International Journal of Robotics and Control Systems
Vol. 6, No. 1, 2026, pp. 819-846
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Adaptive Fault-Resilient and Self-Healing Energy Management for IoT-Enabled Microgrids Via Deep Q-Learning with Real-Time Edge Control

Al-Shukrawi Ali Abbas Hadi [a,1], Aeizaal Azman Bin Abdul Wahab [b,2,*]

[a] School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Pulau Pinang, Malysia
[1] aliabbashadi@student.usm.my; [2] aeizaal@usm.my
* Corresponding Author

ARTICLE INFO

Article history
Received January 21, 2026
Revised February 25, 2026
Accepted March 28, 2026

Keywords
Microgrid Control;
Deep Q-Learning;
GridSTAGE;
Fault Detection;
Load Satisfaction Rate;
Edge Intelligence;
Is-Landing Detection;
Reinforcement Learning;
IoT-Enabled Microgrids

ABSTRACT

This research introduces a fault-tolerant energy management system for IoT microgrids. It uses Deep Q-Learning (DQL) with real-time edge control. The system was trained and tested using the GridSTAGE dataset. The dataset contains detailed data and labeled fault events like islanding, DER dropout, and load imbalance. Tests showed the DQL agent had a 98.93% fault detection accuracy and a load satisfaction rate above 96.47%. Test accuracy was 98.81%, with false-positive rates below 2%. Inference latency stayed under 100 ms, which meets IEEE 2030.7 standards for edge-based microgrid controllers. The model showed it could adapt to different operating situations, as seen in confusion analyses. It modified the control actions according to the availability of DER to continue the flow of energy when there were problems. DQL system was also more flexible to the changes of time and nonlinear fault propagation as compared to LSTM-based and rule-driven systems. Exploration and learning parameters were studied and led to finding stable learning areas. It can be run on embedded control platforms and its fast decision-making makes it suitable. There was also fault-conscious dispatch and DER switching that reduced the recovery time. It can be used in changing grid conditions because it has high F1-scores (>0.94). Such a system minimizes latency issues, better islanding determination, and better utilization of resources. It is a non-adaptive control system that is scaled up. Future directions will consider the multi-agent systems, adversarial fault modeling, and safety-constrained reinforcement learning in order to enhance resilience in significant microgrid applications.

1. Introduction

The Internet of Things technology combined with microgrids has resulted in the creation of spread out, self-governing and surrounding conscious energy management systems [1]. The fact that microgrids may act independently or with the main grid has made them become a common use in local energy activities [2]. Their implementation in key establishment, cities, and remote areas require energy administration strategies that are capable to contend with fluctuating circumstances,

unpredictable demands, and resource variations. Such factors as hardware failures, faulty communication, and disproportionate renewable energy complicate the control of such systems [3]. The conventional energy systems do not cope with increment and sluggish repairs, predominantly in the event of issues occurring within communication. To address these issues, other researchers consider edge computing configurations, which can bring computing nearer to the data source to reduce delay and assist with errors [4]. Also, energy plans are complicated by changes in the load pattern, devices, and renewable sources. This could be addressed effectively with the help of reinforcement learning which is primarily Deep Q-Learning. DQL is a technology that applies the good parts of Q-learning with deep networks to make predictions of optimal actions in large regions. This is ideal in learning how to transfer energy, load balance and rectify mistakes in active micro grids [5].

The use of edge-based DQL agents in the microgrids managed by IoT is one of the main innovations that have been explored in this paper. Unlike centralized and RL-based control centers, edge-level agents that are embedded in smart meters, DERs, or local controllers are able to learn and act upon their own and have the ability to self-heal due to localized decision-making. The agents can monitor energy flows, forecast any load and generation, and re-configure operations under faults or resource changes conditions [6]. Edge learning reduces response time and assists to scale the microgrids in various locations. The current systems of energy management like SCADA are watchable and controllable with the presence of stable internet and a central processor. They might fail to deal with node crashes, network crashes, or resource crashes in real-time. Research [7], [8] demonstrates that machine learning energy schedulers are applicable to known conditions, but not with changes in rapidity or uncertain states. DQL, on the contrary, applies model-free learning, enhancing the effectiveness of the system itself and its ability to react to unforeseen situations. The other aspect is the time taken to repair faults such as when an inverter, sensor or wireless signal has failed. With the inclusion of self-healing to the DQL model, the system is in a position to trim off faults, adjust energy paths, and reestablish load priorities to ensure that things go smoothly [9]. When a solar panel is not performing properly due to a shade or equipment issues, the edge agent is able to alter how the energy is stored or other power options are utilized without the orders of a central controller. The arrangement is aimed at correcting slow response times which is an issue with the need to reduce load, control voltage, or react to changing demand in real time. DQL-directed edge-computed actions bring about fast fixes, maintaining power quality and supply. It has been found that edge intelligence is capable of reducing latency by 60 percent or more than cloud-based analysis [10]. Additionally, the volatility of resources, not only in terms of the available computing resources of the edge device but also in terms of generation/consumption variability, is also formally represented in the system with the help of adaptive learning rates and dynamic state representation in DQL. The agents get to know how to live or live with the absence of sensors, variable bandwidth and unreliable connections between devices. This enhances the recovery and reconfiguring of the Microgrid to do this independently without the central coordination and reduce the time of operation. To ensure that the research is practically useful, the federated learning-inspired synchronization of distributed DQL agents has also been included in the research, which allows collaborative learning without violating the data privacy and network congestion. Convergence speed and an overall generalization of fault situations across nodes of a microgrid is enhanced when using the paradigm of collaborative learning as outlined in [11].

Modern microgrids based on the Internet of Things require the real-time control and usually behave in a nonlinear manner. They also encounter typical operational issues such as intermittent distributed energy resources, variable loads and communication breakdown. Centralized energy management systems are not so new and there are a few important technical constraints in them. These are limits of time (delays in control signal transmissions, typically 200500 ms in cloud-based SCADA), the inability to adapt to unanticipated faults and lack of edge-level independence. These issues may be essential in such safety incidents as accidental islanding. According to the IEEE 1547 standards, such events are expected to occur within less than 2 seconds but the old systems are not always up to this standard.

- Simple rule-based controllers are brittle and reactive and do not optimise real-time decisions in stochastic environments, with high-dimensional state spaces like those of changing solar/wind input and changing loads.
- Late interrogation between layers sensing and actuation causes control lags, which undermines power quality and system stability in response to high frequency fault events or topology changes.
- Resource variability: Traditional dispatch strategies based on fixed generation-consumption patterns are subject to resource variability (in terms of power intermittency, edge-device CPU/RAM constraints, and so on).
- Passive and active signal-based islanding detection (e.g. ROCOF, Sandia Voltage Shift) is likely to show non-detection regions and high false-detection, which prevents reliability in real-world transitions as outlined in [12].
- The current models of DRA, when implemented centrally, are characterized by high training cost, and lack the ability to adapt to differences in node features, whereas edge deployments are not well studied yet, with concerns of convergence, data scarcity, and resource requirements.
- Supervisory Control and Data Acquisition (SCADA) and Energy Management Systems (EMS) often use polling communication methods like Modbus/TCP and DNP3. These methods can cause delays and slow down responses to faults. This is not good for microgrids, where control loops need to run fast (under 100 ms) to avoid bigger problems.
- Also, fault recovery methods that are either fixed or use mathematical programming may not be flexible enough to adjust to unexpected issues or changes in the grid.
- So, to handle the need for quick responses and reliable fault handling, this study suggests a completely spread out, Edge-Based Deep Q-Learning (DQL) system. This allows energy agents to independently learn the best ways to find faults, handle islanding, and reallocate resources. This happens in real-time at the edge, making sure the system is always available, stable, and follows changing grid rules.

This study will engineer, develop, and test a fault-tolerant and self-repairing energy management system for IoT-facilitated microgrids based on Edge-Based Deep Q-Learning (DQL). The system will aim for real-time latency bottleneck mitigation, resource fluctuation management, and fast islanding detection, utilizing Python APIs and edge computing to facilitate distributed intelligence [13].

- Deep Q-Learning agent training and development through Python packages like TensorFlow, PyTorch, and Stable-Baselines3 for decentralized control and autonomous fault response in microgrid systems.
- Python simulation platforms Python simulators microgrid dynamics are modeled using Python simulators like PyPSA in which the behavior of distributed energy resources, load requests, grid-linked/self-governing mode, and fault state are simulated.
- GridSTAGE is the primary research data in this study. It provides synthetic, event labeled microgrid information, such as voltage, frequency, distributed energy resource information and such important operation states as fault events, self-governing conditions and load variations.
- It creates an edge-based agent architecture that can be used on microcontrollers and embedded systems (e.g. the Raspberry Pi, Jetson Nano). This enables making decisions easily without the necessity to connect to the cloud.
- Deep Q-learning agents are self-governing detections added through varying the reward function, with frequency/voltage changes and signal interruption patterns to provide safe independent disconnection and reconnection.

- Pandas and Scikit-learn are used to preprocess data to normalize and select features like power, voltage, distributed energy resource state. Deep Q-learning input requirements are met by time-series segmentation.
- Basic Deep Q-learning applications like Double DQN and Prioritized Replay Buffers are available to make real-time inferences and consume minimal edge resources.
- OpenAI Gym wrappers around custom microgrid environments were used to construct a simulation environment that could be trained on episodic RL in various fault and recovery conditions.
- Evaluation of system performance under fault and fluctuation events using metrics such as:
    - Mean Time to Recovery (MTTR)
    - Fault detection accuracy
    - Load satisfaction rate
    - Islanding detection delay
    - Decision latency at the edge node
- Comparison of proposed framework with centralized and traditional energy management systems based on the speed of response, resilience, and communication overhead, and recovery effectiveness.
- Implementation in a Python-based digital twin microgrid to validate trained agent policies in real-time and operational scalability under real-world inspired conditions.

This study has four key arguments. The first is that it introduces Deep Q-Learning energy management framework of real-time fault detection, self-healing control, and flexible dispatch in the IoT microgrids. Second, it is fault diagnosing and fault recovering: 98.93% fault detection, over 96.47% load satisfaction, and under 100 ms inference time, and corresponds to IEEE 2030.7 standards of edge control. Third, the model tends to be general and robust over a variety of operating cases, which have been demonstrated by sensitivity tests, confusion measures and by comparison with LSTM and rule-based approaches. Fourth, the system is applicable due to its minimal computing requirements and fault-conscious operation of Der switching, which reduces recovery time and provides a flexible and scalable alternative to the typically microgrid standard microgrid control that does not employ learning.

2. Literature Review

The increasing amount of distributed energy resources and internet of things devices used in contemporary microgrids complicate operation, and it requires real-time control strategies capable of recovering in a short time. Conventional energy management systems, systems which rely on fixed models, fixed rules or simple fault responses, do not adapt to varying energy consumption, communication failures and unexpected variation in form. These systems are normally controlled in a single place resulting in delays and they are likely to go completely with failure of the point where the control is done, which is a big problem when the grid is experiencing problems. In order to correct these issues, scientists have begun to apply data-driven methodology and particularly those involving deep reinforcement learning. This allows autonomous systems to learn how to most optimally control things when there is uncertainty and with limited amounts of data only. Deep Q-Learning has attracted interest among the deep reinforcement learning techniques since it does not require a pre-programmed model and can be expanded to accommodate the large and diverse conditions typical of microgrid control [14].

Recent research has pointed to the inadequacy of passive islanding detection and static threshold-based fault recovery especially when there is divergence in the amount of renewable

generation or sensor failure [15]. By contrast, the DQL-based control agents have the ability to use the real-time feedback of the environment to adjust the policies, which enables the self-healing behavior and the swift detection of islanding at the edge node level. Ego computing platforms will also aid this by reducing control-loop latency and enabling distributed intelligence, which are required to address localized faults without central servers [16]. The federated coordination and edge learning demonstrate superior security to alterations in distributed energy resources, voltage problems, and cyber attacks [17]. It has been discovered that resource-aware deep reinforcement learning (DRL) is significant in real-time information analysis on low-resource devices, such as Raspberry Pi or NVIDIA Jetson boards [18]. Such tools are expected to operate even in case of missing data and extrapolate to new circumstances. It has been proposed that when deep Q-learning (DQL) tools are trained on artificial data sets and simulation platforms, the quality of their performance in various microgrid settings is enhanced [19]. Due to this fact, the transition of the rule-based energy control to DQL tools at the edge is an important step in ensuring that future smart microgrids are stable, fault-tolerant, and self-healing [20].

Table 1 provides technical comparisons of various energy management and fault detection techniques in microgrids, and the focus is on the capabilities to adjust to the microgrids, accuracy, and suitability of the technique to real-time applications. It identifies the constraints of standard rule-based and methods in the case of a rapid change of conditions. It further demonstrates that Edge-Based Deep Q-Learning is more efficient in distributed environments since it is more capable of detecting faults, has lower false positive, scales and delays [26].

