ISSN 1840-4855
e-ISSN 2233-0046

Original scientific article
http://dx.doi.org/10.70102/afts.2025.1834.528

ANALYZING THE STABILITY OF SMART GRIDS USING POLICY-BASED REINFORCEMENT LEARNING MODEL

S. Mahendran[1*], B. Gomathy[2]

[1]Department of Electronics and Communication Engineering, KGiSL Institute of Technology, Coimbatore, Tamil Nadu, India. e-mail: 1mahendran207@gmail.com, orcid: https://orcid.org/0000-0002-4518-5200*
[2]Department of Computer Science and Engineering, PSG Institute of Technology and Applied Research, Coimbatore, Tamil Nadu, India. e-mail: bgomramesh@gmail.com, orcid: https://orcid.org/0000-0002-0418-2150

Received: September 08, 2025; Revised: October 22, 2025; Accepted: November 26, 2025; Published: December 30, 2025

SUMMARY

The stability of smart grids (SG) plays a critical role in improving the stability of power supply, particularly when system failures or sensor breakdowns could occur and result in a lack of input data. This paper provides a new method of prediction of smart grid consistency by using a Gradient Policy prediction model, which is based on reinforcement learning to tackle the problem of incomplete input features. With the help of deep neural networks, the model predicts the stability of the four-node star network even in the case of incomplete information. The suggested model is assessed based on the statistical measures of R-values and Mean Squared Error (MSE), and the findings show considerable improvement in the prediction of stability. Precisely, the model attained an R-value of 0.97 and an MSE value of 125, which is higher in predictive accuracy and stability than conventional methods. Also, an ablation study was done to evaluate how the absence of data affected the performance of the prediction. The results indicate that the model is capable of detecting and offsetting the lost input variables, which is why it is a valid instrument in predicting smart grid stability. The subsequent study will aim at expanding this approach to include other, more nonlinear variables, such as price elasticity and consumer response time, to improve the level of prediction.

Key words: *smart grid, reinforcement learning, stability prediction, deep neural networks, missing data imputation, gradient policy model, power system forecasting.*

INTRODUCTION

There are several issues with the traditional power grid that is founded on the generation of power by means of fossil fuel burning, which include cybersecurity, loss of power through one-way communication, and privacy concerns [1]. As the cost of energy increases and the climate changes, there is a need to move to renewable sources of energy so as to increase the sustainability and efficiency of the grid. One response to this is the creation of smart grids that enable two-way communication between the grid devices and an optimization of the grid devices by use of sophisticated infrastructure and a digital sensor, amongst other things [2]. With a smart grid, consumers may produce and store energy, and hence, can have a more dynamic and responsive grid than the traditional grid, where consumers are charged only depending on the amount of electricity usage [3].

The smart grid stability is vital and controlled in a decentralized manner with methods like the decentralized Smart Grid Control (DSGC) method, where there are four nodes in the form of a star that predict instability with the aid of a four-node star network with the help of a different set of differential equations [4]. The stability of the grid is controlled by fluctuation of the electricity prices and change in consumer response time, and hence is a dynamic and time-sensitive process that needs precise stability calculation [5][6]. Conventional approaches, such as statistical models such as the Kalman filters, autoregressive moving averages (ARMA), and Markov chains, have difficulties in sustaining stability prediction in the presence of uncertainties and other non-linear variations in power load [7]. These models are applicable in a limited environment, but not applicable in complex environments such as operations of a smart grid, where renewable energy and variable load complicate the process of predicting the environment [8][9].

It has been discovered that deep learning models, but especially neural networks, can provide a more stable solution to smart grid stability prediction because the model allows processing of large and complex datasets without preprocessing limitations [10] [13]. These models are able to deal with nonlinear relationships between variables and provide high accuracy in the process of training and testing. But the current machine learning models do not typically consider the missing data, and it can have a severe effect on the accuracy of predictions, particularly in cases where sensors become unresponsive, or data is lost due to failed sensors or lost data [12] [14] [15]. These deficiencies are essential since any neglect of the missing data or the incorrect approach to the given information can result in erroneous forecasts and grid disruption. The suggested system presented in this paper will solve the given challenge and utilize the deep learning methods that specifically address the issue of missing data to make the predictions of the stability more reliable and accurate. The model improves the stability and performance of the smart grid by addressing any missing data of inputs instead of either ignoring it or assigning it to average values [11]. To sum up, although deep learning models have great potential in enhancing the stability of smart grids forecasting, the management of missing data is important to achieve a high level of accuracy and reliability in the real world. The missing data is to be addressed through the proposed system, which is why it will be a more effective solution to the modern smart grids. Inspired by earlier research, the proposed work initiates a modern technique to forecast smart grid consistency using a four-node star network and a DNN by applying absolute and lost input data. The significant benefits of this proposed research work are listed below:

- The smart grid consistency of a four-node star network is predicted by employing a classic FFNN containing the entire input dataset.
- The lost input data caused by network interruption, sensor failures, and other system malfunctions is identified using the sub-neural networks. Then, the recognized missing data is used to predict the stability of the smart grid system.
- Four case studies are used to analyze the accuracy and efficiency of the proposed model with at least one missing data point.

The paper is organized in the following way: Section 2 presents a review of the related works and outlines the limitations of the traditional approaches and the promise of the machine learning methods in the field of predicting the stability in smart grids. Section 3 explains the methodology, which specifies the proposed model and mathematically formulates the four-node star network to be used in predicting stability. Section 4 gives the experimental results and discussion, which compares the performance of the proposed model based on the statistical values like R-values and Mean Squared Error (MSE). Lastly, Section 5 closes the paper and outlines the most important findings, as well as the recommendations for future research, especially the incorporation of nonlinear variables, including price elasticity and response times, in order to expand further on the prediction capabilities of the model.

RELATED WORKS

The smart grid (SG) facilitates bidirectional electricity flow between providers and consumers, utilizing advanced telecommunication, data, and power technologies integrated with the existing power system [16]. It incorporates automation for efficient energy supply, storage, malfunction detection, and grid flexibility, supported by renewable energy sources (RES) and hybrid systems [17]. Key components include renewable resources, smart data systems, security systems, storage, sensors, and grid lines [18].

