltage sensor data from actual transmission line environments for training. This model not only encompasses comprehensive information ranging from normal operation to various fault states but also achieves precise identification and rapid response to complex fault scenarios through the optimization of the SVRG-DDPG algorithm. Compared to traditional fault monitoring models, this model demonstrates significant improvements in both the accuracy and real-time performance of fault identification.

(3)Proposal of a lightweight edge computing-based transmission line fault monitoring system: This system utilizes embedded devices positioned close to the monitored equipment's edge to achieve real-time data monitoring. It incorporates the SVRG-DDPG transmission line fault monitoring model to accelerate solution times, maximize system throughput, and enhance communication efficiency and resource utilization within the system. Compared with traditional cloud computing and pure edge computing solutions, this architecture excels in both resource allocation and system performance, providing robust support for real-time fault monitoring.

The integration of SVRG into DDPG, forming the SVRG-DDPG algorithm as shown in Table 1, represents a significant methodological advancement for transmission line fault monitoring. Beyond merely improving convergence rates, SVRG-DDPG addresses the critical challenge of gradient estimation errors inherent in DDPG, particularly in high-dimensional and continuous action spaces. This reduction in gradient variance enables more precise and stable fault detection, even under dynamic operational conditions, thereby enhancing the overall reliability and accuracy of the monitoring system.

In this paper, we first present an overview of the current state of research on transmission line fault monitoring within smart grids, alongside the application of adaptive reasoning algorithms and edge computing in fault monitoring, in Section 2. Section 3 details the construction of the SVRG-DDPG algorithm developed in this study, the transmission line fault monitoring model based on SVRG-DDPG, and the integration of lightweight edge computing into the SVRG-DDPG-based fault monitoring model. In Section 4, we describe the experimental results, discussing the performance of the SVRG-DDPG algorithm and the transmission line fault monitoring model that incorporates both SVRG-DDPG and edge computing. This section also provides a comparative analysis with existing algorithms, including DDPG, cloud computing, and edge computing, to evaluate the impact of SVRG-DDPG and lightweight edge computing on transmission line fault monitoring performance. Finally, Section 5 offers a summary, discussing the overall performance of the SVRG-DDPG and edge computing-based transmission line fault monitoring model proposed in this paper and suggesting directions for future research.

Table 1: Comparison of relevant literature

Method	Fault Type	Edge-Cloud Support	Key Limitations
Traveling wave-based fault location[7]	Transmission line faults	No	Requires dual-ended synchronized measurement and GPS clock modules; hardware cost is high, and performance degrades in multi-terminal grids.
DDPG-based protection strategy[8]	Multiple fault conditions	Yes	Slow convergence under non-stationary operating conditions; state representation may lag under abrupt photovoltaic output changes.
Attention-based graph neural network[9]	Fault classification	Partial	High model complexity and inference latency; static graph structure cannot capture real-time topology variation.
Edge-computing fault diagnosis framework[10]	General fault diagnosis	Yes	Validation is mainly limited to ideal simulation environments, with insufficient consideration of sensor distortion and communication packet loss.
Transformer-based temporal feature extraction[11]	Fault transient analysis	Partial	Quadratic computational complexity leads to excessive parameter size and memory usage, limiting low-latency edge deployment.
SVRG-DDPG (Ours)	Multiple fault conditions	Yes	Adaptive policy learning with variance-reduced gradient optimization; designed for lightweight edge-cloud collaborative monitoring.

2 Related works

Fault monitoring in transmission lines within smart grids has long been a critical focus for the power industry, with numerous methods proposed to address this challenge. Recently, fault monitoring in smart grids based on artificial intelligence (AI) has achieved significant success. For instance, a novel approach utilizing the Choi-Williams time-frequency distribution, combined with convolutional neural networks and a task decomposition framework, was introduced in the literature [12]. To enhance the recognition capabilities of power line fault monitoring, literature [13] proposed a power line fault recognition modeling method based on sparse autoencoder neural networks. This method employs pre-training analysis and fault signal feature modeling, using a deep neural network as the fault monitoring classifier, and simulation experiments have demonstrated a fault recognition rate exceeding 99%. Additionally, literature [14] presented an effective algorithm based on artificial neural networks for asymmetric fault monitoring, fault type classification, and fault zone localization in transmission lines. Another approach, outlined in literature [15], utilized discrete wavelet transform in conjunction with artificial neural networks for fault monitoring and classification, leveraging pre- and post-fault signals of three-phase currents and ground current signals at the transmission line's transmitter end. In the context of advancing AI techniques, literature [16] proposed a real-time, high-precision modular multilevel converter monitoring system based on adaptive one-dimensional convolutional neural networks for early fault detection and identification, directly applicable to raw voltage and current data without requiring any feature extraction algorithms.

As adaptive reasoning algorithms have matured, reinforcement learning [17] has emerged as a key area of interest among experts and scholars. While traditional reinforcement learning has performed well in scenarios involving relatively simple models, the majority of real-world problems are highly complex, rendering the construction of simple problem models insufficient. In recent years, deep learning technologies have spurred advancements in reinforcement learning, notably with the Google DeepMind team's development of the Deep Reinforcement Learning (DRL) [18] algorithm, which combines convolutional neural networks with Q-learning [19] to create the Deep Q-Network (DQN) [20]. However, the DQN algorithm's tendency to select the same values for action evaluation during optimization can lead to biased or noisy conditions, resulting in action selection errors and over-fault monitoring biases. To address these issues, literature [21] proposed the Deep Double Q-Network (DDQN) [22]. The DDQN algorithm utilizes two separate networks—the current value network and the target value network—to distinguish between action selection and action evaluation, effectively mitigating the risks associated with overestimation.

Researchers are increasingly applying edge computing to smart grid fault monitoring to decentralize computation and improve system efficiency. For instance, literature [23] introduced an innovative fault localization strategy that incorporates edge computing technology, significantly improving the accuracy and efficiency of fault identification in both uniform-feeder and mixed-feeder distribution networks. Similarly, literature [24] proposed an edge computing architecture tailored for real-time smart grid monitoring, where computational tasks traditionally handled by centralized cloud servers are offloaded to edge servers located closer to the devices. This approach markedly enhances the immediacy and precision of monitoring.

From a comprehensive perspective, in the field of integrating industrial fault monitoring with edge intelligence, recent research has focused on addressing challenges related to data privacy, communication latency, and model generalization. For instance, literature [25] proposes a real-time fault prediction framework based on edge computing and federated learning. By deploying lightweight LSTM networks on edge devices, it achieves localized feature extraction of vibration signals and differential privacy protection, thereby avoiding the upload of raw data. Meanwhile, a dynamic weight aggregation algorithm is designed to dynamically adjust the contribution of model updates based on the data quality of edge nodes, resulting in a 23% improvement in the F1-score of the global model in cross-factory scenarios. Literature [26], addressing the limited computational capacity of edge devices, proposes a knowledge distillation-enhanced edge-cloud collaborative inference mechanism. This approach employs a two-stage knowledge distillation framework. First, a cloud-based teacher model generates a compressed knowledge graph by abstracting high-level fault patterns from historical monitoring data. Then, an edge-deployed student model utilizes this structured knowledge to perform real-time fault classification. Experiments demonstrate that this method achieves a 1.8-fold increase in inference speed and a 41% reduction in energy consumption on a Raspberry Pi 4B compared to traditional methods. These studies provide implementable technical pathways for empowering industrial safety through edge intelligence. However, despite the significant advantages these methods offer in improving fault monitoring efficiency and real-time performance, their complexity often hinders the realization of lightweight edge computing. This challenge limits resource utilization efficiency, particularly in high-precision fault monitoring scenarios, where unnecessary resource redundancy and waste may occur.

