An Intelligent Fault Prediction And Self-Healing Framework For Electrical Automation Systems Using Deep Reinforcement Learning

Songhua Cao

Shijiazhuang College of Applied Technology, Department of electrical and electronic engineering, Shijiazhuang, 050800, China

Corresponding author. E-mail: SonghuaCao@outlook.com

Received March 29, 2026; accepted May 27, 2026

Modern smart grids have electrical automation systems that need smart mechanisms to predict faults early and recover fast to maintain the system stability and reliability. Nonetheless, the traditional fault detection methods have a weakness in that they cannot effectively represent the temporal relationships in power system signals, slow recovery processes, and excessive system downtime in response to disturbances. To overcome these difficulties, the present paper proposes an intelligent fault prediction and self-healing system, which combines a Temporal Convolutional Network (TCN) and Deep Reinforcement Learning (DRL). The model then predicts possible faults through the TCN model that is trained on the IEEE-39 bus power system data with the time-series data on voltage, frequency and power flow parameters under different operating conditions. The DRL agent subsequently becomes aware of the best control measures to curb flaws and reinstate stability in the system. The suggested model is able to withstand the shortcomings of conventional approaches due to its ability to integrate precise learning of temporal patterns with the adaptability of decision-making. The experimental findings indicate that the proposed model, which gives 98.2% accuracy, 97.5% precision, 98.8% recall, and 98.1% F1-score and has an ROC-AUC of 0.985, indicates high accuracy and precision. Also, the system demonstrates an average recovery time of 120 ms and a voltage deviation of 0.02 p.u., which is much higher than the grid resilience and operational reliability. Overall, the proposed framework provides an effective and intelligent solution for predictive fault management and automated self-healing in modern electrical automation systems.

Keywords: Fault Prediction, Self-Healing Power Systems, TCN, DRL, Smart Grid Automation, Electrical Automation Systems

http://dx.doi.org/10.6180/jase.202610_33.033

1. Introduction

Modern industrial and energy infrastructure is made up of power systems [1]. Industry 4.0 and IIoT raise connectivity and fault risks; edge computing improves containment, making fault diagnosis essential [2]. TCN–DRL combines prediction and control, enabling fault detection and self-healing in one framework. These models improve accuracy and adaptability but only provide prediction, not corrective action [3]. DRL is effective for complex systems where traditional methods fail [4]. ML-based optimisation improves stainless steel–reinforced concrete beam design efficiency [5]. Reliability-based model enhances serviceability deflection prediction for RC beams [6]. ML/DL applications focus on design optimisation, not integrated prediction task and control [7]. The inverse ML framework predicts optimal geometry and materials for perforated beams [8]. Reliability analysis improves buckling and safety prediction of steel CHS beam-columns [9]. DRL-based self-healing enables real-time prediction task, detection, and recovery, improving reliability and reducing downtime. Develop a TCN–DRL framework for coordinated fault detection and control in smart grid systems.

- Develop a DRL-based self-healing strategy for optimal fault handling and stability.
- Develop an integrated prediction–control system for automated fault recovery.

Section 2 presents the materials and methods; Section 3 covers results and discussion; Section 4 provides the conclusion.

2. Materials and methods

Predictive maintenance has also been attracting a lot of concern due to its IoT capability. Dui et al. [10] proposed an IoT-based predictive maintenance model for real-time fault detection using sensor data analytics. To reduce sensor faults, Cai and Zhou [11] developed a self-healing fault-tolerant control framework using redundancy for system stability. Transformers improve fault detection via long-range dependencies Baadji et al. [12]. GNNs capture spatial–temporal relationships for better fault detection Ngo et al. [13]. DRL enables real-time fault detection and self-healing to improve reliability and reduce downtime.

