Research on Algorithm Driven Intelligent Management and Control Technology for Future Power Grid

Jun Li[1,2], Qi Fu[2] and Pei Ruan[3,*]

[1]Department of electronic engineering ,North China Electric Power University ,Baoding, Hebei, 071000, China
[2]Zhejiang Huayun Information Technology Co., LTD，Hangzhou, Zhejiang，310000, China
[3]Zhejiang University school of continuing education，Hangzhou, Zhejiang, 310000, China

Abstract

An ever-more crucial architecture for both present and future electrical systems is a Power Grid (PG) that spans multiple areas comprising interlinked transmission lines, which may effectively reallocate energy resources on an extensive level. Preserving system equilibrium and increasing operating earnings are largely dependent on how the PG dispatches power using a variety of resources. The optimization techniques used to solve this dispatch issue today are not capable of making decisions or optimizing online; instead, they require doing the entire optimization computation at every dispatch instant. Herein, a novel Mutable Galaxy-based Search-tuned Flexible Deep Convolutional Neural Network (MGS-FDCNN) is presented as an online solution to targeted coordinated dispatch challenges in future PG. System optimization can be achieved using this strategy using only past operational data. First, a numerical model of the targeted coordination dispatch issue is created. Next, to solve the optimization challenges, we construct the MGS optimization approach. The effectiveness and accessibility of the suggested MGS-FDCNN approach are validated by the presentation of experimental data relying on the IEEE test bus network.

1. Introduction

Modern technology has allowed automotive manufacturers to make significant strides in improving the safety of both pedestrians and passengers [1]. Power Grid (PG) development continues amid concerns about fossil fuel depletion, climate change and global warming, despite their essential role in modern society since the 19th century [2]. Smart grid development incorporates renewable energy sources like solar and wind power, enhancing the economic, sustainable and efficient operation of cutting-edge electric energy systems [3]. Smart grids utilize knowledge, cyber-secure communication and computer information for productivity, safety, versatility, ecological sustainability and affordability, integrating renewable energy for reduced pollution [4]. Energy production and consumption are evolving to meet smart energy networks and electrification plans, necessitating technological and policy reforms for sustainable operations. Large-scale networks can be created, but smaller, islanded grids face challenges. Power networks rely on accurate predictions and storage options, while diesel generators balance production and demand but can counteract peak demand [5]. The generation and consumption of energy are changing to suit the demands of digitalization programs and intelligent electrical networks. Energy utilities require legislative and technological advancements to operate in a

*Corresponding author. Email: ruanpei0623@163.com

sustainable and renewable manner [6]. Distribution of Renewable Energy Sources (RES), forecasting and the deployment of smart grids could all undergo major changes as a result of the application of Machine Learning (ML) in energy distribution. Though ML is yet in its infancy, it is already a vital tool for the energy industry because it can change the energy landscape and increase the effectiveness of electrical power distribution [7]. Urban systems are utilizing distributed energy resources to reduce fossil fuel reliance, operating in matrix-based or island modes and providing precise voltage and frequency calculations to individual or corporate clients in rural or urban areas [8]. In electrical power networks, managing architectural levels requires the right tools and procedures. A standard language for modelling and database models has been established to facilitate coordination and interoperability [9]. Smart grid architecture model mapping is crucial for use-case leadership, technical assistance and enhanced model functionality. Potential security vulnerabilities are identified using ontologies and automated engineering approaches. A tool chain designed according to international standards is used to represent common information interchange in electrical power networks [10]. Rechargeable Electric Vehicles (REEs) can exhibit intermittent behaviour that can result in rapid voltage variations, which can compromise power quality. For this reason, managing electrical energy in power delivery systems is essential for distributed Medium-Voltage (MG) systems. Due to their greater resistive and reactance ratios, Medium-level Voltage (MV) and Low-level Voltage (LV) configurations are more prone to voltage aberrations in supply lines. Because of their sudden spikes in electrical consumption, Plug-in rechargeable Electric Vehicles (PEVs) also present a challenge to grid operations. A wider altitude management approach is the multi-agent system technique [11]. A critical first step in combating climate change and minimizing the depletion of fossil fuels is the conversion of conventional power sources into micro-grids powered by renewable energy. Micro-turbines, solar energy, wind power and other MG systems are being developed to reduce the uncertainty surrounding wind power and other RES [12].

Aim: The research's objective is to use an online solution search to distribute the most efficient possible electric power from the power generated grid to the power consuming receiver. The study proposed an MGS-FDCNN approach, in which MGS looks for the highest power in the MRPG while FDCNN analyses the power grid model based on the power level. Control Technology is revolutionizing PG management by enhancing performance, incorporating RES and reducing cyber security threats. This technology, utilizing artificial intelligence and machine learning, lays the groundwork for a more intelligent and robust energy infrastructure, ensuring sustainability and dependability in power systems.

Contributions

- To coordinate dispatch in a Multi-Regional Power Grid (MRPG), the study team created a unique structure called a MGS-FDCNN. The structure optimizes system information, enabling rapid real-time dispatch plans with outputs ranging from one to two seconds.

- Focused dispatch optimization in MRPGs is tackled by the MGS-FDCNN techniques. It improves the ability to avoid local optima by combining the classic e-greedy approach with the Boltzmann technique.

- The research compares a centralized learning approach and a hierarchical learning system, finding that the developed technique outperforms the centralized method in terms of effectiveness and efficiency.

The next part of the study is Section 2: Related work, Section 3: Managing multi-regional PG dispatch synchronization, Section 4: MGS-FDNN for dispatcher decision, Section 5: Case study analysis and Section 6: Conclusion.

2. Related Work

The study [13] presented a data-driven system that creates local controls that mimic the ideal behaviour without requiring any connection by utilizing historical data, sophisticated optimization methods and ML approaches. The study also presented the optimized local control's performance on an unbalanced, LV, three-phase distribution network. The paper [14] presented an Artificial Intelligence (AI) architecture that also highlights the Industry 4.0 and discusses its characteristics, requirements and limitations. A special emphasis was placed on the smart grid as it examined the key ML and Deep Learning (DL) algorithms utilized in Industry 4.0. Additionally, the article addresses big data, cyber security, scalability and other issues related to data analysis in the context of the new industrial era. The article [15] proposed an ML-based secure Demand-Side Management (DSM) engine for an Internet of Things (IoT)-enabled grid, utilizing classifiers to predict dishonest entities and a resilient model to limit intrusions, thereby ensuring efficient energy use and reducing smart grid power consumption. The paper [16] provided an in-depth introduction to AI, its applications in fault detection, stability control, security evaluation and stability assessment in smart grids, highlighting practical issues like large data requirements, unbalanced learning, AI interpretability, transfer learning obstacles and communication resilience. The study [17] presented a reinforcement learning-based framework for Home Energy Management (HEM) to improve Demand Response (DR) in households. The framework addressed a finite Markov decision process with discrete time steps using neural networks and Q-learning techniques. Extreme learning machines anticipate uncertainty using real data on solar photovoltaic generation and electricity prices to reduce the cost of electricity and the discontent caused by DR.