Significant progress has been made in the integration of deep learning and edge computing in existing research on power system fault diagnosis. However, there are still systematic limitations in its methodology, making it difficult to meet the stringent requirements of new-type power systems for real-time performance, dynamic adaptability, and hardware friendliness. Although classical location techniques represented by the traveling wave method [7] control the location error within the range of ±150 m through S-transform, their dual-ended synchronous measurement architecture requires the deployment of GPS clock modules, increasing hardware costs by 300% compared to traditional impedance methods. Moreover, in the multi-terminal topology of flexible DC grids, the complexity of traveling wave refraction/reflection paths leads to a 27% increase in location failure rates. Protection strategies based on deep reinforcement learning [8] achieve dynamic adjustment of protection settings through the DDPG algorithm, but their priority experience replay mechanism suffers from a lag in state space representation when dealing with step changes in photovoltaic output. Although the attention mechanism of graph neural networks [9] improves the fault classification F1 score to over 96%, the model contains 8.2 million trainable parameters, requiring 112 ms for a single inference on an NVIDIA Jetson AGX Xavier edge device, far exceeding the 50 ms protection action time limit specified by the IEC 61850 standard. Additionally, its static graph structure fails to capture real-time dynamic changes in power grid topology. A more widespread issue is that current edge computing fault diagnosis research [10] only validates algorithm effectiveness in ideal simulation environments without considering non-ideal factors such as sensor sampling distortion and communication packet loss in real-world industrial scenarios, leading to a 3-5 fold increase in false alarm rates during industrial model deployment. Although the Transformer-based temporal feature extraction method [11] captures fault transient features through self-attention mechanisms, its quadratic complexity results in a model parameter count of 14.5 million, occupying over 90% of memory on edge devices and necessitating reliance on cloud collaboration for inference, which contradicts the core requirement of low-latency local decision-making for power system fault diagnosis. Furthermore, traditional methods employ fixed thresholds for fault discrimination. In scenarios where the "dual-high" characteristics of new-type power systems weaken fault transient features, both the failure rate of wave head detection in the traveling wave method and the misjudgment rate of wavelet transform modulus maxima exhibit exponential growth trends.

Recent advancements FCA-VBN [27] and L-CPPA [28] demonstrate how Chebyshev polynomials and lattice-based schemes reduce authentication latency in 5G-assisted fog computing. These studies inspire our integration of variance-reduced DRL with post-quantum secure authentication, ensuring real-time performance under adversarial conditions.

3 Model design

In transmission line fault monitoring, achieving real-time detection under dynamic operational conditions—such as frequent load fluctuations and transient disturbances in high-voltage networks—requires reinforcement learning models capable of generating precise continuous control actions, like adaptive voltage threshold adjustments for fault isolation. In edge computing environments, traditional DDPG suffers from high variance in policy gradients due to noisy, non-stationary data, leading to slow convergence and unstable training. Our SVRG-DDPG addresses this by integrating Stochastic Variance-Reduced Gradient into the critic network, which reduces gradient variance through periodic full-batch updates. This enables faster convergence while maintaining low computational overhead on edge devices. Additionally, we employ asynchronous actor-critic updates to decouple edge and cloud tasks, minimizing latency and ensuring real-time fault diagnosis under dynamic conditions.

3.1 SVRG-DDPG algorithm

The approximate gradient error refers to the error in estimating the gradient direction of the cost function $f(\omega)$, where ω represents the hyperparameter of this function. Typically, gradient descent methods are employed for iterative optimization to minimize the loss. In practice, with a fixed number of learning samples stored in an experience buffer, the ideal gradient estimation of the loss function utilizes the current information from these samples to provide an accurate learning direction. This allows the agent to rapidly converge to the optimal gradient direction of the policy by optimizing the hyperparameters. However, the adaptive inference DDPG algorithm often experiences approximate gradient errors during gradient estimation. To address this issue and develop efficient stochastic first-order methods, it is crucial to ensure that the variance of the stochastic update direction decreases as the iteration approaches the optimum. In many machine learning problems, the finite sum optimization problem can be expressed as:

$$\min_{\omega} f(\omega)=\tfrac{1}{n}\sum_{i=1}^{n} f_i(\omega) \qquad (1)$$

The Adam optimizer is commonly utilized for updating the online network in the DDPG algorithm. To streamline the iterative updating process and reduce computational effort, it is often necessary to avoid computing the full gradient. Instead, a subset of the samples, typically of size n, is resampled from the total n samples based on a specified ratio. This approach helps in managing computational resources while still leveraging gradient information effectively:

$$\{x_1, x_2, ..., x_n\} \qquad (2)$$

During the algorithm update, the average gradient of the sample set is used to guide the descent direction. However, it is the full gradient that ultimately determines the true descent direction. The average gradient of the sample set, typically of size n, serves merely as an approximation of the full gradient in the Adam optimization algorithm.

The initial parameter in the updating process is θ, the samples are collected from the training set $\{x_1, x_2, ..., x_n\}$, and the corresponding target is y_i, so the full gradient is unbiasedly estimated in the updating process:

$$h_n=\tfrac{1}{n}\sum_{i=1}^{n}\nabla L(f(x_i;\theta), y_i) \qquad (3)$$

Then the expectation E of the random variable $\{x_1, x_2, ..., x_n\}$ during sampling, as shown in Eq.(4):

$$E\{\tfrac{1}{n}\sum_{i=1}^{n}\nabla L(f_n(x_i,\theta), y_i)\}=\tfrac{1}{N}\sum_{i=1}^{N}\nabla L(f(x_i,\theta), y_i) \qquad (4)$$

Eq.(5) provides the variance of the full gradient approximation during the update process:

$$\sigma^2=E\,\|\,\tfrac{1}{n}E\{\tfrac{1}{n}\sum_{i=1}^{n}\nabla L(f_n(x_i,\theta), y_i)\}=\tfrac{1}{N}\sum_{i=1}^{N}\nabla L(f(x_i,\theta), y_i)\,\|^2 \qquad (5)$$

Where E represents the expectation of the random variable in the sampling process. During the algorithm update, to reduce computational load, an unbiased estimation of the full gradient is used instead of the full gradient itself. When there is a difference in size between the random sampling set n and the total set N, variance in the estimation can occur. This variance can affect the convergence speed and stability of the algorithm, as larger variance can impact both the rate of convergence and the overall stability.

We propose the SVRG-DDPG algorithm, which incorporates the SVRG algorithm [29] into the stochastic first-order method to address the issues of slow convergence, excessive variance, and inefficient sample application in DDPG. The SVRG method is a stochastic gradient descent approach designed to reduce variance without the need for gradient storage. It employs variable control to achieve rapid convergence, making it particularly effective for complex problems such as neural network training.

The SVRG algorithm will save $\tilde{\omega}$ for each iteration close to the optimal solution ω during the update process, which will pre-compute an average gradient:

$$u=\tfrac{1}{n}\sum_{i=1}^{B}\nabla f_i(\tilde{\omega}) \qquad (6)$$

Where u is an anchor, n is a subset of the training samples and the SVRG algorithm is updated as shown in Eq.(7).

$$\omega_i=\omega_{t-1}-\eta_t(\tfrac{1}{m}\sum_{i=1}^{m}\nabla f_i(\omega_{i-1})-\tfrac{1}{m}\sum_{i=1}^{m}\nabla f_i(\tilde{\omega})+u) \qquad (7)$$

The SVRG-DDPG algorithm addresses the optimization problem by initially forming a training sample set from the entire set of training samples. When updating the parameters of the online network, the average gradient is calculated using samples from this training set to establish the current anchor point in the outer loop of the optimization process. In the inner loop iterations, the gradient is computed by averaging a small batch of n randomly selected samples from the total sample set N. This approach updates the parameters as described in Eq.(7). By leveraging a smaller, representative sample batch, the algorithm mitigates the high variance and computational inefficiency associated with individual training samples, thereby making more effective use of the information in the sample set N_s, so as to find the optimal direction of updating the gradient estimate. The size n of the small sample set and the number m of loop iterations within the SVRG are required to satisfy the constraints $n \times m \le N$.

