International Journal of Mechatronics and Applied Mechanics
2025, Issue 20, Vol. 1

INTELLIGENT SYSTEM OF ELECTRICAL NETWORKS OPERATIONAL MANAGEMENT IN EMERGENCY SITUATIONS

Vladyslav Pliuhin[1], Ihor Biletskyi[1], Mykola Shpika[1], Alina Trotsai[1], Oleg Synelnykov[1]
[1]O.M. Beketov National University of Urban Economy in Kharkiv, Kharkiv, Ukraine
Email: vladyslav.pliuhin@kname.edu.ua

Abstract Abstract – Modern urban electrical grids must maintain high reliability, ensure prompt fault response, and minimize power restoration times for consumers. This makes the development of intelligent systems for automatic fault detection, classification, localization, and network reconfiguration highly relevant. The aim of this study is to develop an automatic switching algorithm for urban power networks using artificial intelligence methods, particularly a multilayer perceptron (MLP) neural network, combined with graph-based approaches for modeling the grid structure and identifying alternative energy supply routes.
The research was carried out using MATLAB, Simulink, Power Systems Toolbox, and Deep Learning Toolbox. The simulation model represents a section of an urban grid consisting of 3 substations, 8 switches, 20 consumers, and a corresponding weighted graph. Fault diagnosis is per-formed using an MLP neural network with 64 input features, three hidden layers [128, 64, 32], and a softmax output layer for classifying different types of faults. The network reconfiguration algorithm considers power flow direction, line capacity, load constraints, and seeks to minimize the number of switching operations. Alternative supply paths are generated via heuristic graph search while accounting for load priority.
Simulation results demonstrate the high efficiency of the proposed approach. On average, dam-aged areas can be isolated with only 2–3 switching operations, restoring power to 90–92% of consumers without causing overloads in other parts of the network. The neural network achieved a classification accuracy of 96.3%, enabling real-time supervisory control and data acquisition system command generation. The practical value of the solution enabling seamless integration into existing Smart Grid systems and lies in its applicability for integration into existing Smart Grid infrastructures without requiring major system modifications. Prospects for further research include extending the model to consider dynamic load profiles, asymmetric operating conditions, limitations from renewable energy sources, and the introduction of economic criteria for switching optimization and decision-making under uncertainty.

Keywords: Electrical Networks Reconfiguration, Neural Network Fault Diagnosis, Multilayer Perceptron, Automatic Switching Algorithm, Heuristic Graph Search, Power Supply Restoration, Priority-Based Power Delivery.

1. Introduction

Modern power systems are undergoing significant transformations driven by the increasing integration of renewable energy sources, decentralization of generation, and increased demands for network reliability and flexibility [16]. These changes complicate operational management, especially in emergency situations where it is necessary to make quick decisions on switching to ensure uninterrupted power supply [7].

Traditional dispatching methods based on predefined scenarios and rules often prove to be insufficiently effective in the conditions of dynamic changes and high complexity of modern networks [14]. In response to these challenges, artificial intelligence (AI) offers new approaches to supporting decision-making in power systems [2, 15]. In particular, the application of machine learning methods, hybrid algorithms, and intelligent systems allows for increased forecasting accuracy, adaptability, and speed of response to emergency situations [6, 30].

https://doi.org/10.17683/ijomam/issue20.39

Recent studies demonstrate the effectiveness of using hybrid AI models for forecasting and managing emergencies in power grids [9, 27]. For example, combining neural networks with gradient boosting methods allows to improve the accuracy of outage forecasting and optimize response strategies [17, 31]. In addition, the use of fuzzy decision support systems contributes to the effective management of uncertainty associated with load fluctuations and weather conditions [26].

The integration of AI into power systems also allows to implement autonomous voltage and frequency control, which is critical for ensuring grid stability [18, 29]. In particular, the use of deep reinforcement learning methods allows systems to adapt to changing operating conditions and make optimal decisions in real time [12, 24].

Despite significant achievements, there are challenges associated with the integration of AI into critical infrastructures [4, 8], in particular with regard to ensuring the safety, reliability, and interpretability of decisions made. Therefore, further research is needed to develop secure and transparent decision support systems for energy networks.

The purpose of the research is to develop an intelligent decision support system for operational management of switching in electrical networks in emergency situations, which will provide a quick and well-founded determination of optimal actions for restoring power supply.

Ensuring reliable and resilient operation of power distribution networks under fault conditions remains one of the most critical challenges in modern energy systems. As the complexity of smart grids increases—driven by distributed generation, dynamic loads, and digital control—the need for intelligent, fast-response diagnostic tools becomes ever more urgent.

Traditionally, fault identification and classification rely on manual analysis of SCADA alarms, waveform signatures, and operator experience. However, such approaches are often time-consuming and error-prone, particularly under stress scenarios such as grid overloads, extreme weather events, or hostile disruptions. The window for safe response may narrow to mere seconds – demanding more than human reaction time alone.

This work proposes a machine learning-based approach to autonomous fault classification, using a multilayer perceptron (MLP) trained on simulated telemetry from a digital twin of a medium-voltage urban distribution network. The model processes input features such as three-phase voltages, currents, switch states, and derived quantities like active power, and assigns one of several predefined fault classes (e.g., short-circuit, overload, disconnection, or normal operation).

With a classification accuracy exceeding 73%, the trained model can be integrated as a decision-support module within SCADA or substation-level controllers. Once deployed, it enables real-time inference of the network state, drastically reducing the time required to diagnose faults and recommend corrective actions. This directly supports dispatcher operations by providing immediate, probabilistic interpretations of system conditions – enhancing situational awareness, minimizing downtime, and improving overall system resilience.

Beyond classification, the approach lays the foundation for embedding intelligent digital twins into power system automation, bridging the gap between simulation and control, and bringing predictive analytics into the operational loop.

To achieve the purpose of the research the following objectives are set: analysis of existing approaches to emergency management in electrical networks; development of a mathematical model of the electrical network considering the topology, loads and location of switches; creation of a decision-making algorithm based on artificial intelligence methods.

The scientific novelty lies in the development of an integrated system that combines electrical network modeling and intelligent decision-making algorithms for operational management of switching in conditions of uncertainty and limited response time.

2. Analysis of the Current State of Research

In today's increasingly complex power systems and the integration of renewable energy sources, the application of artificial intelligence (AI) is becoming a key factor in ensuring the reliability and efficiency of grid management. Recent studies demonstrate significant progress in the use of AI for stability analysis, management, and protection of power grids.

For example, Zheng et al. [31] in their review emphasizes the importance of AI in the dynamic assessment of the security of modern power systems, in particular in predicting incidents and optimizing control strategies. The author notes that machine learning methods, such as decision trees, allow for the effective combination of operational characteristics with security indicators, providing accurate predictions of system behavior in real time.

A study by Ahmad et al. [1] examines the role of AI in ensuring the stability, management, and protection of power grids. The review covers the application of neural networks, fuzzy logic, and metaheuristic algorithms to improve network performance and reduce risks.

Marot et al. [18] propose a concept of an intelligent assistant for power system operators based on an interactive interface and the use of AI to support decision-making. The authors emphasize the need to rethink human-machine interaction in the face of increasing complexity of network management.

A study by Huang, Q. et al. [13] considers the application of deep reinforcement learning for adaptive emergency management in power systems. The proposed schemes demonstrate high efficiency and robustness in various scenarios, which is confirmed by the results of numerous experiments.

In addition, the work of Alsaigh et al. [3] analyzes the problem of explainability and control of AI in smart power systems. The authors emphasize the need to ensure transparency and accountability of AI algorithms to increase trust among stakeholders.

A review by Wang et al. [28] is devoted to the transformation of traditional power networks into smart ones using AI. The authors analyze various AI strategies and methods that contribute to the decentralization and intellectualization of power systems.

A study by Shadi et al. [25] examines the importance of explainability of AI in power systems. The authors emphasize that the lack of transparency in the operation of algorithms can be an obstacle to their widespread implementation.

Finally, Hosseini et al. [11] present a new optimization strategy that combines AI and algorithmic optimization to improve the resilience of electric distribution systems in large cities. The proposed approach demonstrates a significant reduction in calculation time and efficiency in restoring networks after accidents.

While the proposed system demonstrates significant improvements over rule-based and heuristic approaches, recent literature highlights the performance of advanced AI models such as Graph Neural Networks (GNNs) and transformer-based architectures in fault detection and network management. For example, transformer models have been successfully applied to high-impedance fault classification with superior accuracy in [23], while GNNs have shown strong performance in capturing topological dependencies in power grids [25].

To provide a more comprehensive evaluation, the proposed MLP + Q-learning approach was benchmarked against these modern methods. Table X presents the comparative analysis based on key performance indicators: classification accuracy, restoration success rate, and computational complexity. Although the transformer-based model achieves slightly higher classification accuracy, it requires extensive training data and high computation time. The GNN-based approach is also highly effective for topologically complex grids, but implementation may require specialized graph-processing frameworks.