Smart distributed generation (SDG) enables efficient power generation near consumers, while energy storage systems (ESS) and renewable storage systems (RSS) help control frequency and voltage fluctuations, especially during emergencies [19][20]. ESS batteries enhance the production of renewable energy by increasing the grid reliability and efficiency [21][22]. Optimization of ESS and renewable production requires the synchronization of the grid.

The smart grids utilize distributed automation to exchange information and integrate the system [23]. Smart metering is an automated meter reading (AMR) and advanced metering infrastructure (AMI) smart meter that lets the consumer monitor and control the electricity usage and costs [24][25]. Phasor measurement units (PMUs) and sensor networks play an important role in measuring the health of the grid, and PMUs are used to monitor the state of the system and provide immediate reactions [26]. The grid makes use of dynamic data flows and storage control by using frequency monitoring systems [27]. Swarm intelligence, machine learning, and game theory are used to optimize grid management and predict the behavior [28][29][30]. This summary highlights the smart grid's components and technologies, emphasizing its reliance on advanced systems for real-time monitoring, optimization, and efficiency improvements.

Other previous investigations in smart grid stability prediction have largely been based on conventional models such as Kalman filters, ARMA, and Markov chains that are weak in addressing dynamic settings, as well as the incompleteness of data. However, the approaches come in handy in constant conditions and do not handle variations in unpredictable modern and renewable-based smart grids. The new studies have turned their focus on machine learning and deep learning models that are capable of describing nonlinear relationships and incomplete data better. Nevertheless, the problem of the lack of input data that is vital to the real-life applications of smart grids is still not taken into account by the vast majority of existing strategies. The gap in this paper will be addressed by introducing a reinforcement learning-based Gradient Policy prediction model, with deep neural networks, to predict the smart grid stability even with missing data. Our model is the first of its kind, unlike the traditional model, which ignores or assigns values to the missing concepts and, as such, offers a better and more dependable dynamic solution to the current smart grid systems.

METHODOLOGY

In this paper, a reinforcement learning-based method of predicting the stability of smart grids is suggested, even when some data are missing. The algorithm includes obtaining meaningful features out of the grid data and filling gaps in the input with the help of deep reinforcement learning. Fully Connected Neural Network (FCNN) is employed to produce a Gradient Policy model, which reduces prediction errors through the gradual update of weights through a gradient descent mechanism. It forecasts the best actions according to the state of the grid in order to achieve stability, it compares the stability of the grid by looking at aspects such as load and response time, and it is real-time adaptable. Also, the prediction of the missing data is done with the aid of the trained model, which guarantees further optimal functioning of the grid.

This Figure 1 characterizes the design of a system that reinforces a learning-based method in predicting smart grid stability. The system comprises several elements: smart grid environment, feature extraction and missing data processing, FCNN Gradient Policy model, state-action prediction, system stability prediction, and missing data prediction. The architecture will be optimized to achieve functional performance of the smart grid by resolving the problems of missing data and the correct prediction of the system stability by applying the methods of reinforcement learning. The best policy obtained through the model is the one that will make the smart grid very efficient and stable, even when there is a change in the input data.

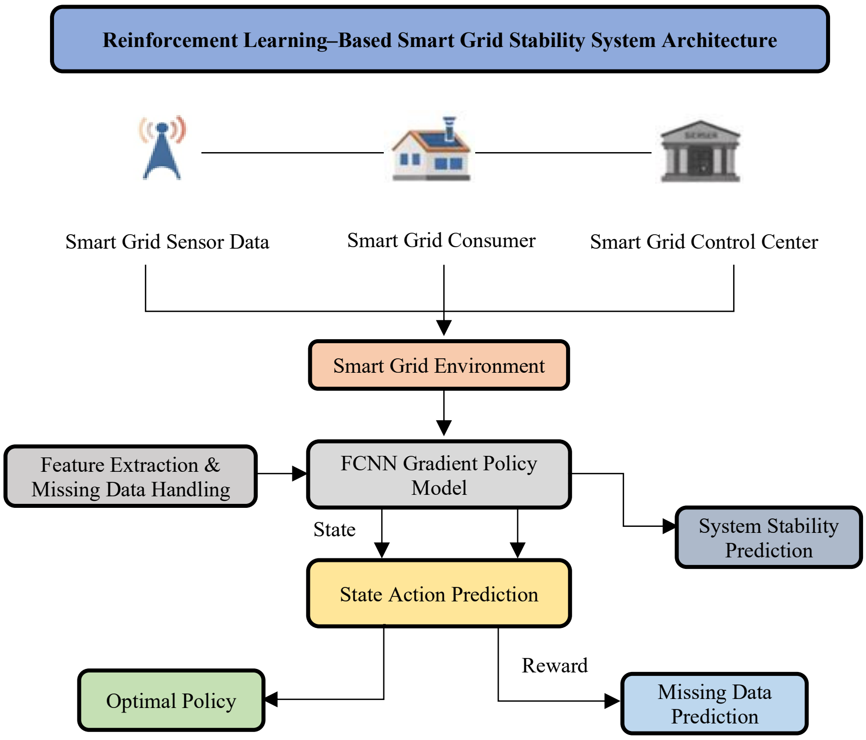

Figure 1. Smart grid stability system architecture based on reinforcement learning

The mathematical representation of the star framework is described in section 3.1, which is constructed based on the equations of motion, and the grid frequency is bonded to the electricity cost.