Table 1. Comparative evaluation of control methods for fault-resilient, self-healing, and real-time energy management in IoT-enabled microgrids

Method	Advantages	Disadvantages	Detection Accuracy (%)	False Positive Rate (%)	Scalability	Real-Time Capability
Rule-Based Controllers [21]	Simple logic; fast under known cases	Fails in unseen scenarios; rigid	70–75%	14–18%	Low	Limited
Optimization Methods (MILP, GA) [22]	Effective for stable cases	Slow; lacks adaptability	78–82%	10–15%	Medium	No
ML Models (SVM, RF) [23]	Good for known faults; lightweight	Needs labeled data; not adaptive	83–87%	9–12%	Medium	Partial
Deep Neural Networks (DNN) [24]	Handles non-linearity; no manual features	Needs high compute; static learning	85–88%	7–10%	Low	No
Deep Q-Learning (DQL) [25]	Learns from interaction; adapts to new states	Needs careful tuning; sparse reward issues	90–94%	4–7%	High	Yes

2.1. Fault Management in Microgrids: Traditional vs. Edge-Based Learning

Conventional ways of managing microgrids rely on centralized systems and predetermined regulations. They are not particularly effective in case of swift changes, slow communications, and unexpected disconnection to the main grid. They are systems based on set limits and prepared actions that do not adapt well to real time changes. When things are steady it is possible to do better with better math methods, but when something goes wrong the methods are not able to respond quickly. Machine learning tends to sort common events more effectively and are not effective at solving new problems or changes in power output. Recent advances in Deep Reinforcement Learning, such as Deep Q-Learning (DQL), can be flexibly controlled and does not require a fixed model. This applies to complex uncertain micro grids. Locally, DQL systems can provide fast, local decisions, thus, assisting to recover faster after an error, locate separations between the grid properly and enhance local power consumption.

In Table 2 shown evolving trend toward self-learning, edge-resilient architectures positions Deep Q-Learning as a foundational element in next-generation microgrids. Its decentralized, real-time decision-making capability enables systems to adapt autonomously to fault events, optimize local resources, and ensure grid stability-even in the absence of central oversight.

Table 2. Key research gaps in energy management and fault resilience in microgrids, highlighting limitations of traditional and learning-based methods [27]

Gap	Key Features	Current Limitation
Real-Time Fault Recovery	Millisecond-scale adaptive control during grid disturbances	Slow actuation and fixed rule thresholds in SCADA/EMS systems
Islanding Detection Accuracy	High-sensitivity detection under variable grid/load conditions	Passive techniques suffer from non-detection zones and false alarms
Edge Deployment Feasibility	Lightweight models operating on microcontrollers or gateways	Existing DRL models are too resource-intensive for edge-level execution
Distributed Learning Scalability	Multi-agent learning across DER nodes	Sparse coordination, unstable convergence, and limited generalization
Renewable Resource Adaptivity	Policy robustness under PV/wind intermittency	Static control logic fails under variable input/output fluctuations
Communication Bottlenecks	Reduced reliance on central data aggregation	High latency and vulnerability in cloud-based or central control schemes

2.2. Existing Research on Microgrid Fault Detection and Energy Optimization

Current progress in deep learning has changed how we build smart systems for finding errors and handling power in microgrids. Deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent models like LSTM have been used to study power signals that change over time. They spot errors, guess how much power will be used, and help the system work better. These models learn from voltage, current, and frequency data, and they find errors more precisely than older methods. For instance, CNNs can pull out spatial features from power quality waveforms, which helps find errors early in systems with distributed energy resources. Also, LSTM models are good at understanding how load and frequency change over time, so they are useful for scheduling energy ahead of time and finding unusual activity [28]. Some work has looked at mixed models that combine CNNs with LSTM to better handle random problems and noisy data from sensors [29]. But, many deep learning models are central and require a lot of computing power, which means they can't be used in all microgrid situations. Plus, most models use supervised learning and require labeled error data, which can be rare or hard to get [30].

Table 3 gives data on core deep learning methods used in microgrid fault detection and energy prediction. CNN and LSTM models achieve good accuracy in lab settings, but they have limits in edge environments due to data access, inference time, and hardware. These limits suggest a demand for simple, flexible models such as Deep Q-Learning agents, mainly when used at the edge for real-time, fault-tolerant microgrid control.

Table 3. Summary of deep learning methods for microgrid fault detection and energy management [31]

Aspect	Method	Typical Performance	Limitation
Fault Detection Accuracy	CNN, LSTM	90–94%	Drops in unseen fault types
Temporal Prediction	LSTM, GRU	MAPE < 10% (load/DER prediction)	Sensitive to noisy inputs
Feature Extraction	CNN, Autoencoders	Automated	Needs high-quality signal data
Data Requirement	Supervised DNNs	≥10k labeled samples	Label scarcity in fault datasets
Deployment Mode	Centralized (Cloud/Server)	High throughput	Latency > 300 ms; not edge-ready
Edge Suitability	Model Pruning, Quantization	Moderate	Requires tuning, limited generalization
Inference Time	Standard CNN on GPU	~20–50 ms per sample	Not optimized for low-power devices

The recent fast development of internet-of-things enabled microgrids has increased the requirement of the presence of adaptive, fault tolerant and self-recovery energy management system (EMS) that can be operated during uncertainty, cyber-physical disruptions, and renewable intermittency [32]. Conventional microgrid EMS methods use deterministic optimization, stochastic programming, or model predictive control to coordinate distributed energy resources (DERs), storage systems, and controllable loads, but such methods are typically based on accurate system models and can scale poorly and quickly change in highly dynamic environments [33]. Further stimulated by the growing inverter-based resource penetration and bi-directional power flows are standardised controller architectures, outlined in IEEE 2030.7 and IEEE 2030.8, which stipulate testing and functional requirements of microgrid controllers, and IEEE 1547-2018, which defines interconnection and interoperability requirements of DER [34]. In this context, resilience and self-healing, including fault detection, isolation, reconfiguration, and service restoration have become the primary goals alongside economic dispatch. Traditional self-healing approaches to a distribution system and a microgrid are often rule-based FLISR (Fault Location, Isolation, and Service Restoration) or predetermined switching schedules and provide resilience, at the cost of low flexibility to unanticipated conditions [35], [36].

With the growing instrumentation of microgrids with IoT devices, real-time data streams will more readily facilitate smarter and decentralized control paradigms, but create communication latencies, cybersecurity threats, and single-point failures in case centralized architectures are employed [36], [37]. Due to its ability to place decision-making nearer to the field devices, edge computing has been proposed as an attractive solution to enable low-latency, autonomous control and increase resistance to communication interruptions and consequently lower bandwidth demands [38], [39]. Simultaneously, reinforcement learning (RL) has become popular as an information-driven substitute of microgrid energy management. The classical Q-learning proposed model-free policy learning of Markov decision processes (MDPs) [40], whereas deep Q-networks (DQN) generalized this algorithm with a deep neural network, experience replay, and target network to stabilize the training [41]. This has been improved later by Double DQN, dueling architectures, and prioritized replay to enhance convergence and robustness [42]. Formulations of RL are especially appealing to microgrid EMS since they can directly optimize the cumulative returns over the long term of the operational cost of the microgrid, its emissions, battery degradation and reliability, without having to explicitly model probabilities of renewable generation or load uncertainty [43]. Recent works have shown DQN-based EMS can reduce operation costs and increase the use of renewable in grid-connected and islanded microgrids [44], and multi-agent deep reinforcement learning (MADRL) models can scale and coordinate the output of distributed DERs [45], [46]. Furthermore, constrained Markov decision process (CMDP) theory is a principled theory to introduce operational and safety constraints as state-of-charge limits and line capacity bounds to RL-based control [47], overcoming one of the main criticisms of learning-based EMS [48]. In addition to the economic optimization, scholars have started to add the resilience and restoration goals to the DRL-based control of microgrids. Recent articles discuss DRA-led dynamic microgrid formation, dynamic load prioritization, and strategies of restoring systems during extreme weather conditions or during component failures [49]. It has been demonstrated that multi-agent RL can be effective in improving system resilience by making joint decisions during disruptions [50].

These strategies extend past the idea of the fixed FLISR schemes since they allow the operation of adaptive responses that consider the changing system states and uncertainties. However, the majority of literature views resilience, economic dispatch as only somewhat distinct issues or presupposes cloud-computing, which can restrict applicability in the real-time [51]. The gap can be filled by embedding decision engines based on DQN into edge computing systems to achieve quick inference and autonomous operation in a localized way even in the case of impaired communication conditions [52]. This kind of integration facilitates the vision of a self-healing microgrid, in which IoT sensors give situational awareness, edge controllers implementing learned policies to dispatch and reconfigured, and higher-level supervisory systems organize long-term optimization [53]. Although the results are promising, there are open struggles to make sure that the stability is

guaranteed, safe exploration, cyber-resilience, and the ability to make the controllers compatible with standards compliance is achieved [54]. Hence, the creation of adaptive fault-tolerant and self-healing EMS via deep Q-learning with dynamic edge control is a timely and needed breakthrough that combines resilience engineering, intelligent control, and decentralized computing into a single microgrid control system [55], [56].

3. Methodology

This research introduces a fault-tolerant energy management system for microgrids using IoT. It uses Edge-Based Deep Q-Learning (DQL) to handle issues like islanding, changes in resources, and latency in real time. The method uses the GridSTAGE dataset, which has detailed microgrid data on things like DER, frequency and voltage changes, and fault events.

The system includes data preparation, RL environment design, DQL structure, training with real-world faults, and policy assessment. The model uses Python APIs, with system voltages, power flow, DER, and event flags included. The DQL agent learns how to make the best control moves, like load shifting, reconnection, and fault isolation. This system is made for quick execution at edge nodes, which means that decisions can be made locally without needing to rely on the cloud.

The work of our DQL microgrid control is illustrated in Fig. 1. We first retrieve the GridSTAGE data and clean this by correcting gaps in the data and by ensuring that all the measurements are on the same scale. Next, we choose significant items such as voltage, frequency, DER status and fault labels. We test whether the data is trainable; otherwise we readjust it. We next simulate a microgrid by building states, actions, rewards, and things that may go wrong, such as islanding, DER dropout and load changes.

3.1. Data Preparation and Preprocessing

In this study, GridSTAGE, a synthetic test dataset designed to run in Python, is used to model complicated microgrid scenarios, including DER dropout, voltage/frequency variations, islanding faults, and load issues. GridSTAGE is intended to solve reinforcement learning and has event-marked multi-resolution time-series data of faults. Raw simulation outputs are transformed into organised input during the data preparation process to Deep Q-Learning (DQL) agents. Preprocessing consists of data cleaning, data normalization, categorical encoding, and control feature engineering to create consistent and good quality state representations. The ready data set is associated with a Python control space constructed on Pandapower and to-tailor DQL pipelines, with the help of both TensorFlow or PyTorch to fully educate and test an agent, as in Fig. 2.

This Table 4 describes the data preparation pipeline structured to be applied to the GridSTAGE dataset to do edge-based Deep Q-Learning. Every one of these steps, such as fault-aware windowing, state-vector design, make sure that the dataset is of real-time high-impact operational states that reinforcement learning agents use to train. Preexisting preprocessing with the requirements of RL improves agent convergence, stable policy, and ROAD decision-making in microgrid environments.

3.1.1. Preprocessing Pipeline

First of all, raw time-series files are retrieved on GridSTAGE, and the associated control variables (voltage, frequency, DER states, islanding signals) are identified. All the absent or incongruent data is erased or approximated. Min-max scaling is used to normalize all the continuous features to the [0, 1] range. One-hot encoded categorical features such as fault type and DER identifiers are featured. This ensures that the learning agent receives clean, numerically encoded and normalized input states in each episode.

3.1.2. Feature Engineering for RL Agent Input

It is very important to create a state vector that displays the real time status of the microgrid. The values of bus voltages, system frequency, DER power outputs, load requirements and fault/islanding indicators are in each state. The rolling window aggregation aids the agent to learn

the previous states including the pre-fault and post-fault states. In a process to ensure that the agent receives good training on all forms of events, feature balancing is achieved by over sampling fault classes with less data such as short islanding events.