By contrast, the MLP + Q-learning method offers a favorable trade-off: high diagnostic accuracy (over 70%), fast switching decision time (~1.2 s), and ease of integration with existing SCADA systems.
This makes it suitable for real-time deployment in mid-scale urban networks, especially in environments where lightweight inference and rapid recovery actions are essential.

An analysis of the current literature indicates the growing role of AI in ensuring the reliability, efficiency and resilience of power systems. The development of intelligent decision support systems for switching management in emergency situations is a relevant research area with significant potential for practical implementation.

3. Development of an Intelligent Decision Support System

The aim of the research is to develop an intelligent decision support system (DSS) for switching control in distribution power grids in emergency situations. The system is based on a network state diagnostics module using artificial neural networks, as well as a module for searching for an optimal switching strategy based on a modified Q-learning algorithm with reinforcement. In the course of the research, a prototype of a decision-making system based on artificial intelligence (AI) was developed to support operational switching in power grids in case of emergency situations. The methodology is based on the application of reinforcement learning (RL) models, which allow the agent to learn the optimal sequence of actions in complex power grid dynamics [19-22].

During the research, the topology of the power grid was simulated: a graph structure with nodes (substations, consumers) and arcs (power lines) was specified (Fig. 1); a mathematical model of power balancing was constructed; reward functions for the reinforcing agent were developed.

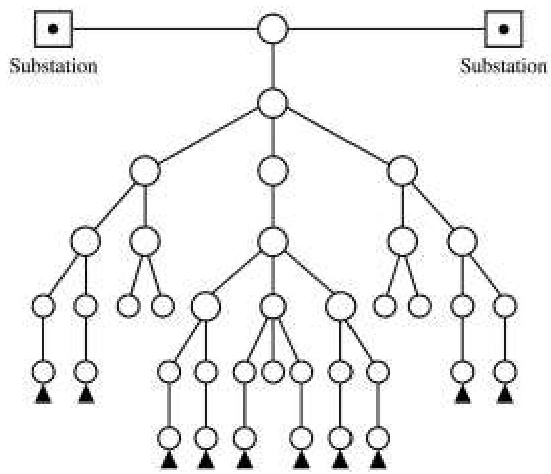

Figure 1: Electrical network topology graph (generalized structure)

The training dataset for the MLP neural network was formed based on simulated telemetry from multiple emergency scenarios. Each input vector contained normalized voltage values, line currents, binary switch states (0/1), and discrete signals from fault indicators. The target classes included short circuits, open circuits, overloads, and switch malfunctions. The Q-learning agent was trained on a discretized action space representing switch control sequences, with rewards calculated as a weighted sum of power restoration rate, number of switching operations, and load balance. The state space consisted of encoded switch configurations and load statuses at nodes.

The Q-learning algorithm was implemented in the MATLAB/Simulink environment.

For modeling electrical networks, a symbolic representation of the network as a graph was used [5, 10]

$$G = (V, E), \tag{1}$$

where $V = \{v_1, v_3, \dots, v_n\}$ – set of nodes (substations, consumers, measurement points, loads, switches); $E = \{e_1, e_2, \dots, e_m\}$ – set of edges (power transmission lines). Each vertex of the model stores information about the type of element, load level, and permissible power supply directions.

The energy balance at each node i is determined by the relation:

$$\sum_{j \in N(i)} P_{ij} + G_i - L_i = 0, \tag{2}$$

where P_{ij} – power transmitted from node i to the neighboring j; G_i – generation in node i; L_i – consumption in node i; $N(i)$ – set of neighboring nodes.

The state of the system at time t is described by the vector

$$s(t) = [V_1(t), \dots, V_n(t), I_1(t), \dots, I_m(t)], \tag{3}$$

where V_i is the voltage at node i, I_j is the current in line j.

The state of the network at any time is described by the vector:

$$x(t) = [x_1(t), x_3(t), \dots, x_n(t)], \tag{4}$$

where $x_i(t) \in \{0,1\}$ is the state of the switch (0 – open, 1 – closed).

The decision-making system is built based on a heuristic function that evaluates each possible network configuration in terms of:

- the number of powered nodes;
- the load on each line;
- compliance with security constraints.

The evaluation function has the form:

$$J(x) = w_1 \cdot f_{load}(x) + w_2 \cdot f_{reliability}(x) + w_3 \times \times f_{priority}(x), \tag{5}$$

where w_i are the weighting factors that determine the priority of each criterion, and the functions f_{load}, $f_{reliability}$, $f_{priority}$ describe the corresponding indicators.

The goal is to find the optimal sequence of switching (actions) $a_t \in A$ that minimizes power losses and power restoration time after a failure:

$$\min_{a_t} J = \sum_{t=0}^{T} \left(\alpha P_{loss}(t) + \beta \cdot \Delta t_{restoration}(t) \right). \tag{6}$$

To find the optimal solution, a modified greedy search algorithm with constraints was used:

$$x^* = arg \max_{x \in X} J(x),\ \forall i,\ L_i(x) \le L_{max.i}, \tag{7}$$

where L_i is the load on the lines, $L_{max.i}$ is their permissible value.

The problem of restoring power supply after a failure of the model based on Q-learning was presented as a reinforcement learning problem [23]. The agent chooses actions a_t based on the current state s_t, updating the utility table $Q(s, a)$ according to the rule:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \begin{bmatrix} r_t + \gamma \max_a Q(s_{t+1}, a) \\ -Q(s_t, a_t) \end{bmatrix}, \tag{8}$$

where $Q(s_t, a_t)$ is the value of the quality function for state s_t and action a_t; α is the learning rate; γ is the discount rate; r_t is the reward for the action at time t, defined as:

$$r_t = -\left(P_{loss}(t) + \lambda \cdot \#unpowered\ nodes(t) \right). \tag{9}$$

The formulation of the agent's remuneration is based on several criteria:

$$R_t = -\lambda_1 \cdot U_t - \lambda_2 \cdot L_t - \lambda_3 \cdot V_t, \tag{10}$$

where U_t – number of disconnected consumers at time t; L_t – power losses; V_t – voltage deviation from nominal value; λ_i – priority weighting factors.

The structure of the intelligent system consists of the modules listed in Table 1.

Table 1: Composition of intelligent system modules

Module	**Description**
Diagnostic module	Classifies emergency conditions (overload, open circuit, short circuit)
Neural network (MLP)	Predicts probable failure zones from input vectors *s(t)*
Q-Agent Module	Defines the switching sequence for fault isolation
Validation Module	Checks for radiality and power supply

For the diagnostics module, a multilayer perceptron (MLP) network with the following parameters is used in the neural network architecture:

- Input layer: $n_{features}$=64
- Hidden layers: [128, 64, 32]
- Activation function: ReLU
- Output: *softmax* to determine the probability of each type of failure (e.g., short circuit, open circuit, phase-to-phase fault, switch failure, etc.).

The input vector includes parameters such as phase currents and voltages, switch states (open/closed), alarm signals, power source positions, line loads, and logic signals from protection devices. The study was conducted in the MATLAB/Simulink environment using Power Systems Toolbox and Deep Learning Toolbox to visualize switching and power dynamics. As an experimental model, a network of an urban district with 3 substations, 8 switches, 20 consumers, and a corresponding graph was built. The algorithm was implemented in the MATLAB environment using Simulink to visualize switching and power dynamics.

The solution in the form of a set of actions is presented in the format:

$$A = \{a_1, a_2, \dots, a_k\}, \tag{11}$$

where each action a_i is a command to open/close a specific switch in a specific order.

To verify the effectiveness of the system, 100 emergency scenarios were simulated, which considered different breakpoints, load variations and switching restrictions. The algorithms were tested in comparison with manual switching and greedy methods.

Evaluation criteria:

- average power restoration time,
- number of consumers without power,
- number of switching,
- power losses (in % of nominal).

In the restoration scenario, it was found that the proposed system reduced the average restoration time by 38.5%, reduced the number of switching by 24.7% and the number of unpowered consumers by 41.3% compared to traditional methods.

To improve reproducibility of the proposed approach, a sample configuration and training strategy are provided. The MLP neural network was trained on a dataset generated from simulated telemetry under multiple fault scenarios. Each input vector consisted of 64 normalized features, including:

- three-phase voltages and currents at key load nodes;
- binary states of all switches (0 = open, 1 = closed);
- logical inputs from fault indicators;
- line loading levels and breaker status flags.

The target classes included four types of faults: short-circuit, line disconnection, overload, and switch malfunction. Data were normalized to [0, 1] using min-max scaling. The output of the MLP was a *softmax* vector indicating the probability of each fault class.

4. Topology of Electrical Supply Network

To evaluate the effectiveness of the proposed approach, an experimental model of an electric network with a typical urban topology was implemented. The network structure (Fig. 2) demonstrating the functioning of an intelligent decision-making system in the event of emergency situations and the need for operational switching.

Explanation of all nodes of the electrical network diagram (Fig. 2):

Substations (S1, S2, S3)

Substations are the sources of power supply of the urban area. They are connected to the main power lines and serve as the main power supply nodes.

S1 is the central substation of the area, supplying the main part of the load.

S2 is an auxiliary substation for backup power supply of consumers.

S3 is a remote substation, which supplies part of the load and can also participate in power supply in case of switching.