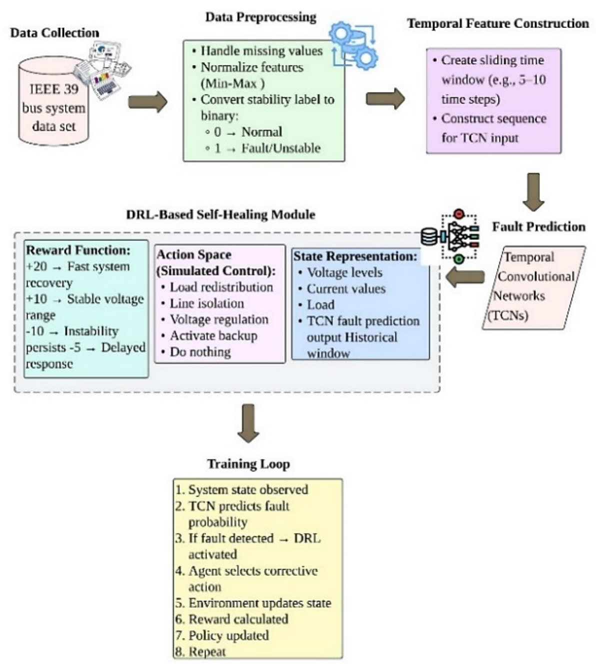

Fig. 1. Proposed DRL-Based Fault Prediction and Self-Healing Framework Architecture

Figure 1 shows TCN prediction and DRL self-healing with mean imputation. Missing voltage, current, and load data are filled using mean imputation for stability and low cost, though it may weaken temporal dependencies. Normalization scales features (often to 0–1) to improve model performance. Standardization (zero mean, unit variance) and Min-Max scaling (fixed range) improve model stability, convergence, and fault detection performance.

The normalization to the standardized feature distribution is a normalization step that alters the feature distribution to enhance the model learning performance, as shown in Eq. (1).

$$\widehat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{1}$$

where x_{ij} is the original value of feature j for sample i, μ_j is the mean of feature j, and σ_j is its standard deviation.

Binary classification is used to distinguish normal and faulty operating states for simplified and reliable decision-making. Multi-class fault classification can be explored to improve detailed fault type interpretation. After preprocessing and normalization, data is structured into sliding time-window sequences to capture temporal patterns for model input. Each feature vector contains electrical measurements that describe the system state at that time instance. Short windows improve detection speed, while longer windows enhance feature learning; thus, an optimal window is chosen to balance responsiveness and stability. The individual feature vector is defined as a combination of electrical parameters, as expressed in Eq. (2).

$$x_t = [V_t, I_t, L_t] \tag{2}$$

where V_t represents the bus voltage measurements at time $t_1 I_t$ denotes the line current values, and L_t indicates the load parameters of the electrical system.

The temporal sequences are organized into a 3D tensor for training the TCN-based fault prediction model (Eq. 3).

$$T \in \mathbb{R}^{N \times k \times F} \tag{3}$$

The tensor $T \in \mathbb{R}^{N \times k \times F}$ represents the input dataset where N is the number of samples, k is the time window length, and F is the number of features such as voltage, current, and load.

Figure 2 shows a TCN with dilated convolutions and residual connections for fault prediction, followed by pooling and dense layers for output.

$$F(s) = \sum_{i=0}^{k-1} w(i) x_{s-d \cdot i} \tag{4}$$

$F(s)$: output feature; $w(i)$: filter weights; x_{s-di}: input; k: kernel size; d: dilation factor in (Eq. 4).