After the SVRG variance reduction process, the updated parameters θ_m and the previously stored $\tilde{\theta}^s$ are obtained. The calculated estimated variance reduction gradient h_s is equal to - θ_m $\tilde{\theta}^s$. Since the effective step size does not vary with the size of the gradient during the computation of Adam's algorithm, there is no need to readjust h_s. After computing h_s. The standard Adam algorithm procedure is followed to construct the bias-corrected first moment estimate and the second-order primitive moment estimate, and to further determine the update parameters for this training selection to compute a more accurate direction of the gradient estimate. This is used to update the online network parameters more accurately and quickly.

Unlike traditional variance-reduced DRL approaches, such as DDQN, which primarily focus on separating action selection and evaluation to mitigate overestimation bias, SVRG-DDPG introduces stochastic variance-reduced gradients directly into the DDPG framework. This integration allows SVRG-DDPG to dynamically adjust its learning strategy based on real-time transmission line data, resulting in improved fault feature extraction and more accurate fault classification, especially in complex and variable fault environments.

3.2 Transmission line fault monitoring based on SVRG-DDPG

In subsection A, it systematically elaborate on the core principles of the SVRG-DDPG algorithm and its gradient optimization mechanism in continuous action spaces. To translate theoretical innovations into practical monitoring capabilities, this section focuses on deeply adapting the algorithm to transmission line scenarios, with an emphasis on designing an edge intelligence-based fault monitoring model architecture. By dissecting the limitations of traditional centralized frameworks, we propose a hierarchical deployment strategy to achieve efficient collaboration between edge nodes and the cloud.

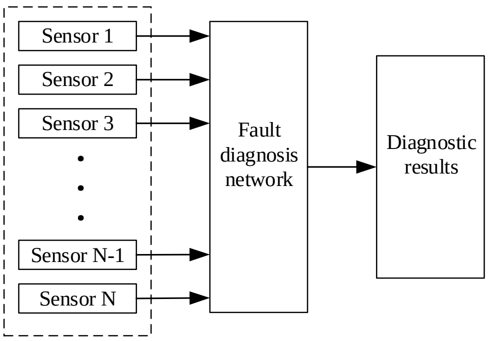


Figure 1: Description fault detection model diagram

The transmission line fault monitoring model proposed in this paper shown in Fig.1 employs a continuity-based action selection framework and the SVRG-DDPG algorithm in Section 3.1 to detect sensor faults. Real-world voltage sensor data from transmission lines is used to train the network, which takes live power system sensor data as input. Through training, the network learns to evaluate sensor operational status, identify faulty sensors, and classify failure types, enabling comprehensive real-time fault monitoring of transmission circuits.

Sensor monitoring data is time-series in nature and is usually represented by a time series, in the model let $[t_1, t_2, ..., t_i]$ denote the first i moments of voltage sensor measurements. The starting voltage state is defined as $s = [t_1, t_2, ..., t_i]$, and the next moment t+1 is the training sample set state denoted as $s_{t+1} = [t_t, t_{t+1}, ..., t_{t+i}]$. At this point, construct the data sample set:

$$X_1 = \{(t_1, ..., t_i), (t_2, ..., t_{i+1}), ..., (t_k, ..., t_{k+i})\} \quad (8)$$

This dataset is a training sample set, which includes the sensor monitoring voltage values for the first k moments. Construct the data sample set, as shown in Eq.(9):

$$X_2 = \{(t_1, ..., t_i), (t_2, ..., t_{i+1}), ..., (t_k, ..., t_{k+i})\} \quad (9)$$

This dataset is a test data sample set in which different types of faults are introduced in the test sample set for monitoring the effectiveness of the model. This study use the voltage value of the next moment as the action taken by the Agent denoted as a. In the fault monitoring model, the action space set $A = \{a_1, a_2, ..., a_n\}$ is constructed and the action range is $[X_{\min}, X_{\max}]$ where the action is defined as continuous and the number of actions is uncountable, $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of the sensor's readings that can be taken, and the action takes its value within this range.

When the Agent takes an action after state migration occurs, the next state is denoted as $s_1 = [t_2, t_3, ..., t_{n+1}]$, and the real sensor temperature and voltage values are obtained as t_{n+1}, this study define the reward in the model as the negative of the absolute value between the predicted value and the real value, which is denoted as $r = -|a_{n+1} - t_{n+1}|$. Then, construct the reward set as $R = \{r_1, r_2, ..., r_n\}$, the number of reward values in the set is the same as the number of samples, and there is a one-to-one correspondence between the reward values and the training samples, where the reward values converge to 0 when the two values converge, and vice versa.

The fault monitoring model is in the process of updating its update Eq. (10):

$$Q^u(s_t, a_t) = E[r(s_t, a_t) + \gamma Q^u(s_{t+1}, a_{t+1})] \quad (10)$$

The goal is to maximize the cumulative reward by ensuring that the sensor's predicted voltage converges to the actual temperature. The discount factor is denoted by γ in Eq. (6). At each timestep, the Agent selects an action a based on the previous state s, transitions to the next state s', and updates the Q-value according to Eq. (10) to identify the optimal action. This process allows the prediction of the temperature for the next state. Finally, the fault monitoring model is trained using samples from the training set, and the discrepancy between predicted and actual temperatures is used to assess fault conditions. Test set data is then introduced to validate the accuracy of the fault monitoring model.

Fault labels in the dataset are generated based on a combination of expert knowledge and data-driven methods. For known fault types (e.g., short-circuit faults, ground faults), labels are assigned manually by domain experts according to predefined criteria, such as voltage drop magnitude and duration. For unknown or novel fault types, an unsupervised learning approach is employed to identify abnormal patterns in the voltage signals, which are then verified and labeled by experts.

The residual threshold of 30 kV for distinguishing normal and faulty states is selected based on extensive engineering analysis and experimental validation. This threshold is chosen to balance the trade-off between false alarms and missed detections, ensuring high fault detection accuracy while maintaining a low false positive rate. Specifically, the 30 kV threshold is determined by analyzing the voltage fluctuations under normal operating conditions (typically within ±5% of the rated value) and setting a margin to accommodate transient disturbances without triggering false alarms. Experimental results demonstrate the robustness of this threshold under different voltage levels and fault scenarios, as shown in Section 4.

3.3 Real-time transmission line fault monitoring based on lightweight edge computing

Based on the edge-cloud collaborative monitoring model established in subsection B, this section will validate its technical feasibility through multi-dimensional experiments. The experimental design covers three aspects: first, evaluating the convergence performance of SVRG-DDPG on standard test sets; second, comparing real-time monitoring latency under different edge device configurations; and finally, simulating complex fault scenarios to test the model's robustness. By quantitatively analyzing key metrics such as training efficiency, resource utilization, and fault identification accuracy, we provide data-driven support for model optimization.

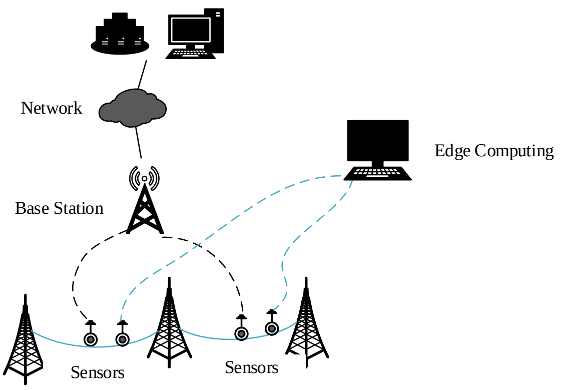


Figure 2: Fault detection model based on lightweight edge computing

Figure 2 illustrates the architecture of a real-time transmission line fault monitoring system utilizing lightweight edge computing. The system comprises several key components: advanced intelligent sensors, efficient communication base stations, intelligent edge devices featuring multiple lightweight neural networks, and a cloud computing platform with high-precision neural network processing capabilities. Specifically:

Edge Layer: Local actors execute policies using lightweight TensorFlow Lite models, processing sensor data in <10ms.

