International Journal of Engineering and Modern Technology (IJEMT) E-ISSN 2504-8848
P-ISSN 2695-2149 Vol 10. No. 10 2024 www.iiardjournals.org Online Version

Hybridization of Mayfly-Pelican Optimization Algorithm for Selection of CNN Optimal Hyper-Parameters

Ganiyu Adedayo Ajenikoko [1], Isaiah Gbadegeshin Adebayo[2] and Bolarinwas Samson Adeleke[3]
[1,2,3]Department of Electronic and Electrical Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria
Corresponding Email: ajeedollar@gmail.com
DOI:10.56201/ijemt.v10.no10.2024.pg84.113

Abstract

This work develops Pelican Mayfly Algorithm (PMA) to minimize CNN high computational requirement to the minimum by the selection of its optimum parameters. PMA was designed by applying pelican exploration model to improve the attraction process of MA as deterministic process and to establish a balance between exploration and exploitation in MA. PMA was applied to optimize CNN hyper-parameters to develop hybridized CNN-PMA, and CNN-PMA was applied to South Western Nigeria electrical network for detection and classification of electrical faults. MAPE, MNE, RMSE, SNR and PSNR and confusion matrix were used as performance metrics. PMA achieved the optimum CNN architecture as follows: 1-convolutional-layer, filter size of 6 x 6, number of filters per layer is 128 and 256-batch-size with recognition-rate of 99.53%. PMA selected optimal parameters of CNN timely and accurately. CNN-PMA performed better in detection and classification of faults in SWN electrical network compared to CNN, CNN-MA and some other selected models.

Key words: *Convolutional Neural Network (CNN), Pelican Mayfly Algorithm, Hyper-parameters.*

1 INTRODUCTION

Convolutional Neural Network is a competent Artificial Intelligence (AI) algorithm in computer system for specific application such as: expert system, natural language processing, image recognition, machine vision as well as speech recognition (Tang *et al*., 2019). CNN is a special type ANN that has convolutional layers in replacement of linear map by ANN. Convolutional layers make use convolutional filters (Mozo *et al*., 2018). CNN belong to a class of feed-forward neural network which has convolutional operation and deep structure. CNN multi-layer neural network consists of many convolutional layers as well as pooling layers alternately together with one or more full connection layers connected for classification of image features generated by the previous layers (Chen *et al*., 2018). CNN has important advantages in processing of large amount of data with less computational cost. Hence, it is used in solving various engineering problems (Jing *et al*., 2017: Bracale *et al*., 2017). CNN has five main parts, they are: input layer, convolutional layer, pooling layer, full connection layer and output layer (Lu *et al*., 2019: Samet *et al*., 2021: Chen *et al*., 2018: Bukhari *et al*., 2020: Afrasiabi *et al*., 2019: Pan *et al*., 2019: Hatata *et al*., 2022).

Although, CNN has better opportunity of processing large data with small computational cost (Chen *et al*., 2018: Jing *et al*., 2017). However, CNN possess high computational requirement and has difficulty with small data (Zhao *et al*. (2020). In view of these, application of strong optimization technique reduces computational requirement of CNN by selection of its optimum hyper-parameters.

Optimization is a technique in AI, applied to obtain the best solution among different possible solutions under some constraint functions (Yang and Karamanoglu, 2016). Ogundoyin and Kamil (2021) categorized optimization techniques as: deterministic and stochastic. Several optimization techniques had been developed by different researchers and applied in different fields, for example: Kennedy and Eberthart, 1995 developed PSO, Geem and Kim, 2001 developed Harmony search, Yang, 2008 developed Firefly Algorithm and Storn and Price, 1997 developed Differential Evolution.

Recently, Mayfly Algorithm (MA) optimization was developed by Zervoudakis and Tsafarakis (2020). Its principle of operation is rooted in the mayflies social and mating process, however there is unbalance in both exploration and exploitation of MA. Similarly, Pelican Optimization Algorithm (POA) was developed by Trojovsky and Dehghani (2022). POA modeled the strategy and behavior of pelicans during hunting (Marchant, 1990). Their behaviors and strategies when hunting made them skilled hunters, and the design of POA replicated the modeling of pelicans' strategy (Perrins and Middleton, 1985: Anderson, 1991).