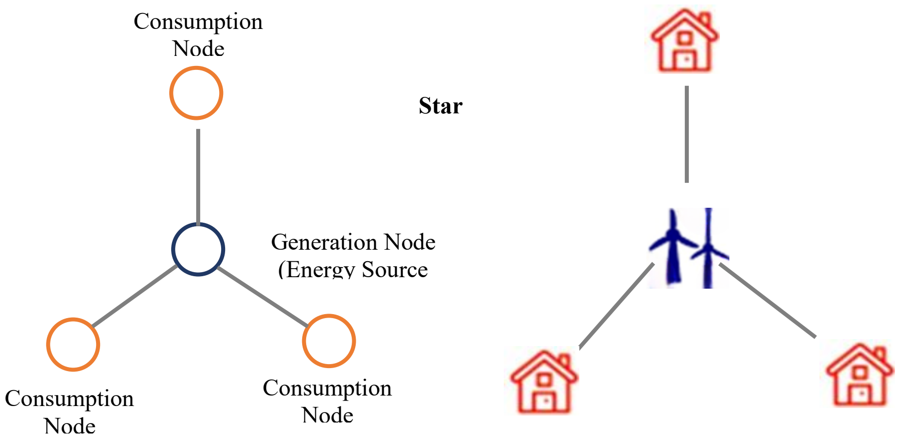

Figure 2. Smart grid

Mathematical modelling

Figure 2 presents the mathematical structure of the decentralized smart grid control acquired from the star network framework. The network structure contains three end users (i.e., user nodes) and one electricity generator in the middle (i.e., production node). There are two portions in the mathematical representation constructed with the assumption that there are no external disturbances and no uncertainties. The first part illustrates the load dynamics and the production unit developed using equations of motion. The second portion is related to the fusion of grid frequency with the power price.

The energy conservation law is applied in the first phase of the mathematical modelling. According to this law, the power balance equation is represented as follows:

$$P^s = P^a + P^d + p^t \qquad (1)$$

In the above equation (1), P^s indicates the power produced by the resource. In Eq. (1), dissipated power generated from the turbine is represented as P^d, which is directly proportional to the square of the angular velocity specified as:

$$P^d = K_j \left(\delta_j(t)\right)^2 \qquad (2)$$

In Equation (2), where, the node index (load index or generator index) is given as j, the friction coefficient of the j^{th} node is shown as K_j, and the rotor angle of the j^{th} node is $\delta_j(t)$, and it can be described as:

$$\delta_j(t) = \omega t + \theta_j(t) \qquad (3)$$

In Equation (3), where the frequency of the grid is ω, and the relative rotor angle is represented as θ_j. Likewise, in Eq. (1), P^a represents the gathered kinetic energy, and P^t is the transmitted power and is defined as:

$$P^a = \frac{1}{2} M_j \frac{d}{dt} \left(\delta_j(t)\right)^2 \qquad (4)$$

Here, the j^{th} node's moment of inertia is given as M_j and P_{jm}^{max}, which indicates the line's maximum capacity between the nodes j and m. Applying Eq. (2), (4), and (5) in (1), P_j^s is derived as:

$$P_j^s = \frac{1}{2} M_j \frac{d}{d_t} \left(\delta_j(t)\right)^2 + K_j \left(\delta_j(t)\right)^2 - \sum_{m-1}^{4} P_{jm}^{max} \sin(\delta_m - \delta_j) \quad (5)$$

Then, employing $\delta_j(t)$ derived using Eq. (3) in (6), $\frac{d^2}{dt^2}\theta_j(t)$ is derived as follows:

$$\frac{d^2}{dt^2}\theta_j(t) = P_j - \alpha_j \frac{d}{dt}\theta_j(t) + \sum_{m-1}^{4} K_{jm} \sin\left(\theta_m - \theta_j\right) \qquad (6)$$

Where produced or consumed energy is P_j, the damping constant is α_j, and the coupling strength between the nodes j and m is given as K_{jm}. These coefficients are evaluated as follows: in Equation (7)

$$P_j = \frac{P_j^s - K_j\omega^2}{M_j}, \alpha_j = \frac{2K_j}{M_j}, K_{jm} = \frac{p_{jm}^{max}}{M_j\omega} \qquad (7)$$

The last phase in the proposed model comprises the fusion of grid frequency ω to the power price, encouraging end users to modify their consumption or generation. Thus, the power price p_j for the j^{th} node is calculated.

$$P_j = p_\omega - c_1 \int_{t-T_j}^{t} \frac{d}{dt}\theta_j\left(t - \tau_j\right) dt \qquad (8)$$

In equation (8), where the electricity cost is given as p_ω at $d\theta_j/dt = 0$, the proportionality coefficient is represented as c_1, T_j is the average time, and τ_j represents reaction times. The electricity used or generated $\hat{P}_j(p_j)$ at price p_j is defined as:

$$\hat{P}_j(p_j) \approx P_j + c_j(p_j - p_\omega) \quad (9)$$

In Equation (9) Where, c_j is a coefficient directly proportional to the elasticity cost. According to the star network illustrated, the algebraic sum of power used or produced is assumed to be zero. Thus, the presumption is illustrated as: Equation (10)

$$\sum_{j=1}^{4} P_j = 0 \quad (10)$$

Therefore, the last dynamic equation of decentral smart grid control (DSGC) for the four-node star structural design network is developed by applying (7), (9), and (10) in Eq. (11) as:

$$\frac{d^2}{dt^2}\theta_j(t) = P_j - \alpha_j \frac{d}{dt}\theta_j(t) + \sum_{m-1}^{4} K_{jm}\sin(\theta_m - \theta_j) - \frac{\gamma_j}{T_j}\left(\theta_j\left(t-\tau_j\right) - \theta_j(t-\tau_j-T_j)\right) \quad (11)$$

Where, $\gamma_j = c_1 * c_j$.

Stability Analysis

During the initial phase of evaluating the dynamic stability of the network near the grid's steady-state function, the network's stable points are calculated by solving $\frac{d^2}{dt^2}\theta_j = 0$ and $\frac{d}{dt}\theta_j = 0$ and it can be given as:

$$\theta_j(t), \frac{d}{dt}\theta_j(t) = (\theta_j^*, \omega_j^*) \quad (12)$$

According to the given equation (12), sufficient coupling strength coefficient K_{jm} indicates the existence of fixed points in the network which broadcast electricity from the source nodes to the end users' nodes. Furthermore, the grid frequency of the j^{th} node ω_j is $\frac{d}{dt}\theta_j(t)$ and it is assumed as zero. Therefore, $\omega_j^* = 0$ is computed as zero. The fixed points in the network always depend on the value of the relative rotor angle θ_j, which is examined to predict the system's stability. In the next step, the Jacobian matrix is applied to identify the eigenvalues, which are used to forecast the consistency of the network. Thus, in Equation (13), the following calculation is used to determine the Jacobian matrix J.