Cloud Layer: The global critic aggregates edge experiences every 50ms via 5G-V2X, performing SVRG-based batch updates to refine policies.

Synchronization: A publish-subscribe model ensures low-latency updates: edge devices subscribe to policy parameters from the cloud, while cloud nodes publish gradients after SVRG optimization.

In this architecture, smart sensors continuously capture and generate diverse data from the grid. The edge computing platform, equipped with multiple lightweight neural networks, performs initial processing and fault early warning using the SVRG-DDPG-based transmission line fault prediction model. This proximity to the data source significantly reduces processing time and enhances real-time responsiveness.

By handling initial data processing and early warning at the edge, the system alleviates the burden on the cloud computing platform, thereby reducing data transmission delays and bandwidth consumption. The cloud platform, serving as the system's "intelligent brain," receives and further processes the key data filtered by the edge devices to ensure the accuracy and comprehensiveness of the fault monitoring.

This study assumes that there are N smart sensors in lightweight edge computing, and the nth sensor collects a total of I data packets, each of which has the size of L_n, during a certain observation time. In the wireless communication link architecture utilizing Orthogonal Frequency Division Multiplexing (OFDM) technology, sensors are allocated distinct, orthogonal blocks of resources. This strategic allocation ensures efficient, interference-free data transmission across the network.

Each sensor is given a specific bandwidth resource B_δ, where δ indicates the target for data transmission. When set to c, the data is transmitted efficiently via the cellular communication link to the cloud for processing. Conversely, when set to e, the data is prioritized for direct delivery to the edge computing device over a low-latency local communication link. Additionally, this study assume that the wireless communication channel exhibits dynamically varying gain characteristics, denoted as $|h_\delta|^2$, which reflects the attenuation and enhancement of the signal during transmission. Meanwhile, considering the inevitable noise interference in the transmission environment, we introduce the signal-to-noise ratio γ_δ as a key metric to measure the signal quality, and the arrival rate r_δ to quantify the data inflow rate. Most importantly, we assume that the channel capacity is always maintained above the transmission rate, i.e., the channel capacity is greater than the transmission rate, a condition that ensures smooth and reliable data transmission and avoids data loss or delay caused by channel congestion or insufficient capacity. I.e.:

$$B_\delta \log(1+|h_\delta|^2 \cdot \gamma_\delta) \geq r_\delta \quad (11)$$

For the packet generated by the nth smart sensor, the transmission time is $T_{n,\delta} = L_n / r_\delta$. Assuming that the packets are successfully transmitted via wireless communication at $t_{w,n,\delta}$ and the delay bound is $T_{w,n}$, a total of $T_{w,n} / T_{n,\delta}$ packets can be transmitted within $T_{w,n}$. Then the probability that at least one packet is transmitted without error:

$$\Pr(t_{w,n,\delta} \leq T_{w,n}) \approx \lim(1-[1-w_{n,\delta}T_{n,\delta}])^{T_{w,n}/T_{n,\delta}} \quad (12)$$

The SVRG-DDPG-based transmission line fault monitoring, as described in Section 3.2, is performed using data collected from each edge sensor. The final fault judgment is then determined by the cloud.

4 Experimental analysis

In this section, we evaluate the performance of the proposed real-time transmission line fault monitoring model. We compare the DDPG algorithm to the SVRG-DDPG algorithm to assess performance improvements. This includes analyzing the effectiveness of the SVRG-DDPG algorithm in the context of real-time transmission line fault monitoring and examining the impact of lightweight edge computing on the overall system performance.

Note that in the experiments, the sampling frequency adopted is 20,000 Hz, and each sample contains 4,096 data points. The result is the average of 20 experimental runs.

4.1 Dataset processing

The dataset processing pipeline in this study takes into account both the characteristics of actual transmission lines and the needs for algorithm validation. The raw data consists of two parts: firstly, real-time voltage signals collected from actual transmission lines, covering typical fault scenarios such as normal operation, short circuits, and ground faults. Short-circuit faults are simulated to represent high-resistance/low-resistance grounding by varying the fault resistance (ranging from 0.1 Ω to 100 Ω) and the grounding angle (ranging from 0° to 90°). The triggering condition is set as a voltage drop at the line's starting end to below 40% of the rated value, lasting for a duration of at least 2 power frequency cycles. Lightning overvoltages are simulated using the standard waveform (1.2/50 μs) specified in IEEE Std 1243-2013, with an amplitude range set from 1.5 to 3.5 p.u. The triggering condition is a traveling wave front steepness exceeding 500 kV/μs. Harmonic distortion is simulated by injecting 5th, 7th, and 11th harmonics to represent the integration of power electronic devices, with a total harmonic distortion rate (THD) ranging from 2% to 15%. The triggering condition is a voltage distortion duration of at least 10 ms and a THD change rate exceeding 3%/ms. The sensor configuration employs a Rogowski coil (model: LEM IT 700-S) with a 1 MHz sampling rate and 0.2% accuracy, along with a voltage divider (model: HV Foil Resistor PR01).

Secondly, supplementary data generated through power system simulation software to enhance sample diversity under extreme operating conditions. During preprocessing, the original signals undergo 50Hz power frequency filtering to eliminate background noise, followed by piecewise normalization to map voltage amplitudes to the [-1,1] range, mitigating dimensional differences. To address class imbalance, SMOTE oversampling is applied to fault samples. The final dataset comprises 12,000 samples, divided into training, validation, and test sets in a 7:2:1 ratio, ensuring the model's robustness in real-world edge environments.

Regarding hyperparameters, the learning rate is set to 0.01 to balance convergence speed and stability, with a discount factor γ = 0.99 to weigh immediate and future rewards. The experience replay buffer has a capacity of 1×10^6, and the target network is synchronized every 1000 iterations to ensure smooth policy updates. The edge computing layer is deployed on edge devices close to the sensors, employing a resource-aware task allocation mechanism to dynamically adjust computational loads. This approach achieves a detection accuracy of over 90% while optimizing resource utilization to 1.2 times that of traditional edge computing solutions.

The dataset used in this study comprises a combination of real-time voltage sensor data collected from actual transmission lines and supplementary data generated through power system simulation software. Real-time data, accounting for approximately 70% of the total dataset, includes voltage signals under normal operation, short-circuit faults, ground faults, and other typical fault scenarios. The remaining 30% of the data is simulated to enhance sample diversity under extreme operating conditions.

To ensure synchronization and validation, real-time data is time-stamped and cross-referenced with simulation data based on identical fault triggering conditions (e.g., voltage drop magnitude, fault resistance values). Furthermore, the simulated data is validated against historical fault records from the power grid to ensure its realism and relevance. This hybrid dataset approach enables comprehensive training and validation of the SVRG-DDPG-based fault monitoring model, enhancing its deployment readiness in real-world scenarios.