Despite the good performances of existing optimization techniques, development of new methods to achieve better results in term of accuracy, precision and Signal to noise ratio is on increase. Beside these, optimization problems in different fields require different approach because of their wide evolvement and enlargement. New optimization algorithms need to be developed from time to time to cope with the advancement in the field of computational intelligence optimization. In addition, according to 'No Free Lunch' (NFL) theorem: no single optimization algorithm could solve optimization problem in different fields (Wolpert and Macready, 1997). Hence, there is need for modification, enhancement or hybridization of existing methods or development of new methods for a better performance. In view of these, a new optimization method: PMA that can select optimum parameters of CNN is developed by combining MA and POA optimization techniques.

The main contribution of this paper is to carry out hybridization of Mayfly and Pelican optimization algorithms in order to develop a novel optimization technique that can select optimal hyper parameters of CNN. The main contributions of this work are thus summarized as follows: i) development of PMA, ii) development of CNN-PMA, iii) simulation of CNN-PMA, iv) detection of electrical faults and fault classification in 330kV electrical network and synchronous generator (SG) using CNN-PMA model and v) the performance evaluation of CNN-PMA compared with CNN-MA and CNN.

The remainders of this work are arranged as follows: Section Two presents the problem formulations while section Three shows the proposed hybridized model. Section Four discusses and presents results obtained while section Five concludes the work.

2 PROBLEM FORMULATION

Mathematical modeling for the proposed algorithms is presented in this section.

2.1 Mayflies Algorithms

Mayflies survive to adults after hatching, the position of each mayfly indicated a potential solution to a problem in the search space. Optimization algorithm development is as follows: two sets of mayflies are produced randomly indicating male and female populations in $d-$dimensional vectors: $x = (x_1, x_2, x_3, x_4 \dots\dots\dots x_d)$ and $y = (y_1, y_2, y_3, y_4, \dots\dots\dots y_d)$ respectively. Their performances are tested on the predefined objective function $f(x)$. Their velocity $v = (v_1, v_2, v_3, v_4 \dots\dots. v_d)$ is the change of position of mayfly. The direction of each mayfly is determined by individual flying experiences known as personal best position $(pbest)$ as well as the best position gained by any other mayflies of the swarm known as global best $(gbest)$. Assuming x_i denotes the initial position of mayfly 'i' at time $step\ t$, if there is change in the position by a velocity v_i^{t+1} to the new position as stated in Equation 1 (Zervoudakis and Tsafarakis, 2020).

$$x_i^{t+1} = x_i^t + v_i^{t+1} + v_i^{t+2} + v_i^{t+3} + \dots v_i^{t+n} \qquad 1$$

Male mayflies always perform nuptial dance a few meters above water as well as moving constantly with low speeds. Then, male mayfly velocity is calculated using Equation 2 (Zervoudakis and Tsafarakis, 2020).

$$v_{ij}^{t+1} = v_{ij}^t + a_1 e^{-\beta r_p^2}\left(pbest - x_{ij}^t\right) + a_2 e^{-\beta r_g^2}\left(gbest - x_{ij}^t\right) \qquad 2$$

where v_{ij}^t is the velocity of mayfly i in dimension $j = 1,2,3 \dots\dots. n$ at $step\ t$

x_{ij}^t is the position of mayfly in dimension j at $step\ t$

β is a fixed visibility coefficient used to limit mayfly visibility to others

r_p is the Cartesian distance between $x_i\ and\ pbest$

r_g is the Cartesian distance between $x_i\ and\ gbest$

$a_1\ and\ a_2$ are positive attraction constant used to scale the contribution of the cognitive and social component respectively.

Best female in the swarm is attracted to the best male why second-best female to second best male and so on. Then, their velocities are calculated as:

$$v_{ij}^{t+1} = \begin{cases} v_{ij}^t + a_2 e^{-\beta r_{mf}^2}\left(x_{ij}^t - y_{ij}^t\right) & if\ f(y_i) > f(x_i) \\ v_{ij}^t + fl * r & if\ f(y_i) \le f(x_i) \end{cases} \qquad 3$$

v_{ij}^{t+1} is the velocity of female mayfly$i$ in dimension $j = 1, \dots\dots n\ at\ time\ step\ t$

y_{ij}^t is the position of female mayfly i in dimension $j = 1, \dots\dots n\ at\ time\ step\ t$

a_2 is the positive attraction constant, $\beta$is the fixed visibility coefficient,

r_{mf} is the Cartesian distance between male and female mayflies,

fl is a random walk coefficient used when a female is not attracted by a male and

r is a random value in range $(-1, 1)$

The gravity coefficient g helps in achieving a sufficient balance between exploration and exploitation. Hence, male mayfly velocity i in Equation 2 is modified as:

$$v_{ij}^{t+1} = g * v_{ij}^t + a_1 e^{-\beta r_p^2}\left(pbest - x_{ij}^t\right) + a_2 e^{-\beta r_g^2}\left(gbest - x_{ij}^t\right) \qquad 4$$