$$J = \begin{pmatrix} \frac{\partial}{\partial\theta_j}\left(\frac{d}{dt}\theta_m\right) & \frac{\partial}{\partial\omega_j}\left(\frac{d}{dt}\theta_m\right) \\ \frac{\partial}{\partial\theta_j}\left(\frac{d}{dt}\omega_m\right) & \frac{\partial}{\partial\omega_j}\left(\frac{d}{dt}\omega_m\right) \end{pmatrix} \quad (13)$$

The network stability is predicted using the eigenvalues λ derived from the given Jacobian matrix. Infinite numbers of solutions are present in the determined matrix. But $Re(\lambda) \geq 0$ (a real positive component) exists only in a definite number of solutions, which is used to predict the network's stability. The stability is also indicated by the negative real part ($Re(\lambda) < 0$). Therefore, the permanence of the network is summarized as: Equation (14)

$$Stability = \begin{cases} stable & if\ Re(\lambda) < 0 \\ unstable & if\ Re(\lambda) \geq 0 \end{cases} \quad (14)$$

Prediction

This section proposes applying the online Deep Reinforcement Learning (DRL) technique to execute optimal design resource allocation at various aggregation levels. Fig 2 depicts the standard structure of our proposed method. Unlike the conventional RL model, the power consumption patterns can be extracted automatically using DRL, which is RL connected with DNNs of k hidden layers. Over a given input distribution, the proposed technique implements the DNN as a black box form from a general perspective with better generalization abilities. It is provided as follows in Equation (15):

$$Input\ (data) \rightarrow DNN_{(k)} \rightarrow Output\ (data\ evaluation) \quad (15)$$

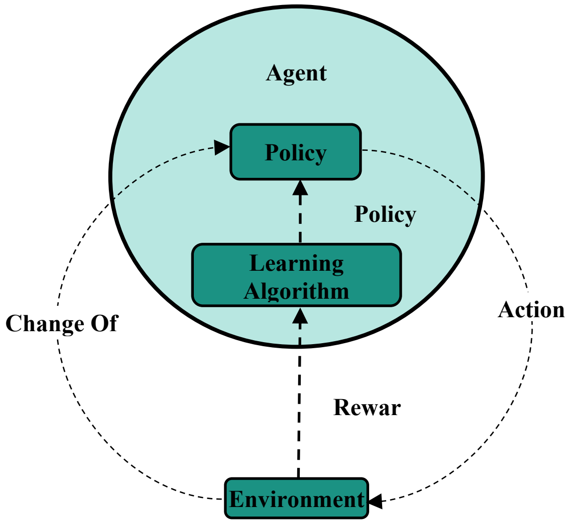

Figure 3. RL learning model

As the Figure 3 shows, the reinforcement learning model will involve the interaction between the State and the environment as the Agent takes on certain actions that cause the state to change. The agent is operated according to the Policy but modified by the Learning Algorithm considering the rewards that the environment would provide. The agent can also advance a better decision-making process with this continuous loop.

Two DRL models are introduced in the next section of this research, namely Deep Policy Gradient (DPG) and Deep Q-learning (DQN).

$$DRL = \begin{cases} Input\ (data) \rightarrow DNN_{(k)} \rightarrow Output\ (s,a) & Q - learning \\ nput\ (data) \rightarrow DNN_{(k)} \rightarrow Output\ p(s,a) & Policy \end{cases} \quad (16)$$

In Equation (16) comparison, the policy-based technique can explicitly parameterize the policy $\pi(a|s;\ \theta)$ and revise the parameters θ by calculating estimated gradient ascent on the predictable long-term reward than the value-based techniques like deep Q-learning.

Deep Q-learning (DQN)

To update the parameters, Q-learning variant is used to train the deep neural network with stochastic gradient descent. Initially, deep Q-network with parameters θ replaces the value function in the general RL model provided by the biases and weights of DNN as $Q(s,a,\theta) \approx Q\pi\ (s,a)$. The objective function by MSE in Q −values is defined using the above approximation technique.

Q-learning gradient

The data applied to the standard Q −learning are sequential, and this causes oscillation in the results of the neural network. To avoid the issue of instability distribution and correlated data, the proposed technique applies a familiar replay methodology which arbitrarily samples earlier transitions mini-batch (s_t, r_t, a_t, s_{t+1}) from the gathered dataset D. This process helps to smooth the function of training distribution over a lot of traditional data.

$$L(\theta) = E[(r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}, \theta) - Q(s_t, a_t, \theta)^2] \quad (17)$$

$$\frac{\partial(\theta)}{\partial\theta} = E[(r+\gamma] \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}, \theta) - Q(s_t, a_t, \theta))\frac{\partial Q(s_t, a_t, \theta)}{\partial\theta} \quad (18)$$

The Deep RL method is incorporated directly in Eq, (17) (18). Then, $maxat + 1\ Q(st + 1, at + 1, \theta)$ is augmented by the binary action vector at ∈ A P. As a result, m d=1 at $P - d, t$ can be controlled optimally. This research aims to encapsulate the time window in the reward function $rt(\lambda_t^+, \lambda_t^-, P_{i,d}^-)$ rather than implementing the limitation on the time window needed by a particular device d and the consumer satisfaction is considered.

Deep Policy Gradient (DPG)

It was revealed that the convergence time is reduced by policy gradient techniques in continuous games. Instead of computing $Q(s_t, a, \theta), \forall a \in A$ as in deep Q-learning, neurons in the last output layer make use of deep policy gradient with θ parameters from a structural point of view, which can calculate the prospect to take action in a particular state s_t which can be shown as $p(a|s_t, \theta), \forall a \in A$. The outcome of the above function can depict the advantage of using deep policy gradient instead of deep Q-learning. DPG can execute multiple functions concurrently in the game, utilizing the available probability for all actions. In the context of policy gradient, the estimated optimization issue was explained in Eq. (19), which is comparable to maximizing the sum predictable for the parameterized model using the policy π, and it can be described as:

$$maximize\ E_{x\sim p(x|\theta)}\ [R|\pi] \quad (19)$$

In the context of deep policy gradient, the deep neural network contains the parameterized model, making it a probability density function with its inputs, i.e. $f(x)$, resulting in Eq. (20) for the subsequent optimization issue.