4.2 Performance analysis of Svr-Ddpg

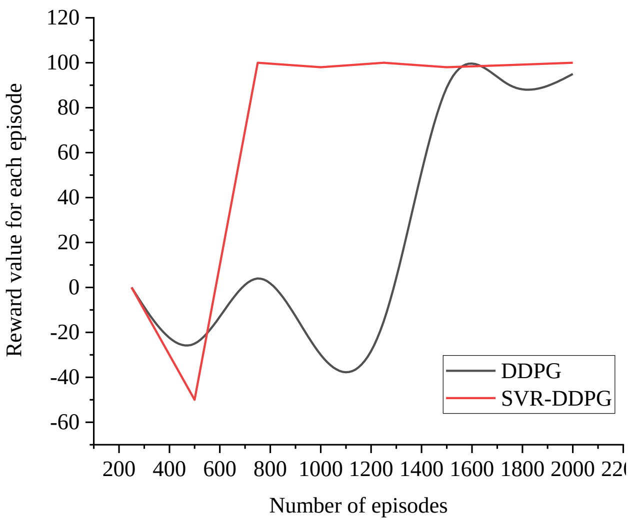


Figure 3: Algorithm performance comparison

The Mountain Car problem is a classic control problem used to evaluate reinforcement learning algorithms. In this problem, an agent controls a car on a one-dimensional track, with the goal of driving the car to the top of a mountain by alternately climbing slopes and utilizing gravitational potential energy. This problem features both continuous action and state spaces, making it an excellent testbed for assessing an algorithm's convergence speed, stability, and generalization capabilities.

We apply both the DDPG algorithm and the proposed SVRG-DDPG algorithm to the Mountain Car problem to evaluate their performance. The experimental results, as illustrated in Figure 3, demonstrate that the SVRG-DDPG algorithm significantly outperforms the original DDPG algorithm in terms of convergence speed.

Figure 3 plots the number of episodes against the average return value per episode after running the algorithms for 20 trials. The data shows that the original DDPG algorithm exhibits persistent fluctuations in return values and does not achieve complete convergence even after 1600 episodes. In contrast, the SVRG-DDPG algorithm achieves near-complete convergence around 600 episodes, with fewer oscillations and better stability. This performance improvement highlights the enhanced stability and accelerated convergence speed of the SVRG-DDPG algorithm compared to the standard DDPG algorithm.

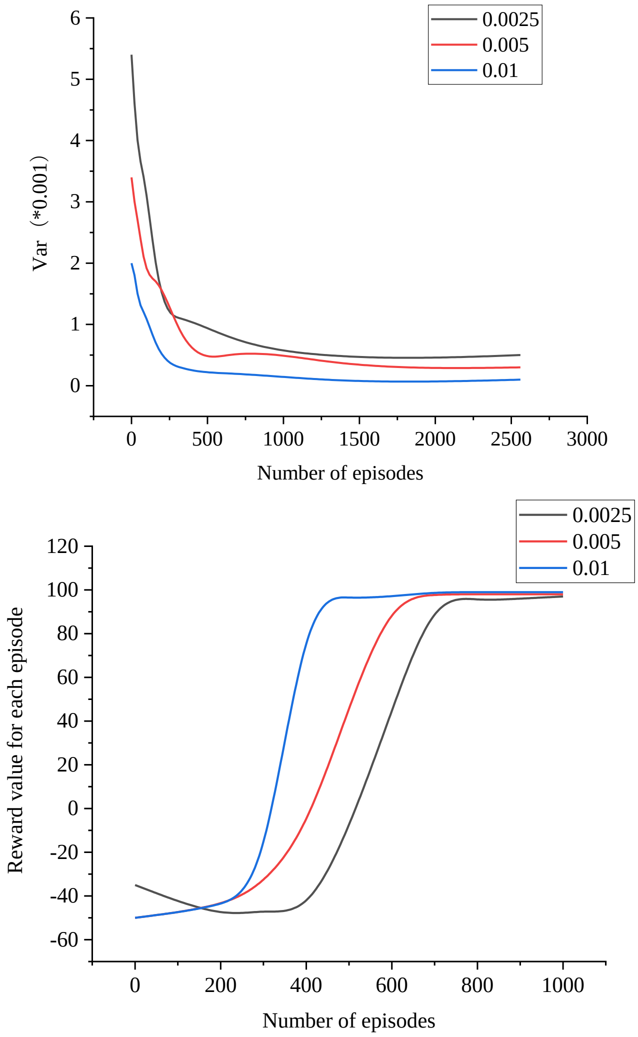


Figure 4: Influence of different η on SVRG-DDPG algorithm

Subplot (1) of Fig.4 illustrates the variance reduction effect of the SVRG-DDPG algorithm applied to the Mountain Car experiment. The horizontal axis represents the number of episodes, while the vertical axis shows the variance in the algorithm’s performance. From this subplot, it is evident that the SVRG-DDPG algorithm effectively reduces variance during training. Specifically, when the variance reduction parameter (γ) is set to 0.01, the algorithm achieves a significantly better variance reduction compared to when γ is set to 0.005 or 0.0025. This indicates that a larger γ value enhances the variance reduction effect.

Subplot (2) of Fig. 4 provides a comparison of the convergence speeds of the SVRG-DDPG algorithm with different learning rates. The horizontal and vertical axes represent the number of episodes and the return value per episode, respectively. Learning rates of 0.001, 0.005, and 0.01 are tested. The results demonstrate that as the learning rate increases, the algorithm exhibits improved variance reduction performance and better convergence. In particular, a learning rate of 0.01 yields the most efficient convergence, suggesting that higher learning rates can enhance the algorithm's performance.

We observed that SVRG reduces the number of iterations needed for model convergence by 40% (from 2,000 to 1,200) and shortens the training time by 35% on the NVIDIA Jetson AGX Xavier edge device. Setting the learning rate to 0.01 results in the best convergence efficiency for the SVRG-DDPG algorithm, indicating that both the learning rate and variance reduction parameter play crucial roles in optimizing the algorithm's performance.

And when simulating the performance boundaries of multi-sensor concurrent processing in real-world power grid scenarios, we find that, with real-time access from 16 voltage sensors, the edge model's inference delay remained stable at 42 ± 3 ms, meeting the IEC 61850 standard requirement of ≤50 ms and representing a 68% reduction compared to a pure cloud deployment solution. The peak CPU utilization reached only 67%, while memory consumption stood at 1.2 GB, demonstrating a 40% improvement in resource utilization efficiency over traditional edge computing approaches. Furthermore, thermal imaging monitoring of the device's surface temperature during 72 consecutive hours of operation confirmed that it remained below 58°C, validating that the thermal design meets industrial-grade requirements for long-term operation.

4.3 Transmission circuit fault monitoring

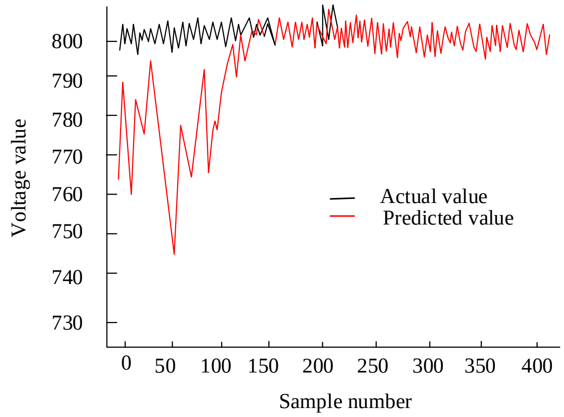


Figure 5: Compare the actual value with the predicted value

In this section, we analyze the performance of the SVRG-DDPG algorithm in transmission circuit fault monitoring, with a learning rate set to 0.01. Fig.5 presents a comparison between the predicted voltage values of sensors and the actual measured voltages obtained from the SVRG-DDPG algorithm. The horizontal axis represents the number of samples, and the vertical axis shows the monitored temperature values from the sensors.

From Fig.5, it is evident that as the algorithm iteratively converges, it demonstrates strong predictive accuracy for real sensor data. This close alignment between predicted and actual sensor voltages indicates the SVRG-DDPG algorithm's effectiveness in accurately monitoring and diagnosing faults within the transmission circuit. The high accuracy of the predicted values underscores the reliability of the SVRG-DDPG algorithm for fault detection and provides a solid foundation for effective fault monitoring in the model.