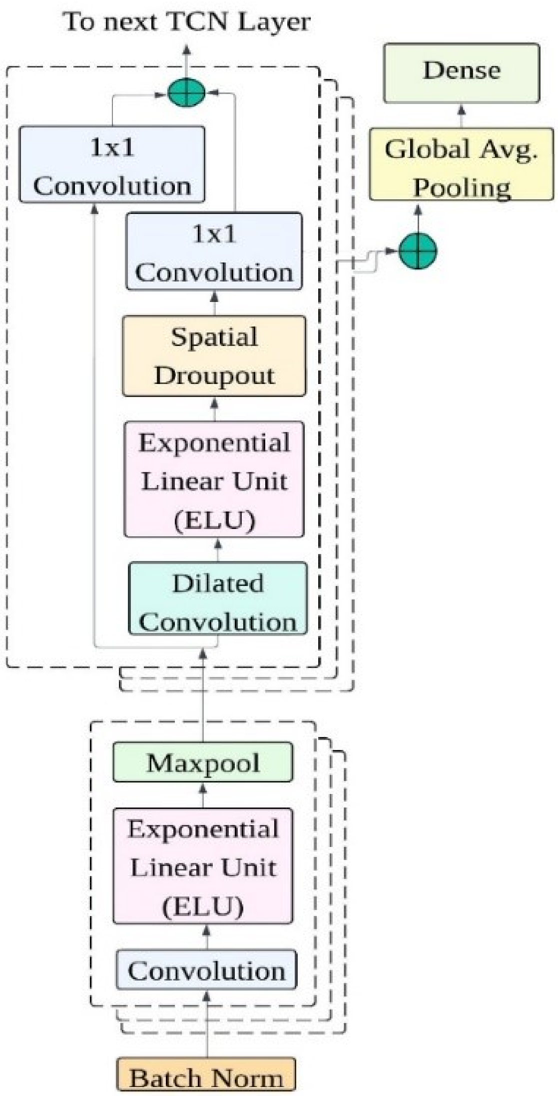

Fig. 2. Architecture of the TCN for Fault Prediction

Dilated convolutions expand the receptive field, letting TCN capture short- and long-term dependencies. Compared to RNNs, LSTMs, and Transformers, TCN enables parallel processing, stable gradients, and efficient long-sequence modeling. Residual connections in TCN improve gradient flow and training stability (Eq. 5).

$$y = \mathcal{F}(x) + x \tag{5}$$

Here, x is the input, F(x) is the transformation (dilated convolution, activation, dropout), and y is the residual block output. Residual connections improve gradient flow and stabilize deep TCN training. A reward mechanism penalizes voltage deviations, guiding the DRL agent toward stable fault handling and voltage control. In order to train the model to classify binary faults (normal or unstable) Binary Cross-Entropy (BCE) loss function will be used as defined in Eq. (6).

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\widehat{y}_i\right) + (1-y_i)\log\left(1-\widehat{y}_i\right)\right] \tag{6}$$

N: number of samples; y_i: true label; $\widehat{y}_i$: predicted fault probability from TCN.

TCN outperforms LSTM and Transformers with parallel processing, lower cost, and stable training for fault prediction, while normalization improves gradient stability and convergence. The trained TCN outputs fault probability and binary classification (0/1), which are used by the DRL module for self-healing actions in the system. TCN has O(L·k·C^2) time complexity and depth-dependent space complexity; its parallel structure reduces inference delay compared to LSTM, enabling faster training, larger receptive fields via dilated convolutions, and more stable fault prediction. TCN uses dilation for fault prediction, while DRL learns self-healing actions via reward feedback (Figure 3).

The state representation of the electrical system at the time step t is defined in Eq. (7).

$$S_t = [V_t, I_t, L_t, P_t, H_t] \tag{7}$$

Here, S_t is the system state at time t, including voltage V_t, current I_t, load L_t, TCN-based fault probability P_t, and historical window H_t.

Sensor uncertainty is handled via imputation and normalization for reliable DRL states, while adding stability indicators improves fault diagnosis. The DRL agent selects actions from the state to maintain stability (Eq. 8).

$$a_t = \pi\left(S_t\right) \tag{8}$$

a_t is the action, π the policy; actions include balancing, isolation, voltage control, backup, or no change.

Reward (Eq. 9) favors stability and penalizes instability.

$$R_t = \begin{cases} +20, & \text{fast system recovery} \\ +10, & \text{stable voltage range} \\ -10, & \text{instability persists} \\ -5, & \text{delayed response} \end{cases} \tag{9}$$

The reward function uses stability indicators like voltage deviation, faults, and recovery performance to guide learning, providing useful feedback for control decisions. The DRL agent learns the optimal control strategy by updating its action-value function using the Q-learning rule (Eq. 10).