$$maximize\ E_{x\sim p(x|\theta)}\ [f(x)] \quad (20)$$

The $f(x)$ is used as a score function yield by the unbiased gradient estimation as presented in Eq. (21):

$$\nabla_\theta E_x[f(x)] = \nabla_\theta \int dx\, p(x|\theta) f(x) = \int dx\, \nabla_\theta\, p(x|\theta) f(x) \quad (21)$$

Where, the output data for the first-order partial derivative is depicted as $\partial/\partial\theta = \nabla_\theta$, naturally, the gradient description of Eq. (21) is given by considering the samples of $x_i \sim p(x|\theta)$ and the approximate gradient is calculated, such that $g\hat{}_i^\theta = f(x_i)\,\nabla_\theta\, log\, p(x_i\,|\theta)$. The log probability of a specific sample xi is increased when moving in the direction of $\hat{g}_i$, which is proportional to the reward connected with $f(x_i)$. This practically shows a good sample. The samples $\tau = (s_0, a_0, r_0, \ldots, s_{T-1}, a_{T-1}, r_{T-1})$ are gathered in a trajectory and the reward is provided at the closing stages of the game in policy gradient. It is required to compute the density $p(\tau\,|\theta)$ concerning θ and differentiate the result to find the trajectory gradient as follows in Equation (22):

$$p(\tau|\theta) = p(s_0) = \prod_{t=0}^{T-1}(\pi(a_t|s_t, \theta) p(s_{t+1}|s_t, a_t)) \quad (22)$$

Here, log probability is taken for Eq. (23), and as a result,

$$\log p(\tau|\theta) = \log p(s_0) + \sum_{t=0}^{T-1}[\log \pi(a_t|s_t, \theta) + \log p(s_{t+1}|s_t, a_t)]) \quad (23)$$

Computing the derivative of Eq. (24) concerning θ directs to:

$$\frac{\partial}{\partial\theta} \log p(\tau|\theta) = \frac{\partial}{\partial\theta} \sum_{t=0}^{T-1} \log \pi(a_t|s_t, \theta) \quad (24)$$

At last, experts can formulate the gradient update gˆ θ τ for parameters θ after using a trajectory τ as: Equation (25)

$$\hat{g}_{\tau}^{\theta} \propto R_T \frac{\partial}{\partial \theta} \sum_{t=0}^{T-1} \log \pi(a_t | s_t, \theta) \tag{25}$$

Algorithm: Smart Grid Stability Prediction Using Reinforcement Learning

1. **Input:**

- Smart grid data: $D = \{X_t, Y_t\}$, where $X_t$represents the grid features at time step t, and $Y_t$represents the corresponding stability values.
- Missing data indicator: M_t, where $M_t = 1$if data is missing at time step t, and $M_t = 0$otherwise.

2. **Initialization:**

- Initialize the neural network with random weights $W_0$and biases b_0.
- Set learning rate η, number of episodes E, and discount factor γ.

3. **For each episode $e = 1$to E:**

0. **For each time step $t = 1$to T:**

- If $M_t = 1$(missing data):
- Predict missing values X_t^{pred}using the model based on previous observations.
- Extract current grid state features X_t.
- Compute the action $A_t$using the Gradient Policy model:
- $A_t = \arg\ \max_A\ Q(X_t, A_t; \theta)$, where $Q$is the state-action value function.
- Apply the predicted action $A_t$to update the grid's state.
- Observe the next state $X_{t+1}$and reward R_t, which corresponds to the stability prediction error.
- Update the model's weights using the policy gradient algorithm:
- $\theta_{new} = \theta_{old} + \eta \cdot \nabla_\theta J(\theta)$, where $J(\theta)$is the objective function.

4. **Output:**

- The learned model parameters θ^*that minimize the stability prediction error.

5. **Post-Training:**

- Test the model on new data to evaluate its performance.
- Use the trained model to predict the stability of the smart grid, even with missing data.

The algorithm is a mix of reinforced learning and deep neural networks that is capable of managing missing data to determine the stability of smart grids in real-time. The model is continuously enhanced with its prediction capacity continuously as it exists in the smart grid environment, offering a solid solution to the contemporary energy systems.

In Table 1 Deep Q-learning and deep policy gradient are trained based on the offline repository dataset. The environment is built using the fixed base loads and several possibilities of the unspecified loads are taken into account. The rewards are awarded depending on whether the calculated results are similar to the optimization target. The model is able to learn correct choices as the training advances. The offline data is further optimized, which provides a more suitable approach in the real-life application. The main benefit of such a method is that once the model is trained on offline data, it is also dynamically adapted to changes in an online setting, which enhances the accuracy and performance.

NUMERICAL RESULTS AND ANALYSIS

Table 1. Software, tools, and dataset details used in the smart grid stability prediction model

Software/Tool	Details
Programming Language	Python 3.8
Deep Learning Framework	TensorFlow 2.4
GPU	NVIDIA RTX 3090
Operating System	Ubuntu 20.04 LTS
IDE/Editor	Visual Studio Code 1.60
Dataset	Pecan Street Smart Grid Test Bed, which includes real-time data on power consumption, generation, and response times from various consumers.
Data Preprocessing	Missing data handling using deep reinforcement learning models for missing feature prediction.
Other Libraries	NumPy, pandas, matplotlib (for data analysis and visualization), scikit-learn (for evaluation metrics)

Dataset details: The training and testing data has been obtained in the Pecan Street Smart Grid Test Bed, which uploads real-time data on the power consumption, generation, and consumer response time. It has more than 10,000 data points and covers such critical aspects of electricity demand, rate of generation, changes in loads, consumer behaviors, and response time. This data is employed to simulate different conditions of the smart grid such as data loss to determine how the model performs in predicting the stability of the grid.

Parameter Initialization

The parameters applied in the experiments include: **Learning rate (η) = 0.001, ** Number of Episodes (E) =1000 and a Discount Factor (γ) =0.99. The Batch Size was 64 and the FCNN architecture (with 3 hidden layers) comprised of 128, 64, 32 neurons with ReLU activation and linear output. Adam optimizer was used, and the model was trained during 50 epochs. These environments were selected in order to have accuracy and training efficiency.

Scalability evaluation

Three case studies that comprised of different number of consumers (10, 20 and 48 buildings) of the Pecan Street SG test bed were used to test the efficiency of proposed model. The results were analyzed using deep Q-learning and deep policy gradient. The model showed scalable gains such as minimization of peaks and minimization of costs with high gains in performance where most consumers were concerned with the cost minimization. These findings are shown in Tables 2, 3 and 4.