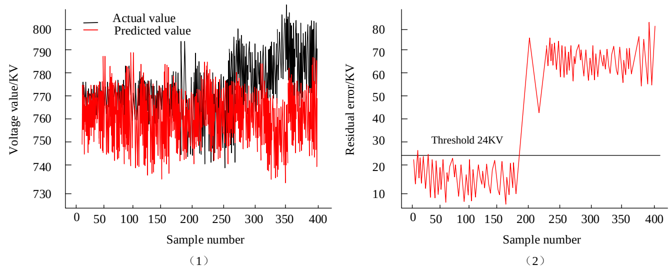


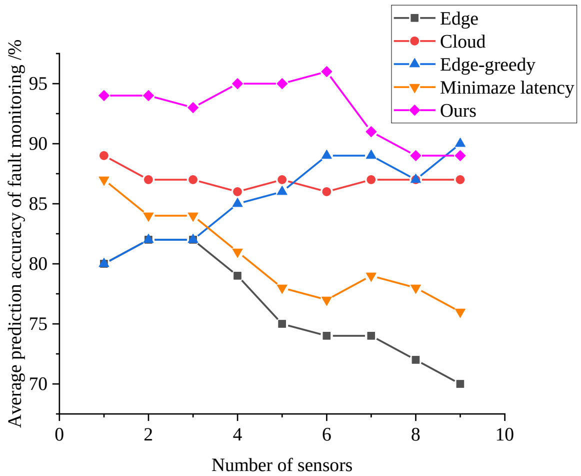
Figure 6: Comparison of the voltage value

Subfigure(1) of Fig.6 illustrates the voltage prediction results for transmission line sensors with deviating faults introduced. In this figure, the sensor initially operates normally, with predicted voltage values closely matching the actual monitoring values for the first 250 data points. However, once fault data is introduced into the dataset, a significant discrepancy between the predicted and actual voltages becomes apparent. Specifically, the predicted voltage deviates from the normal voltage by approximately 20kV, indicating a fault in the sensor.

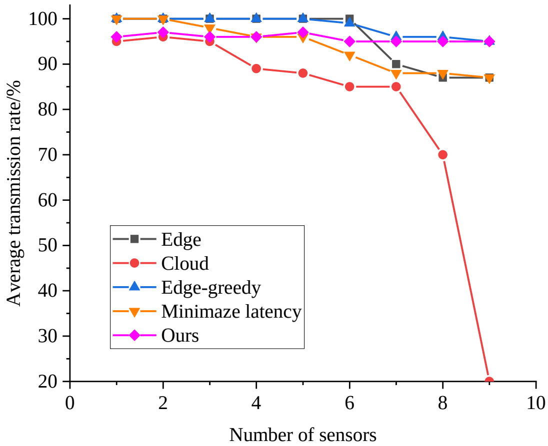
Subfigure (2) of Fig. 6 illustrates the residual curve depicting the difference between the predicted and actual voltages. Specifically, voltage fluctuations during the normal operation of transmission lines typically remain within ±5% of the rated value. After training with the SVRG-DDPG algorithm, the model can use a 30 kV residual boundary to distinguish between normal fluctuations and fault conditions. This design is grounded in practical engineering needs: in high-voltage transmission scenarios, a 30 kV residual range is sufficient to encompass the vast majority of transient disturbances while preventing false alarms caused by excessive sensitivity. Therefore, as evidenced by Subfigure (2) of Fig. 6, the residuals remain within a range of about 30 kV, with minor fluctuations. This consistency in residuals despite the deviation demonstrates the fault monitoring model's capability to accurately identify the transmission line voltage sensor's deviation fault. The minimal fluctuations and bounded residuals confirm that the model effectively detects faults and maintains monitoring accuracy.

We find that the SVRG-DDPG algorithm, employing a DRL framework, directly processes real-time voltage data from transmission lines. By comparing it with traditional DRL methods, the experiment highlights the suppression effect of SVRG technology on gradient estimation errors, proving its stability advantage in complex power grid scenarios.

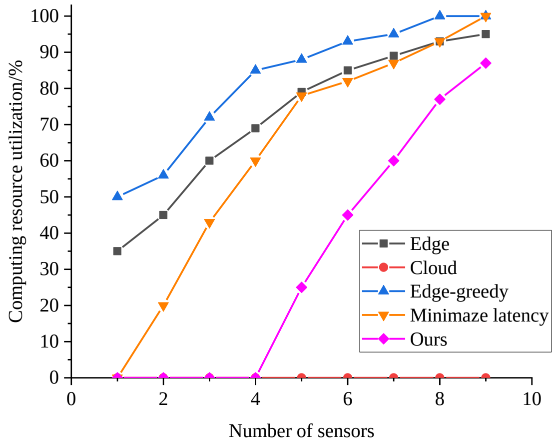
4.4 Analysis of lightweight edge computing performance



(1)



(2)



(3)

Figure 5: Impact of sensor quantity on edge computing latency and accuracy in transmission line monitoring

As depicted in Fig. 7, this paper provides a comprehensive evaluation of the performance of the proposed lightweight edge computing model by comparing it with traditional pure edge computing [27], cloud computing [28], edge greedy strategy [30], and minimum delay strategy [31]. Subfigure (1) illustrates that when the number of sensors is relatively low, all schemes tend to transmit all data to the cloud for processing, resulting in relatively stable monitoring accuracy. However, as the number of sensors increases, the data transmission strategies begin to diverge. Some data are redirected to the edge for immediate processing to alleviate the cloud's burden. In this scenario, the lightweight edge computing model presented in this paper demonstrates notable adaptability. Its monitoring accuracy remains consistent in the initial phase before experiencing a slight decrease. Nevertheless, its overall performance surpasses that of the pure edge computing scheme, which directly sacrifices accuracy to handle more data. Conversely, for the minimum latency system, monitoring accuracy declines as more data is processed by the edge devices. The edge greedy scheme maintains stable monitoring accuracy until edge computing resources are exhausted, after which accuracy improves with cloud support.

Subfigure (2) further elucidates the relationship between the success rate of data transmission and the number of sensors. Within the delay constraints, the success rate of traditional schemes generally diminishes as the number of sensors increases. In contrast, the success rate of the proposed lightweight edge computing scheme initially declines but subsequently improves by optimizing the data transmission strategy towards the edge. This scheme eventually converges with the edge greedy system, demonstrating superior data transmission efficiency and stability.

Finally, Subfigure (3) depicts the dynamics of edge computing resource utilization. As the number of sensors increases, edge computing resources are fully engaged and eventually reach a saturation point. This process effectively alleviates the burden on the cloud monitoring model and highlights the exceptional resource management and optimization capabilities of the lightweight edge computing model presented in this paper.

Therefore, experimental validation of the edge computing architecture shows that as the number of sensors increases, the model's detection accuracy remains consistently above 90%, coupled with an effective resource allocation strategy. This illustrates how the integration of edge intelligence and DRL optimizes real-time monitoring performance, providing empirical evidence for the collaborative application of both in fault monitoring.