The author of [18] presented an intelligent model-free Reinforcement Learning (RL), Deep Neural Network (DNN) and Multi-Micro Grid (MMG) energy management approach. The DSO wanted to optimize energy profit while lowering the demand-side Peak-To-Average Ratio (PAR). The DSO employed a DNN without direct access to user data to preserve user privacy. The DSO then used a Monte Carlo approach from RL to choose its retail pricing strategy, optimizing choices based on forecasts. The efficacy of the

suggested data-driven deep learning approach in resolving power system issues with incomplete or ambiguous data was demonstrated.

3. Managing Multi-Regional Power Grid (MRPG) Dispatch Synchronization

This section introduces the MRPG management dispatching system and organizational structure. Additionally, several restrictions and purposes are covered.

3.1. The multi-regional power grid's architecture

The MRPGs is N_{region}, transmission line between the power generation transmitter and power-consuming receiver is $N_{\text{trans-lines}}$, two types of power sources (solar and wind units), and two kinds of loads (standard load and transportable adjustable load) make up the MRPG shown in Fig. 1. The network schedules the outputs of windmills and solar units according to their ability to generate power beyond local use. Transmission lines send excess power to fulfil local demand while receiving grids use the power supplement to meet load demand.
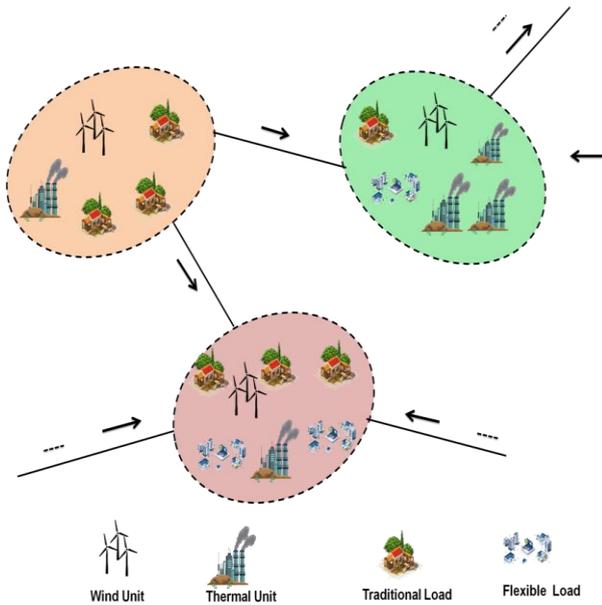
Figure 1. Architecture of a MRPG

3.2. Integrating dispatch system

The study investigates two aspects of integrated dispatch, focusing on the crucial role of electricity transmissions between regions in managing power communications within a single regional PG.

- The synchronization between the regional power systems' electricity generation and consumption. The MRPG dispatcher is responsible for arranging the transmission-line power transmission to enable to synchronization of power resources among several regional PGs.
- Power sources and mobile loads are synchronized by the regional power groups. To preserve a balance between domestic consumption and demand, the dispatchers of every regional PG is in charge of scheduling the production of different generators of electricity and the needs of adjustable loads.

The interaction between local power generation and consumer demand determines when regional grids can transfer or receive electrical power throughout various hours of operation. The dispatch and operation procedure consists of six steps and it is based on the assumption that transmitting and receiving activities in each region will remain unchanged during a dispatch period.

1. **Monitoring:** The local grid data, such as solar unit output, load demands, timeline power transmission, forecasting for wind energy and load in time frame k, is monitored by regional grid dispatchers. Additionally, they monitor system data from every regional grid, such as wind power forecasts, load forecasts and solar unit output.
2. **Analyzing:** The dispatchers collect the data and process the information to make a dispatch decision.
3. **Forecasting:** The transmission-line dispatcher plans a transmission-line electricity strategy for the cycle k based on the entire load.
4. **Scheduling:** The transmission-line distribution plan and regional data are used by regional grid managers to arrange the area's financial dispatch plans. Each economical dispatch plan includes the results of various power sources as well as the dispatch quantities of the adjustable loads for cycle k.
5. **Implementation:** The dispatch operations get carried out in cycle k.
6. **Evaluation:** After the dispatch plan is executed, the system information is obtained by the transmission line dispatcher and the regional grid dispatchers. The effectiveness of the dispatch plans can then be assessed.

The procedure involves anticipating grid information for future decision periods and scheduling dispatch strategies based on current grid information, using a multi-regional PG's information and decision system as well as implement during condensed operating periods. Fig. 2 shows the temporal link between these six processes.
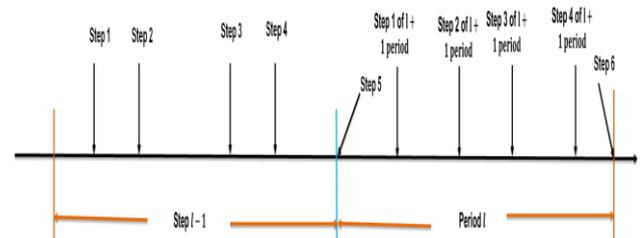
Figure 2. Dispatch Procedure

The power balancing constraint for the local PG m can be expressed as follows in the schedule stage:

$$p_m^k(l) - \sum_{j=1}^{M_m^{jk}} P_{m,j}^{jk}(l) + \sum_{j=1,j\neq m}^{M_{region}} P_{m,j}^{sk}(l) =$$
$$\sum_{j=1}^{M_m^{sg}} P_{m,j}^{sg}(l) + \sum_{j=1}^{M_m^x} P_{m,j}^x(l) \tag{1}$$

The values $p_{m,j}^{sk}(l) > 0$ indicate that power is being sent from regional grid m to j in this case, and $p_{m,j}^{sk}(l) < 0$ indicates that power is being communicated in the reverse direction.

The study focuses on the scheduling of power sources and flexible loads to balance power generation and demand based on load demand forecasts. However, the plan and actual demand can vary due to uncertainties in wind power and load needs. The transmission-line power schedule is not adjusted for short-term dispatch durations and automatic generation control units and solar units compensate for these variances. The operational stage's power balance can therefore be expressed as follows:

$$\bar{P}_m^k(l) - \sum_{j=1}^{M_m^{jk}} P_{m,j}^{jk}(l) + \sum_{j=1,j\neq m}^{M_{region}} P_{j,m}^{sk}(l) =$$
$$\sum_{j=1}^{M_m\ sg} \bar{P}_{m,j}^{sg}(l) + \sum_{j=1}^{M_m^x} \bar{P}_{m,j}^x(l) \tag{2}$$

$$\bar{P}_{m,j}^x(l) = \min\{P_{m,j}^x(l) + P_{m,j}^{x,max}(l)\} \tag{3}$$

The wind unit j 's real maximum output in the area m, as well as its planned and actual production plans are denoted independently by $\bar{P}_{m,j}^x(l)$, $P_{m,j}^x(l)$, and $\bar{P}_{m,j}^{x,max}(l)\}$. The MRPG could continually preserve an equilibrium level of functioning by repeating and consistently carrying out every plan and operational stage for each period.