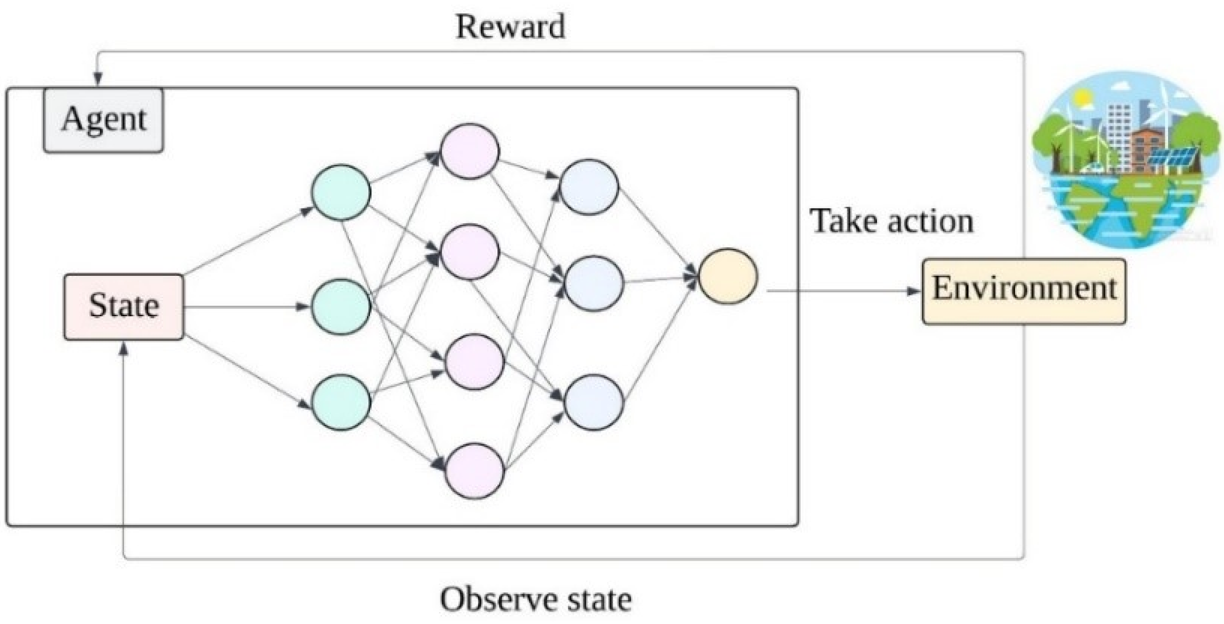

Fig. 3. DRL Framework for Fault Prediction and Self-Healing in Electrical Systems

Algorithm 1. DRL-Based Self-Healing Control

Input : System state $S_t = [V_t, I_t, L_t, P_t, H_t]$, action space A, reward function R

Output: Optimal control policy π and corrective action a_t

1. Observe current electrical system state S_t
2. Receive fault prediction probability P_t from the TCN model
3. Construct state representation $S_t = [V_t, I_t, L_t, P_t, H_t]$
4. Initialize DRL policy π
5. each time step t
6. Select action a_t from action space A using policy π
7. Execute corrective control action in the environment
8. Observe next system state S_{t+1}
9. Compute reward R_t based on system stability and recovery
10. Update policy π using reward feedback
11. Return optimal policy π and selected corrective action a_t

$$Q(S_t, a_t) \leftarrow Q(S_t, a_t) + \alpha\left[R_t + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, a_t)\right] \tag{10}$$

Where $Q(S_t, a_t)$ represents the action-value function for the state S_t and action a_t, α denotes the learning rate, γ is the discount factor controlling future rewards, and R_t represents the immediate reward obtained after executing the action. Q-learning is used for its simplicity and suitability. A target network and periodic updates stabilize training, while prioritized replay improves learning from rare faults, enhancing convergence and fault recovery. Self-healing enhances system resiliency and reliability by enabling fast, intelligent responses to predicted faults.

Algorithm 1 uses DRL to select optimal control actions based on system state and fault prediction for self-healing.

3. Results and discussion

The framework operates in a cyclical process where system states are monitored and processed by the TCN to estimate fault probability. Temporal correlations in power system signals (voltage, frequency, and power flow) are captured by TCN using dilated convolutions to model short- and long-term fault patterns for accurate prediction. The agent updates its policy to maximize cumulative reward (Eq. 11).

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R_t\right] \tag{11}$$

$J(\pi)$ is the expected reward; DRL maximizes it for fault handling and stability using the IEEE 39-Bus dataset (70/15/15 split).

The framework was implemented in Python using PyTorch and TensorFlow on a Windows 10 system with Intel i5, 16 GB RAM, and SSD. The TCN–DRL model uses 4 layers (k=3), lr 0.001, batch 64, 20 epochs, and DRL settings γ= 0.99, exploration 0.1, buffer 10,000. The TCN–DRL framework is computationally efficient and suitable for scalable electrical automation. Standardized terminology ensures clarity and consistent performance evaluation.