And female mayfly i velocity in Equation 3 is modified as:

$$v_{ij}^{t+1} = \begin{cases} g * v_{ij}^t + a_2 e^{-\beta r_{mf}^2}\left(x_{ij}^t - y_{ij}^t\right) & if\ f(y_i) > f(x_i) \\ g * v_{ij}^t + fl * r & if\ f(y_i) \le f(x_i) \end{cases} \qquad 5$$

Gravity coefficient g is a constant in the range of (0, 1),

$$g = g_{max} - \frac{g_{max} - g_{min}}{iter_{max}} \, x \, iter \qquad 6$$

Where g_{max}, g_{min} are maximum and minimum values of the gravity, $iter$ is the latest iteration of the algorithm, and $iter_{max}$ is the maximum number of iterations.

2.2 The Pelican Optimization Algorithm

Details mathematical formulations of POA are presented in Trojovsky and Dehghani, 2022. Each population member indicates candidate solution, and the optimization problem variables were according to their position within the space. At starting stage, Equation 7 indicated population members at the lower and upper bound of the problem (Trojovsky and Dehghani, 2022).

$$x_{i,j} = l_j + rand \cdot (u_j - l_j), \quad i = 1, 2, \dots N, j = 1, 2, \dots m \qquad 7$$

Where $x_{i,j}$ is the value of the j_{th} variable specified by the i_{th} candidate solution,

N is the number of population member, m is the number of problem variable,

$rand$ is a random number in interval (0, 1), l_j is the j_{th} lower bound, and u_j is the j_{th} upper bound of problem variables.

Hunting strategy is modeled in two stages;

(i) Moving toward prey (exploration phase) (**phase 1**)

(ii) Winging on the water surface (exploitation phase) (**phase 2**)

In exploration phase, the pelicans locate the prey and move towards it. This concept is mathematically simulated in Equation 8

$$x_{i,j}^{p_1} = \begin{cases} x_{i,j} + rand \cdot (p_j - I \cdot x_{i,j}), & F_p < F_i \, ; \\ x_{i,j} + rand \cdot (x_{i,j} - p_j), & else \end{cases} \qquad 8$$

Where $x_{i,j}^{p_1}$ is the new status of the i_{th} pelican in the j_{th} dimension based on phase 1,

p_j is the location of prey in the j_{th} dimension, and F_p is its objective function value.

I is a number that can be randomly equal to 1 or 2, and randomly selected for each iteration and for each member.

In exploitation phase, after the pelicans reach the surface of the water, they spread their wings and move the fish to a shallow area for collection. The behavior of pelicans during hunting is simulated mathematically in Equation 9.

$$x_{i,j}^{p_2} = x_{i,j} + R \cdot \left(1 - \frac{t}{T}\right) \cdot (2 \cdot rand - 1) \cdot x_{i,j} \qquad 9$$

Where $x_{i,j}^{p_2}$ is the current status of the i_{th} pelican in the j_{th} dimension based on phase 2,

R is a constant equal to 0.2, $R \cdot \left(1 - \frac{t}{T}\right)$ is the neighborhood radius of $x_{i,j}$,

t is the iteration counter, and T is the maximum number of iterations.

Hence, POA converges to solutions closer to the global optimal based and effectively updating to accept or reject the new pelican position.