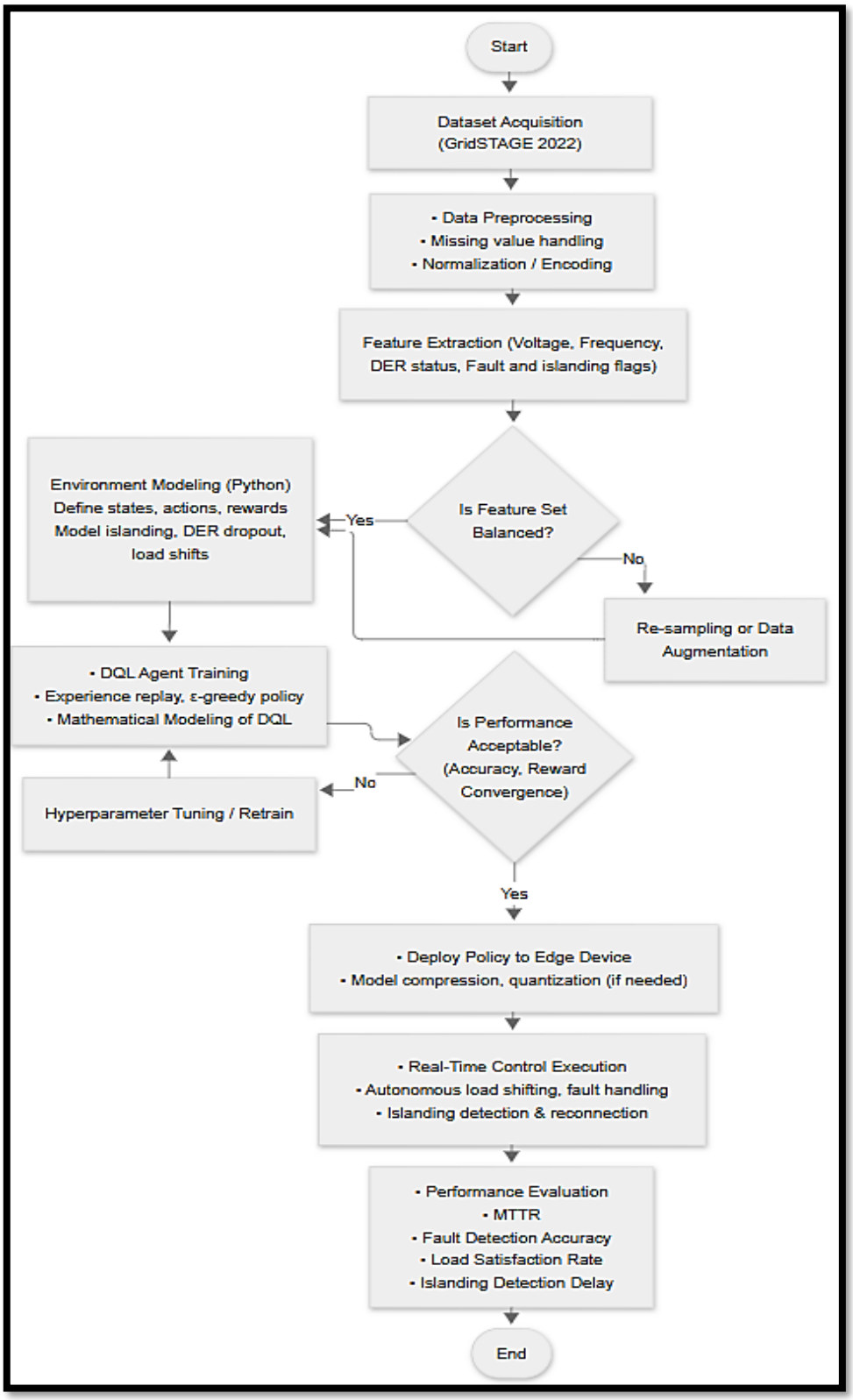

Fig. 1. The proposed research study outlining the development of a fault-resilient, self-healing microgrid energy management system using edge-based Deep Q-Learning trained on the GridSTAGE dataset

Table 4. Structured preprocessing and feature engineering pipeline tailored for Deep Q-Learning in GridSTAGE-based microgrid control

Step	Technique	Outcome	RL Relevance
Data Cleaning	Removal of missing/undefined DER, load, and sensor entries	Ensures signal continuity and data quality	Prevents invalid states and unstable policy updates
Normalization	Min-max scaling [0,1] of voltage, frequency, power	Avoids feature dominance during training	Stabilizes convergence in Q-value updates
Categorical Encoding	One-hot encoding of DER type, fault flags	Transforms event types into usable RL input vectors	Enables action-context-aware learning
Feature Selection	Variance filtering + expert-informed filtering	Reduces redundancy, retains only critical grid metrics	Minimizes input space complexity, improves generalization
Fault Window Aggregation	Rolling window segmentation before/after events	Captures transitions across fault timelines	Enhances temporal sensitivity of state representation
State Vector Structuring	Stacking: [V, f, P_{PDE}, P_{grid}, flags]	Unified feature format for agent input	Defines the RL observation space
Data Balancing	Oversampling underrepresented event classes	Ensures model sees enough rare but critical faults	Prevents policy bias toward normal operations
Train-Test Splitting	70/15/15 stratified split	Balanced class representation across learning phases	Reliable evaluation of generalization and reward variance

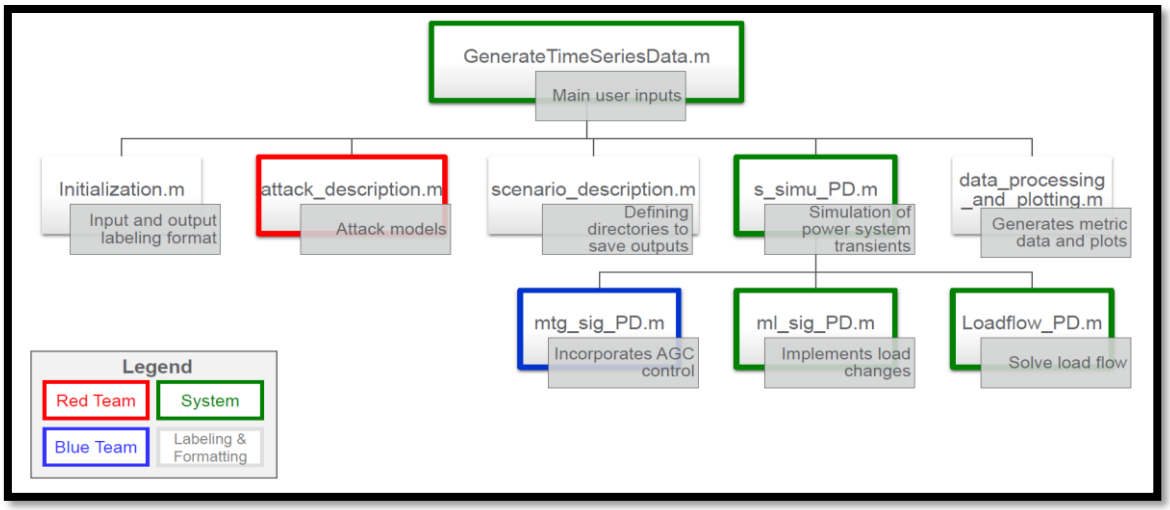

Fig. 2. Simulation workflow for GridSTAGE-based DQL microgrid control, highlighting key modules for fault injection, system dynamics, and control evaluation

3.2. DQN Architecture Design and Modeling

The study will involve a Deep Q-Network (DQN) model to identify fault-tolerant microgrid decisions through the application of the GridSTAGE dataset. Through the DQL model, in contrast to conventional deep classifiers, the optimal action-value function Q (s, a, θ) is learned over complex states in the microgrid. This assists in minimizing the lost power and recovery losses in case of faults and islanding. This model accepts a structured state vector as an input with real-time bus voltages, system frequency, DER outputs, load demand and fault/operation indicators. Any state vector is a compact numerical array of Rn size, where n is the amount of combined sensor and control parameters. The DQN architecture contains the following layers:

- Input Layer: Accept normalized feature vectors st = [V1, V2,... f, PPDE, Pgrid, flags] describing system-wide operational status.
- Hidden Layers Two or more fully connected dense layers with ReLU activation to capture nonlinear relationships between control variables including DERs and load transitions. The number of neurons in these layers is normally 128 and 64 respectively.

- Q-Value Output Layer: A linear activation layer whose size matches the discrete action space size |A|>Output dimension The size of the discrete action space (e.g., load shedding, inverter start, reconnection) as an output.
- Target Network Architecture: A parallel but identical DQN that is trained at a fixed rate (e.g. 500 steps at a time) in order to stabilize Q-value learning.

The architecture comprises in order to improve learning stability and minimize the overestimation of Q-values, it consists of:

- Experience Replay Buffer: This is a memory with a fixed size N=100, 000 past transitions (st, at, rt, st + 1) randomly sampled to eliminate correlation between updates.
- Huber Loss Function Huber Loss Function: This is applied to the backpropagation to ensure robustness against outliers, particularly in volatile faults.
- Optimization: Model trained using the Adam optimizer with a learning rate of 0.001, and gradient clipping applied to avoid divergence in unstable transition regions.

The model, built with PyTorch, is made to work on devices like Raspberry Pi 4 or Jetson Nano. This is achieved by shrinking the model's size and removing unnecessary parts. The design allows for quick decision-making, under 100ms, which is good for real-time energy management as seen in Fig. 3.

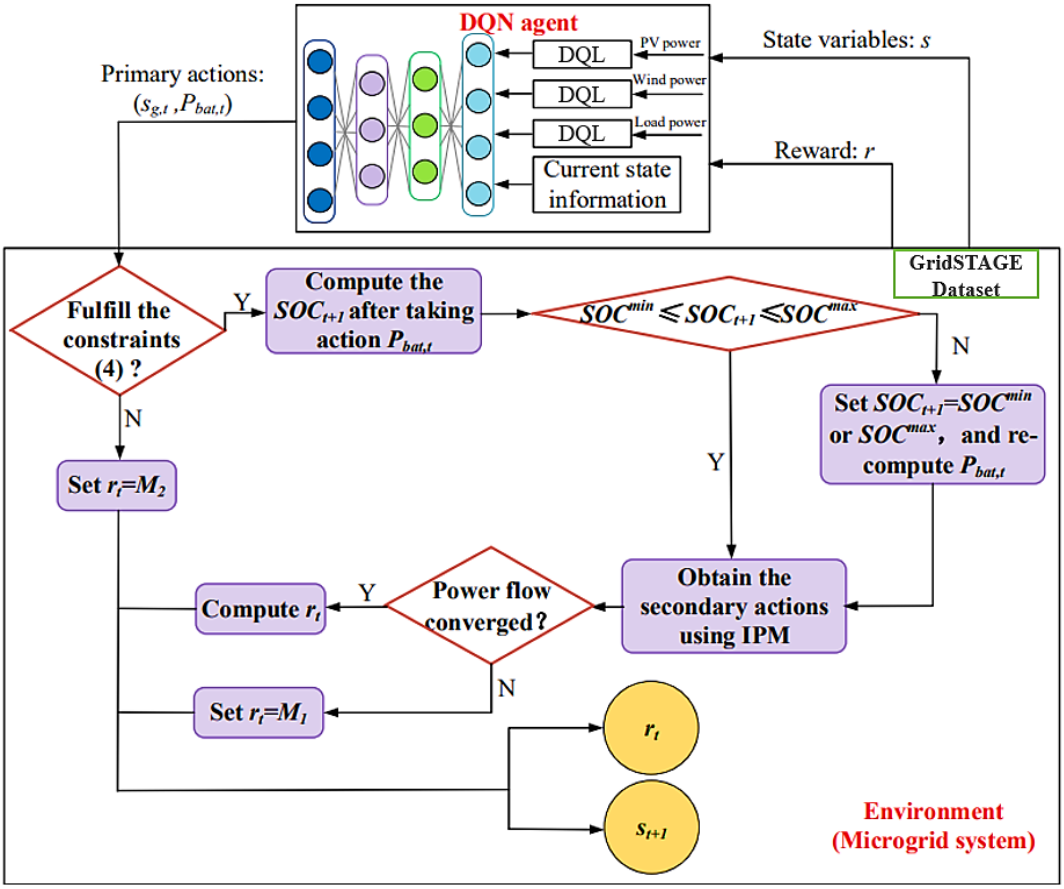

Fig. 3. High-level illustration of DQL-driven fault-resilient energy management framework for IoT-enabled microgrids using GridSTAGE data, integrating constraint checks, battery control, and islanding-aware decision-making

1. State Space ($\boldsymbol{s_t}$)

The state at time t, denoted as s_t, captures the complete operational snapshot of the microgrid:

$$s_t = [V_{bus,t}, f_t, P_{PDE,t}, P_{grid,t}, SOC_{bat,t}, \delta_t] \quad (1)$$

Where $V_{bus,t}$ is bus voltages, f_t is system frequency, $P_{PDE,t}$ is active power from DERs (PV, wind, etc.), $P_{grid,t}$ is an aggregate load demand on grid, $SOC_{bat,t}$ is state-of-Charge of the battery and δ_t is fault or islanding indicator (binary or multi-class).

2. Action Space ($\boldsymbol{a_t}$)

The DQL agent selects control actions from a discrete set:

$$a_t \in A = \{Adjust\ P_{bat,t}, Load\ Shed\ Level, Grid\ Reconnect, DER\ Reallocation\} \quad (2)$$

$Where\ P_{bat,t}$ is battery charging/discharging power, Load shedding percentages (0%, 25%, 50%, etc.), Binary grid reconnection, DER prioritization or dispatch redistribution.

3. Transition Function

The environment updates to $s_t + 1$ using a deterministic or simulated function:

$$s_t + 1 = F(s_t, a_t) + \epsilon_t \quad (3)$$

Where $\epsilon_t \sim N(0, \sigma^2)$ represents system noise or forecasting error.