The performance is evaluated using Accuracy, Precision, Recall, F1-score, ROC-AUC, Recovery Time, and Voltage Deviation in (Eqs. 12-17).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

$$F1\ \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

$$\text{Voltage Deviation} = |V_{\text{observed}} - V_{\text{nominal}}| \tag{16}$$

$$\text{Recovery Time} = t_{\text{recovery}} - t_{\text{fault}} \tag{17}$$

TP, TN, FP, and FN define outcomes for performance evaluation. The model achieves high accuracy (AUC 0.985, F1 98.1%) with fast recovery and low voltage deviation.

Figure 4 shows high accuracy with minimal errors, indicating reliable fault detection and clear separation of normal and faulty states.

Figure 5 shows TCN achieves the highest AUC (0.985), outperforming other models. TCN–DRL has low execution delay and efficient decision-making. It improves voltage stability, fault recovery, and grid resilience, ensuring continuous operation under faults.

Figure 6 shows TCN–DRL achieves fastest recovery (~120 ms) among methods.

Figure 7 shows strong TCN–DRL performance with reliable fault prediction and self-healing.

Table 1 shows TCN–DRL has lowest failures, highest recovery, and least voltage deviation.

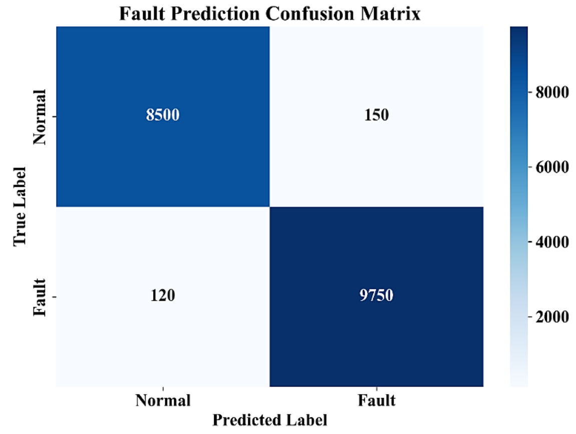

Fig. 4. Fault Prediction Confusion Matrix

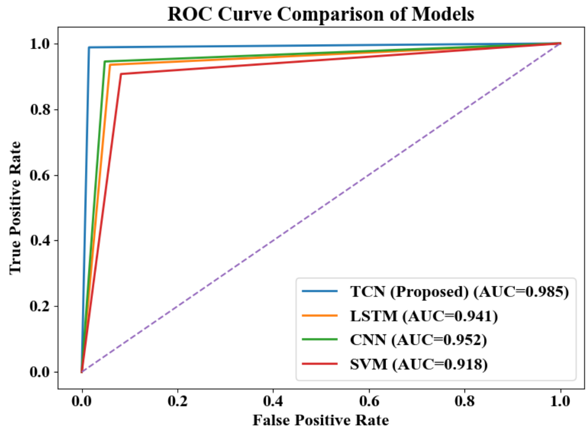

Fig. 5. ROC Curve Comparison of Fault Prediction Models

Table 2 shows that the proposed TCN–DRL model outperforms existing methods with higher accuracy and better system performance, ensuring more reliable fault detection and recovery. Deployment may face challenges in communication latency, scalability, and coordination, affecting response time and stability in large-scale systems.

The model is evaluated over 10 runs, reporting mean, standard deviation, and 95% CI, showing consistent and stable performance Table 3. The TCN–DRL framework improves fault prediction and stability but requires high-quality data and higher training resources.

4. Conclusion

The paper presents a TCN–DRL framework for accurate fault prediction and fast self-healing, improving grid reliability and resilience. The study uses the IEEE 39-Bus dataset; further statistical analysis is needed. Results are limited to benchmark data, and future work will test noisy

Table 1. Ablation Study of the Proposed TCN-DRL Framework

Configuration	Avg Failure Rate	Recovery Rate	Voltage Deviation (p.u.)
Conventional	0.125	0.45	0.15
TCN Only	0.052	0.685	0.08
Proposed TCN-DRL	0.008	0.982	0.02

Table 2. Comparative Analysis Between Proposed Work and Existing Models

Author	Models Used	Metrics & Values	Limitations
Eldeghady et al. [14]	BPNN, BPNN-PSO	Prediction Accuracy 87.8% (BPNN), 95% (BPNN-PSO); Convergence steps: 350 (BPNN) and 250 (BPNN-PSO)	Requires optimization tuning; training complexity increases with system parameters.
Arellano-Espitia et al. [15]	Stacked Autoencoder-based Deep Learning	Fault classification accuracy reported 92.03%, depending on the operating condition datasets	Model performance is highly dependent on hyperparameter tuning and large labelled datasets.
Proposed Work	TCN-DRL	Accuracy = 98.2%, Precision = 97.5%, Recall = 98.8%, F1-Score = 98.1%, ROC-AUC = 0.985, Recovery Time 120 ms, Voltage Deviation 0.02 p.u.	Performance depends on the quality of the power-system dataset