3.3. Objective and Restrictions

Restrictions: The restrictions taken into account during this research can be stated as follows:

$$P_{m,j}^{sg,down} \leq P_j^{m,sg}(l) \leq P_{m,j}^{sg,vo} \tag{4}$$

$$P_{m,j}^{sg,qc} \leq \left(P_{m,j}^{sg}(l) - P_{m,j}^{sg}(l-1)\right) V_{m,j}^{sg}(l-1)V_{m,j}^{sg}(l) \leq P_{m,j}^{sg,qv} \tag{5}$$

$$P_{j,down}^{sk} \leq P_j^{sk}(l) \leq P_{j,up}^{sk} \tag{6}$$

$$P_{j,qc}^{sk} \leq P_j^{sk}(l) - P_j^{sk}(l-1) \leq P_{j,qv}^{sk} \tag{7}$$

$$0 \leq P_{m,j}^{jk}(l) \leq P_{m,j}^{jl,up} \tag{8}$$

The solar unit j's electricity production and power ramping constraints are indicated by equations (4) and (5). The transmission line j's power ramping and power spectrum restrictions are shown by equations (6) and (7), equation (8)

represents the ability to dispatch's power dispatching quantity limitation.

Objective: The research approaches the dispatch optimization goals from two angles: lowering the solar modules' financial burden and giving up on renewable energy sources. The following could be used to formulate each of these objectives' distinct functions:

$$d_{eco} = \sum_{l=1}^{L} d_{eco}(l) = \sum_{l=1}^{L} \sum_{m=1}^{M_{region}} (\sum_{j=1}^{M_m^{sg}} (d_{j=1}^{sg}(l) *$$
$$p_{m,j}^{sg}(l)) + \sum_{j=1}^{M_m^{jk}} (d_{j=1}^{jk}(l) * P_{m,j}^{jk}(l)) \tag{9}$$

$$d_{cle} = \sum_{l=1}^{L} d_{cle}(l) =$$
$$\sum_{l=1}^{L} \sum_{m=1}^{M_{region}} (\sum_{j=1}^{M_m^x} (e_{cst}^x (P_{m,j}^x(l), \bar{P}_{m,j}^x(l))) \tag{10}$$

$$e_{cst}^x \left(P_{m,j}^x(l), \bar{P}_{m,j}^{x,max}(l)\right) =$$
$$\begin{cases} \bar{P}_{m,j}^{x,}(d) \ max - P_{m,j}^x(l) \ if \ \bar{P}_{m,j}^{x,max}(l) \geq P_{m,j}^x(l) \\ 0 \quad else \end{cases} \tag{11}$$

In this instance, the solar modules negotiate on the electricity price i in the regional PG n is denoted by $d_{m,j}^{sg}$. It is possible to imagine the concept of $d_{DM,j}^{sg}$ for each thermal unit as a component of the multiregional power networks' operating environment because it remains constant during each dispatch period. Disparities between predicted and actual load and wind power values are due to uncertainties in the system's load and source. The economic cost and abandonment cost of wind power also differ, making it challenging for navigators to determine optimal dispatch plans for each period due to nonlinear cost operations. Moreover, the dispatch problem's balanced multi-objective function can be expressed as follows:

$$d = \theta_{eco}d_{eco} + \theta_{cle}d_{cle} \tag{12}$$

The relative weights of the sub-objectives in this case, denoted by d_{eco} and d_{cle}, represent the significance and dispatcher's highest priority for each sub-objective. The research aims to optimize a system's weighted cost by considering dispatcher specifications, system state, economic profit preference and thermal unit bid prices for optimal operational profit.

3.4. Wind turbines and load estimates

The research utilizes Markov models to optimize wind power and load, generating data and operational samples to characterize uncertainties in these factors, despite their model-free nature, which does not directly impact the optimization strategy. A load forecast comprises standard and flexible components, with some flexible loads as a regular load that needs to be supplied during operational stages. The local electrical grid n's load during time k can therefore be determined using the following formula:

$$P_m^k(l) = P_{m,tra}^k(l) + \Delta\bar{P}_{m,tra}^k(l) \tag{13}$$

$$\bar{P}_m^k(l) = P_m^k(l) + \Delta\bar{P}_{m,}^k(l) \tag{14}$$

Here, $P_{m,tra}^k(l)$ represents The region's PG statistics mean load level n and Markov models characterize the transition processes of $\Delta\bar{P}_{m,tra}^k(l)$, and $\bar{P}_m^k(l)$ are assumed to conform to the load model's two-stage variation framework and these two transmission stages are also assumed to fulfil the previously stated Markov models.

4. Mutable Galaxy-Based Search-Tuned Flexible Deep Neural Network (MGS-FDNN) for Dispatcher Decision

The research proposed the Flexible Deep Convolution Neural Network (FDCNN) and the Mutable Galaxy-Based Search (MGS) algorithm for PG dispatch optimization. FDCNN and MGS are powerful online optimization methods for PG dispatch. FDCNN uses historical operational data to predict dispatch decisions, while MGS dynamically adjusts FDCNN's parameters using a galaxy-based search strategy. This allows for optimal dispatch decisions online without intensive offline computations. The combination enhances system equilibrium and operating income, making FDCNN and MGS a powerful tool for optimizing dispatch decisions.

4.1. Flexible Deep Convolutional Neural Network (FDCNN)

The study forecasts areas of high power usage and creates feature maps for power distributor dispatch producers using the FDCNN architecture. The input data is synthesized by the Region of Interest (ROI) pooling layer using transmission restrictions, customer demand and historical grid measurements. The maximum PG is optimized using Softmax classifier and bounding box regression shown in Fig. 3.

Inception V2: Google-Net's state-of-the-art approach, Inception V2, has one pooling layer and seven convolutional layers with varying kernel sizes. It accelerates training and increases classification accuracy by using batch normalization (BN). By normalizing the internal representation of testing data, BN lessens the internal covariate shift brought by differences in the distributions of the input data. Because this method changes the activations of each hidden layer and modifies the weights between layers, it marks a significant development in the field. One can normalize a particular layer's input.
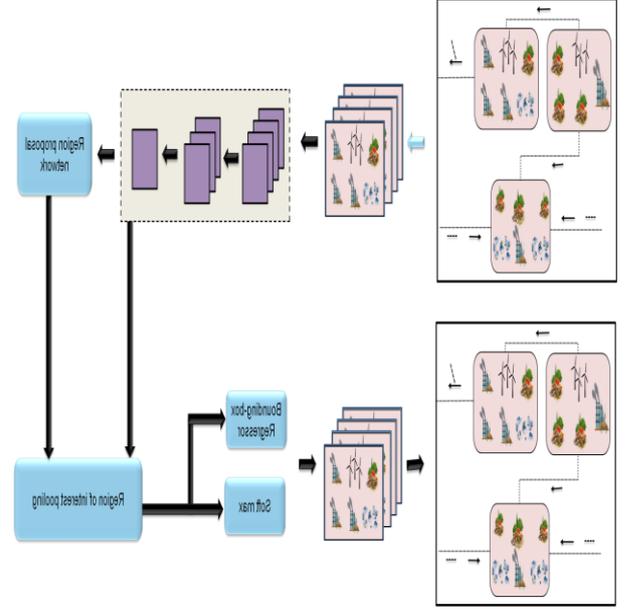
Figure 3. FDCNN Model Architecture