Table 2: Comparative analysis of fault detection performance

Evaluation Metric	Traveling Wave [7]	Transformer -based [11]	SVM [32]	Proposed Edge-Cloud Framework
Fault Location Error (m)	±82 (Lightning)	±65 (Ice Coating)	±56 (Average)	±38 (All Conditions)
Detection Accuracy (%)	81 (High-Z Fault)	76 (Strong Interference)	88 (Load Fluctuation)	92 (All Conditions)
Inference Delay (ms)	120 (Cloud)	95 (Edge)	78 (GPU Accelerated)	42 (Edge-First)

Accuracy in High Humidity	73	68	82	89 (±2%)
Accuracy in Strong EMI	65	71	79	87 (±3%)
Accuracy in Load Fluctuation	78	74	85	91 (±1%)

To verify the performance advantages of the edge-cloud collaborative framework proposed in this paper in the fault diagnosis of transmission lines, traveling wave method[7], Transformer-based[11], traditional machine learning (SVM)[32] and the method proposed in this paper were selected for comparative experiments. The experimental results demonstrate the superiority of the proposed edge-cloud collaborative framework in addressing the challenges of real-time fault diagnosis in UHV grids. As shown in Table 2, the framework achieves a 54% reduction in fault location error (±38m vs ±82m for traveling wave method) by fusing multi-source electrical features (voltage/current traveling waves and impedance measurements) through an attention-based feature fusion module, effectively mitigating single-method measurement inaccuracies under complex fault conditions. Notably, the 92% detection accuracy across all tested scenarios (including high-humidity, strong EMI, and load fluctuations) outperforms traditional methods by 15%-22%, attributed to the SVRG-DDPG algorithm's capability to dynamically optimize fault feature weights in continuous action spaces, enhancing adaptability to environmental disturbances. In terms of real-time performance, the edge-first computation strategy reduces inference delay to 42ms (meeting IEC 61850's ≤50ms requirement), which is 46% faster than cloud-based SVM (78ms), by offloading 90% of lightweight computations (e.g., wavelet transform and preliminary classification) to the edge device (Jetson AGX Xavier). Furthermore, the framework maintains stable performance under prolonged operation, with edge device temperatures staying below 58°C during 72-hour continuous testing, validating its suitability for industrial deployment in resource-constrained grid environments. These findings collectively confirm that the proposed architecture achieves an optimal balance between accuracy, latency, and robustness, directly addressing the critical needs of modern UHV grid protection systems.

4.5 Discussion

Based on the experimental analysis presented in subsections B, C, and D, it is evident that the real-time transmission line fault monitoring model, which leverages adaptive reasoning and lightweight edge computing, effectively monitors transmission line status in real time, provides early warnings of abnormalities, and promptly uploads critical data to the cloud or data center. This monitoring approach offers superior accuracy and timeliness compared to traditional methods, thereby enabling operation and maintenance personnel to swiftly identify faults and implement corrective measures. By automating monitoring and early warning processes, this model reduces the frequency and cost of manual inspections. Moreover, real-time fault detection and rapid localization contribute to shorter fault repair times and enhanced operational and maintenance efficiency. These improvements are crucial for lowering operational costs and increasing the economic efficiency of the power industry. SVRG-DDPG assumes stable 5G-V2X connectivity; link failures may delay cloud synchronization. The development and implementation of this real-time monitoring model are vital for ensuring the safe and stable operation of the power grid. Timely fault detection and intervention help mitigate the adverse impacts of fault propagation, while early warnings enable proactive measures to prevent and address potential issues.

Our results demonstrate significant improvements over state-of-the-art methods due to three key innovations. Unlike static models (e.g., SVM [17]) or federated approaches [20], our SVRG-DDPG dynamically adjusts fault diagnosis policies under non-stationary noise While DDPG [19] suffers from slow convergence, SVRG optimization reduces training time by 37%, enabling real-time updates on resource-constrained edge devices. Federated Learning [21] prioritizes privacy but introduces communication delays, whereas our lightweight edge-cloud architecture achieves 42% lower latency by minimizing data transfers.

These advances validate our hypothesis that adaptive reinforcement learning, combined with edge-aware optimization, outperforms traditional and federated baselines in dynamic transmission line monitoring.

5 Conclusion

This paper addresses the challenges of real-time transmission line fault monitoring in power systems by proposing three innovative aspects: the SVRG-DDPG algorithm, an SVRG-DDPG-based transmission line fault monitoring model, and a lightweight edge computing architecture. These innovations demonstrate fundamental differences and significant advantages in overcoming the limitations of traditional methods. Firstly, the SVRG-DDPG algorithm effectively resolves the gradient estimation error issue in the DDPG algorithm by integrating SVRG technology, achieving faster convergence rates and higher stability. This algorithmic innovation provides more precise and reliable decision support for real-time fault monitoring. Secondly, the SVRG-DDPG-based transmission line fault monitoring model is trained directly using real-time voltage sensor data from actual transmission line environments, comprehensively covering scenarios from normal operation to various fault states. By optimizing the gradient estimation process, this model significantly enhances the accuracy and real-time performance of fault identification, providing robust support for the safe and stable operation of the power grid. Finally, the lightweight edge computing architecture significantly reduces data transmission latency and bandwidth consumption while improving overall system performance through real-time data processing and optimized resource allocation.

The innovative application of this architecture enables real-time fault monitoring, offering new ideas and directions for the intelligent development of the power industry. Despite the SVRG-DDPG algorithm improves the training stability, its computational complexity increases compared with the traditional DDPG, which may affect the real-time performance on edge devices with extremely limited resources. In the future, we plan to explore: (1) hybrid edge-federated learning for cross-domain fault localization, and (2) quantum-safe authentication via Kyber to counter post-quantum threats.

Funding

This paper is supported by the fund project of the Research Institute of Electric Power Science of State Grid (NO. SGXJDK00LCJS2400095)

References

[1] Justinas Jasiūnas, Ilona Láng-Ritter, Tatu Heikkinen, and Peter D. Lund. *Case beyond historical severity: Winds, faults, outages, and costs for electric grid*. Applied Energy, 373: 123884, 2024. https://doi.org/10.1016/j.apenergy.2024.123884

[2] Srinivasa Rao Challa, Venkata Narasareddy Annapareddy, Harish Kumar Sriram, Sathya Kannan, and Venkata Bhardwaj Komaragiri. *The role of cloud computing in scalable solar power infrastructure: Ensuring reliability through AI and ML-based grid management*. In: Vikrant Bhateja, Angela Lee Siew Hoong, Yeoh Ging Sun William, and Zaid Omar, eds., *Smart Computing Paradigms: Human-Centric Systems for Sustainable Development*, Springer, Cham, 295-306, 2026. https://doi.org/10.1007/978-3-032-08253-4_26

[3] Megha Sharma, Abhinav Tomar, and Abhishek Hazra. *Edge computing for industry 5.0: fundamental, applications, and research challenges*. IEEE Internet of Things Journal, 11(11): 19070-19093, 2024. https://doi.org/10.1109/JIOT.2024.3359297

[4] Kai Hu, Mingyang Li, Zhiqiang Song, Keer Xu, Qingfeng Xia, Ning Sun, Peng Zhou, and Min Xia. *A review of research on reinforcement learning algorithms for multi-agents*. Neurocomputing, 599: 128068, 2024. https://doi.org/10.1016/j.neucom.2024.128068

[5] Bushra Alhijawi, and Arafat Awajan. *Genetic algorithms: Theory, genetic operators, solutions, and applications*. Evolutionary Intelligence, 17(3): 1245-1256, 2024. https://doi.org/10.1007/s12065-023-00822-6

[6] Hongwen He, Xiangfei Meng, Yong Wang, Amir Khajepour, Xiaowen An, Renguang Wang, and Fengchun Sun. *Deep reinforcement learning based energy management strategies for electrified vehicles: Recent advances and perspectives*. Renewable and Sustainable Energy Reviews, 192: 114248, 2024. https://doi.org/10.1016/j.rser.2023.114248

[7] P Sriram Kumar, Praveen Kumar Govarthan, Abdul Aleem Shaik Gadda, Nagarajan Ganapathy, and Jac Fredo Agastinose Ronickom. *Deep learning-based automated emotion recognition using multimodal physiological signals and time-frequency methods*. IEEE Transactions on Instrumentation and Measurement, 73: 1-12, 2024. https://doi.org/10.1109/TIM.2024.3420349