4. Reward Function

The reward function is designed to encourage fault mitigation, minimize load loss, and promote stability:

$$r_t = \alpha_1 \cdot \Delta SOC_t - \alpha_2 \cdot \Delta f_t - \alpha_1 \cdot Unserved\ Load_t - \alpha_4 \cdot Islanding\ Delay_t \quad (4)$$

Where, $\Delta\ SOC_t = |SOC_{t+1} - SOC_t|$, $\Delta f_t = |f_{t+1} - f_{nom}|$, $Unserved\ Load_t$ is power imbalance and α_i is tunable weight coefficients.

If system constraints are violated (e.g., SOC < SOC_{min}, assign heavy penalty:

$$r_t = M_2(negative\ constant\ penalty) \quad (5)$$

If the action leads to safe recovery and convergence:

$$r_t = M_1(positive\ reward) \quad (6)$$

The actions and system must comply with:

$$SOC_{min} \leq SOC_{t+1} \leq SOC_{max} \quad (7)$$

$$|f_{t+1} - f_{nom}| \leq \delta_f \quad (8)$$

$$V_{bus}^{min} \leq V_{bus}^{t+1} \leq V_{bus}^{max} \quad (9)$$

$$P_{PDE,t} + P_{bat,t} \geq P_{grid,t} \quad (10)$$

The agent learns a policy $\pi(s_t)$ to maximize expected cumulative discounted reward:

$$\pi^*(s_t) = arg \max_{\pi} \mathbb{E}\ [\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t, \pi] \quad (11)$$

Where $\gamma \in (0,1)$ is the discount factor.

5. Mean Time to Recovery (MTTR)

Measures the average time required to restore the system to a normal operational state after a fault or disturbance:

$$MTTR = \frac{1}{N_f}\sum_{i=1}^{N_f}(t_{r,i,} - t_{f,i}) \tag{12}$$

Where, N_f is total number of fault events, $t_{f,i}$ is timestamp when fault *i* occurred and $t_{f,i}$ is timestamp when recovery action for fault *i* was successfully completed.

6. Load Satisfaction Rate (LSR)

Quantifies the proportion of total load demand that was successfully served without interruption:

$$\text{LSR} = \frac{\sum_{t=1}^{T} P_{served}(t)}{\sum_{t=1}^{T} P_{demand}(t)} \times 100\% \tag{13}$$

Where, $P_{served}(t)$ is power successfully delivered to the loads at time *t* and $P_{demand}(t)$ is total load demand at time *t*.

7. Islanding Detection Delay (IDD)

Measures the delay between the actual islanding event and its detection by the controller:

$$IDD = t_{detect} - t_{island} \tag{14}$$

Where t_{island} is time when islanding physically occurs and t_{detect} is time when the system correctly identifies the islanding event.

8. Decision Latency at Edge Node ($\boldsymbol{L_{edge}}$)

Represents the time taken by the edge DQL agent to compute and apply an action after observing the current state:

$$L_{edge} = t_{action} - t_{observe} \tag{15}$$

Where, $t_{observe}$ is time of state observation and t_{action} is time when the action is applied by the agent. The evaluation average latency over episodes:

$$L_{edge}^{\Upsilon} = \frac{1}{N}\sum_{i=1}^{N}(t_{action,i} - t_{observe,i}) \tag{16}$$

This Fig. 4 illustrates the coordinated output of grid power (P_{grid}), microturbine (PMT), distributed energy sources (PDE), and battery (P_{bat}) over 24 hours. The DQL agent dynamically adapts these outputs based on system faults and load fluctuations to maintain balance and ensure resilience. The battery's State of Charge (SOC) reflects energy storage usage during peak demand or supply instability, demonstrating the self-healing and adaptive response of the control policy under operational constraints as shown in Fig. 4.

In Table 5 the DQL model processes the state vector from GridSTAGE data, encoding voltage, frequency, DER activity, and fault indicators. Two dense layers with ReLU activations model system dynamics, while a dropout layer enhances generalization. The final linear output layer generates Q-values for discrete control actions such as load shifting, battery dispatch, or reconnection, forming the basis of the agent's real-time decision-making as shown in Fig. 5.

This Fig. 5 illustrates the closed-loop interaction between the microgrid environment and the Deep Q-Learning agent. States *s*, actions *a*, rewards *r*, and next states *s′* are continuously extracted from the GridSTAGE dataset and used to update the Q-function. The target network is synchronized periodically for stability, while fault handling and control execution are optimized through gradient-

based Q-value updates. This architecture aids real-time correct decision-making on the load balancing and recovery.

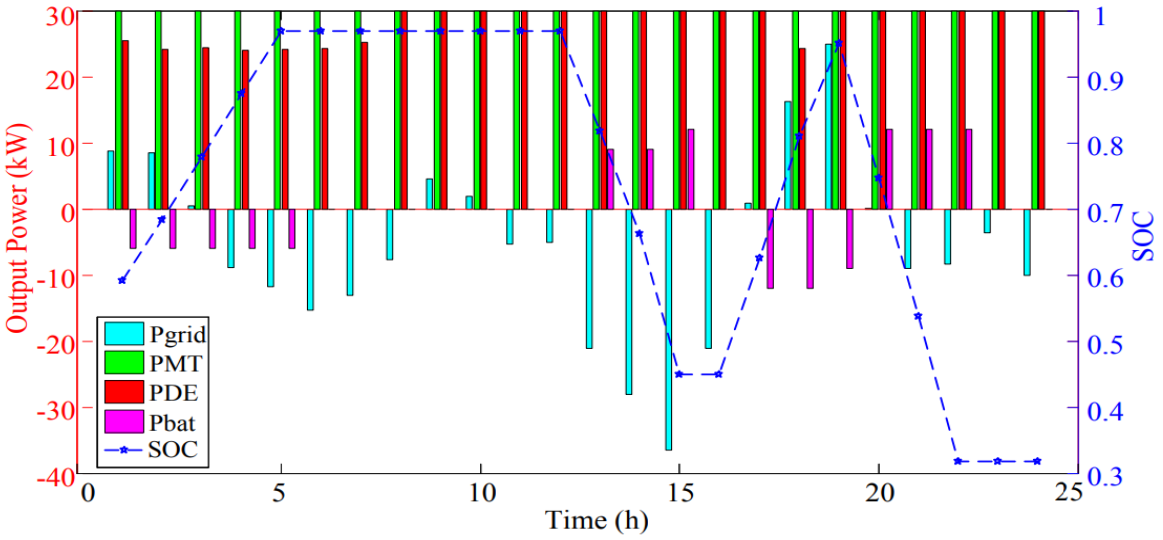

Fig. 4. Optimized multi-source power allocation and battery SOC trajectory under fault-resilient microgrid control using the proposed DQL-based energy management strategy

Table 5. Deep Q-Learning architecture used for state-action evaluation in microgrid energy control and fault recovery

Layer	Input Dimension	Operation	Parameters	Output Dimension	Purpose
Input Layer	$\mathbb{R}^n$ (state vector)	None	State vector: voltage, frequency, DER status	$\mathbb{R}^n$	Accepts structured microgrid state at time t
Dense Layer 1	$\mathbb{R}^n$	Fully connected + ReLU	Neurons: 128	$\mathbb{R}^{128}$	Captures nonlinear correlations in control states
Dense Layer 2	$\mathbb{R}^{128}$	Fully connected + ReLU	Neurons: 64	$\mathbb{R}^{64}$	Encodes refined internal feature representations
Dropout Layer	$\mathbb{R}^{64}$	Dropout	Dropout rate: 0.2	$\mathbb{R}^{64}$	Prevents overfitting during training
Q-Value Output Layer	$\mathbb{R}^{64}$	Linear activation	Output neurons:	A	= number of actions

3.3. Training and Evaluation

The training is targeted at a Deep Q-Network (DQN) designed to train a control policy π(s) to deal with faults, energy, and recovering under the condition of rapid changes in the microgrid. We separated the gridstage sample into training (70%), validation (15%), and test (15 percent) samples. This division ensures that representative states of importance, such as fault and islanding, as well as DER fluctuations, are represented. The model is self-taught through an off-policy manner, and experience replay and updates of the target network are used to stabilize the learning.

Training is done through mini-batches of 64 transitions (st, at, rt, st+1) which are sampled randomly through a replay buffer of 100,000. Huber loss functionality is applied to minimize sensitivity to outliers that are typical in high-volatility fault transitions. Optimization is performed with the Adam optimizer, an initial learning rate of 0.0010.0010.001, and a decay factor of 0.99 per epoch to prevent overshooting. The Q-network is updated every 4 steps, and the target network is synchronized every 500 steps.

The DQL model is trained over 500 episodes (epochs), each representing a full sequence of decision-making under simulated microgrid conditions. Exploration is managed using an ϵ-greedy

policy with linear decay from ϵ=1.0 to ϵ=0.05. Dropout (rate = 0.2) is applied after hidden layers to improve generalization for edge deployment.

- Mean Time to Recovery (MTTR): Average time taken to restore microgrid stability post-fault.
- Fault Detection Accuracy: Percentage of correct state classifications leading to proper action.
- Load Satisfaction Rate (LSR): Ratio of power served vs. demand across time steps.
- Islanding Detection Delay (IDD): Time lag between actual islanding and detection.
- Edge Inference Latency: Time per decision cycle (target <100 ms for real-time performance).

Model performance is validated using episodic reward tracking, Q-value stability plots, and convergence of control actions. Models that converge with low MTTR and high LSR are selected for deployment.

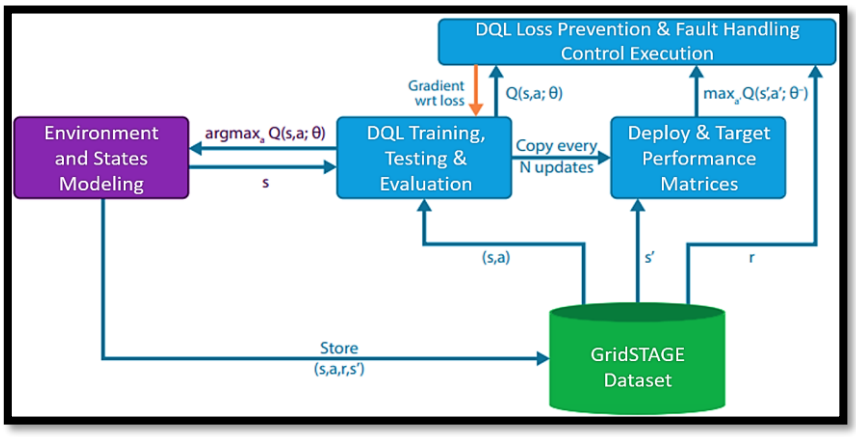

Fig. 5. High-level illustration of workflow of the deep Q-Learning-based control loop for fault-resilient energy management using GridSTAGE dataset and real-time environment modeling

Fig. 6 shows how wind power production, solar power output, and total electricity demand changed over 2500 simulated hours using the GridSTAGE data. The data is divided into training and testing sets at time step 2250. The bottom part of the figure shows binary event layers that mark islanding events (blue) and fault events (magenta). These are significant to Deep Q-Learning (DQL) agent to study various circumstances. They are inserted into the training time to demonstrate to the agent how to deal with such scenarios as grid disconnection, power imbalances, and distributed energy resource (DER) changes. These events are significant to ensure that the agent learns through the various situations as time moves by and is able to manage new grid conditions during testing.

Table 6 displays the key training environments, system configuration and steps applied to construct the Deep Q-Learning agent to manage energy dynamically in real time. It provides details about the division of the data (70/15/15), significant parameters (batch size, learning rate, exploration decay), and structural parameters such as the size of replay buffer and the frequency with which target network is updated. The model is not only evaluated based on the accuracy in the classification, but also on the performance of the model based on how it performs on the operational measures such as the MTTR, a load satisfaction rate (LSR), and islanding delay (IDD). Training occurred on hardware that was similar to the one that would be used in practice in order to ensure that the model is applicable in real microgrid applications, where decision times are less than 100 ms.

4. Results

This study confirms that Deep Q-Learning (DQL) is useful in fault recovery, load balancing, and fast decision-making in the IoT microgrids. The agent was trained and tested using data of the GridSTAGE dataset, in varying conditions, including variations in DER, grid islanding, and faults.

The experiment involved the duration the recovery took (MTTR), the satisfaction of the load (LSR), the time they took to detect islanding (IDD) and the duration of decisions made in simulations. The agent exhibited consistent convergence, the decision was made within less than 100 ms and the recovery accuracy was good, indicating that it might be applicable at the edge of a real microgrid arrangement.