Table 2. Peak minimization comparison

Metrics	**Methods**	**No. of grids**					
		10		**20**		**30**	
		Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**
	-	60	1	125	10	282	15
Peak value [kW]	DQN	50	5.7	107	8	239	13
Optimized peak [kW]	DPG	42	5	94	8	214	13

Table 3. Cost reduction estimation

Metrics	Methods	No. of grids					
		10		20		30	
		Mean	SD	Mean	SD	Mean	SD
	-	60	1	125	10	282	15
Peak value [kW]	DQN	50	5.7	107	8	239	13
Optimized peak [kW]	DPG	42	5	94	8	214	13
Cost [$/day]	-	57	21	119	31	230	39
Min. cost [$/day]	DQN	48	18	94	25	199	33
	DPG	45	16	83	22	168	29

Table 4. Comparison with related works

Techniques	RMSE	MAPE	NRMSE	R-Value
RNN	461.3	37.312	0.625	0.569
ARIMA	327.6	32.18	0.48	0.712
RNN-LSTM	235.56	29.12	0.21	0.895
Bi-LSTM	128.8004	22.4460	0.1074	0.9442
RL	125	19.5	0.09	0.97

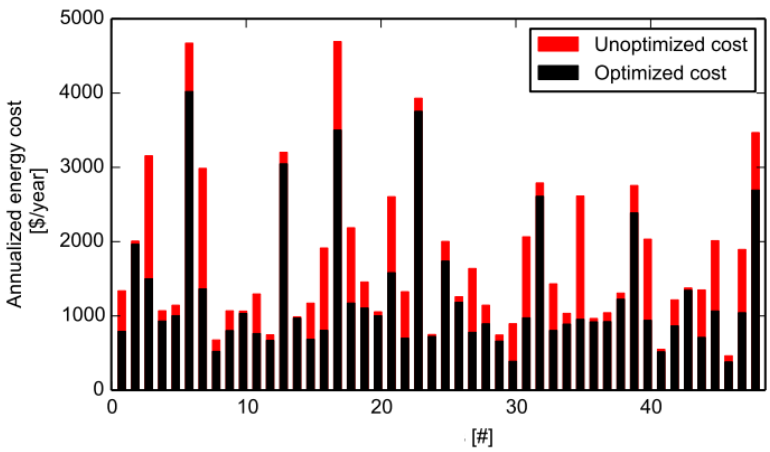

Figure 4. Cost analysis

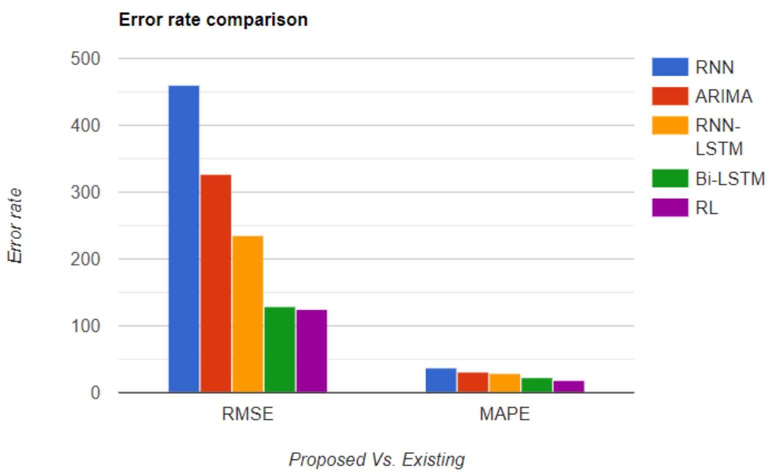

Figure 5. RMSE and MAPE comparison

Similarly, at the grid level, a deep policy gradient is predicted to be more stable and to gain better performance than deep Q-learning. The peak reduction rate of 27% and cost minimization of 27% was achieved by deep policy gradient, while deep Q-learning can reduce the peak by 10% and reduce the cost by 15%, which is comparatively much lesser performance than DPG. To visualize the performance of the deep policy gradient, the optimized and un-optimized annualized power expenses for buildings are depicted in Figure 4. From the experiment, learned that all buildings behave differently. In a few cases, deep policy gradients can reduce the annual cost by half. In other cases, it facilitates by reducing the price by some percentage level.

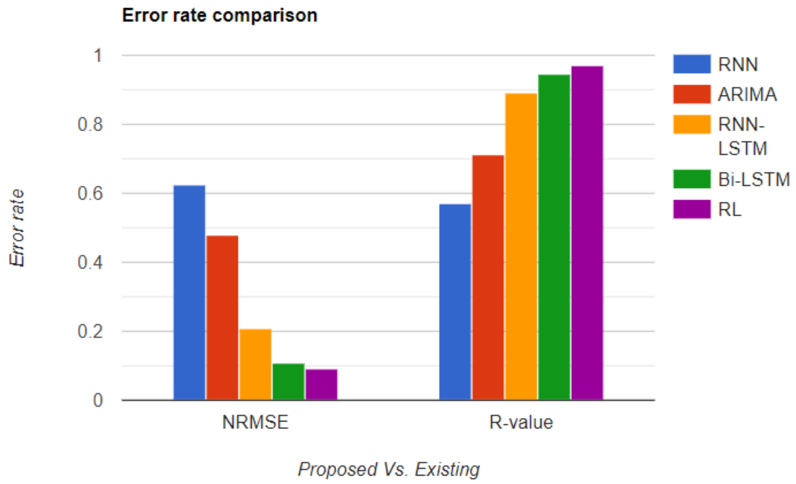

Figure 6. NRMSE and R-value comparison

Convergence capabilities

The convergence is evaluated after the execution of several iterations over episodes. For instance, figure 3 depicts the learning capability of DPG techniques for peak reduction and relative reward function for a structure. Twenty arbitrarily selected days are taken to predict the average value for each episode. At the time of observation, it is clear that the reward function increases rapidly, but after 1000 episodes, it increases gradually. As a result, after one thousand episodes, convergence is achieved using DPG between the optimized average peak value and standard peak value. The error rate comparison is shown in Figure 5 and Figure 6.