$$\widehat{w} = \frac{w - F[w]}{\sqrt{Var[w] + \omega}} \tag{15}$$

Where, respectively, w and $\widehat{w}$ stand for the input data as well as the layer's normalized quantity. The expectation and variance of the input are denoted by $E[w]$ and $Var[w]$, respectively. In addition, ω stands for the inner autocorrelation offset. It can be removed by using BN and after normalization; each layer can have the same input distribution. Parameters η and β are engaged to lessen the influence on every network layer following normalization. The following is an expression for an equation:

$$z_j = \gamma\widehat{w}_j + \beta \tag{16}$$

$$\frac{\partial k}{\gamma\widehat{w}_j} = \frac{\partial k}{\gamma z_j} \cdot \gamma \tag{17}$$

$$\frac{\partial k}{\partial\delta^2\theta} = \sum_{j=1}^{n}\frac{\partial k}{\gamma\widehat{w}_j} \cdot (w_j - \mu_\theta) \cdot \frac{-(\delta_\theta^2 + \omega)^{-3/2}}{2} \tag{18}$$

$$\frac{\partial k}{\partial\mu_\theta} = \left(\sum_{j=1}^{n}\frac{\partial k}{\gamma\widehat{w}_j} \cdot \frac{-1}{\sqrt{\delta_\theta^2 + \omega}}\right) + \frac{\partial k}{\partial\delta^2\theta} \cdot \frac{-2\sum_{j=1}^{n}(w_j - \mu_\theta)}{n} \tag{19}$$

$$\frac{\partial k}{\gamma w_j} = \frac{\partial k}{\gamma\widehat{w}_j} \cdot \frac{-1}{\sqrt{\delta_\theta^2 + \omega}}\right) + \frac{\partial k}{\partial\delta^2\theta} \cdot \frac{-2(w_j - \mu_\theta)}{n} + \frac{\partial k}{\partial\mu_\theta} \cdot \frac{1}{n} \tag{20}$$

$$\frac{\partial k}{\partial\gamma} = \sum_{j=1}^{n}\frac{\partial k}{\gamma z_j} \cdot \widehat{w}_j \tag{21}$$

$$\frac{\partial k}{\partial\beta} = \sum_{j=1}^{n}\frac{\partial k}{\gamma z_j} \tag{22}$$

Here k is the back propagation gradient loss. A mini-batch of size θ has a size of n and the variables w_j and z_j represents the input value w during the mini-batch and the output following the BN process. The variance and mean of the

mini-batch are denoted by $\delta^2\theta$ and μ_θ, respectively. The following can be used to express the BN network z's final output:

$$z = \frac{\gamma w}{\sqrt{U\,b[w]+\omega}} + \beta - \frac{\gamma F[w]}{\sqrt{U\,[bws]+\omega}} \qquad (23)$$

One way that Inception uses BN to normalize the output in every layer is by reducing the number of internal covariate shifts. Conversely, it reduces the amount of characteristics and speeds up the processing rate.

Region Proposal Network (RPN): The Region Proposal Network (RPN) is a method that generates rectangle region suggestions from a grid using a 3x3 window. It generates nine PG area recommendations (anchors) for a fixed-size vector, which is then fed into classification and regression layers to assess foreground and predict offsets. Box regression and softmax are used to calculate anchors' coordinates. This is typically the loss function is displayed:

$$K(o, o*, s, s*) = K_{cls}(o, o*) + \lambda K_{reg}(s, s*) \qquad (24)$$

$$\begin{cases} s_w^* = \frac{(R_w - O_w)}{O_x} \\ s_z^* = \frac{(R_z - O_z)}{O_g} \\ s_x^* = log\left(\frac{R_x}{O_x}\right) \\ s_g^* = log\left(\frac{R_g}{O_g}\right) \end{cases} \qquad (25)$$

Where the bounding box regression loss is represented by K_{reg} and the softmax loss by K_{cls}. λ stands for a loss-balancing coefficient. O and $O*$ represent, respectively, the true and expected labels. Additionally, $O = \{O_x, O_y, Ow, O_h\}$ and $R = \{R_x, R_y, Rw, R_h\}$ and $s = \{s_x *, s_y *, s_w *, s_h *\}$, W stands for both the ground-truth bounding box regression's anticipated offset vectors and $O = \{O_x, O_y, Ow, O_h\}$ and $R = \{R_x, R_y, Rw, R_h\}$ represent the proposal's region width and height, respectively, as well as the ground-truth box's centre coordinates (containing the x and y axes). This fine-tuning step allows obtaining an architecture that is more accurate in identifying maximum PG.

RoI Pooling: The ROI pooling layer in the spatial pyramid pooling network is a streamlined version that incorporates feature maps into each proposal, ensuring region features are located in the same location. This rectangle-like structure excludes noise-producing areas like cities and consumption zones. Therefore, a $(w/7 \times h/7)$ sub-window is utilized to handle region proposals with sizes of $w \times h$ (where w stand for width and h stand for height, respectively) during the max pooling process. Ultimately, despite the variations in proposal sizes, fixed-length outputs with a 7×7 size can be produced.

Classification: The regression process utilizes full-connected softmax and feature maps to categorize MRPG species into groups, with the softmax activation function calculating the probability of class labels for max power

dispatcher distribution decision. The following is an expression for the activation function:

$$z_d = \frac{exp\,w_d()}{\Sigma_{d=1}^d \exp u_d()} \qquad (26)$$

When the whole interconnected connection reaches its final outputs layer, w_d stands for the data imported from the program d the activation softmax function's effect for a class d is denoted by u_d. The symbol d stands for the aggregate amount of classes. Bounding box regression is used to determine the positional offset of each proposal, which can be regressed for more precise item classification.