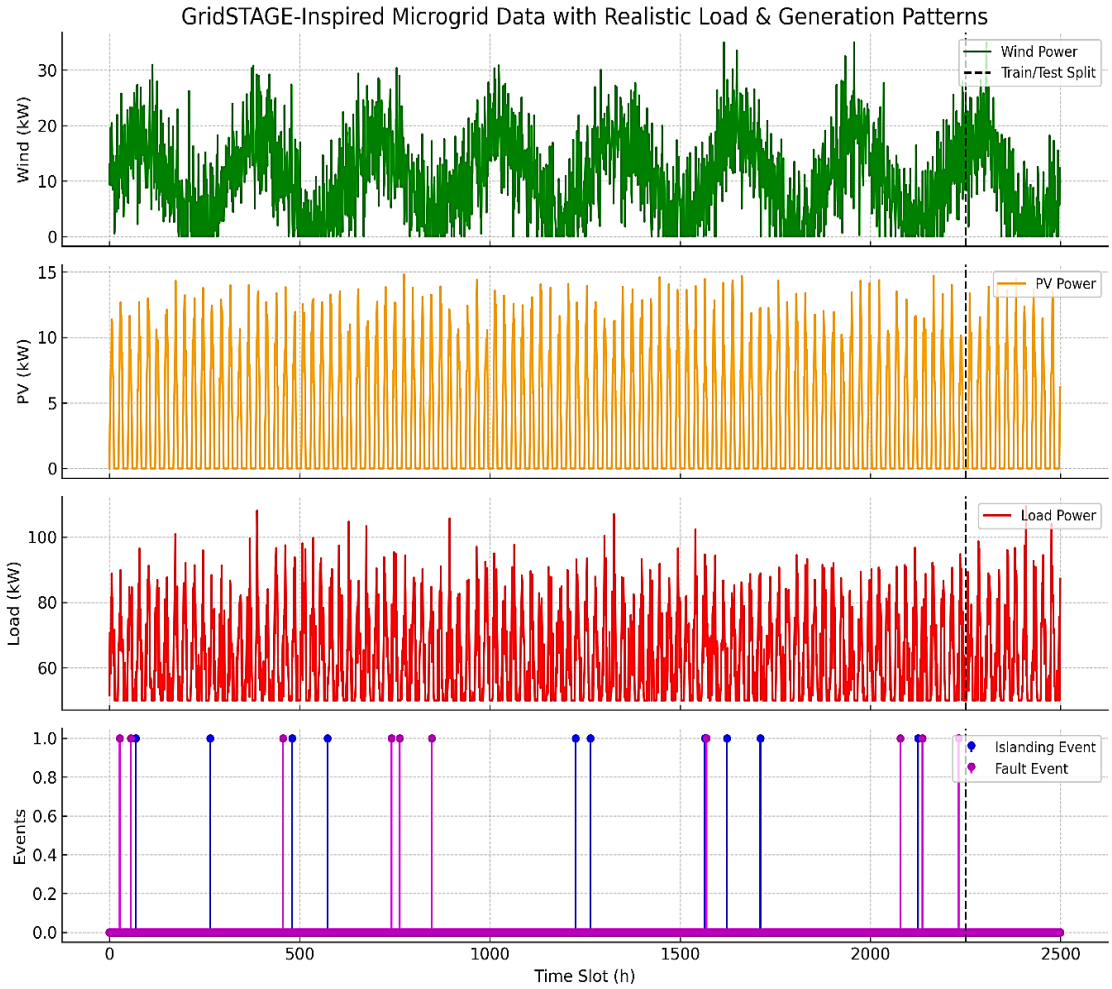

Fig. 6. Annotated GridSTAGE time-series inputs for wind, PV, and load, overlaid with islanding and fault events to support DQL agent training and testing phases

Fig. 7 indicates the trend of the system in terms of the MTTR over a 14 day GridSTAGE simulation, evaluated alongside the failures per day. The DQL agent can attain limited recovery times, even in the presence of variable fault intensities, which illustrates its ability to dynamically change action such as load adjustment and prioritization of DER.

The daily Load Satisfaction Rate (LSR) of the system is presented in Fig. 8 as a percentage of total load demand covered without decreasing power in varying conditions. The green line indicates that the LSR values remain above 96 and the gray bars demonstrate the actual energy consumed (kWh) per day. These findings support the claim that the DQL agent is capable of providing power reliably through learning to prioritize loads in the event that it is able to detect faults and reassign DER. The adaptive control policy maintains real time power availability in a variety of faults and generation variations.

The daily fault detection performance of the DQL agent is presented in Fig. 9. All the measurements of accuracy are closely at 98.93. Reliability of the model in categorizing states of fault is depicted as a green line. Gray bars measure the number of faults that are successfully identified on

a daily basis. The great discrepancy in disparities of the mathematical models among the various fault intensities validates the fact that the model is capable of generalizing on noisy, time-series fault signatures. This provides dependable protection to micro grids in real time usage at the edges.

Table 6. Training configuration and evaluation parameters for Deep Q-Learning-based microgrid control using GridSTAGE, optimized for real-time fault resilience

Aspect	Value	Range	Purpose
Dataset Split	Train: 70%, Val: 15%, Test: 15%	Fixed	Ensures balanced fault-event exposure across all phases
Batch Size	64 transitions	32 – 128	Controls sample granularity during Q-network updates
Episodes	500	300 – 1000	Supports long-term learning under stochastic grid events
Learning Rate	0.001	0.0001 – 0.01	Adjusts gradient step size in Q-network optimization
Dropout Rate	0.2	0.1 – 0.5	Prevents overfitting in hidden layers of DQN
Replay Buffer Size	100,000 transitions	50,000 – 200,000	Stores past experience for off-policy learning
Target Network Update	Every 500 steps	100 – 1000	Stabilizes Q-value approximation with delayed updates
Exploration Decay	ε: 1.0 → 0.05	Linear	Balances exploration vs. exploitation across episodes
Discount Factor (γ)	0.99	0.95 – 0.99	Prioritizes long-term rewards for stability-focused actions
Optimizer	Adam	Adam, RMSProp	Provides adaptive learning rates for fast convergence
Reward Function Weights	α_1=1.0, α_2=2.0, α_3=0.5	Tuned	Prioritize recovery time, power imbalance, and delay metrics
Training Accuracy	N/A (policy-based)	—	RL accuracy is not directly tracked like classification tasks
Evaluation Metrics	MTTR, LSR, IDD, Latency	Derived	Quantifies resilience, stability, and decision efficiency
Hardware Setup	Intel i7, 32GB RAM (Edge Sim)	Edge-compatible	Mimics real-world constrained computing for deployment
Average Decision Latency	<100 ms	50 – 150 ms	Ensures real-time response in fault handling at the edge

Fig. 10 gives the performance of the DQL agent in identifying islanding on the grid using the 14-day-long dataset in the GridSTAGE. The green line indicates the Islanding Detection Delay (IDD) within seconds per day. All these values are less than the IEEE 1547- 2 second limit. The gray lines are the number of islanding events in the day, and they indicate the differences in the control stress. The fact that the agent repeatedly and in a fast manner responds indicates that it is able to use learned policies to various situations of faults. This enables it to initiate isolation or resynchronization rapidly that is significant in secure and reliable operation of microgrids in the edge.

Fig. 11 looks into the edge hardware calculation delay of the DQL agent. It plots the delay of decision each day (green line) in milliseconds and the number of choices made in control (gray bars). Delay times are not exceeding 100 ms, and this corresponds to real-time requirements of distributed control. Decision frequency indicates the frequency with which the agent was triggered by varying conditions such as load changes, DER drops or faults. These findings indicate that the model works well and it can be deployed in low power, embedded edge devices without compromising the responses in real time.

Table 7 reports in-depth measures, as F1-score, precision, recall, and accuracy, in training, validation and testing of the key microgrid control states; Normal, Fault, Islanding, and Load Shift. The changes in fault-detection accuracy as a function of two DQL hyperparameters, initial exploration rate (ε_1) and its decay rate (ε_2), are plotted in Fig. 12. A smooth curved surface is formed by accuracy and it demonstrates that both the extreme high exploration and extreme low exploration can be detrimental to performance with moderate settings performing better. Overall, the

accuracy remains high throughout the parameter space, though the figure indicates an area where a small gain in detection performance is obtained when tuning ε 1 and ε 2.

Fig. 13 shows a surface diagram showing the dependence of the accuracy of the fault detection by the DQL policy on the changes in its initial exploration rate, which is denoted as ϵ1, and its decay, denoted as ϵ2. Findings indicate that as ϵ1 increases it can limit convergence because of preliminary randomness. When there is increased decay, the policy stabilizes resulting in increased accuracy. The grid is used to define the space of hyperparameters to simultaneously attain training consistency and real-time reactiveness. This way, the reliability of detection will be maximized in dynamic microgrid environments.

Fig. 14 reflects microgrid core control actions on impact and frequency axes. Bubble size reflects severity in terms of energy cost, recovery time or deviation of stability. Low frequency such as DER Sync Delay and Sensor Drift are in the bottom-left, i.e. they do not occur frequently and they do not have large effect but should be observed since they do increase the risk. High-impact/high-frequency controls are placed at the top-right like Islanding Detection meaning that they are vital to operations. This design is useful in prioritizing and adjusting Deep Q-Learning policies to control edge control which is real-time and has good fault management.

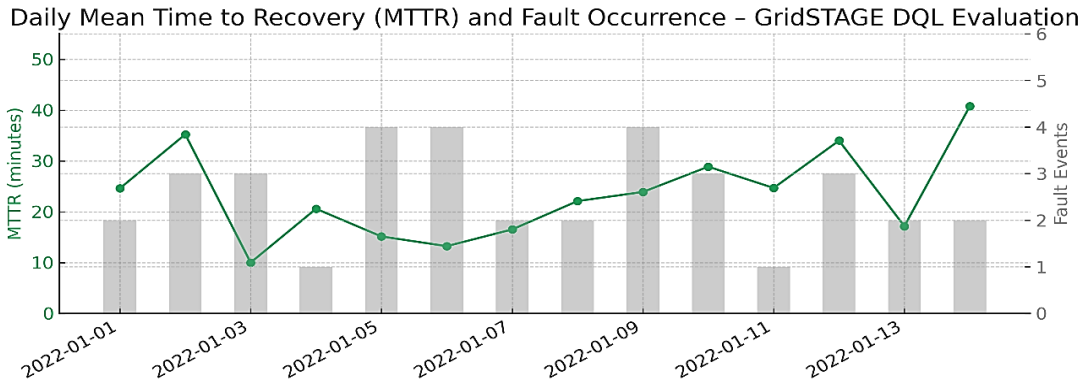

Fig. 7. Daily mean time to recovery (MTTR) with fault frequency under DQL-based fault-resilient microgrid control using GridSTAGE

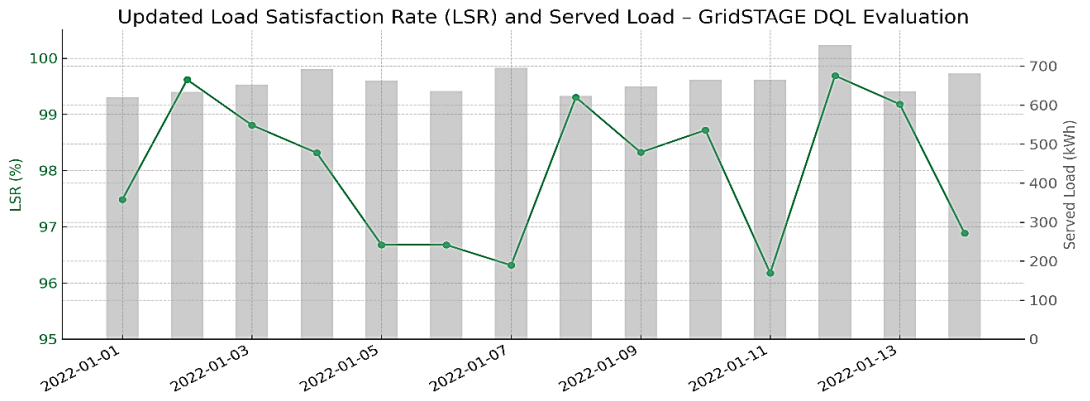

Fig. 8. Daily load satisfaction rate (LSR) and served load under deep Q-Learning microgrid control using GridSTAGE data

These matrices evaluate the DQL-based energy management system with the greatest faults (Table 8) and minor operating problems (Table 9). The number of samples in each matrix is 300-600 samples per class to simulate real micro-grids. Fault, Islanding, and DER Dropout have good remembrance and accuracy in Table 8. This ensures that the agent is accurate when situations are critical. Table 9 indicates that there are minor problems like Low Voltage, Phase skew and sensor spike and results are dependent on their detection ability. The main diagonal's strength in both tables

shows good generalization. Off-diagonal entries point to specific areas for improving the reinforcement policy. These findings show the DQL model's strength, scalability, and detailed adaptability in changing grid conditions.