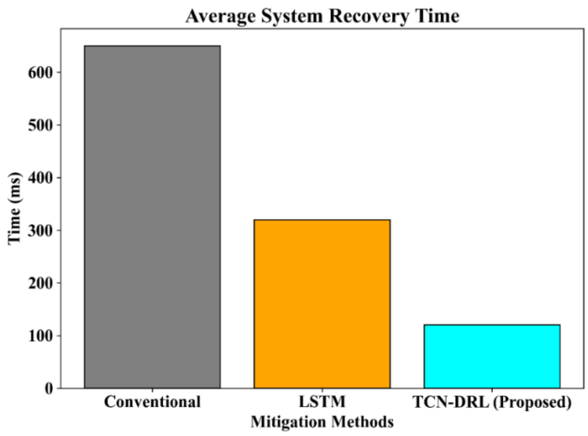

Fig. 6. Average System Recovery Time Comparison

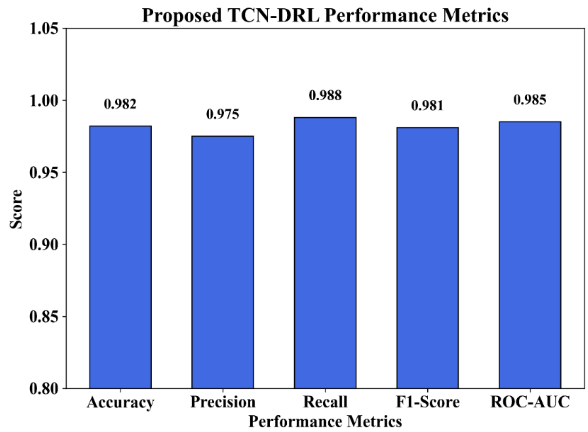

Fig. 7. Overall Performance Metrics of the Proposed TCN-DRL Framework

and practical conditions. The study uses a standard train–test split, and k-fold cross-validation will be considered in future work for improved robustness. The TCN–DRL approach enables intelligent automation in power systems, with future scope in large-scale deployment and improved scalability using edge computing and distributed learning.

5. Declaration

Data availability

Not Applicable

Table 3. Statistical Validation of TCN–DRL Framework

Metric	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Mean	Std Dev	95% CI
Accuracy (%)	98.1	98.3	98.0	98.4	98.2	98.3	98.1	98.2	98.3	98.2	98.21	0.12	±0.07
Precision (%)	97.4	97.6	97.3	97.7	97.5	97.6	97.4	97.5	97.6	97.5	97.51	0.12	±0.07
Recall (%)	98.7	98.9	98.6	99.0	98.8	98.9	98.7	98.8	98.9	98.8	98.81	0.12	±0.07
F1-Score (%)	98.0	98.2	97.9	98.3	98.1	98.2	98.0	98.1	98.2	98.1	98.11	0.12	±0.07
ROC-AUC	0.983	0.986	0.982	0.987	0.985	0.986	0.984	0.985	0.986	0.985	0.985	0.0016	±0.001
Recovery Time (ms)	118	121	122	119	120	121	119	120	121	120	120.1	1.32	±0.82
Voltage Dev (p.u.)	0.021	0.019	0.022	0.020	0.020	0.019	0.021	0.020	0.019	0.020	0.0201	0.001	±0.0006

Conflicts of interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, ornot-for-profitsectors.

Author contribution

Songhua Cao conceptualized the study, designed the methodology, developed the model, conducted experiments, analyzed the results, and wrote and revised the manuscript.

Ethical approval

This article does not contain any studies involving human participants or animals performed by the author. Therefore, ethical approval is not required.

Consent to participate

Not applicable.

Consent to publication

The author consents to the publication of this manuscript.