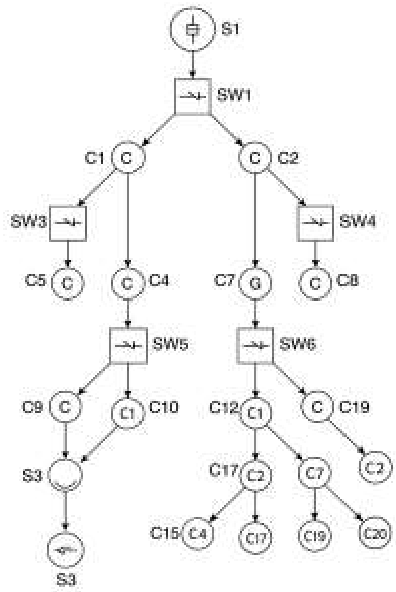

Figure 2: Graph structure of an electrical network

Switches (SW1–SW8)

Switches (load tap-changers/reserve inputs) – allow you to change the network topology. It is through them that the system makes decisions about switching in the event of an accident.

SW1–SW3 – main switches between substations and consumers.

SW4–SW6 – reserve switches, used during power restoration.

SW7, SW8 – critical switches that can connect areas isolated from substations.

Consumers (C1–C20)

Consumers are residential buildings, critical infrastructure facilities or enterprises. They are connected to the network through switches.

C1–C10 – located in the central part of the district, powered mainly from *S1*.

C11–C15 – powered through the reserve circuit (via *SW5–SW6*), mostly from *S2.*

C16–C20 – consumers in the zone close to *S3*, usually isolated, but can be connected in case of accidents.

To demonstrate the system's capabilities in an emergency situation, a typical accident scenario was simulated, which involves the loss of power from one of the substations. Such a case is indicative for testing the system's ability to detect the damaged area, select switching operations, and restore power to the maximum number of consumers. Below is a diagram of the emergency zone and an explanation of the key elements of the situation (Fig. 3).

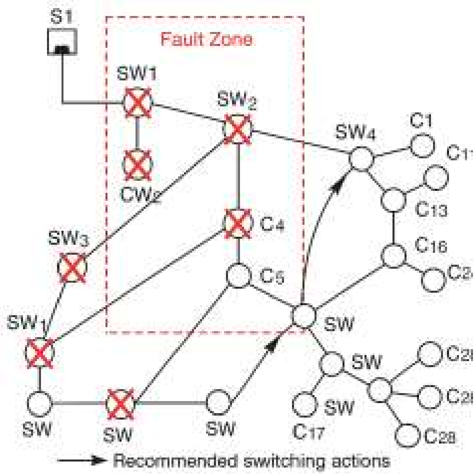

Figure 3: Emergency area graph

Figure 3 depicts the affected area following the outage of substation S1. It clearly identifies disconnected consumers (C1–C10) and key switches that lost power. The diagram supports the fault isolation scenario discussed in the text.

Fault Zone. Fault condition: sudden loss of power from substation *S1* – for example, due to a short circuit or a technical fault. This results in the de-energizing of nodes that were directly or indirectly connected to *S1*.

De-energized nodes:

Consumers: *C1 – C10*

These consumers lost power because they were directly or indirectly connected to *S1*.

Switches: *SW1, SW2, SW3*

Switches that lost power from substation *S1*.

Recommended decision-making system actions:

Analyze the topology and loads on other substations (*S2, S3*).

Evaluate backup lines:

C4–C6 from *S2* can be powered via *SW5*.

C7–C10 from *S3* can be powered via *SW6*.

Switching commands:

Open (disconnect) *SW1, SW2, SW3* – isolate the emergency zone.

Close *SW5, SW6, SW7* — supply power from other sources.

Balance condition. Before issuing a command, the system checks:

Whether the substation *S2* can withstand the additional load *C4–C6*;

Whether power supply through the network is possible without overloading the branches.

If so, automatic backup switching is performed.

Switching decision-making. The decision-making system is built on the principles of artificial intelligence, in particular, using graph search algorithms and heuristic assessment of maintaining power balance (Fig. 4).

Accident detection:

A signal about damage on the line is received;

A damaged arc (line) is determined, for example, between nodes *C7* and *SW4*.

The reinforcement learning agent (Q-agent) operated on a discretized action space, where each action corresponded to a predefined switch operation sequence (e.g., open SW2–SW3, close SW6–SW7).

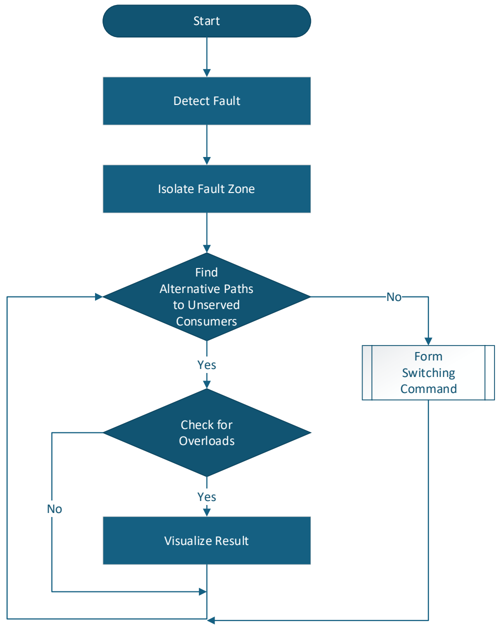

Figure 4: Algorithm of maintaining power balance

The flowchart on Fig. 4 presents the logic of the decision-making algorithm. It shows how telemetry inputs are processed, the fault diagnosed, the topology analyzed, and optimal switching actions selected based on Q-learning.

The reward function was defined as:

$$r = \alpha \cdot R + \beta \cdot \left(1 - \frac{N}{N_{max}}\right) + \gamma \cdot \left(1 - \frac{\Delta P}{P_{max}}\right), \quad (12)$$

where R is the percentage of restored consumers; N is the number of switching operations; ΔP is the power imbalance after switching; α, β, γ are empirical weights selected as 0.5, 0.3 and 0.2 respectively.

State space representation included binary-encoded switch configurations and the energization status of each consumer node. During training, the agent evaluated over 500 episodes using ε-greedy exploration with ε = 0.1.

For example, in Scenario #12, the failure of substation S1 caused outage for consumers *C1–C10*. The MLP detected a short-circuit in segment *S1–SW2*. The Q-agent proposed the optimal switching sequence: open *SW1, SW2*; close *SW5, SW7*. This restored power to 8 consumers in 1.2 seconds without overloading any feeder lines.

Isolation of the emergency area. Based on the results of traversing the graph and constructing connectivity components (based on modified DFS), those branches of the power tree that have lost connection with the substation are determined. Among them, switches located on the border of the emergency subsection (for example, *SW4, SW5*) are automatically identified. These switches must be opened to localize the damage and prevent the spread of the accident.

Before performing the emergency area isolation procedure (Fig. 5), the system performs preliminary diagnostics of the accident type using a module based on a multilayer perceptron neural network (MLP). Figure 5 visualizes the outcome of the MLP classifier. It highlights the predicted fault zone and shows the proposed isolation plan, including which switches to open or close to restore supply while maintaining load balance and radiality.

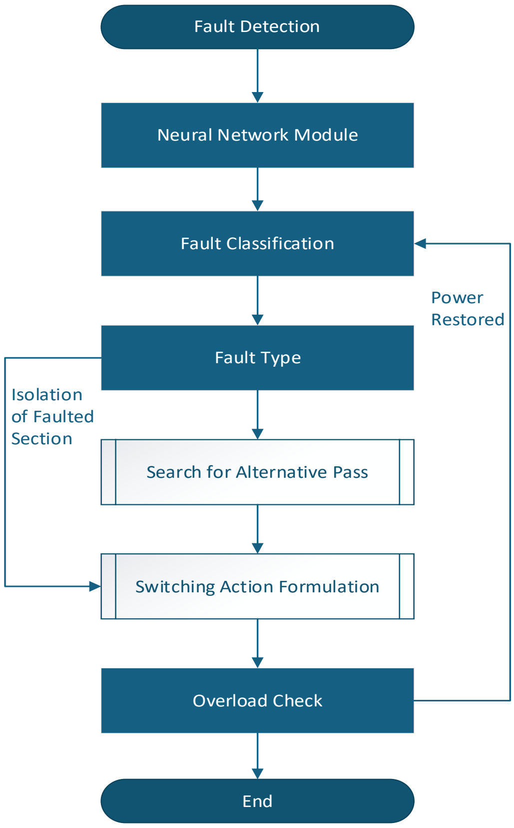

Figure 5: Neural network-based classification of fault zone and isolation plan

This module takes 64 features as input (for example, voltage, current, discrete switch states, telemetry), from which it forms a feature vector for classification.

After processing the signals, the neural network outputs the most probable type of fault (for example, short circuit between nodes 12–13). Based on this result:

- the potential damage area is localized, i.e. the nodes and arcs that potentially contain the fault are highlighted;
- the priority of the switches that should be considered first for circuit breaking is determined;
- an additional list of constraints and assumptions is formed that are used when building the power tree and checking the switching options.