3 Hybridization of MA and POA (PMA)

Pelican Mayfly Algorithm (PMA) is modeled by applying pelican exploration strategy to design the attraction process of standard Mayfly algorithm. The application of pelican's exploration and exploitation behaviors to MA established balance between exploration and exploitation in MA. PMA is applied for optimization of CNN hyper parameters such as: number of layers, number of filters in each layer, filter size and batch size. The Pelican Mayfly Algorithm (PMA) is formulated using Equation 10 to model the attraction process of male and female mayflies as a deterministic process instead of the random process for selection of hyper parameters in the existing mayfly. The updated velocities and solution of male and female using Pelican Exploration Phase is expressed in Equation 10.

$$\left.\begin{array}{l} \text{If } F_P < f(x)_. \\ v_{std} = x_{std} + rand * (x_{mean} - I * x_{std}) \text{ where } rand\epsilon(0,1) \\ \text{else,} \\ v_{std} = x_{std} + rand * (x_{std} - x_{mean}) \text{ where} rand\epsilon(0,1) \\ \text{end} \end{array}\right\} \quad 10$$

where $\boldsymbol{x_{std}}$ and $\boldsymbol{x_{mean}}$ are the search space limits for the fitness function, I is a random number between 1 and 2. F_P is its new objective function value, $f(x)_.$ is the initial objective function value of the males and females mayflies. Given the existing Mayfly Algorithm velocity updates as in Equation 11

$$v_{ij}^{t+1} = g * v_{ij}^t + a_1 e^{-\beta r_p^2}[pbest_{ij} - x_{ij}^t] + a_2 e^{-\beta r_g^2}[gbest_j - x_{ij}^t] \quad 11$$

where $\boldsymbol{\beta}$ is a fixed visibility coefficient which is used to limit a mayfly's visibility to others,
r_p is the Cartesian distance between $\boldsymbol{x_i}$ and $pbest_{ij}$ and,
r_g is the Cartesian distance between $\boldsymbol{x_i}$ and $gbest_j$.
Challenges of imbalance between exploration and exploitation experienced by exiting mayfly algorithm were resolved in this study by modifying the velocity of the female with application of Pelican Exploitation Phase. Mathematically, Equation 12 expressed the Pelican male and female position to converge to a better solution.

$$x_{ij}{}^P = x_{ij}{}^t + R.\left(1 - \frac{t}{T}\right).(2.rand - 1).x_{ij}{}^t \quad 12$$

where $x_{ij}{}^P$ is the latest status position of the *ith* pelican in the *jth* dimension based on pelican exploitation phase, R is a constant, which is equal to 0.2,
$R.\left(1 - \frac{t}{T}\right)$ is the neighbourhood radius of $x_{ij}{}^t$
t is the iteration counter, and
T is the maximum number of iterations.

The coefficient $R.\left(1 - \frac{t}{T}\right)$ indicated the radius of the neighbourhood of the population members of male and female mayfly and improve exploitation power of PMA. PMA convergence to solutions closer to the global optimal based on the usage concept as expressed in Equation 13.

$$v_{ij}^{t+1} = g * v_{ij}^t + a_1 e^{-\beta r_p^2}[pbest_{ij} - x_{ij}^p] + a_2 e^{-\beta r_g^2}[gbest_j - x_{ij}^p] \quad 13$$

The algorithmic steps for the PMA technique used to achieve optimized CNN parameters selection is described in Algorithm 1 The output from the CNN parameters selection is the most significant balanced parameters used by CNN for feature extraction and classification.

Algorithm 1: Pelican May-Fly Algorithm

Step 1: Assign initial values of the male mayfly population $\boldsymbol{x_{ij}^0}$ (i=1,2, 3,4 …, N) and velocities $\boldsymbol{v_{ij}^0}$ $(i = 1,2,3,4, \dots \dots .V)$,
Assign initial values of the female mayfly population $\boldsymbol{y_{ij}^0}$ (i=1, 2,3,4 …, M),
$\boldsymbol{Max_{iter}}$ =max.no of iteration
Step 2: Set iteration t = 1

Step 3: Compute the objective function values of males and females' mayflies as $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}_i^t)$. where $\boldsymbol{f}: \boldsymbol{R}^n \rightarrow \boldsymbol{R}$ is the objective function which evaluates the quality of a solution

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{k=2}^{m}\left[\sum_{i=1}^{n}\left(x_{i,k-1} - x_{i,k}\right)^2\right]$$

Where $\boldsymbol{x}_i^t$ denote the CNN parameters at *i =1,2,3,4 ..., n* and *k =2,3, 4,5..., m*

Step 4: Locate the $Pbest$ for each male and female as $\boldsymbol{P}_{best,iD}^t = \boldsymbol{x}_i^t$ and G$best$ as $\boldsymbol{G}_{best,iD} = \boldsymbol{min}\{\boldsymbol{P}_{best,iD}^t\}$