4.2. Mutable Galaxy-Based Search (MGS) optimization for Optimal Dispatch Decision in the Power Grid

The MGS is a meta-heuristic optimization technique that uses a navigator to search for the ideal power state in power generation state space. It has a significant advantage over other variable neighborhood search algorithms, as it searches the maximum and minimum power state spaces from multiple regions are depicted in Fig. 4.
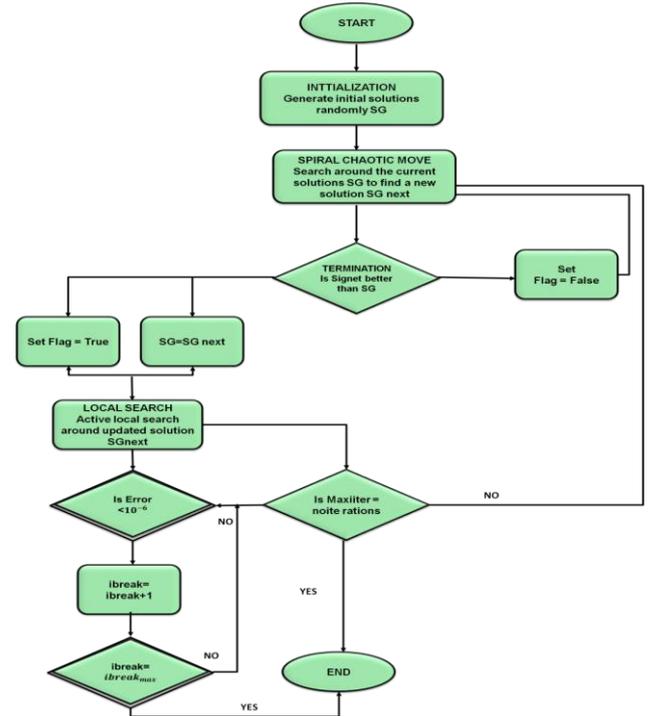
Figure 4. MGS Flow Chart

Even so, the MGS algorithm recognizes neighborhood structure and variables, but users must first identify neighborhood sets and their parameters due to unknown number, size and structure. Algorithm 1 depicts the MGS Pseudocode.

Algorithm 1: MGS Pseudocode

Galaxy-based Search Algorithm
$T \leftarrow GenerateInitia\,lSolution$
$T \leftarrow LocalSearch(TH)$
$For\ s = 1: max - iteration$
   •$Flag \leftarrow False$
•$(T, Flag) \leftarrow SpiralChaoticMove(T, Flag)$
•$If\ (Flag)\ then$
   o $T \leftarrow LocalSearch(T)$
End For
Return T

Using the Generate Initial Solution routine, the MGS first generates an initial solution with the L dimension ($T = (T1, T2, \ldots, TL)$). The LS method then receives this solution. This process yields an L-dimensional solution. Subsequently, the algorithm's primary portion the Spiral Chaotic Move (SCM) and Local Search (LS) operations are the two fundamental procedures that make up the MGS. These steps are carried out until the cancellation requirement is satisfied. Algorithm 1 depicts the pseudo-code of MGS, executed repeatedly until the termination condition is met. When the Flag variable is False in the first main section, the SCM method receives it together with the current solution, SG. If the Flag is true after the SCM operation is completed, the LS procedure is triggered.

SCM Procedure: In the MGS, the SCM process has an exploratory function (Algorithm 2). The L-dimension solution (T) and a flag variable are fed into the algorithm. The approach also includes a loop that represents Algorithm 2 and it is the maximum replayed for each MaxRep repetition. The Navigator in SCM uses the Spiral Chaotic Move to lead the solution space, ensuring an exploratory search procedure by avoiding local optima and suboptimal areas. The jth component of the subsequent solution is generated in the following manner for each loop iteration:

$$TNext_j \leftarrow T_j \mp NextChaos().q.Cos(\theta_j), j = 1,2,..K \quad (27)$$

The jth component of the current solution and the term $NextChaos()$ is represented by T_j in equation (27).

Algorithm 2: SCM Procedure
The procedure of Spiral Chaotic Move
// input: T is the best solution at the moment with K components, where k is the total amount of solution elements and Si is the ith element in solution T. // output: The initial solution that outperforms the supplied solution is the output, denoted by SNext. In this instance, an improved approach has been identified, as shown by the Flag being set to true.. // function f is fitness function. // parameters: $\Delta\theta = 0.01$, $r = 0.001$, Δq and $\Delta\theta$ are set using function $NextChaos()$ in each
$call\ , \Delta q_{max} = 1, \Delta\theta_{max} = 0.1$
$\Delta q \leftarrow , \Delta q_{max} * NextChaos()$
$\Delta\theta \leftarrow , \Delta\theta_{max} * NextChaos()$
$For\ j = 1\ to\ K$
$\theta_j \leftarrow (2. NextChaos() - 1) * \pi$
$Rep \leftarrow 0$

$While\ Rep < MaxRep$
  •$For\ j = 1\ to\ K$
$TNext_j \leftarrow T_j + NextChaos().q.Cos(\theta_j)$
•$If\ (\ e(TNext) \geq e(T)\ )\ then$
$Flag \leftarrow True$
Go to the End procedure
•$For\ j = 1\ to\ K$
$TNext_j \leftarrow T_j - NextChaos().q.Cos(\theta_j)$
  • $If\ (\ e(TNext) \geq e(T)\ )\ then$
$Flag \leftarrow True$
Go to the End procedure
   •$q \leftarrow q + \Delta q$
•$For\ j = 1\ to\ K$
$\theta_J \leftarrow \theta_J + \Delta\theta$
$If\ (\theta_J > \pi)\ then\ \theta_J \leftarrow -\pi$
•$Rep \leftarrow Rep + 1$
End while
End Procedure

The $TNext_j$ is one of the arms of the spiral power generation grid with the core T because $q.Cos(\theta_j)$ mimics an arm of the spiral power generation grid. A chaotic sequence is produced by the logistics one-dimensional map. The $NextChaos()$ function returns a chaos number between 0 and 1. Algorithm 3 depicts the $NextChaos()$ function's pseudo code. In each cycle iteration, r, and θi in equation (27) change in the following way:

$$q \leftarrow q + \Delta q \quad (28)$$

$$\theta_j \leftarrow \theta_j + \Delta\theta, \ j = 1,2,..,K \quad (29)$$

Algorithm 3: $NextChaos()$ Procedure
Procedure NextChaos()
   $Z_{s+1} \leftarrow \lambda.Z_s.(1 - Z_s)$
Return Z_{s+1}
End Procedure // λ is control parameter (i.e $\lambda = 4$), $Z_0 \in [0,1] - - \{0, 0.25, 0.5, 0.75, 1\}, m$

The $NextChaos()$ method sets variables Δq and $\Delta\theta$ in the initial call of the SCM process as follows:

$$\Delta q \leftarrow \Delta q_{max} \times NextChaos() \quad (30)$$

$$\Delta\theta \leftarrow \Delta\theta_{max} \times NextChaos() \quad (31)$$

The process comes to an end and the loop closes if the next solution ($TNext$) is superior to the present solution (T); If not, up to the MaxRep iteration, the next area is searched. The simulated code for the SCM process is shown in Algorithm 4.

LS Procedure: In the MGS, the LS process plays an exploitative function. The process searches from the closest neighborhood to the furthest neighborhood to improve the answer discovered by the SCM procedure. The LS Navigator employs a power level adjustment mechanism to investigate nearby neighborhoods of the current solution to identify the best power dispatcher to enhance the performance of the

MRPG dispatcher. To guarantee a targeted search process and take advantage of the local structure of the solution space, it employs a chaotic sequence produced by the $NextChaos()$ function to ascertain the direction of the regional grid and power generation level in the MRPG.

If the input area is L-dimension, the LS process in algorithm 4 gets an input vector as $T = (T_1, T_2, \dots, T_L)$; initially, as the optimal electrical solution; it looks at this input vector, and then, based on equation (30), it attempts to examine the neighbouring elements with radius r1 regarding the T1 portion (i.e. $T = (T_{q_1}, T_2, \dots, T_k)$).

$$T_{q_1} = T_1 \mp \alpha \Delta T \tag{32}$$

The search step size is determined by $\alpha \Delta T$ in equation (31), where $T1$ is the initial component of the current solution.