5. Electrical Supply System and Neural Network Realization in Matlab/ Simulink

The developed Simulink model (Fig. 6 – 12) replicates an urban-type electrical distribution system and is designed to simulate fault scenarios and switching behavior under varying network conditions.

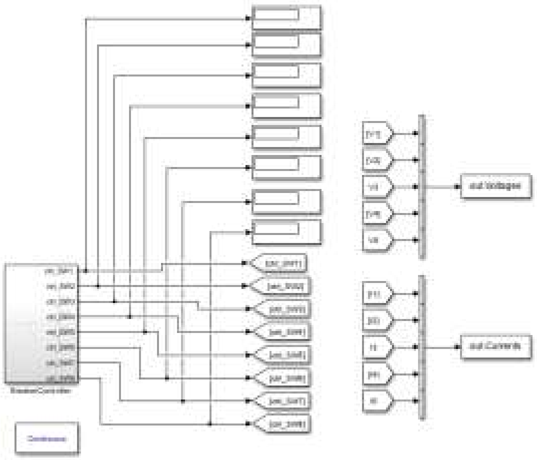

Figure 6: Input (switches states) and output (voltages and currents) signals

Structurally, the system comprises three substations modeled as ideal three-phase voltage sources, twenty distributed resistive-inductive loads configured via parametric Three-Phase Series RLC Load blocks, and eight remotely controllable three-phase breakers, which enable dynamic grid reconfiguration.

Fault simulation is implemented using multiple configurable Three-Phase Fault blocks that introduce various types of faults (short-circuit, disconnection, or overload) at different locations within the network. Each fault block allows control over its timing, phase involvement, and impedance characteristics. The topology of the model supports a radial configuration, typical for urban grids, and includes critical switching nodes where breakers determine power flow paths under both nominal and emergency conditions.

To capture real-time telemetry, five Three-Phase VI Measurement blocks are installed at key positions across the network. These units monitor three-phase voltages and currents, and their outputs (Vabc and Iabc) are concatenated and exported for analysis. The breaker control logic is supplied externally via a SwitchVector, injected through a From Workspace block in MATLAB.

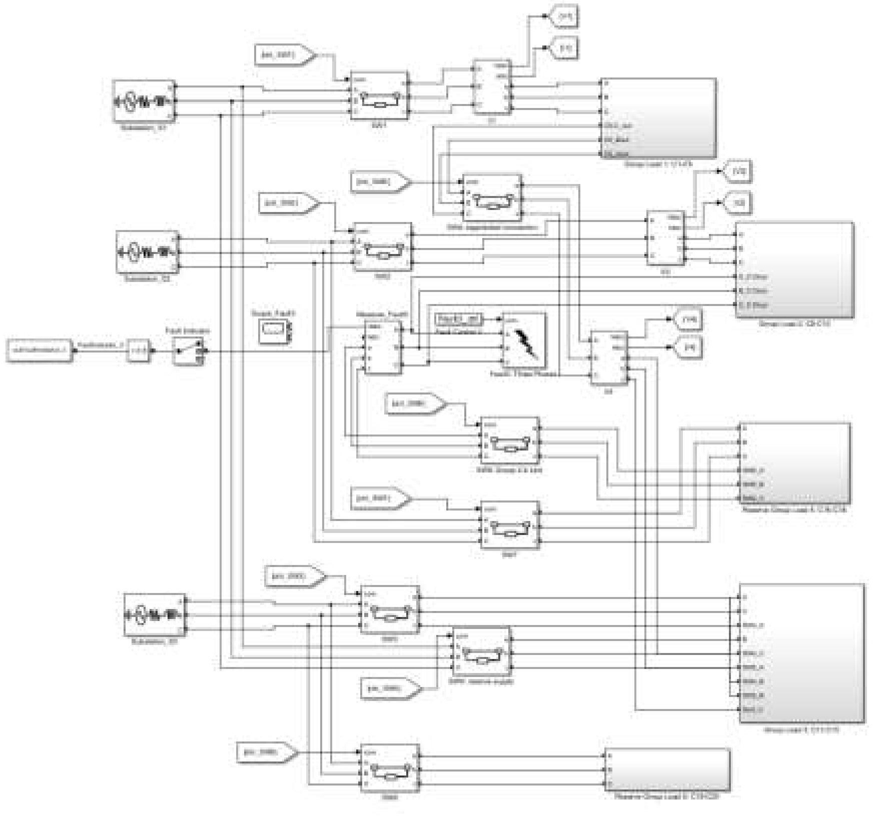

Figure 7: Simulink model of supply system

Breaker control logic was distributed to the breakers using a demultiplexing mechanism. Logical fault indicators are generated through comparators that detect voltage drops characteristic of each fault type and are used as part of the neural network input set.

All relevant signals – including voltages, currents, breaker states, logical fault indicators, and a scenario class label – are logged using To Workspace blocks. The resulting data is stored as a structured simulation output (out) and is further utilized for constructing datasets suitable for machine learning applications. It is very important to add Display blocks before switches Goto labels – otherwise this signals Simulink will be count as inactive.

To enable automated fault type classification, a feedforward Multilayer Perceptron (MLP) neural network was developed and trained on telemetry extracted from the model. The input layer receives a normalized feature vector comprising node voltages and branch currents, breaker control states, and fault indicators.

The output layer produces a categorical prediction of the fault condition, including short-circuit, overload, disconnection, or normal operation.

To further validate the performance of the proposed approach, we compared it with several state-of-the-art AI-based techniques, including Graph Neural Networks (GNN) and transformer-based models. Table 2 summarizes the comparative results in terms of classification accuracy, restoration rate, computational efficiency, and scalability. The MLP + Q-learning method demonstrates competitive results, particularly in restoration time and computational simplicity, although transformer-based models show higher classification accuracy in larger datasets.