Step 5: Determine gravity coefficient: The gravity coefficient g may be a fixed value in the range of [-1, 1], or it may be gradually reduced over the iterations, making the algorithm to achieve few worst and best specific areas as displayed in equation

$$\boldsymbol{g} = \boldsymbol{g}_{std} - \frac{(\boldsymbol{g}_{std} - \boldsymbol{g}_{mean}) * (\boldsymbol{iter}_{max} - \boldsymbol{iter} + \mathbf{1})}{\boldsymbol{iter}_{max}} - \boldsymbol{iter}$$

where $\boldsymbol{g}_{std}$ and $\boldsymbol{g}_{\text{mean}}$ are the standard deviation and mean values respectively, $\boldsymbol{iter}$ is the initial iteration of the algorithm and $\boldsymbol{iter}_{max}$ is the maximum number of iterations.

Step 6: Modify velocities and solution of males and females' mayflies

Using Pelican Exploration Phase $(\boldsymbol{x}_i)$

If $\boldsymbol{F}_P < \boldsymbol{f}(\boldsymbol{x})$

$\boldsymbol{v}_{std} = \boldsymbol{x}_{std} + \boldsymbol{rand} * (\boldsymbol{x}_{mean} - \boldsymbol{I} * \boldsymbol{x}_{std})$ **where** $\boldsymbol{rand}\ \boldsymbol{\epsilon}(\mathbf{0}, \mathbf{1})$

else

$$\boldsymbol{v}_{std} = \boldsymbol{x}_{std} + \boldsymbol{rand} * (\boldsymbol{x}_{std} - \boldsymbol{x}_{mean})\ \textbf{where}\ \boldsymbol{rand}\ \boldsymbol{\epsilon}(\mathbf{0}, \mathbf{1})$$

end

Where $\boldsymbol{x}_{std}$ and $\boldsymbol{x}_{mean}$ are the search space limits for the fitness function, *I* is a random number which is equal to 1 or 2. $\boldsymbol{F}_P$ is its new objective function value, $\boldsymbol{f}(\boldsymbol{x})$ is the initial objective function value

$$\boldsymbol{v}_{ij}{}^{t+1} = \begin{cases} \boldsymbol{v}_{std}, & \boldsymbol{if}\, \boldsymbol{v}_{ij}{}^{t+1} > \boldsymbol{v}_{std} \\ -\boldsymbol{v}_{std}, & \boldsymbol{if}\, \boldsymbol{v}_{ij}{}^{t+1} < -\boldsymbol{v}_{std} \end{cases}$$

$$\boldsymbol{x}_{ij}{}^{P} = \boldsymbol{x}_{ij}{}^{t} + \boldsymbol{R}.\left(\mathbf{1} - \frac{\boldsymbol{t}}{\boldsymbol{T}}\right).(\mathbf{2}.\boldsymbol{rand} - \mathbf{1}).\boldsymbol{x}_{ij}{}^{t}$$

Where $\boldsymbol{x}_{ij}{}^{P}$ is the current status position of the ith pelican in the jth dimension based on pelican exploitation phase, R is a constant, which is equl to 0.2, $\boldsymbol{R}.\left(\mathbf{1} - \frac{\boldsymbol{t}}{\boldsymbol{T}}\right)$ is the neighbourhood radius of $\boldsymbol{x}_{ij}{}^{t}$ while, $t = iteration\ counter$, and $T = maximum\ number\ of\ iteration.$

$$\boldsymbol{v}_{ij}{}^{t+1} = \boldsymbol{g} * \boldsymbol{v}_{ij}{}^{t} + a_1 \boldsymbol{e}^{-\beta r_p^2}\left[\boldsymbol{pbest}_{ij} - \boldsymbol{x}_{ij}{}^{P}\right] + a_2 \boldsymbol{e}^{-\beta r_g^2}\left[\boldsymbol{gbest}_j - \boldsymbol{x}_{ij}{}^{P}\right]$$

Where

$\boldsymbol{\beta}\ is\ a\ fixed\ visibility\ coefficient\ which\ is\ used\ to\ limit\ a\ mayfly's\ visibility\ to\ others$, $r_p$is the Cartesian distance between $\boldsymbol{x}_i$ and $\boldsymbol{pbest}_{ij}$, and $\boldsymbol{r}_g$ is the Cartesian distance between $\boldsymbol{x}_i$and $\boldsymbol{gbest}$. The distances are calculated as:

$$\|\boldsymbol{x_i} - \boldsymbol{X_i}\| = \sqrt{\sum_{j=1}^{n} (\boldsymbol{x_{ij}} - \boldsymbol{X_{ij}})^2}$$