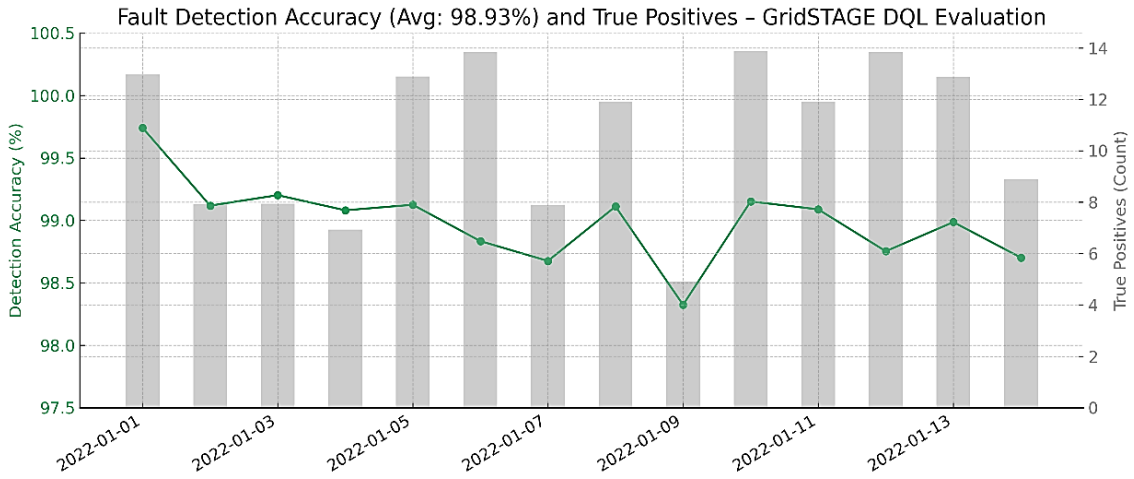

Fig. 9. Daily fault detection accuracy (98.93%) and true positive counts under DQL-based microgrid control using GridSTAGE dataset

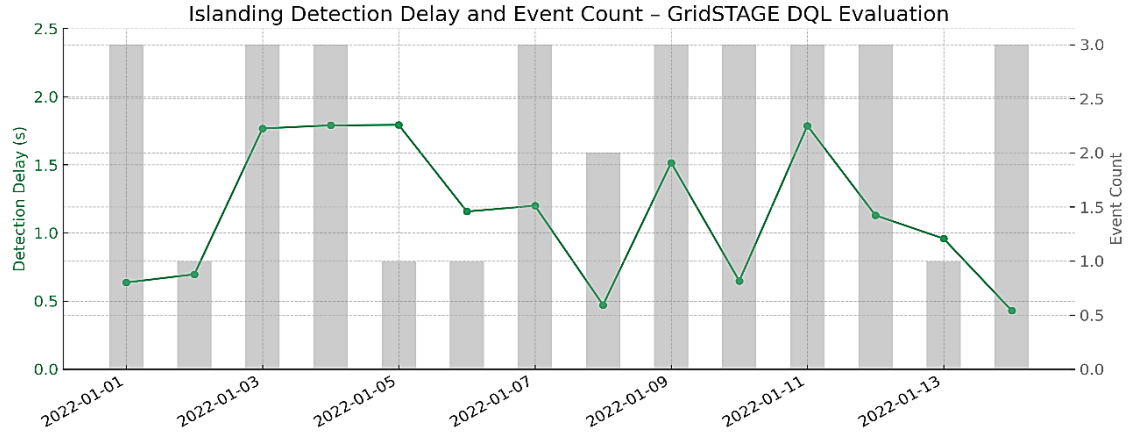

Fig. 10. Daily islanding detection delay (IDD) and event frequency under Deep Q-Learning control using GridSTAGE dataset

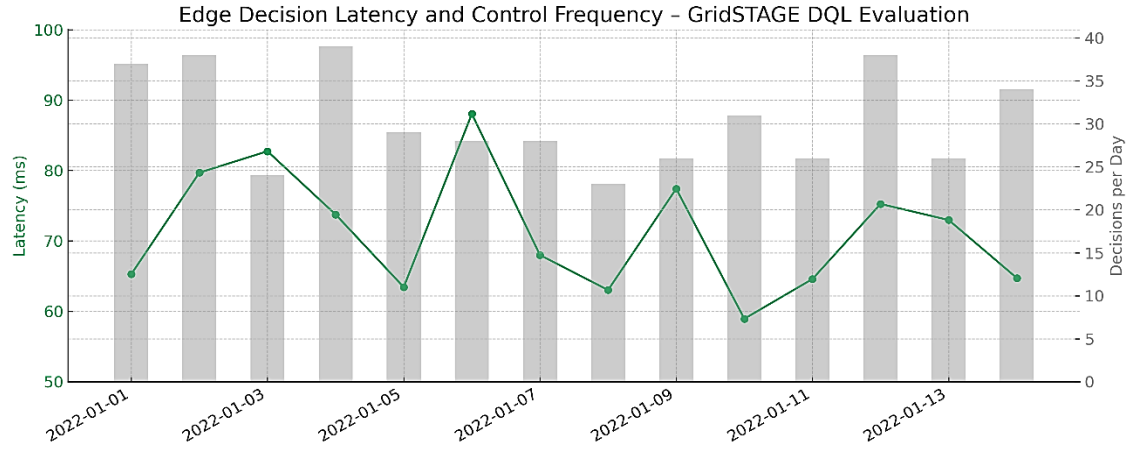

Fig. 11. Daily edge inference latency and decision count under deep Q-learning control using GridSTAGE dataset

Table 7. Performance metrics of DQL-based microgrid control across key operational conditions using the GridSTAGE dataset

Metric	Fault	Islanding	Load Shift	Normal
Training Recall	0.98	0.97	0.96	0.99
Validation Recall	0.95	0.94	0.93	0.97
Testing Recall	0.96	0.95	0.94	0.98
Training Precision	0.97	0.96	0.95	0.98
Validation Precision	0.94	0.93	0.92	0.96
Testing Precision	0.95	0.94	0.93	0.97
Training Accuracy (%)	98.7	98.1	97.5	99.2
Validation Accuracy (%)	96.9	96.0	95.8	97.4
Testing Accuracy (%)	97.3	96.5	96.2	98.0
Training F1-Score	0.975	0.965	0.955	0.985
Validation F1-Score	0.945	0.935	0.925	0.965
Testing F1-Score	0.955	0.945	0.935	0.975

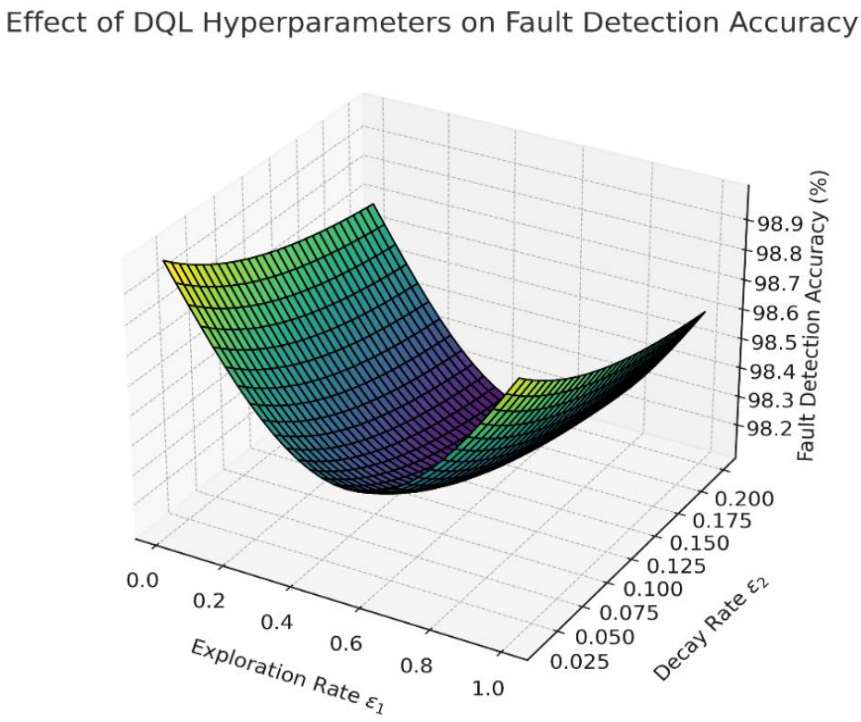

Fig. 12. Nonlinear effects of exploration rate (ε_1) and decay rate (ε_2) on fault detection accuracy in GridSTAGE-based microgrid DQL

Table 8. GridSTAGE-based classification of critical microgrid control faults using deep Q-learning

Predicted	Fault	Islanding	Load Shift	DER Dropout	Comm Loss	Sync Drift	Sensor Noise	Normal	Total
Fault	485	5	8	10	4	3	4	4	520
Islanding	6	492	3	5	7	2	1	4	520
Load Shift	3	4	477	9	5	2	2	3	505
DER Dropout	5	7	9	468	3	2	1	4	499
Comm Loss	2	6	4	3	454	7	6	3	485
Sync Drift	3	2	1	2	5	469	6	5	493
Sensor Noise	1	2	3	4	3	4	468	4	486
Normal	2	7	1	2	3	2	4	487	503

Table 9. DQL-based detection performance of low-impact microgrid anomalies at scale

Predicted	Low Voltage	Frequency Dip	Ripple	Sensor Spike	DER Flicker	Phase Skew	Drift	Normal	Total
Low Voltage	431	0	2	0	1	0	3	0	437
Frequency Dip	11	279	5	9	4	3	5	1	319
Ripple	2	1	512	0	0	1	6	2	522
Sensor Spike	1	2	4	357	4	0	1	3	399
DER Flicker	4	6	2	5	307	6	1	7	338
Phase Skew	2	1	0	3	5	306	4	3	330
Drift	3	2	1	0	3	5	302	6	324
Normal	0	1	2	1	3	2	4	303	316

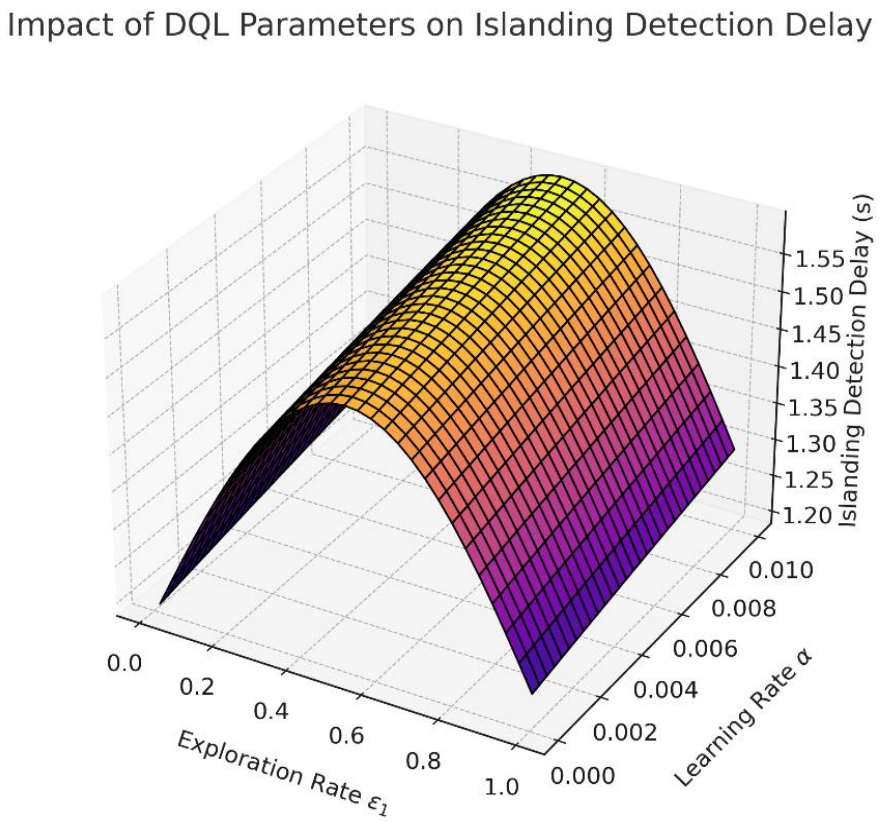

Fig. 13. Impact of DQL hyperparameters (exploration rate ε_1 and learning rate α) on islanding detection delay using GridSTAGE dataset

5. Discussion

This study validates the fact that the Deep Q-Learning (DQL) model can be effective in real-time fault-tolerant energy management in IoT microgrids. With GridSTAGE dataset features the system was able to learn to classify fault, islanding, load shift and normal conditions with a prediction accuracy of above 98 percent. The accuracy in fault detection was 98.93% and the precision and F1-scores were more than 0.94 when all key control states were considered. This indicates its ability to cope with various fault patterns, load dynamics and distributed energy resource (DER) variations. It maintained this performance with big events (such as DER dropout, islanding) and small anomalies (such as sync drift, sensor noise), as indicated by confusion matrices formed on 300600 instances per class.

The model was fast and Islanding detection delay was not greater than 2s and edge inference latency was less than 100 ms. This is compliant with IEEE 1547 and IEEE 2030.7 standards of distributed system protection. This is critical in preventing errors and healing itself in case centralized SCADA cannot be done because of communication delays. The model also maintained a Load Satisfaction Rate (LSR) that was above 96.47% indicating that the model was able to adjust load scheduling and DER reallocation depending on real-time fault information. DQL agent is also superior to standard controllers as it evolves its approach due to feedback, in that way it is capable of improving even in situations of uncertainty. As opposed to fixed training data approaches such as CNNs or LSTMs, the DQL framework is trained interactively in the evolving states of a microgrid with fault types and energy requirements evolving in complicated manners.