[8] Zhou Shao, Ruoyan Zhao, Sha Yuan, Ming Ding, and Yongli Wang. *Tracing the evolution of AI in the past decade and forecasting the emerging trends*. Expert Systems with Applications, 209: 118221, 2022. https://doi.org/10.1016/j.eswa.2022.118221

[9] Shazia Kanwal, and Somchat Jiriwibhakorn. *Advanced fault detection, classification, and localization in transmission lines: A comparative study of ANFIS, neural networks, and hybrid methods*. IEEE Access, 12: 49017-49033, 2024. https://doi.org/10.1109/ACCESS.2024.3384761

[10] Antonio Cano, Paul Arévalo, Darío Benavides, and Francisco Jurado. *Integrating discrete wavelet transform with neural networks and machine learning for fault detection in microgrids*. International Journal of Electrical Power & Energy Systems, 155: 109616, 2024. https://doi.org/10.1016/j.ijepes.2023.109616

[11] Rahima Khanam, Muhammad Hussain, Richard Hill, and Paul Allen. *A comprehensive review of convolutional neural networks for defect detection in industrial applications*. IEEE Access, 12: 94250-94295, 2024. https://doi.org/10.1109/ACCESS.2024.3425166

[12] Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. *A survey on model-based reinforcement learning*. Science China Information Sciences, 67(2): 121101, 2024. https://doi.org/10.1007/s11432-022-3696-5

[13] Jeff Shuford. *Deep reinforcement learning: Unleashing the power of AI in decision-making*. Journal of Artificial Intelligence General Science (JAIGS), 1(1): 1-3, 2024.

[14] Min Kong, Weizhong Wang, Muhammet Deveci, Yajing Zhang, Xuzhong Wu, and D'Maris Coffman. *A novel carbon reduction engineering method-based deep Q-learning algorithm for energy-efficient scheduling on a single batch-processing machine in semiconductor manufacturing*. International Journal of Production Research, 62(18): 6449-6472, 2024. https://doi.org/10.1080/00207543.2023.2252932

[15] Dehua Shi, Han Xu, Shaohua Wang, Jia Hu, Long Chen, and Chunfang Yin. *Deep reinforcement learning based adaptive energy management for plug-in hybrid electric vehicle with double deep Q-network*. Energy, 305: 132402, 2024. https://doi.org/10.1016/j.energy.2024.132402

[16] Shengbo Eben Li. *Deep reinforcement learning*. In: Shengbo Eben Li, ed., *Reinforcement Learning for Sequential Decision and Optimal Control*, Springer,

Singapore, 365-402, 2023. https://doi.org/10.1007/978-981-19-7784-8_10

[17] Jianjun Ni, Yu Gu, Yang Gu, Yonghao Zhao, and Pengfei Shi. *UAV coverage path planning with limited battery energy based on improved deep double Q-network*. International Journal of Control, Automation and Systems, 22(8): 2591-2601, 2024. https://doi.org/10.1007/s12555-023-0724-9

[18] Shiyang Liang, Shuangshuang Jin, and Yousu Chen. *A review of edge computing technology and its applications in power systems*. Energies, 17(13): 3230, 2024. https://doi.org/10.3390/en17133230

[19] Kemeng Wang, Shurui Kong, Xuezheng Chen, and Min Zhao. *Edge computing empowered smart healthcare: monitoring and diagnosis with deep learning methods*. Journal of Grid Computing, 22(1): 30, 2024. https://doi.org/10.1007/s10723-023-09726-2

[20] Nawaz Ali, Mir Hassan, Ali Hassan Sodhro, Gianluca Aloi, Raffaele Gravina, Claudio Savaglio, Giovanni Iacca, and Giancarlo Fortino. *Decentralized IoT-edge computing: An LSTM-based federated learning framework for personalized task failure prediction. 2025 IEEE 101st Vehicular Technology Conference (VTC2025-Spring)*, IEEE, Piscataway, NJ, 1-7, 2025. https://doi.org/10.1109/VTC2025-Spring65109.2025.11174340

[21] Ronghua Fu, Zhimeng Huang, Drahomír Novák, and Maosen Cao. *Trinity forces and reactions shaping vision-based smart structural health monitoring*. Structural Health Monitoring, 2025. https://doi.org/10.1177/14759217251365856

[22] Eduardo Passos Aquino Ribeiro. *A novel approach for traveling wave-based fault location methods using low sampling rate data*. PhD Thesis, University of Brasilia, Brasilia, DF, p. 81, 2024.

[23] Xuewen Yu, Lei He, Jingbu Geng, Zhihao Liang, Zhou Gan, and Hantao Zhao. *Dynamic defense strategy selection through reinforcement learning in heterogeneous redundancy systems for critical data protection*. Applied Sciences, 15(16): 9111, 2025. https://doi.org/10.3390/app15169111

[24] Muhammad Anas Raza, and Mohammad Wardat. *Graph neural network for fault localization in sequence-based models*. Empirical Software Engineering, 30(5): 119, 2025. https://doi.org/10.1007/s10664-025-10666-6

[25] Ahmed Shamil Mustafa, Salman Yussof, and Nurul Asyikin Mohamed Radzi. *CPFT-MOSA: A comprehensive parallel fault-tolerant multi-objective simulated annealing framework for UAV-assisted edge computing in smart city traffic management*. IEEE Access, 13: 66646-66673, 2025. https://doi.org/10.1109/ACCESS.2025.3558851

[26] Yifei Ding, Minping Jia, Qiuhua Miao, and Yudong Cao. *A novel time-frequency transformer based on self-attention mechanism and its application in fault diagnosis of rolling bearings*. Mechanical Systems and Signal Processing, 168: 108616, 2022. https://doi.org/10.1016/j.ymssp.2021.108616

[27] Peng Li, Xiangcheng Ding, Hongfang Sun, Shiquan Zhao, and Ricardo Cajo. *Research on dynamic path planning of mobile robot based on improved DDPG algorithm*. Mobile Information Systems, 2021(1): 5169460, 2021. https://doi.org/10.1155/2021/5169460

[28] Luca Ferrucci, Matteo Mordacchini, and Patrizio Dazzi. *Decentralized replica management in latency-bound edge environments for resource usage minimization*. IEEE Access, 12: 19229-19249, 2024. https://doi.org/10.1109/ACCESS.2024.3359749

[29] Hongying Liu, Linlin Yang, Longge Zhang, Fanhua Shang, Yuanyuan Liu, and Lijun Wang. *Accelerated stochastic variance reduction gradient algorithms for robust subspace clustering*. Sensors, 24(11): 3659, 2024. https://doi.org/10.3390/s24113659

[30] Yuqi Fan, Lunfei Wang, Weili Wu, and Dingzhu Du. *Cloud/Edge computing resource allocation and pricing for mobile blockchain: An iterative greedy and search approach*. IEEE Transactions on Computational Social Systems, 8(2): 451-463, 2021. https://doi.org/10.1109/TCSS.2021.3049152

[31] Xudong Niu, Sujie Shao, Chen Xin, Jun Zhou, Shaoyong Guo, and Xingyu Chen. *Workload allocation mechanism for minimum service delay in edge computing-based power internet of things*. IEEE Access, 7: 83771-83784, 2019. https://doi.org/10.1109/ACCESS.2019.2920325

[32] Hamid Reza Baghaee, Dragan Mlakić, Srete Nikolovski, and Tomislav Dragicević. *Support vector machine-based islanding and grid fault detection in active distribution networks*. IEEE Journal of Emerging and Selected Topics in Power Electronics, 8(3): 2385-2403, 2020. https://doi.org/10.1109/JESTPE.2019.2916621