Algorithm 4: LS Procedure

Procedure LocalSearch (T)

// input: T is the present solution with K elements such that $T_j i$ denotes the element jth of solutionT

// output: TNext is the output of the LS //e(.) is the fitness function

// parameters: ΔT is a fixed parameter a is a dynamic parameter, $\Delta\alpha$ is 0.5, ΔT_{max} is 2, and

$L - Max$ *indicates the number of times the LS can recursively look for better options surrounding an element.*
$$\Delta T \leftarrow \Delta T_{max} \times NextChaos()$$
Repeat for j=1toK
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□$\alpha \leftarrow 1, l \leftarrow 0$
While l < kMax
$$TK_j \leftarrow T_j - \alpha \times \Delta T$$
$$TV_j \leftarrow T_j - \alpha \times \Delta T$$
If (e(TK) < e(T)(and e(TV) < e(T)) then
Go to End Repeat
If (e(TV) > e(T)) then
$$T_j \leftarrow TV_j$$
$$TK_j \leftarrow TV_j$$
$$\alpha \leftarrow \alpha + 0.01 \times NextChaos()$$
$$l \leftarrow l + 1$$
Else If $\left(e(TK) > e(T)\right)$then
$$T_j \leftarrow TK_j$$
$$TV_j \leftarrow TK_j$$
$$\alpha \leftarrow \alpha + 0.01 \times NextChaos()$$
$$l \leftarrow l + 1$$
Else
$\alpha \leftarrow \alpha + 0.05 \times NextCh()$ ⌉□
$l \leftarrow l + 1$
End while
$$TK_j \leftarrow T_j, \ TV_j \leftarrow T_j,$$
End repeat
$$TNext \leftarrow T$$
End Procedure

α rises in the following manners as the method proceeds:

$$\alpha \leftarrow \alpha + 0.01 \times NextChaos() \tag{33}$$

During the LS process, ΔT remains constant, but the $NextChaos()$ function modifies it as the MGS progresses in the following ways:

$$\Delta T \leftarrow \Delta T_{max} NextChaos() \tag{34}$$

If a improved outcomes is discovered in the r1 neighbouring area, it is chosen as the best option at the moment (i.e., $T_1 \leftarrow T_{q_1}$), and the neighborhood radius is increased (using equation 31) and the procedure for searching proceeds with the new radius (Algorithm 5).

Algorithm 5: Mutable MGS Procedure

Procedure Mutable MGS

// input: T is the present solution with K apparatus such that T_j denotes the component jth of the solutionT

// output: TNext is the output of the LS//e(.) is the fitness function

// parameters: The parameterΔT is both fixed and dynamic, $\Delta\alpha$ is 0.5, $\Delta T_{max}$is 2, and LMax indicates the number of times the LS can look around a component before finding a better solution.
$$\Delta T \leftarrow \Delta T_{max} \times NextChaos()$$
Repeat for j=1toK
$$\alpha \leftarrow 1, l \leftarrow 0$$
While l < kMax
$$TK_j \leftarrow T_j - \alpha \times \Delta T$$
$$TV_j \leftarrow T_j - \alpha \times \Delta T$$
If (e(TV) > e(T) then
$$T_j \leftarrow TV_j$$
$$TK_j \leftarrow TV_j$$
$$\alpha \leftarrow \alpha + 0.01 \times NextChaos()$$
$$l \leftarrow l + 1$$
Else If (e(TK) > e(T)) then
$$T_j \leftarrow TK_j$$
$$TV_j \leftarrow TK_j$$
$$\alpha \leftarrow \alpha + 0.01 \times NextChaos()$$
$$l \leftarrow l + 1$$
Else
$\alpha \leftarrow \alpha + 0.05 \times NextCh()$ ⌉
$l \leftarrow l + 1$
End while
$$TK_j \leftarrow T_j, \ TV_j \leftarrow T_j,$$
End repeat
$$TNext \leftarrow T$$
End Procedure

If a better solution is found, neighborhoods surrounding the second component are identified using the LS technique in algorithm 5. However, it's not always possible to find a better solution in lower components, preventing surveys of populations around higher components.

5. Performance Analysis

This work recreated an updated IEEE 300 bus system to evaluate the effectiveness of a proposed optimization strategy. The system was divided into three sections: two with DLC flexible loads and one with wind power units. Thermal and erratic loads exist in all areas. Markov process models were developed using data from two dispatch centres, with two transmission lines estimated to have a capacity of 1200 MW. The prices of the thermal units in the first–third areas were set at 115/185/245, 255/215/295 and 235/185/285 ¥ per MWh, respectively, and the units were divided into three groups within the regional PGs. Table 1 lists the MGS-FDCNN's architecture and activation mechanism for each of the three navigators. Each navigator's learning rate was configured to decay exponentially, with a declining parameter of 0.99 for each process in the learning procedure. A Python application built on Tensorflow was used to run the simulations.

Table 1. Parameters of the Model

Parameters	Hidden layer	Number of Neural	Activation function
FDCNN of the LS Navigator	5	120, 150, 170, 200	
FDCNN of the Transmitting Navigator	4	120, 170, 120,	softmax
FDCNN of the Receiving Navigator	4	120, 170, 120	

Fig. 5 (a), which displays the weighted costs of the navigators, illustrates the learning optimization processes of the navigators. Navigators learn more effectively and eventually converge, with weighted costs dropping quickly in the initial stages. Memorizing examples helps them quickly recognize inappropriate activities and steer clear. The curve rises as search for better ways to proceed, indicating high levels of synchronization of optimization throughout the learning process. The three LS navigators' power imbalance costs are depicted in Fig. 5 (b) and progressively decline and eventually converge to almost nil. Uncertainty in wind power and load causes more significant oscillations in the process for the transmitting navigator. Fig. 5 (c) shows navigators' economic outlays, which increased before levelling down. To preserve power balance and reduce imbalances, use more thermal units, sacrificing economy to lower costs while maintaining weighted costs.
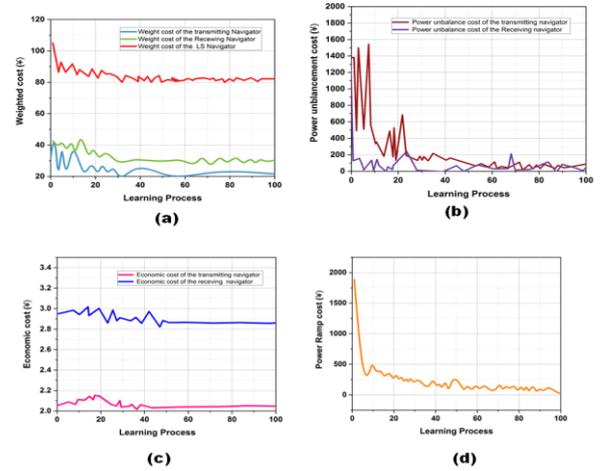
Figure 5. Learning-Optimisation Curve of the Navigators (a) Weight Cost, (b) Power Cost, (c) Economic Cost, and (d) Validation Number