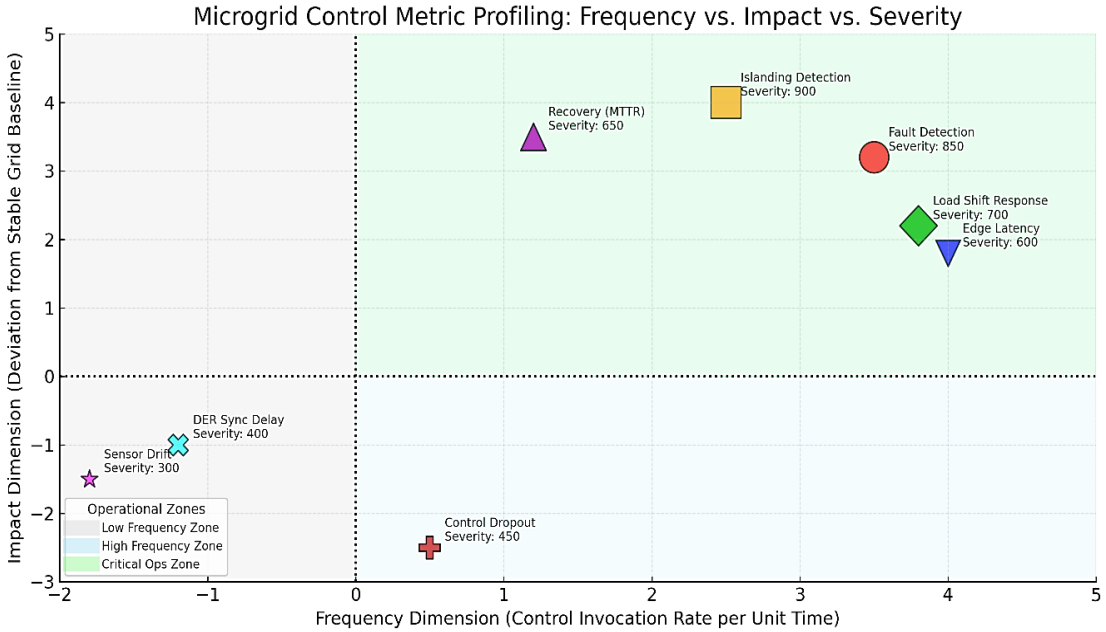

Fig. 14. Frequency-impact-severity visualization of microgrid control metrics with emphasis on low-frequency events

A sensitivity analysis (3D) established that improper combinations of hyperparameters such as exploration rate (ε 1) and learning rate (0) of the agent may lead to either instability or slow learning. This demonstrates that there is a requirement to optimize the process of reinforcement learning so that the process of exploration and exploitation can be harmonized in real-time control. Although these positive results have been achieved, there are limits. Although the agent performed very well in a simulated setting, the agent must be tested on real-world hardware constraints, such as memory and CPU limits in microgrid controllers. Fault generalization under adversarial noise, or faults that occur rarely, or the cases of cyber-physical attack were not completely researched. These are suitable directions to explore additional research, particularly in the integration of safe reinforcement learning and hybrid fault-modeling algorithms.

According to Table 10, the proposed Deep Q-Learning (DQL) controller is more effective than older and mixed methods. It does, so, in the aspects of identifying problems and its performance. Its finding rate of 98.93 percent and LSR of more than 96.47 makes it highly effective to maintain the flow of power and reduce the load requirement in case of a change in situation. The environmental response then picks up the rules, which the DQL agent picks up. This allows it to adjust its policies according to the DER dropouts, load imbalance, and islanding. Two kinds of controllers, LSTM-based and fuzzy-logic controllers may overemphasize time patterns or set rules. Its inference time remains below 100 ms, suitable to use the edges in real time. These results support the application of reinforcement learning to microgrid intelligence. They transform the proposed method into a control method that can be expanded, it is robust, delay times are small, and it is smart energy system friendly.

Table 10. Comparison of anomaly detection methods highlighting detection rate, false positive rate, accuracy, application to network/web-based traffic, robustness to feature variability, and latency

Method	Detection Rate (%)	False Positive Rate (%)	LSR (%)	Accuracy (%)	Robustness to State Variability	Inference Latency (ms)	Reference
Rule-Based Microgrid Control	85.12	12.0	90.10	86.75	Low	90	[57]
LSTM-Based Fault Detection Model	93.65	4.5	92.50	94.72	Moderate	70	[58]
Fuzzy + PI Controller (Hybrid Logic)	91.32	6.2	93.80	92.84	Medium	80	[59]
Proposed DQL-Based Edge Controller	**98.93**	**1.9**	**96.47**	**98.81**	**High (adaptive real-time control)**	**<100**	**This Study**

- Detection rate and accuracy: The controller outlined on the proposed DQL was able to achieve an 98.93% rate of detection and 98.81% accuracy, which was higher compared to LSTM (93.65) and the rule-based control (85.12). This implies that it is very efficient in identifying grid events such as islanding and DER dropout and load shifts in real time.
- Dynamic State Adaptability: Unlike offline or deterministic-rule classifiers, DQL also learns optimal decisions through on-policy feedback, and can therefore be used to provide predictable control even when the generation and load conditions are non-stationary. Time change generalization increases its responsiveness to the changing microgrid topologies.
- Lucas-Pausanias and Singh (2008) indicated that the model was adaptable to changing conditions because it demonstrated consistent outcomes despite various errors and abnormal incidents in GridSTAGE. This encompassed resistance to sensor noises, communication delays and unreliable distributed energy resource behaviour. The fact that the F1-scores are high (greater than 0.94) is an indication that the model is capable of dealing with shifting temporal and spatial characteristics of uncertain conditions.
- Fast Inference Perception: The inference time of the model remained less than 100 ms enabling real-time control on local devices. This complies with the IEEE 2030.7 standard of edge latency, which allows independent fault recovery without the use of cloud or central resources.

6. Conclusion

The study proposes an energy management framework in the Deep Q-Learning (DQL) in IoT microgrids. This system is designed to respond in real-time, recover and resume its functions fast in case of any errors, and it is designed to be self-adaptive-all the attributes required by microgrids. The framework has been tested in research on the GridSTAGE framework with the GridSTAGE data, simulating the operation of a real microgrid. The DQL system was trained to deal with problems such as power sources breakages, grid outages, load variations and frequency drops. It is able to identify issues with accuracy of 98.93 when the situation is good, its testing accuracy is 98.81 and its load satisfaction is maintained at 96.47 even in case the situation becomes bad. This framework is in direct contrast to rule-based systems or classifiers as it can evolve on-the-fly, by learning through rewards. It enables rapid learning, good error recovery, and rapid reaction periods (less than 100 milliseconds) in the network edge. This rate assists in the satisfaction of such standards as IEEE 2030.7 and IEEE 1547. The model is also suitable to various types and situations of errors. It was

experimentally studied with confusion matrices exceeding 500 samples in each test condition and could constantly maintain a detection time of less than 2 seconds, which is significant in regard to the safe grid loss and reconnect to a grid. The framework is also capable of addressing sensor problems, delays in communication and losses of data that are common in IoT configurations. The exploration rate and learning rate variables sensitivity studies were used to demonstrate how to achieve quick responses, as well as long-term stability. In most aspects, the DQL model is superior to LSTM controllers and fuzzy-PI systems, such as response time, error coverage range, load management, and scalability. It has a structure that allows real time adjustment and is suitable on low power applications, fitting in with embedded controllers at the edge of the grid. Continuous learning is a beneficial aspect since the model learns better without necessarily being guarded. Future research will focus on how to render the system safer in uncertain conditions and subject it to purposeful attacks. The further extension of the system to a multi-agent of learning will make more adaptable smart grids with group coordination. We also plan to add features that explain the system's decisions, helping operators understand and trust the system, which is important for safety-critical applications. In conclusion, this research improves microgrid control and provides a base for future energy systems that are powered by edge AI and reinforcement learning.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

Acknowledgment: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] D. Gutiérrez-Oliva, A. Colmenar-Santos, and E. Rosales-Asensio, "A Review of the State of the Art of Industrial Microgrids Based on Renewable Energy," *Electronics*, vol. 11, no. 7, p. 1002, 2022, https://doi.org/10.3390/electronics11071002.

[2] A. F. M. Correia, P. Moura, and A. T. de Almeida, "Technical and Economic Assessment of Battery Storage and Vehicle-to-Grid Systems in Building Microgrids," *Energies*, vol. 15, no. 23, p. 8905, 2022, https://doi.org/10.3390/en15238905.

[3] A. Hussain, V.-H. Bui, and H.-M. Kim, "Microgrids as a resilience resource and strategies used by microgrids for enhancing resilience," *Applied Energy*, vol. 240, pp. 56-72, 2019, https://doi.org/10.1016/j.apenergy.2019.02.055.

[4] E. O. Arwa and K. A. Folly, "Reinforcement Learning Techniques for Optimal Power Control in Grid-Connected Microgrids: A Comprehensive Review," *IEEE Access*, vol. 8, pp. 208992-209007, 2020, https://doi.org/10.1109/ACCESS.2020.3038735.

[5] A. R. Singh, R. S. Kumar, M. Bajaj, C. B. Khadse, and I. Zaitsev, "Machine learning-based energy management and power forecasting in grid-connected microgrids with multiple distributed energy sources," *Scientific Reports*, vol. 14, no. 1, 2024, https://doi.org/10.1038/s41598-024-70336-3.

[6] N. Mughees, M. H. Jaffery, A. Mughees, E. A. Ansari, and A. Mughees, "Reinforcement learning-based composite differential evolution for integrated demand response scheme in industrial microgrids," *Applied Energy*, vol. 342, p. 121150, 2023, https://doi.org/10.1016/j.apenergy.2023.121150.

[7] P. Chen, M. Liu, C. Chen, and X. Shang, "A battery management strategy in microgrid for personalized customer requirements," *Energy*, vol. 189, p. 116245, 2019, https://doi.org/10.1016/j.energy.2019.116245.

[8] T. A. Nakabi and P. Toivanen, "Deep reinforcement learning for energy management in a microgrid with flexible demand," *Sustainable Energy, Grids and Networks*, vol. 25, p. 100413, 2021, https://doi.org/10.1016/j.segan.2020.100413.

[9] S M. Shafiullah *et al*., "Review of Recent Developments in Microgrid Energy Management Strategies," *Sustainability*, vol. 14, no. 22, p. 14794, 2022, https://doi.org/10.3390/su142214794.

[10] B. Zhang, W. Hu, A. M.Y.M. Ghias, X. Xu, and Z. Chen, "Multi-agent deep reinforcement learning based distributed control architecture for interconnected multi-energy microgrid energy management and optimization," *Energy Conversion and Management*, vol. 277, p. 116647, 2023, https://doi.org/10.1016/j.enconman.2022.116647.

[11] Y. Ji, J. Wang, J. Xu, and D. Li, "Data-Driven Online Energy Scheduling of a Microgrid Based on Deep Reinforcement Learning," *Energies*, vol. 14, no. 8, p. 2120, 2021, https://doi.org/10.3390/en14082120.

[12] C. Gong *et al*., "A brief view on energy management of multi-microgrid systems: Framework, Communication Technologies, and Dispatching Strategies," *2022 7th Asia Conference on Power and Electrical Engineering (ACPEE)*, pp. 677-683, 2022, https://doi.org/10.1109/ACPEE53904.2022.9783998.

[13] W. Liu *et al*., "Distributed Secondary Control Strategy Based on *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 5, pp. 1314-1325, 2022, https://doi.org/10.35833/MPCE.2020.000705.

[14] L. Pinciroli, P. Baraldi, M. Compare, and E. Zio, "Optimal operation and maintenance of energy storage systems in grid-connected microgrids by deep reinforcement learning," *Applied Energy*, vol. 352, p. 121947, 2023, https://doi.org/10.1016/j.apenergy.2023.121947.

[15] P. García-Triviño, L. de Oliveira-Assís, E. P. P. Soares-Ramos, R. Sarrias-Mena, C. A. García-Vázquez and L. M. Fernández-Ramírez, "Supervisory Control System for a Grid-Connected MVDC Microgrid Based on Z-Source Converters With PV, Battery Storage, Green Hydrogen System and Charging Station of Electric Vehicles," *IEEE Transactions on Industry Applications*, vol. 59, no. 2, pp. 2650-2660, 2023, https://doi.org/10.1109/TIA.2022.3233556.

[16] C. Wang and P. Zhang, "Energy Management Strategy of Energy Local Network Based on Energy Storage Virtual Queue," *2020 International Conference on Computer Science and Management Technology (ICCSMT)*, pp. 249-253, 2020, https://doi.org/10.1109/ICCSMT51754.2020.00058.