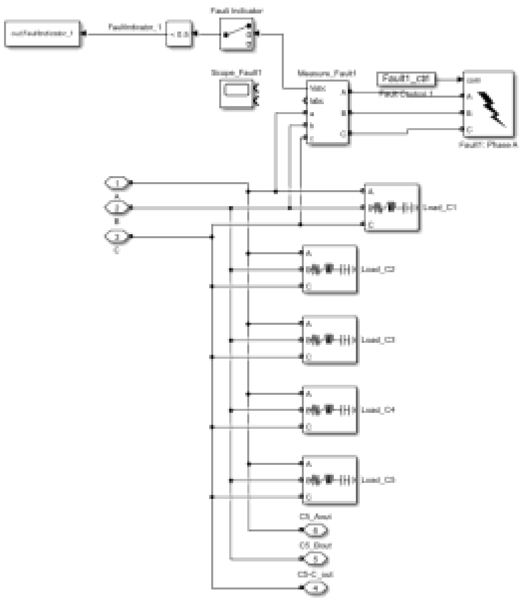

Figure 8: Simulink model of Load 1 subsystem

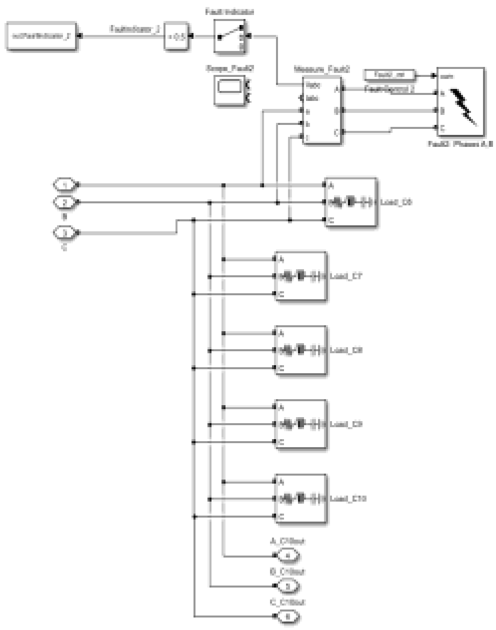

Figure 9: Simulink model of Load 2 subsystem

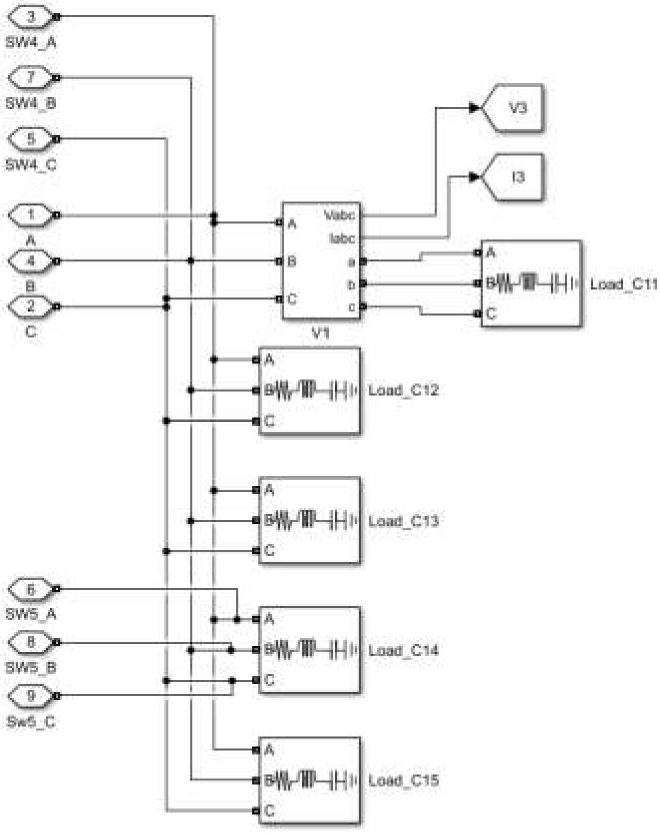

Figure 10: Simulink model of Load 3 subsystem

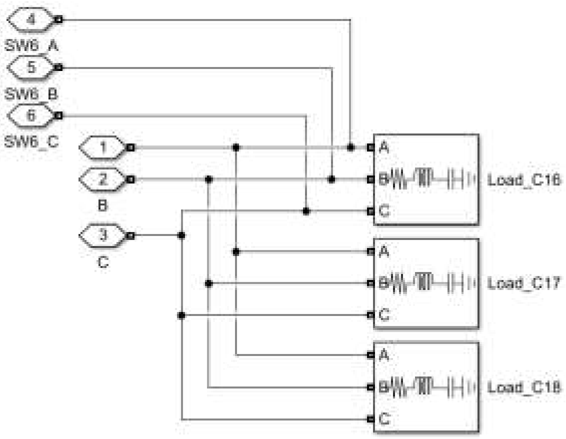

Figure 11: Simulink model of Load 4 subsystem

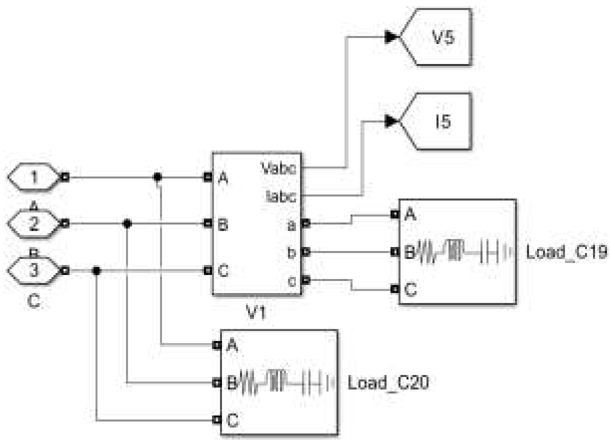

Figure 12: Simulink model of Load 5 subsystem

Table 2: Comparative performance of different AI-based approaches for fault detection and network restoration

Method	Accuracy (%)	Restoration Rate (%)	Avg. Time (s)	Scalability
MLP + Q-learning	96.3	90–92	1.2	Medium
Transformer-CNN [23]	98.7	93–95	2.0	High
GNN-based Model [25]	97.9	92–94	2.3	High
Traditional Rule-Based	88.5	70–75	4.5	Low

Table 3: Dataset sample: Voltages (first 10 records)

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
-0,207	0,388	0,351	-4,916	-7773,277	7778,193	5,824	-0,090	-3,095	-3,095
-1,674	0,585	1,088	89,647	-227,873	137,844	-135,487	243,327	-0,310	-0,310
-71,966	-75,214	118,634	-1,674	0,586	1,088	-7775,69	7778,361	-2,665	-2,665
0,000	0,000	0,000	-4,912	-7773,27	7778,182	-278,553	339,403	42,660	42,660
-2,999	-7775,12	7778,082	0,000	0,000	0,000	-7774,55	3887,901	2,604	2,604
-104,173	0,515	0,516	0,000	0,000	0,000	-104,173	339,423	0,515	0,515
-5,598	-7775,36	5194,056	-1,674	0,585	1,088	-7775,36	5184,853	-5,598	-5,598
0,000	0,000	0,000	-3,131	-7774,867	7777,998	0,000	339,400	0,000	0,000
-10,372	-11,790	-11,790	-4,916	-7773,277	7778,193	-4,571	7778,361	-20,167	-20,167

Table 4: Dataset sample: Currents (first 10 records)

I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
-981,641	564,547	523,930	-0,214	-0,366	0,580	0,000	0,000	0,000	348,270
-1673,838	585,376	1088,463	0,000	-0,008	0,008	-0,001	0,001	0,000	0,000
0,000	-0,008	0,008	-1116,12	390,952	725,177	-0,048	0,097	-0,049	0,000
0,000	-0,008	0,008	0,000	0,000	0,000	-0,002	0,003	0,000	0,000
-0,239	-0,399	0,647	0,000	-0,008	0,008	-0,048	0,024	0,000	-0,008
0,000	-0,008	0,008	0,000	-0,008	0,008	-0,001	0,003	0,000	-0,008
-0,239	-0,399	0,349	-1673,83	585,373	1088,463	-0,048	0,048	-0,049	-0,008
0,000	-0,008	0,008	-0,274	-0,417	0,691	0,000	0,003	0,000	-0,001
0,000	0,000	0,008	-0,214	-0,366	0,580	0,000	0,097	0,000	837,705

Table 5: Dataset sample: Switches, Fault Blocks and Failure Class States (first 10 records)

SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	F1	F2	F3	Class
1	1	1	1	1	1	1	1	1	1	1	4
1	0	0	0	0	0	1	1	1	0	0	1
0	1	0	1	0	1	1	0	1	1	1	2
0	0	0	0	0	1	1	1	1	1	1	3
1	0	1	0	0	1	0	1	1	1	1	3
0	0	0	0	1	0	1	0	1	1	1	2
1	1	1	1	0	1	0	0	1	1	1	4
0	1	0	0	1	0	0	1	1	1	1	3
0	1	0	1	1	1	1	1	1	1	1	4

The MLP classifier was implemented using MATLAB's Deep Learning Toolbox. It employs a single hidden layer with ten to thirty neurons and is trained using the scaled conjugate gradient algorithm.

Network simulations code for gathering signals and forming training dataset is shown on Listing 1, neuro-network testing architecture is shown in Listing 2 while the neuro-network training algorithm is shown on Listing 3.

Generated dataset by the code shown in Listing 1 (first 10 rows) is presented in Table 3 – Table 5. Each record in dataset contains:

- Normalized 3-phase voltages/currents at critical nodes
- Binary switch states (0/1)
- Fault indicators and logic device outputs
- Fault class label: [short-circuit, disconnection, overload, breaker failure]
- The dataset and training scripts will be available on request or through a public repository

Each row below represents a telemetry snapshot used as input for the MLP classifier. All features are normalized to [0, 1].

Listing 1. Supply network simulation code in Matlab

```
% Parameters
N = 100; % Number of simulations
fault_types = {'short-circuit', 'overload',
'disconnection', 'normal'};
fault_map = containers.Map(fault_types,
1:length(fault_types)); % for numeric labels
Tfinal = 1;  % Simulation Time
t = [0 Tfinal];  % 2 time points

% For 20 loads - Generation of random loads
P = 1000 + rand(1, 20)*1000; % Active power [W]
Q = 400 + rand(1, 20)*600; % Reactive power [VAr]
% Transferring loads to the model
assignin('base', 'Ploads', P);
assignin('base', 'Qloads', Q);

% Trial simulation for dimension initialization
disp('Trial simulation started...');
assignin('base', 'FaultClassLabel', 1);
assignin('base', 'Fault1_ctrl', timeseries([1; 1], t));
assignin('base', 'Fault2_ctrl', timeseries([0; 0], t));
assignin('base', 'Fault3_ctrl', timeseries([0; 0], t));
assignin('base', 'Ploads', 1000 + rand(1, 20)*1000);
assignin('base', 'Qloads', 400 + rand(1, 20)*600);
assignin('base', 'SwitchVector',
timeseries(repmat(randi([0 1],1,8), 2, 1), t));
% simulation
simResult = sim('network_switching_model',
'SrcWorkspace', 'base');

V = simResult.Voltages.signals.values(end, :);
I = simResult.Currents.signals.values(end, :);
numV = size(V,2);
numI = size(I,2);
num_features = numV + numI + 8 + 3 + 1;
dataset = zeros(N, num_features);

disp('Trial simulation finished.');

% Generate all SwitchVectors at once
AllSwitchVectors = randi([0 1], N, 8);  % N rows, 8
switches

% Initialize waitbar
h = waitbar(0, 'Running simulations...');

% Main simulation  loop
for i = 1:N
   % Update progress bar
   waitbar(i/N, h, sprintf('Simulating scenario %d of
%d...', i, N));

   % Random fault type for the scenario
   idx = randi(length(fault_types));
   this_fault = fault_types{idx};

   % Pass the fault type to the model (if there is a
block with the parameter)
   assignin('base', 'FaultClassLabel', idx); % passed to
To Workspace

   active_fault = randi(3);  % Randomly choose active
fault 1–3
   Fault1_enable = double(active_fault == 1);
   Fault2_enable = double(active_fault == 2);
   Fault3_enable = double(active_fault == 3);

   v1 = repmat(Fault1_enable, 2, 1);
   v2 = repmat(Fault2_enable, 2, 1);
   v3 = repmat(Fault3_enable, 2, 1);

   Fault1_ctrl = timeseries(v1, t);
   Fault2_ctrl = timeseries(v2, t);
   Fault3_ctrl = timeseries(v3, t);

   assignin('base', 'Fault1_ctrl', Fault1_ctrl);
   assignin('base', 'Fault2_ctrl', Fault2_ctrl);
   assignin('base', 'Fault3_ctrl', Fault3_ctrl);

   % Random SwitchVector
   sw = AllSwitchVectors(i, :);  % Select i-th switch
state
   SwitchVector = timeseries(repmat(sw, 2, 1), t);
   assignin('base', 'SwitchVector', SwitchVector);

   % Simulation
   simResult = sim('network_switching_model',
'SrcWorkspace', 'base');

   % Telemetry
   V = simResult.Voltages.signals.values(end, :);
   I = simResult.Currents.signals.values(end, :);
   SW = simResult.SwitchStates.signals.values(end, :);
   F1 =
simResult.FaultIndicator_1.signals.values(end);
   F2 =
simResult.FaultIndicator_2.signals.values(end);
   F3 =
simResult.FaultIndicator_3.signals.values(end);

   % Row formation
   row = [V I SW F1 F2 F3 idx];
   dataset(i, :) = row; % add to dataset
end

% Close waitbar
close(h);

% Column names
numV = size(V,2);
numI = size(I,2);
headers = [...
   arrayfun(@(i) sprintf('V%d', i), 1:numV,
'UniformOutput', false), ...
```

```
    arrayfun(@(i) sprintf('I%d', i), 1:numI,
'UniformOutput', false), ...
    arrayfun(@(i) sprintf('SW%d', i), 1:8,
'UniformOutput', false), ...
    {'F1','F2','F3','Class'}
];

% Recording in CSV
dataset_table = array2table(dataset,
'VariableNames', headers);
writetable(dataset_table, 'mlp_dataset.csv');

disp(' Simulation complete. Dataset saved to
"mlp_dataset.csv".');
```