The transmission line constraint validation quantity gathers from 10,000 samples during the learning process and progressively decreases to almost 0 in the end stages. Three learning modes are where comparisons are displayed. The LS navigator is used in a centralized mode, optimizing the system by combining all four navigators under a hierarchical structure. The learning rate in this mode is similar to the LS navigator's, but convergence is worse in the centralized mode shown in Fig. 6 (a-c).
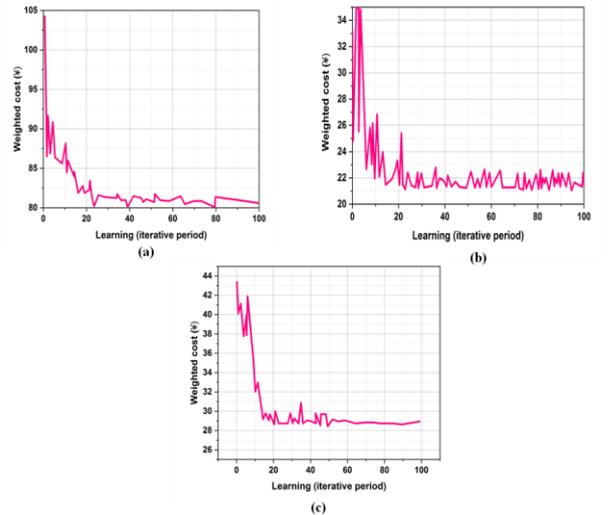
Figure 6. Every Navigator's Optimizing Curve (a) LS Navigator with MGS-FDCNN, (b) Transmitting Navigator with MGS-FDCNN and (c) Receiving Navigator with MGS-FDCNN

The initial mode performs worst. The centralized mode shows more fluctuation since it has wider system state, action and policy spaces. Because of this, finding an effective policy with the same amount of samples becomes difficult, underscoring the hierarchical mode's effectiveness shown in Fig. 7 (a-c).
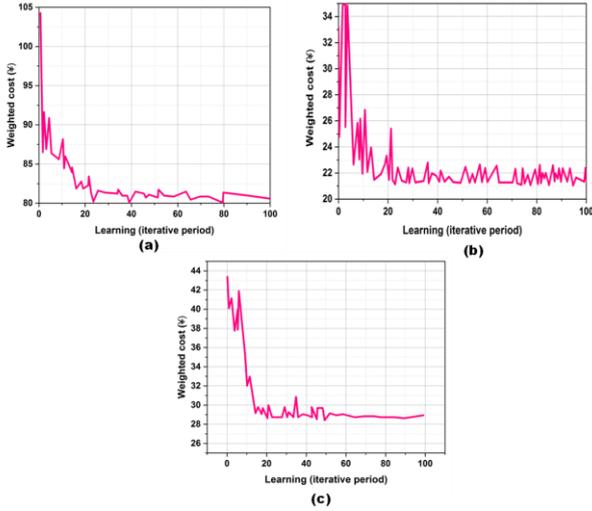
Figure 7. Optimisation Curves of each Navigator (a) LS Navigator with MGS-FDCNN, (b) Transmitting Navigator with MGS-FDCNN and (c) Receiving Navigator with MGS-FDCNN
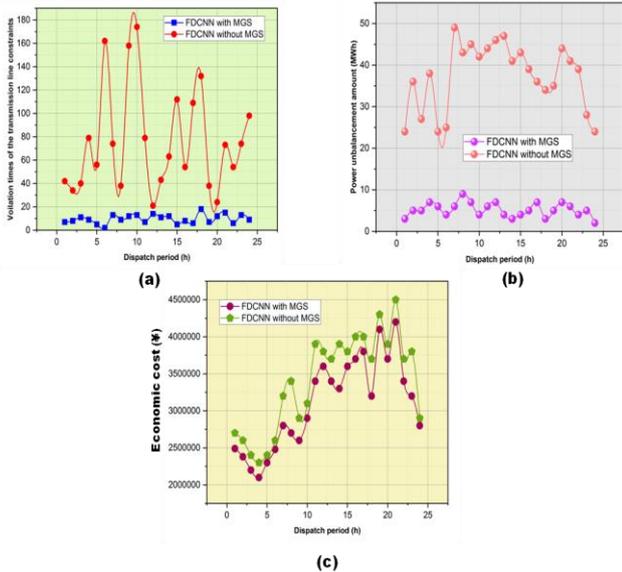
Figure 8. Process Outcomes (a) Power Ramp Cost, (b) Economic Cost, and (c) Power Unbalance Cost

The operating costs of dispatch policies developed using MGS-FDCNN optimization techniques are displayed in Fig. 8 (a-c). As can be shown in Fig. 7, the exact model and the suggested approach have comparable running costs. The MGS-FDCNN generates policies without optimization, resulting in lower operational expenses and less economic cost difference compared to the first FDCNN without optimization, supporting the efficacy of the hierarchical learning approach in Figs 6 and 7.

Figs. 9 and 10 (a-b), The MRPG's daily operation data should be presented alongside final dispatching guidelines to ensure the efficiency of the hierarchy technique. The regional power networks' load information is displayed in Fig. 9; each region's actual loads and predicted loads are different.
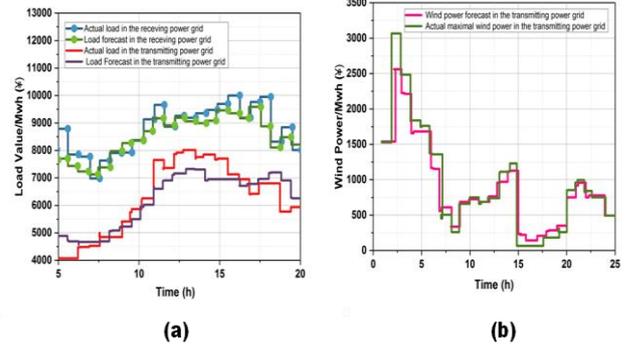
Figure 9. Statistics on load and wind energy for the regional PGs (a) Load Information, (b) Wind Power

The wind power system experiences significant variation in the second receiving region due to high load. All three navigators use higher power prices for dispatch command generation, demonstrating cost-effective dispatch command generation in Fig. 10, but the least expensive units cannot always produce the most power.
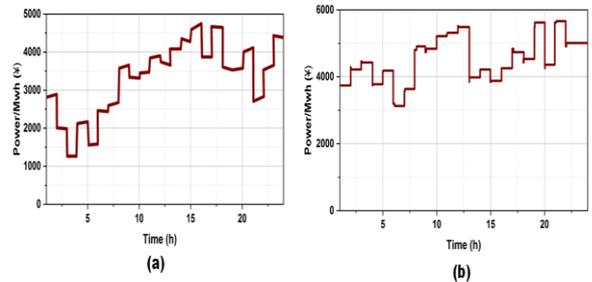
Figure 10. Data is generated by thermal generators in each of the regional electricity grids. (a)Transmitting PGs, (b) Receiving PGs

Navigators face unpredictable loads and wind power influences, leading to varying real and predicted loads. Selecting the cheapest units can reduce electricity consumption, but if the difference is large, balancing thermal units can fail due to capacity ramp and operational range restrictions. This highlights the influence of other factors on decision-making eras and the effects of different decision-making eras.