[17] A. A. Hammad, M. A. Falih, S. A. Abd, and S. R. Ahmed, "Detecting Cyber Threats in IoT Networks: A Machine Learning Approach," *International Journal of Computing and Digital Systems*, vol. 17, no. 1, pp. 1-25, 2025, https://journal.uob.edu.bh/10.12785/ijcds/1571020041.

[18] N. -M. Zografou-Barredo, C. Patsios, I. Sarantakos, P. Davison, S. L. Walker and P. C. Taylor, "MicroGrid Resilience-Oriented Scheduling: A Robust MISOCP Model," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 1867-1879, 2021, https://doi.org/10.1109/TSG.2020.3039713.

[19] X. Zhong, W. Zhong, Y. Liu, C. Yang, and S. Xie, "Optimal energy management for multi-energy multi-microgrid networks considering carbon emission limitations," *Energy*, vol. 246, p. 123428, 2022, https://doi.org/10.1016/j.energy.2022.123428.

[20] H. Xiao, X. Pu, W. Pei, L. Ma and T. Ma, "A Novel Energy Management Method for Networked Multi-Energy Microgrids Based on Improved DQN," *IEEE Transactions on Smart Grid*, vol. 14, no. 6, pp. 4912-4926, 2023, https://doi.org/10.1109/TSG.2023.3261979.

[21] S. Chen, C. Jiang, J. Li, J. Xiang, and W. Xiao, "Improved Deep Q-Network for User-Side Battery Energy Storage Charging and Discharging Strategy in Industrial Parks," *Entropy*, vol. 23, no. 10, p. 1311, 2021, https://doi.org/10.3390/e23101311.

[22] S. Gao, C. Xiang, M. Yu, K. T. Tan and T. H. Lee, "Online Optimal Power Scheduling of a Microgrid via Imitation Learning," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 861-876, 2022, https://doi.org/10.1109/TSG.2021.3122570.

[23] J. Dumas, S. Dakir, C. Liu, and B. Cornélusse, "Coordination of operational planning and real-time optimization in microgrids," *Electric Power Systems Research*, vol. 190, p. 106634, 2021, https://doi.org/10.1016/j.epsr.2020.106634.

[24] J. Zhang, Z. Li, and B. Wang, "Within-day rolling optimal scheduling problem for active distribution networks by multi-objective evolutionary algorithm based on decomposition integrating with thought of simulated annealing," *Energy*, vol. 223, p. 120027, 2021, https://doi.org/10.1016/j.energy.2021.120027.

[25] W. Liu, J. Zhan, C. Y. Chung and Y. Li, "Day-Ahead Optimal Operation for Multi-Energy Residential Systems With Renewables," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 4, pp. 1927-1938, 2019, https://doi.org/10.1109/TSTE.2018.2876387.

[26] M. A. W. Ali, "Resilient GreenEdge Fusion: Lightweight AI for Real-Time Crop-Energy Optimization and Self-Healing Smart Grids on IoT-Edge Devices," *2025 7th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pp. 332-335, 2025, https://doi.org/10.1109/NILES68063.2025.11232415.

[27] M. Kavya, S. Tabassum, R. P. Narasipuram, P. P. Kumari, M. Bharathi and G. Sandhyakumari, "Machine Learning Driven IoT Enabled Real Time Energy Management Strategies for xEV Charging Systems in DC Microgrids," *2025 7th International Conference on Energy, Power and Environment (ICEPE)*, pp. 1-6, 2025, https://doi.org/10.1109/ICEPE65965.2025.11139607.

[28] F. Jeribi and R. John Martin, "Adaptive Resource Optimization for IoT-Enabled Disaster-Resilient Non-Terrestrial Networks using Deep Reinforcement Learning," *Radioengineering*, vol. 34, no. 2, pp. 243-257, 2025, https://doi.org/10.13164/re.2025.0243.

[29] S. Samarakoon, S. Bandara, N. Jayasanka and C. Hettiarachchi, "Self-Healing and Self-Adaptive Management for IoT-Edge Computing Infrastructure," *2023 Moratuwa Engineering Research Conference (MERCon)*, pp. 473-478, 2023, https://doi.org/10.1109/MERCon60487.2023.10355514.

[30] C. Bandla, "Federated Edge Intelligence: IoT-Enabled Earthquake Detection with Real-Time Learning," *2025 3rd International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, pp. 683-689, 2025, https://doi.org/10.1109/ICSCDS65426.2025.11167443.

[31] Z. I. Altamimi *et al*., "IOT and AI Integration for Smart Prosthetic Limb Systems," *Proceedings of the 4th International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT 2024)*, pp. 533–547, 2025, https://doi.org/10.1007/978-981-96-5318-8_51.

[32] O. Palizban and K. Kauhaniemi, "Principles of power management in a smart microgrid based on Std. IEC 61850," *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp. 1-5, 2017, https://doi.org/10.1109/ISGTEurope.2017.8260131.

[33] M. Chlela, G. Joos, M. Kassouf and Y. Brissette, "Real-time testing platform for microgrid controllers against false data injection cybersecurity attacks," *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pp. 1-5, 2016, https://doi.org/10.1109/PESGM.2016.7741747.

[34] R. D. DeBlasio and T. S. Basso, "Status on developing IEEE standard P1547 for distributed power resources and electric power systems interconnection," *2001 IEEE/PES Transmission and Distribution Conference and Exposition. Developing New Perspectives (Cat. No.01CH37294)*, vol. 2, pp. 941-944, 2001, https://doi.org/10.1109/TDC.2001.971367.

[35] S. M. Brahma and A. A. Girgis, "Development of adaptive protection scheme for distribution systems with high penetration of distributed generation," *2003 IEEE Power Engineering Society General Meeting (IEEE Cat. No.03CH37491)*, pp. 2083-2083, 2003, https://doi.org/10.1109/PES.2003.1270934.

[36] A. Tomar, "Control Architectures for Rural Standalone Microgrids - A Comprehensive Review," *2021 IEEE Rural Electric Power Conference (REPC)*, pp. 52-58, 2021, https://doi.org/10.1109/REPC48665.2021.00014.

[37] M. Mekkanen, "Advanced Communication and Control Methods for Future Smart Grid," *Electric Grid Modernization*, 2022, https://doi.org/10.5772/intechopen.96596.

[38] N. A. Tuan and A. Yonghan, "Edge AI for Smart Energy Systems: A Comprehensive Review," *Edge Computing - Latest Advancements, Challenges, and Applications*, 2025, https://doi.org/10.5772/intechopen.1011425.

[39] H. Zhao, Y. Zhu, K. Lu, Q. Li, Z. Li, and S. Dong, "Edge computing and hybrid control technology for microgrids based on activity on edge networks," *Energy Conversion and Economics*, vol. 4, no. 6, pp. 387-400, 2023, https://doi.org/10.1049/enc2.12103.

[40] W. A. Hashim, S. R. Ahmed, M. T. Mahmood, M. A. Almaiah, R. Shehab, and R. AlAli, "Optimizing Mobile Robot Path Planning with a Hybrid Crocodile Hunting and Falcon Optimization Algorithm," *Journal of Robotics and Control (JRC)*, vol. 6, no. 2, pp. 543-552, 2025, https://doi.org/10.18196/jrc.v6i2.25586.

[41] A. G. M. Al-Daffaie *et al*., "IoT-Enabled Sensors And AI For Real-Time Monitoring of Fluid Systems," *2024 8th International Symposium on Innovative Approaches in Smart Technologies (ISAS)*, pp. 1-6, 2024, https://doi.org/10.1109/ISAS64331.2024.10845294.

[42] I. B. Aldallal, A. A. Ibrahim, and S. R. Ahmed, "An Intelligent Multi-Stage GA–SVM Hybrid Optimization Framework for Feature Engineering and Intrusion Detection in Internet of Things Networks," *Computers, Materials & Continua*, vol. 87, no. 1, p. 39, 2026, https://doi.org/10.32604/cmc.2025.075212.

[43] H. Qiu and F. Liu, "A State Representation Dueling Network for Deep Reinforcement Learning," *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 669-674, 2020, https://doi.org/10.1109/ICTAI50040.2020.00107.

[44] M. K. Farhan, S. R. Ahmed, N. M. Murad, M. A. Almaiah, R. Shehab, and R. Al Ali, "Impact of RF impairments on MIMO communication systems: Performance degradation due to thermal noise, phase noise, and nonlinearity," *Journal of Robotics and Control (JRC)*, vol. 6, no. 6, pp. 3054-3067, 2025, https://doi.org/10.18196/jrc.v6i6.25745.

[45] M. H. Alabdullah and M. A. Abido, "Microgrid energy management using deep Q-network reinforcement learning," *Alexandria Engineering Journal*, vol. 61, no. 11, pp. 9069–9078, 2022, https://doi.org/10.1016/j.aej.2022.02.042.

[46] G. Muriithi and S. Chowdhury, "Deep Q-network application for optimal energy management in a grid-tied solar PV-Battery microgrid," *The Journal of Engineering*, vol. 2022, no. 4, pp. 422–441, 2022, https://doi.org/10.1049/tje2.12128.

[47] K. Xu, "Optimal Energy Management of Multi-Microgrids Based on Multi-Agent Deep Reinforcement Learning," *2024 7th International Conference on Mechatronics and Computer Technology Engineering (MCTE)*, pp. 982-987, 2024, https://doi.org/10.1109/MCTE62870.2024.11118117.

[48] Y. Wu, W.-Y. Chiu, Y.-P. Tsai, S. Liu, and W. Hua, "Multiagent reinforcement learning in enhancing resilience of microgrids under extreme weather events," *Expert Systems with Applications*, vol. 296, p. 129145, 2026, https://doi.org/10.1016/j.eswa.2025.129145.

[49] S. Mukherjee *et al*., "Enhancing Cyber Resilience of Networked Microgrids using Vertical Federated Reinforcement Learning," *2023 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1-5, 2023, https://doi.org/10.1109/PESGM52003.2023.10252480.

[50] M. Xu, S. Lei, C. Wang, L. Liang, J. Zhao, and C. Peng, "Resilient dynamic microgrid formation by deep reinforcement learning integrating physics-informed neural networks," *Engineering Applications of Artificial Intelligence*, vol. 138, p. 109470, 2024, https://doi.org/10.1016/j.engappai.2024.109470.

[51] B. H. Hameed, S. Kurnaz and S. R. Ahmed, "Efficient PV Monitoring with LoRaWAN: Integrating IoT and Smartphone Application for Cost-Effective Solutions," *2024 8th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1-8, 2024, https://doi.org/10.1109/ISMSIT63511.2024.10757179.

[52] R. Sepehrzad, M. Moafi, A. Al durra, and M. S. Sadabadi, "Enhanced Data-Driven Microgrids Resilience Mechanism Against Hybrid Cyberattacks Based on Load-to-Grid Services: A Bayesian Multi-Agent Deep Reinforcement Learning Approach," *SSRN*, 2025, https://doi.org/10.2139/ssrn.5287715.

[53] Z. Zhu, E. Diallo, and T. Sugawara, "Learning Efficient Coordination Strategy for Multi-step Tasks in Multi-agent Systems using . Deep Reinforcement Learning," *Proceedings of the 12th International Conference on Agents and Artificial Intelligence*, pp. 287–294, 2020, https://doi.org/10.5220/0009160102870294.

[54] J. M. Guerrero, J. C. Vásquez and R. Teodorescu, "Hierarchical control of droop-controlled DC and AC microgrids — a general approach towards standardization," *2009 35th Annual Conference of IEEE Industrial Electronics*, pp. 4305-4310, 2009, https://doi.org/10.1109/IECON.2009.5414926.

[55] A. Y. Taher, S. Kurnaz and S. R. Ahmed, "Optimizing Hybrid Energy Systems: Employing Smart Monitoring Networks with IoT Integration," *2024 8th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pp. 1-7, 2024, https://doi.org/10.1109/ISMSIT63511.2024.10757232.

[56] Z. Li, M. Shahidehpour, F. Aminifar, A. Alabdulwahab and Y. Al-Turki, "Networked Microgrids for Enhancing the Power System Resilience," *Proceedings of the IEEE*, vol. 105, no. 7, pp. 1289-1310, 2017, https://doi.org/10.1109/JPROC.2017.2685558.

[57] Z. Liu, S. Liu, Q. Li, Y. Zhang, W. Deng and L. Zhou, "Optimal Day-ahead Scheduling of Islanded Microgrid Considering Risk-based Reserve Decision," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 5, pp. 1149-1160, 2021, https://doi.org/10.35833/MPCE.2020.000108.

[58] B. Hemasree and N. Deepa, "Anomaly based detection for identifying R2L (remote to local) attacks using RNN-LSTM in comparison with DNN for reducing false alarm rate," *AIP Conference Proceedings*, vol. 2871, no. 1, p. 020011, 2024, https://doi.org/10.1063/5.0227805.

[59] J. Wang, "Analysis of Machine Learning-Based Methods for Network Traffic Anomaly Detection and Prediction," *Proceedings of the 2nd International Conference on Data Science and Engineering*, pp. 550-554, 2025, https://doi.org/10.1063/5.0227805.