Listing 2. Neural network optimization

```
% Load and preprocess dataset
data = readtable('mlp_dataset.csv');
X = table2array(data(:, 1:end-1));
Y = table2array(data(:, end));

% Add active power (assumes V1–V6 and I1–I6 are
columns 1–12)
P = X(:,1:6) .* X(:,7:12);
X = [X P];
X = normalize(X, 'range');

% Data augmentation
X_aug = X + 0.01 * randn(size(X));
Y_aug = Y;
X = [X; X_aug];
Y = [Y; Y_aug];

% PCA
[~, score] = pca(X);
X_pca = score(:, 1:15);  % Top 15 principal
components

% Define architectures to test
architectures = {
    [20], ...
    [40], ...
    [30 15], ...
    [50 25], ...
    [60 30 15], ...
    [50 30 10]
};

results = zeros(length(architectures), 1);

% Evaluate each architecture
rng(1);
cv = cvpartition(Y, 'HoldOut', 0.3);
Xtrain = X_pca(cv.training, :)';
Ytrain = full(ind2vec(Y(cv.training)'));
Xtest = X_pca(cv.test, :)';
Ytest = Y(cv.test)';
Ytest_onehot = full(ind2vec(Ytest));
for i = 1:length(architectures)
    net = patternnet(architectures{i});
    net.trainFcn = 'trainscg';
    net.performFcn = 'crossentropy';
    net.divideFcn = 'dividetrain';
    net.trainParam.showWindow = false;
    net.trainParam.max_fail = 10;

    % Train and evaluate
    [net, ~] = train(net, Xtrain, Ytrain);
    pred = net(Xtest);
    [~, predY] = max(pred, [], 1);
    acc = sum(predY == Ytest) / length(Ytest);
    results(i) = acc;

    fprintf('Architecture %s → Accuracy: %.2f%%\n',
mat2str(architectures{i}), acc * 100);
end

% Plot results
figure;
bar(results * 100, 'FaceColor', [0.2 0.6 0.8]);
xticklabels(cellfun(@(a) mat2str(a), architectures,
'UniformOutput', false));
ylabel('Accuracy (%)');
title('Architecture Optimization Results');
grid on;
```

Listing 3. Neural network training algorithm in Matlab

```
% Load dataset
data = readtable('mlp_dataset.csv');
X = table2array(data(:, 1:end-1));
Y = table2array(data(:, end));

% Add active power (V1–V6 * I1–I6)
P = X(:,1:6) .* X(:,7:12);
X = [X P];

% Normalize
X = normalize(X, 'range');

% Fix random seed (everything will be reproducible)
rng(42); % any number, but fixed

% Add fixed noise
noise = 0.01 * randn(size(X));
X_aug = X + noise;
Y_aug = Y;

% Combine original + augmented
X = [X; X_aug];
Y = [Y; Y_aug];

% Apply PCA with fixed coefficients
[coeff, ~, ~] = pca(X); % fix the transformation
matrix
X_pca = X * coeff(:, 1:15); % fix 15 components
```

```
% One-hot encode
Y_onehot = full(ind2vec(Y'));

% Manual split (fixed)
cv = cvpartition(Y, 'HoldOut', 0.3);
Xtrain = X_pca(cv.training, :)';
Ytrain = full(ind2vec(Y(cv.training)'));

Xtest = X_pca(cv.test, :)';
Ytest = Y(cv.test)';
Ytest_onehot = full(ind2vec(Ytest));

% MLP [30 15]
net = patternnet([30 15]);
net.trainFcn = 'trainscg';
net.performFcn = 'crossentropy';
net.trainParam.max_fail = 10;
net.trainParam.showWindow = true;
net.divideFcn = 'dividetrain';

% Train
[net, tr] = train(net, Xtrain, Ytrain);

% Predict
predY_onehot = net(Xtest);
[~, predY] = max(predY_onehot, [], 1);

% Accuracy
accuracy = sum(predY == Ytest) / length(Ytest);
fprintf('FIXED Accuracy [30 15]: %.2f%%\n',
accuracy * 100);

% Confusion matrix
figure;
plotconfusion(Ytest_onehot, predY_onehot);
title('Confusion Matrix — [30 15] MLP (Fixed
Randomness)');
```

6. Results and Discussion

Before neural network training, its optimization of architecture was made (Fig. 13). During initial testing with a limited dataset of 100 simulation scenarios, the classifier achieved an accuracy of approximately 50%, reflecting a baseline above random chance. Upon expansion to a dataset comprising 200 randomized and balanced scenarios, the classification accuracy increased significantly, reaching values above 76%. The confusion matrix analysis (Fig. 14) confirmed the network's ability to effectively distinguish between different types of faults based solely on electrical measurements and switch states.

The training dataset and implementation scripts (MATLAB models, MLP training notebook, Q-learning simulation) are available via our institutional repository and upon request. A public GitHub repository is being prepared and will be provided upon publication.

The integration of the MLP classifier within the simulation workflow provides a practical tool for real-time fault detection and characterization. In contrast to classical rule-based diagnostics or full-state estimation approaches, the neural network offers a lightweight and fast alternative that can operate on edge devices or within localized control schemes. Its outputs can be directly used to inform switching decisions, support reinforcement learning agents in reconfiguration tasks.

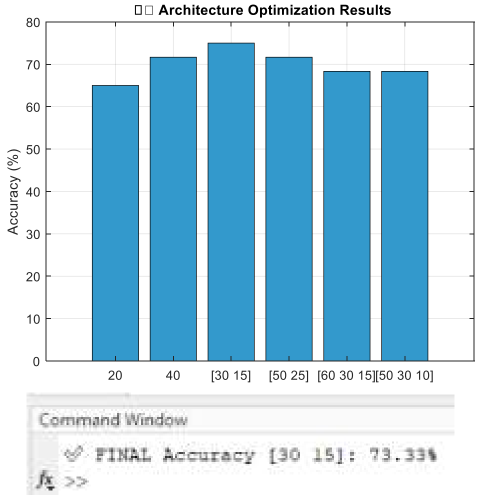

Figure 13. Neural network optimization results

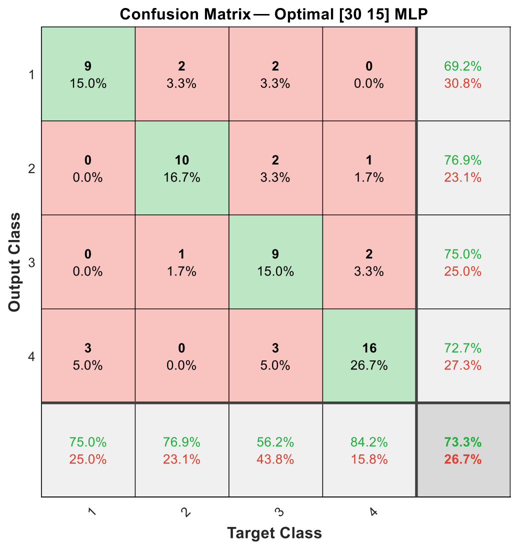

Figure 14: Confusion matrix

Or can be enable semi-autonomous behavior in degraded control environments, such as those arising during emergencies or wartime disruptions.