References

[1] S. Gil, G. Zapata-Madrigal, R. García-Sierra, and L. Cruz Salazar, (2022) *"Converging IoT protocols for the data integration of automation systems in the electrical industry"* **Journal of Electrical Systems and Information Technology** ***9***(1): 1. DOI: 10.1186/s43067-022-00043-4.

[2] A. Al Shahrani, M. Alomar, K. Alqahtani, M. Basingab, B. Sharma, and A. Rizwan, (2023) *"Machine learning-enabled smart industrial automation systems using internet of things"* **Sensors 23**(1): 324. DOI: 10.3390/s23010324.

[3] F. Almasoudi, (2023) *"Enhancing power grid resilience through real-time fault detection and remediation using advanced hybrid machine learning models"* **Sustainability** ***15***(10): 8348. DOI: 10.3390/su15108348.

[4] Y. Li, H. He, A. Khajepour, Y. Chen, W. Huo, and H. Wang, (2024) *"Deep reinforcement learning for intelligent energy management systems of hybrid-electric powertrains: Recent advances, open issues, and prospects"* **IEEE Transactions on Transportation Electrification** ***10***(4): 9877–9903. DOI: 10.1109/TTE.2024.3377809.

[5] M. Rabi, I. Abarkan, S. Sarfarazi, F. Ferreira, and A. Alkherret, (2025) *"Automated design and optimization of concrete beams reinforced with stainless steel"* **Structural Concrete**: DOI: 10.1002/suco.70332.

[6] M. Rabi, (2025) *"Serviceability-based deflection of RC beams with stainless steel reinforcement: A revised design approach and reliability assessment"* **Results in Engineering** ***27***: 105696. DOI: 10.1016/j.rineng.2025.105696.

[7] I. Abarkan, M. Rabi, S. Sarfarazi, and F. Ferreira, (2024) *"Machine learning for optimal design of circular hollow section stainless steel stub columns: A comparative analysis with Eurocode 3 predictions"* **Engineering Applications of Artificial Intelligence** ***132***: 107952. DOI: 10.1016/j.engappai.2024.107952.

[8] S. Sarfarazi, R. Shamass, M. Rabi, I. Abarkan, F. Ferreira, and K. Tsavdaridis, (2026) *"Inverse machine learning for the design of perforated beams: Parent section and material prediction"* **Engineering Applications of Artificial Intelligence** ***164***: 113275. DOI: 10.1016/j.engappai.2025.113275.

[9] M. Rabi, (2025) *"Investigation on the buckling behavior of normal steel CHS beam–columns: A revised design approach with reliability analysis"* **Buildings** ***15***(10): 1708. DOI: 10.3390/buildings15101708.

[10] H. Dui, X. Dong, L. Chen, and Y. Wang, (2023) *"IoT-enabled fault prediction and maintenance for smart charging piles"* **IEEE Internet of Things Journal** ***10***(23): 21061–21075. DOI: 10.1109/JIOT.2023.3285206.

[11] M. Cai, X. He, and D. Zhou, (2024) *"Self-healing fault-tolerant control for high-order fully actuated systems against sensor faults: A redundancy framework"* **IEEE Transactions on Cybernetics** ***54***(4): 2628–2640. DOI: 10.1109/TCYB.2023.3285903.

[12] B. Baadji, S. Belagoune, and S. Boudjellal, (2025) *"Transformer-based deep learning networks for fault detection, classification, and location prediction in transmission lines"* **Network: Computation in Neural Systems** ***36***(4): 1837–1857. DOI: 10.1080/0954898X.2024.2393746.

[13] Q.-H. Ngo, B. Nguyen, J. Zhang, K. Schoder, H. Ginn, and T. Vu, (2025) *"Deep graph neural network for fault detection and identification in distribution systems"* **Electric Power Systems Research** ***247***: 111721. DOI: 10.1016/j.epsr.2025.111721.

[14] G. Eldeghady, H. Kamal, and M. Hassan, (2023) *"Fault diagnosis for PV system using a deep learning optimized via PSO heuristic combination technique"* **Electrical Engineering** ***105***(4): 2287–2301. DOI: 10.1007/s00202-023-01806-6.

[15] F. Arellano-Espitia, M. Delgado-Prieto, V. Martinez-Viol, J. Saucedo-Dorantes, and R. Osornio-Rios, (2020) *"Deep-learning-based methodology for fault diagnosis in electromechanical systems"* **Sensors** ***20***(14): 3949. DOI: 10.3390/s20143949.