6. Conclusion

This research examines the planning and dispatching challenge of an MRPG in a situation of uncertainty. The research seeks to enhance forecasting performance and command-making performance in real-world procedures by concentrating on online optimizations and determination strategies for the system under consideration. Next, a design and introduction of an MGS-FDCNN approach is made. The results of the simulation demonstrate the design method's availability and good performance. An MGS algorithm with discrete actions is used as the FDCNN in this work. To get a more accurate optimization, the research might eventually discover a practical method to integrate both constant and

discontinuous action MGS algorithms for the LS and SCM navigators. Moreover, in an unidentified situation, the FDCNN optimized by MGS alone might not be able to produce a workable command.

Limitation and Future scope

Methods that combine data-driven and model-based algorithms in the future can be able to address this issue and strengthen the resilience of learning-based optimization techniques. In this study, the fixed weights of the sub-objectives are used to search for the best dispatch plan. Another difficult issue in the future will be determining the ideal priorities and the accompanying coefficients dynamically by the unique environment.

Acknowledgement:

References

[1] Lipu, M.H., Hannan, M.A., Karim, T.F., Hussain, A., Saad, M.H.M., Ayob, A., Miah, M.S. and Mahlia, T.I., 2021. Intelligent algorithms and control strategies for battery management system in electric vehicles: Progress, challenges, and future outlook. Journal of Cleaner Production, 292, p.126044.https://doi.org/10.1016/j.jclepro.2021.126044

[2] Rocchetta, R., Bellani, L., Compare, M., Zio, E. and Patelli, E., 2019. A reinforcement learning framework for optimal operation and maintenance of power grids. Applied energy, 241, pp.291-301. https://doi.org/10.1016/j.apenergy.2019.03.027

[3] Jiang, W., Wang, X., Huang, H., Zhang, D. and Ghadimi, N., 2022. Optimal economic scheduling of microgrids considering renewable energy sources based on energy hub model using demand response and improved water wave optimization algorithm. Journal of Energy Storage, 55, p.105311. https://doi.org/10.1016/j.est.2022.105311

[4] Chapaloglou, S., Nesiadis, A., Iliadis, P., Atsonios, K., Nikolopoulos, N., Grammelis, P., Yiakopoulos, C., Antoniadis, I. and Kakaras, E., 2019. Smart energy management algorithm for load smoothing and peak shaving based on load forecasting of an island's power system. Applied Energy, 238, pp.627-642. https://doi.org/10.1016/j.apenergy.2019.01.102

[5] Butt, O.M., Zulqarnain, M. and Butt, T.M., 2021. Recent advancement in smart grid technology: Prospects in the electrical power network. Ain Shams Engineering Journal, 12(1), pp.687-695. https://doi.org/10.1016/j.asej.2020.05.004

[6] Ahmad, T., Madonski, R., Zhang, D., Huang, C. and Mujeeb, A., 2022. Data-driven probabilistic machine learning in sustainable smart energy/smart energy systems: Key developments, challenges, and future research opportunities in the context of smart grid paradigm. Renewable and Sustainable Energy Reviews, 160, p.112128. https://doi.org/10.1016/j.rser.2022.112128

[7] Roy, K., Mandal, K.K. and Mandal, A.C., 2019. Ant-Lion Optimizer algorithm and recurrent neural network for energy management of microgrid connected system. Energy, 167, pp.402-416. https://doi.org/10.1016/j.energy.2018.10.153

[8] Suresh, V., Muralidhar, M. and Kiranmayi, R., 2020. Modeling and optimization of an off-grid hybrid renewable energy system for electrification in rural areas. Energy Reports, 6, pp.594-604. https://doi.org/10.1016/j.egyr.2020.01.013

[9] Panda, D.K. and Das, S., 2021. Smart grid architecture model for control, optimization, and data analytics of future power networks with more renewable energy. Journal of Cleaner Production, 301, p.126877. https://doi.org/10.1016/j.jclepro.2021.126877

[10] Khan, M.W., Wang, J., Ma, M., Xiong, L., Li, P. and Wu, F., 2019. Optimal energy management and control aspects of distributed microgrid using multi-navigator systems. Sustainable Cities and Society, 44, pp.855-870. https://doi.org/10.1016/j.scs.2018.11.009

[11] Sun, S., Fu, J., Wei, L. and Li, A., 2020. Multi-objective optimal dispatching for a grid-connected micro-grid considering wind power forecasting probability. IEEE Access, 8, pp.46981-46997. https://doi.org/10.1109/ACCESS.2020.2977921

[12] Karagiannopoulos, S., Aristidou, P. and Hug, G., 2019. Data-driven local control design for active distribution grids using off-line optimal power flow and machine learning techniques. IEEE Transactions on Smart Grid, 10(6), pp.6461-6471. https://doi.org/10.1109/TSG.2019.2905348

[13] Kotsiopoulos, T., Sarigiannidis, P., Ioannidis, D. and Tzovaras, D., 2021. Machine learning and deep learning in smart manufacturing: The smart grid paradigm. Computer Science Review, 40, p.100341. https://doi.org/10.1016/j.cosrev.2020.100341

[14] Babar, M., Tariq, M.U. and Jan, M.A., 2020. Secure and resilient demand side management engine using machine learning for IoT-enabled smart grid. Sustainable Cities and Society, 62, p.102370. https://doi.org/10.1016/j.scs.2020.102370

[15] Shi, Z., Yao, W., Li, Z., Zeng, L., Zhao, Y., Zhang, R., Tang, Y. and Wen, J., 2020. Artificial intelligence techniques for stability analysis and control in smart grids: Methodologies, applications, challenges, and future directions. Applied Energy, 278, p.115733. https://doi.org/10.1016/j.apenergy.2020.115733

[16] Xu, X., Jia, Y., Xu, Y., Xu, Z., Chai, S. and Lai, C.S., 2020. A multi-navigator reinforcement learning-based data-driven method for home energy management. IEEE Transactions on Smart Grid, 11(4), pp.3201-3211.https://doi.org/10.1109/TSG.2020.2971427

[17] Du, Y. and Li, F., 2019. Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning. IEEE Transactions on Smart Grid, 11(2), pp.1066-1076. https://doi.org/10.1109/TSG.2019.2930299

[18] Nallaperuma, D., Nawaratne, R., Bandaragoda, T., Adikari, A., Nguyen, S., Kempitiya, T., De Silva, D., Alahakoon, D. and Pothuhera, D., 2019. Online incremental machine learning platform for big data-driven smart traffic management. IEEE Transactions on Intelligent Transportation Systems, 20(12), pp.4679-4690. https://doi.org/10.1109/TITS.2019.2924883