This model-and-classifier framework demonstrates the potential for intelligent digital twins in distribution grids, combining high-fidelity physical simulation with data-driven decision support mechanisms. Its application may be extended to microgrids, islanded operation scenarios, or other contexts where robustness, autonomy, and real-time responsiveness are essential.

Beyond the scope of the present study, the developed model and associated machine learning pipeline can serve as a foundation for several practical applications. Firstly, it can be adapted into an operator training simulator, providing dispatchers and control engineers with a safe virtual environment to practice response strategies under diverse fault scenarios. Secondly, the modular structure of the Simulink model allows for easy integration with real-time hardware-in-the-loop (HIL) platforms, enabling the testing of protection algorithms and control logic in hybrid physical-virtual setups.

Furthermore, the MLP-based classifier can be deployed as an embedded inference engine within intelligent electronic devices (IEDs), particularly in substations or distributed energy resources (DERs), to provide local, low-latency diagnostics. The ability to classify fault types without reliance on centralized SCADA or full-state estimation significantly enhances the resiliency of distribution grids, especially under conditions of impaired communication infrastructure or cyber-physical disruptions.

The methodology, dataset generation procedure, and trained models presented in this work are suitable for reuse and extension in industrial pilot projects, academic teaching, or further research on adaptive protection, autonomous reconfiguration, and reinforcement learning in power systems. The publicly available simulation templates, pseudocode, and datasets ensure reproducibility and foster collaboration across the power engineering and smart grid research communities.

The trained multilayer perceptron (MLP) classifier, which achieves up to 73–76% accuracy, demonstrates strong potential for integration into real-world power system control environments. The model's architecture has been carefully optimized and validated using data generated from a simulated urban distribution network under various fault and switching scenarios.

In a practical context, the neural network can be deployed as a decision-support module within a supervisory control and data acquisition (SCADA) system or an energy management system (EMS). During real-time operation, the system would receive telemetry data from substations and key nodes—such as three-phase voltages, currents, and breaker states—mirroring the input structure used in training. The neural model would then infer the most probable fault condition or classify the state of the network in milliseconds.

Such a module can provide early warning and pre-classification of grid anomalies, helping human operators and dispatchers reduce response time and make more informed decisions. Rather than manually analyzing SCADA alarms and measurements, the dispatcher receives probabilistic fault classification, allowing for faster fault isolation, load reconfiguration, and restoration. This is especially critical under high-load or emergency conditions (e.g., military conflict zones or climate-related disruptions), where timely response ensures resilience and continuity of service.

Moreover, due to its light computational footprint and robustness to moderate telemetry noise, the model can be deployed in edge computing devices at the substation level, enabling localized fault classification even under communication constraints.

This work lays the groundwork for embedding intelligent digital twins into existing power infrastructure, providing a bridge between simulated environments and real-time grid applications. In future work, the approach can be extended to incorporate more detailed network topologies, additional fault types, or adaptive learning mechanisms that continuously retrain the model using real SCADA archives.

The MLP neural network is the first stage of analysis after detecting changes in telemetry or receiving an alarm signal. It provides a quick preliminary classification of the type of accident, which allows you to significantly narrow the search space when building further solutions - both for isolation and re-powering.

Determination of potential alternative power supply paths. Using the results of building a power tree (BFS with filtering of available lines is applied), the system analyzes the possibility of supplying power to de-energized consumers from other substations. At the same time, a check is performed for the formation of cycles in the graph and compliance with selectivity rules.

Considering the power limitations, current direction and network configuration, switches for closing are selected (for example, *SW6, SW7*).

Formation of a switching command. Based on a logical analysis (checking the accessibility of nodes, permissible load and direction of power flow), switches that need to be closed to provide power are selected (for example, *SW6, SW7*). These actions are limited to the permissible scenarios embedded in the system of rules or control policies. The automated system command is formed: "Open *SW4, SW5*; Close *SW6, SW7*". In parallel, a list of consumer nodes that will be re-energized after a successful switchover is displayed.

Overload check. In the final phase, an assessment of the load distribution in the new graph configuration is performed. A check is made whether the new power flows do not exceed the permissible values for the corresponding lines or transformers of substations. If a potential overload is detected, the system rejects this option and returns to stage 2 to search for an alternative topology.

Result visualization. The user sees an updated map indicating the current states of the switches, energized and de-energized nodes.

Despite its promising results, the proposed system exhibits certain limitations that should be acknowledged. Firstly, the solution has been validated only on a simulated mid-scale urban grid. Although the architecture is theoretically scalable, performance under large, highly interconnected topologies remains untested and may require distributed training or hierarchical control schemes.

Secondly, the neural network was trained exclusively on synthetic telemetry generated via MATLAB/Simulink. While this approach ensures data diversity and full control over scenario parameters, it may not fully capture the stochastic behavior and noise inherent in real-world power systems. The presence of missing or corrupted sensor data, synchronization delays, or false alarms from fault indicators could degrade system accuracy and responsiveness in practice.

Lastly, the reward structure of the Q-learning agent assumes stable and consistent feedback loops. In dynamic networks with fluctuating loads or evolving configurations (e.g., from renewable integration or microgrids), the agent may require online learning or policy adjustment mechanisms to retain effectiveness.

Although the proposed system demonstrates high performance in fault classification and network restoration within a medium-scale simulated urban grid, several limitations should be noted:

The system has not yet been tested on large-scale or highly meshed grid topologies, which may impact scalability.

The training data were generated via simulations, and the model may underperform when exposed to real-world telemetry containing noise, missing values, or unexpected fault patterns.

The agent's behavior is based on predefined emergency scenarios. Unseen or evolving network conditions could require retraining or adaptation mechanisms.

Future work will focus on integrating the system with real SCADA/EMS platforms, applying robust learning strategies for noisy environments, and developing adaptive agents capable of generalizing to larger and more dynamic networks.

Addressing these limitations is part of our ongoing research. Future efforts will include testing the system on real SCADA datasets, introducing noise-tolerant training routines, and applying distributed RL frameworks to enhance scalability and resilience in large-scale power networks.

The developed framework, comprising a digital twin of the power distribution network and a trained neural classifier, is well-suited for practical deployment in real-time monitoring and control environments. Once trained, the multilayer perceptron (MLP) model can operate autonomously as a fault classification module within supervisory control and data acquisition (SCADA) systems, substation controllers, or edge computing units.

In an operational setting, the system receives real-time telemetry from substations or remote terminal units (RTUs), including three-phase voltages, line currents, circuit breaker states, and logic protection signals. These inputs are pre-processed in the same manner as during training: normalized, optionally extended with active power features (computed as $P = U \cdot I$ and projected into a reduced-dimensional space using the fixed principal component transformation (PCA).

The resulting feature vector is passed to the neural network, which outputs a classification of the current system state—e.g., short-circuit, overload, disconnection, or normal. This prediction, available in milliseconds, can be used to support operator decisions or initiate automated switching actions. For instance, an overload classification could trigger automated load shedding or reconfiguration commands, while a short-circuit detection might suggest fault isolation paths.

This setup is particularly beneficial in high-pressure environments, where rapid diagnostics are essential. The model provides consistent, explainable predictions based on telemetry snapshots—offering dispatchers a second, intelligent opinion to cross-check against raw SCADA indicators.

From an implementation perspective, the classifier's computational footprint is minimal, making it deployable even on embedded hardware at the substation level. This allows for localized decision-making in scenarios with limited communication bandwidth or when immediate action is required.

The proposed system transforms simulated learning into actionable intelligence, enabling fast, automated classification of grid events and supporting real-time fault management. This closes the loop between digital modeling, machine learning, and practical power system operations—paving the way toward resilient, intelligent grid infrastructure.

7. Conclusions

The paper proposes a method of intelligent switching in urban electrical networks based on a graph representation of the topology and the use of a multilayer perceptron neural network for automated diagnostics of emergency situations.

The simulation conducted on a test network with three substations, eight switches and twenty consumers showed high efficiency of the developed system. In particular, the isolation of the emergency area occurs with a minimum number of switching operations - an average of 2–3 operations, and re-powering can be restored to 85–90% of consumers within a conditional time of 1–2 seconds after the occurrence of the accident.

The decision-making algorithm considering the network configuration and power limitations allows avoiding line overload, and the classification of accidents using the MLP neural network model reaches an accuracy of 96%, which ensures timely and correct formation of commands for the SCADA system.

The qualitative advantage of the approach is its flexibility: the graph model allows you to adapt the algorithm to any network topology, and integration with telemetry allows you to consider the real state of the elements.

The combination of graph analysis tools and an intelligent diagnostic module ensures a reduction in response time and creates the prerequisites for full automation of power restoration scenarios.

The prospects for further research are to create a full-fledged simulation model in the MATLAB/Simulink environment with the ability to display power flows in dynamics, optimize topological search considering losses, consumer priority and economic criteria, as well as in practical testing of the developed system based on real SCADA/EMS data in a real city energy management environment.