Computational time requirements

The longed state spaces are handled by both variants of DRL, facilitating improved accuracy and performance. Unlike the present optimization methods, like particle swarm optimization, deep RL knows how to identify optimal control action. Using this prediction, it can make decisions within a limited period (in milliseconds). In contrast, some optimization approach needs the expensive optimization technique to run repeatedly to make every decision.

Metrics Formulae

In order to determine performance of the suggested model to predict smart grid stability the measures applied are the following:

R-value (Correlation Coefficient):

R -value represents the magnitude and direction of the linear association between the forecasted and the real values of stability. It is calculated as:

$$R = \frac{n\sum(X_i Y_i) - \sum X_i \sum Y_i}{\sqrt{[n\sum X_i^2 - (\sum X_i)^2][n\sum Y_i^2 - (\sum Y_i)^2]}} \quad (26)$$

In Equation (26) Where: X_i are the predicted values, Y_i are the actual values, n is the number of data points. An increased R-value (near to 1) is a better predictor.

Mean Squared Error (MSE):

MSE is used to measure the squared differences between the predicted value and the actual value with the smaller the MSE the higher the accuracy. It is calculated as:

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (X_i - Y_i)^2 \quad (27)$$

In Equation (27) Where: X_i are the predicted values, Y_i are the actual values, n is the number of data points.

Precision, Recall, and F1-Score (for classification models):

Such metrics are applied in the evaluation of the classification performance of the model especially when it is binary in nature or when it is a multi-class prediction. **Precision:** The amount of accuracy of true predictions.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (28)$$

Recall: Measures the ability of the model to identify all the relevant cases.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (29)$$

F1- Score: The harmonic average of Precision and Recall, which give a balance between Precision and Recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \quad (30)$$

In Equation (28)(29) Where: TP= True Positive, FP= False Positive, FN= False Negative. These measures give a holistic analysis of how well of a model is functioning to predict the stability of the smart grid system taking into consideration the quality of the predictions, as well as the capability of the model to manage the uncertainties in the data.

In Equation (30) Where: Precision is the proportion between actual positives and the number of predicted positive. Recall represents the proportion of true positives to the all actual positives. F1-Score (0 to 1) = 1 which means that it has performed optimally. This measure is especially effective in cases of unequal data, because it uses both false positives and false negatives, which are a more comprehensive analysis of the model work.

A comparison between the two prediction methods was done in terms of the effect of missing data management on the performance of prediction. Variations of the model were also put into test, the complete model was compared with the model without the missing data prediction. The findings revealed that the stability and accuracy of the model were hugely enhanced by the treatment of missing data as indicated by the R-value and MSE of the complete model of 0.97 and 125 respectively. It means that the missing data management increases the predictive capability of the model in practice.

CONCLUSION

A new reinforcement learning-based model to predict smart grid stability, especially when missing data is present, is introduced in this paper. The suggested model which employs the deep neural networks to predict missing data and forecast the stability has shown a great improvement compared to the conventional methods. The most important statistics measures such as the variable, R-value and mean squared error (MSE) were taken to determine the model performance and the result was a R-value of 0.97 and a mean squared error of 125, which belongs to the high predictive power and stability of a model. These findings underscore the fact that the model could manage missing input data, which, in turn, guaranteed a further optimal functioning of the smart grid. The ablation experiment also revealed that missing data handling should be included in the prediction procedure. Through comparison of the full model with the models that did not predict missing data, the full model was found to be far much better to the rest, and the importance of data completeness in making accurate predictions is therefore important in this study. The importance of these results is in the fact that the model can be dynamically adjusted to the real conditions of the smart grid where the loss of data because of sensor failures or network disruption is frequent. The flexibility enables the model to be employed in real-time application to enhance resilience and stability of the current smart grids. Further research on the model will be done to expand its use of nonlinear input variables, including price elasticity and consumer response times, so as to further develop its predictive features and application to more complex grid systems.

REFERENCES

[1] Salem FM, Khairy R, Ali IA. An elliptic curve-based lightweight mutual authentication scheme for secure communication in smart grids. International Journal of Information Technology. 2025 Sep;17(7):4389-99. https://doi.org/10.1007/s41870-024-01813-1

[2] Liu T, Song Y, Zhu L, Hill DJ. Stability and control of power grids. Annual Review of Control, Robotics, and Autonomous Systems. 2022 May 3;5(1):689-716. https://doi.org/10.1146/annurev-control-042820-011148

[3] Tiwari S, Sharma A, Bhardwaj B, Singh P, Tripathi S. Predicting Smart Grid System Stability Using Machine Learning Techniques for Renewable Energy Sources. In2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS) 2023 Nov 1 (pp. 624-630). IEEE. https://doi.org/10.1109/ICTACS59847.2023.10390513

[4] Massaoudi M, Abu-Rub H, Refaat SS, Chihi I, Oueslati FS. Accurate smart-grid stability forecasting based on deep learning: Point and interval estimation method. In2021 IEEE Kansas Power and Energy Conference (KPEC) 2021 Apr 19 (pp. 1-6). IEEE. https://doi.org/10.1109/KPEC51835.2021.9446196

[5] Pamije LK, Barhani D. Lightweight hardware/software co-design framework for real-time reconfigurable accelerators in IoT devices. SCCTS Transactions on Reconfigurable Computing. 2026;3(1):19-28.