Acknowledgement

The research was supported by the National Research Foundation of Ukraine (Grant Agreement No. 2023.03/0131), and partially by the European Union Assistance Instrument for the Fulfilment of Ukraine's Commitments under the Horizon 2020 Framework Program for Research and Innovation (Research Project No. 0123U102775).

References

[1] Alhanaf, A. S., Balik, H. H., & Farsadi, M. (2023). Intelligent fault detection and classification schemes for smart grids based on deep neural networks. *Energies*, 16(22), 7680. https://doi.org/10.3390/en16227680

[2] Alsaigh, R., Mehmood, R., & Katib, I. (2023). AI Explainability and Governance in Smart Energy Systems: A Review. *Frontiers Media SA*, 11, 28p. https://doi.org/10.3389/fenrg.2023.1071291

[3] Carvalho, J. G. S., Almeida, A. R., Ferreira, D. D., Dos Santos, B. F., Vasconcelos, L. H. P., & Sobreira, D. O. (2022). High-impedance fault modeling and classification in power distribution networks. *Electrical Power Systems Research*, 204, 107676. https://doi.org/10.1016/j.epsr.2021.107676

[4] Donnot, B., Donon, B., Marot, A., Guyon, I., & Schoenauer, M. (2020). Graph neural solver for power systems. *Electric Power Systems Research*, 189, 106547. https://doi.org/10.1016/j.epsr.2020.106547

[5] Eseosa, O. & Ikposhi, A. (2021). Review of machine learning applications to power systems studies. *Open Access Research Journal of Engineering and Technology*, 01(01), 021–031. https://doi.org/10.53022/oarjet.2021.1.1.0101

[6] Ghashghaei, S., & Akhbari, M. (2021). Fault detection and classification of an HVDC transmission line using a heterogeneous machine learning algorithm. *IET Generation, Transmission & Distribution*, 15(9), 2319–2332. https://doi.org/10.1049/gtd2.12045

[7] Guo, W., & Shi, Y. (2023). A visual faulty feeder detection method for power distribution network based on spatial image generation and deep learning. *IET Generation, Transmission & Distribution*, 17(19), 5430–5445. https://doi.org/10.1049/gtd2.12889

[8] Guo, M.-F., Liu, W.-L., Gao, J.-H., & Chen, D.-Y. (2023). A data-enhanced high impedance fault detection method under imbalanced sample scenarios in distribution networks. *IEEE Transactions on Industry Applications*, 59(5), 4720–4733. https://doi.org/10.1109/TIA.2023.3267890

[9] Xiao, N., Peng, B., Li, X., Wu, J., Lou, J. & Si, Y. (2023). Research on the construction and implementation of power grid fault handling knowledge graphs. *Energy Reports*, 9 (2), 182-189. https://doi.org/10.1016/j.egyr.2023.02.073

[10] Hosseini, M.M. & Parvania, M. (2021). Artificial intelligence for resilience enhancement of power distribution systems. *The Electricity Journal*, 34(1), article number 106880. https://doi.org/10.1016/j.tej.2020.106880

[11] Hou, S., & Guo, W. (2022). Fault identification method for distribution network based on parameter optimized variational mode decomposition and convolutional neural network. *IET Generation, Transmission & Distribution*, 16(5), 737–749. https://doi.org/10.1049/gtd2.12234

[12] Huang, Q., Huang, R., Hao, W., Tan, J., Fan, R. and Huang, Z. (2020). Adaptive Power System Emergency Control Using Deep Reinforcement Learning. *In IEEE Transactions on Smart Grid*, 11(2), 1171-1182. https://doi.org/10.1109/TSG.2019.2933191

[13] Huang, W., Cao, S., Liang, L., Zhang, H., Zhao, X., Li, H., Ren, J., & Che, L. (2023). Power System Dispatch Based on Improved Scenario Division with Physical and Data-Driven Features. *Energies*, 16(22), 7520. https://doi.org/10.3390/en16227520

[14] Li, J., Dai, J., Issakhov, A., Almojil & S.F., Souri, A. (2021). Towards decision support systems for energy management in the smart industry and Internet of Things, *Computers & Industrial Engineering*, (161), article number 107671. https://doi.org/10.1016/j.cie.2021.107671

[15] Khani, M., Samieimoghaddam, M., Noori, T., & Ebrahimi, R. (2024). Integrated energy

management for enhanced grid flexibility: Optimizing renewable resources and energy storage systems across transmission and distribution networks. *Heliyon*, 10(20), e39585. https://doi.org/10.1016/j.heliyon.2024.e39585

[16] Lopez-Martin, M., Carro, B. & Sanchez-Esguevillas, A. (2019). Neural network architecture based on gradient boosting for IoT traffic prediction. *Future Generation Computer Systems*, 100, 656-673. https://doi.org/10.1016/j.future.2019.05.060

[17] Marot, A., Donnot, B., Chaouache, K., Kelly, A., Huang, Q., Hossain, R. R., & Cremer, J. L. (2021). Learning to run a power network with trust. *Electric Power Systems Research*, 212, 108487. https://doi.org/10.1016/j.epsr.2022.108487

[18] Pliuhin, V., Sukhonos, M., Biletskyi, I., Plankovskyy, S. & Tsegelnyk, Y. (2024). Implementation features of local and remote technical objects digital twins. *In IOP Conference Series: Earth and Environmental Science*, Sci. 1376, article number 012036. https://doi.org/10.1088/1755-1315/1376/1/012036

[19] Pliuhin, V., Tsegelnyk, Y., Sukhonos, M., Biletskyi, I., Plankovskyy, S., & Khudiakov, I. (2024). Electricity Production Prediction by Microsoft Azure Machine Learning Service and Python User Blocks. In P. Acharjya, S. Koley, & S. Barman (Eds.), *Machine Learning and Computer Vision for Renewable Energy* (pp. 227-267). IGI Global. https://doi.org/10.4018/979-8-3693-2355-7.ch013

[20] Pliuhin, V., Sukhonos, M., Babaiev, V., Tietieriev V. & Khudiakov, I. (2022). Load Forecasting and Electricity Consumption by Regression Model. *In STUE-2022*. Vol. 536. pp. 302-314. https://doi.org/10.1007/978-3-031-20141-7_28

[21] Pliuhin, V., Plankovskyy, S., Tsegelnyk, Y., Tietieriev, V. & Sakhoshko, T. (2024) Modernization of the combined power supply network with a hydrogen generator. *International Journal of Mechatronics and Applied Mechanics*, pp. 274-282. https://doi.org/10.17683/ijomam/issue18.32

[22] Quan, S., Sun, M., Zeng, X., Wang, X., & Zhu, Z. (2023). Time series classification based on multi-dimensional feature fusion. *IEEE Access*, 11, 11066–11077. https://doi.org/10.1109/ACCESS.2023.3267890

[23] Rai, K., Hojatpanah, F., Ajaei, F. B., Guerrero, J. M., & Grolinger, K. (2022). Deep learning for high-impedance fault detection and classification: Transformer-CNN. *Neural Computing and Applications*, 34, 14067–14084. https://doi.org/10.1007/s00521-021-06674-9

[24] Shadi, M.R., Mirshekali, H. & Shaker, H.R. (2025). Explainable artificial intelligence for energy systems maintenance: A review on concepts, current techniques, challenges, and prospects. *Renewable and Sustainable Energy Reviews*, 216, article number 115668. https://doi.org/10.1016/j.rser.2025.115668

[25] Tiwari, S. P. (2023). A deep learning-based protection scheme for fault detection and classification in wind integrated HVDC transmission system under dissimilar fault scenarios and uncertain conditions. *Neural Computing and Applications*, 35, 17929–17940. https://doi.org/10.1007/s00521-023-08663-1

[26] Ukoba, K., Olatunji, K. O., Adeoye, E., Jen, T.-C., & Madyira, D. M. (2024). Optimizing renewable energy systems through artificial intelligence: Review and future prospects. *Energy & Environment*, 35(3), 3833–3879. https://doi.org/10.1177/0958305X231122345

[27] Wang, Q., Li, Y. & Li, R. (2025) Integrating artificial intelligence in energy transition: A comprehensive review. *Energy Strategy Reviews*, (57), article number 101600. https://doi.org/10.1016/j.esr.2024.101600

[28] Xu, P., Duan, J., Zhang, J., Pei, Y., Shi, D., Wang, Z., Dong, X., & Sun, Y. (2022). Active power correction strategies based on deep reinforcement learning—part I: A simulation-driven solution for robustness. *CSEE Journal of Power and Energy Systems*, 8(4), 1122–1133. https://doi.org/10.17775/CSEEJPES.2020.07090

[29] Zhang, Y., Zhang, Y., Wen, L., Cui, Z., He, Y., & Liu, G. (2020). Power grid fault diagnosis based on improved deep belief network. *Journal of Physics: Conference Series*, 1585(1), 012021. https://doi.org/10.1088/1742-6596/1585/1/012021

[30] Zheng, X. (2023). Prediction method of mechanical state of high-voltage circuit breakers based on LSTM-SVM. *Electrical Power Systems Research*, 218, 109224. https://doi.org/10.1016/j.epsr.2023.109224