[6] Xia M, Shao H, Ma X, De Silva CW. A stacked GRU-RNN-based approach for predicting renewable energy and electricity load for smart grid operation. IEEE Transactions on Industrial Informatics. 2021 Feb 4;17(10):7050-9. https://doi.org/10.1109/TII.2021.3056867

[7] Li J, Deng D, Zhao J, Cai D, Hu W, Zhang M, Huang Q. A novel hybrid short-term load forecasting method of smart grid using MLR and LSTM neural network. IEEE Transactions on Industrial Informatics. 2020 Jun 5;17(4):2443-52. https://doi.org/10.1109/TII.2020.3000184

[8] Jeyaraj PR, Nadar ER. Computer-assisted demand-side energy management in residential smart grid employing novel pooling deep learning algorithm. International Journal of Energy Research. 2021 Apr;45(5):7961-73. https://doi.org/10.1002/er.6449

[9] Bingi K, Bhukya R, Kasi VR. Fractional Feedforward Neural Network-Based Smart Grid Stability Prediction Model. In Fractional-Order Activation Functions for Neural Networks: Case Studies on Forecasting Wind Turbines' Generated Power 2025 May 24 (pp. 207-235). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-88091-9_7

[10] Wang S, Bi S, Zhang YJ. Locational detection of the false data injection attack in a smart grid: A multilabel classification approach. IEEE Internet of Things Journal. 2020 Mar 27;7(9):8218-27. https://doi.org/10.1109/JIOT.2020.2983911

[11] Khan SU, Hayder IA, Habib MA, Ahmad M, Mohsin SM, Khan FA, Mustafa K. Enhanced machine-learning techniques for medium-term and short-term electric-load forecasting in smart grids. Energies. 2022 Dec 27;16(1):276. https://doi.org/10.3390/en16010276

[12] Mishra N. Machine Learning–Driven Approaches for Efficient Integrated Circuit Design and Optimization. National Journal of VLSI Systems and Integrated Circuit Design. 2026:10-7. https://doi.org/10.34768/amcs-2021-0026

[13] Selim M, Zhou R, Feng W, Quinsey P. Estimating energy forecasting uncertainty for reliable ai autonomous smart grid design. Energies. 2021 Jan 5;14(1):247. https://doi.org/10.3390/en14010247

[14] Alazab M, Khan S, Krishnan SS, Pham QV, Reddy MP, Gadekallu TR. A multidirectional LSTM model for predicting the stability of a smart grid. Ieee Access. 2020 Apr 28;8:85454-63. https://doi.org/10.1109/ACCESS.2020.2991067

[15] Haq EU, Pei C, Zhang R, Jianjun H, Ahmad F. Electricity-theft detection for smart grid security using smart meter data: A deep-CNN based approach. Energy Reports. 2023 Mar 1;9:634-43. https://doi.org/10.1016/j.egyr.2022.11.072

[16] Omar MB, Ibrahim R, Mantri R, Chaudhary J, Ram Selvaraj K, Bingi K. Smart grid stability prediction model using neural networks to handle missing inputs. Sensors. 2022 Jun 8;22(12):4342. https://doi.org/10.3390/s22124342

[17] Bingi K, Prusty BR. Forecasting models for chaotic fractional-order oscillators using neural networks. International Journal of Applied Mathematics and Computer Science. 2021;31(3).

[18] Surendar A. Hybrid Renewable Energy Systems for Islanded Microgrids: A Multi-Criteria Optimization Approach. National Journal of Renewable Energy Systems and Innovation. 2025 Mar 28:27-37. https://doi.org/10.17051/NJRESI/01.01.04

[19] Vashishtha G, Chauhan S, Zimroz R, Yadav N, Kumar R, Gupta MK. Current applications of machine learning in additive manufacturing: a review on challenges and future trends. Archives of Computational Methods in Engineering. 2024 Dec 26:1-34. https://doi.org/10.1007/s11831-024-10215-2

[20] Chandrasekaran K, Selvaraj J, Amaladoss CR, Veerapan L. Hybrid renewable energy based smart grid system for reactive power management and voltage profile enhancement using artificial neural network. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects. 2021 Oct 2;43(19):2419-42. https://doi.org/10.1080/15567036.2021.1902430

[21] Chung HM, Maharjan S, Zhang Y, Eliassen F. Distributed deep reinforcement learning for intelligent load scheduling in residential smart grids. IEEE Transactions on Industrial Informatics. 2020 Jul 6;17(4):2752-63. https://doi.org/10.1109/TII.2020.3007167

[22] Di Piazza A, Di Piazza MC, La Tona G, Luna M. An artificial neural network-based forecasting model of energy-related time series for electrical grid management. Mathematics and Computers in Simulation. 2021 Jun 1;184:294-305. https://doi.org/10.1016/j.matcom.2020.05.010

[23] Chen ZR, Bai YL, Ding L, Qin HY, Bi Q. Intra-hour photovoltaic power point-interval prediction using a dual-view deep neural network. Expert Systems with Applications. 2025 Jan 15;260:125368. https://doi.org/10.1016/j.eswa.2024.125368

[24] Gupta R, Jha SK, Jha P, Chaprana K, Singh SK. Enhancing the accuracy of global horizontal irradiance estimation model using convolutional neural network coupled with wavelet transform. The European Physical Journal Plus. 2024 Oct 1;139(10):924. https://doi.org/10.1140/epjp/s13360-024-05730-x

[25] Gunduz MZ, Das R. Smart grid security: An effective hybrid CNN-based approach for detecting energy theft using consumption patterns. Sensors. 2024 Feb 9;24(4):1148. https://doi.org/10.3390/s24041148

[26] Ali S, Bogarra S, Riaz MN, Phyo PP, Flynn D, Taha A. From time-series to hybrid models: advancements in short-term load forecasting embracing smart grid paradigm. Applied Sciences. 2024 May 23;14(11):4442.

[27] Abbade H, El Fadil H, Lassioui A, Intidam A, Hamed A, El Asri Y, Fhail A, Hasni A. Deep Learning-Based Performance Modeling of Hydrogen Fuel Cells Using Artificial Neural Networks: A Comparative Study of Optimizers. Processes. 2025 May 9;13(5):1453. https://doi.org/10.3390/pr13051453

[28] Shaik NB, Pedapati SR, Othman AR, Bingi K, Dzubir FA. An intelligent model to predict the life condition of crude oil pipelines using artificial neural networks. Neural Computing and Applications. 2021 Nov;33(21):14771-92. https://doi.org/10.1007/s00521-021-06116-1

[29] Tamrakar G, Dusi P. AI-Driven Intelligent Control Frameworks for Enhancing Reliability, Resilience, and Efficiency in Smart Grids. National Journal of Intelligent Power Systems and Technology. 2025 Oct 20:10-6. https://doi.org/10.17051/NJIPST/01.04.02

[30] Ramadevi B, Bingi K. Chaotic time series forecasting approaches using machine learning techniques: A review. Symmetry. 2022 May 7;14(5):955. https://doi.org/10.3390/sym14050955