Where $\boldsymbol{x_{ij}}$ is the j^{th} element of mayfly i and $\boldsymbol{X_{ij}}$ corresponds to $\boldsymbol{pbest_{ij}}$or$\boldsymbol{gbest}$.

$$\boldsymbol{x_i}^{t+1} = \boldsymbol{x_i}^{t} + \boldsymbol{v_{ij}}^{t+1}$$

With $\boldsymbol{x_i}^{0} \sim \boldsymbol{U}(x_{mean}, x_{std})$ male mayfly = $\boldsymbol{y_i}^{t+1} = \boldsymbol{y_i}^{t} + \boldsymbol{v_{ij}}^{t+1}$
With $\boldsymbol{y_i}^{0} \sim \boldsymbol{U}(y_{mean}, y_{std})$ female mayfly
Using roulette wheel selection $\boldsymbol{p_i}$

$$\boldsymbol{p_i} = \boldsymbol{r} \leq \frac{\boldsymbol{f}(\boldsymbol{x_i^t})}{\sum_{i=1}^{N} \boldsymbol{f}(\boldsymbol{x_i^t})}$$

$$\boldsymbol{v_{ij}^{t+1}} = \begin{cases} \boldsymbol{v_{ij}^t} + a_2 \boldsymbol{e}^{-\beta r_{mf}^2 (x_{ij}{}^t - y_{ij}{}^t)} & \boldsymbol{if}\ (y_i) > f(x_i) \\ \boldsymbol{v_{ij}^t} + \boldsymbol{fl} * \boldsymbol{p_i} & \boldsymbol{if} (y_i) \leq f(x_i) \end{cases}$$

Where $\boldsymbol{v_{ij}^t}$ = is the velocity of female mayfly $\boldsymbol{i}$ in dimension j = 1, 2…, at time step t, $\boldsymbol{y_{ij}}^{t}$ = the position of female mayfly $\boldsymbol{i}$ in dimension $\boldsymbol{j}$ at time step $\boldsymbol{t}$, a_2 = positive attraction constant and $\boldsymbol{\beta} =$ fixed visibility coefficient, while r_{mf} = Cartesian distance between male and female mayflies, estimated using:

$$\|\boldsymbol{x_i} - \boldsymbol{X_i}\| = \sqrt{\sum_{j=1}^{n} (\boldsymbol{x_{ij}} - \boldsymbol{X_{ij}})^2}$$

Finally, $\boldsymbol{fl} =$ random walk coefficient, used when a female is not attracted by a male, so it flies deterministically by roulette wheel selection and r = random value in the range of [-1, 1].
Step 7: Compute Solutions: $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x_i^{t+1}})$
where $\boldsymbol{f}: \boldsymbol{R^n} \rightarrow \boldsymbol{R}$ is the objective function which evaluates the quality of a solution
Step 8: Mate the mayflies and Compute offspring

$$\boldsymbol{offspring1} = \boldsymbol{L} * \boldsymbol{male} + (\boldsymbol{1} - \boldsymbol{L}) * \boldsymbol{female}$$
$$\boldsymbol{offspring2} = \boldsymbol{L} * \boldsymbol{male} + (\boldsymbol{1} - \boldsymbol{L}) * \boldsymbol{male}$$

where $\boldsymbol{male}$ and $\boldsymbol{female}$ are the male and female parents respectively, and L = random value in the scope of specific range. Offspring's early velocities are put to be zero
Step 9: Modify $\boldsymbol{Pbest}$ of population using:

$$\boldsymbol{pbest_i} = \begin{cases} \boldsymbol{x_i^{t+1}}, & \boldsymbol{iff}(\boldsymbol{x_i^{t+1}}) > \boldsymbol{f}(\boldsymbol{pbest_i}) \\ \boldsymbol{iskeptthesame}, & \boldsymbol{otherwise} \end{cases}$$

Step 10: Modify $\boldsymbol{Gbest}$ of population using:
The $\boldsymbol{gbest}$ position at $time = step\ t$, is defined as

$$\boldsymbol{gbest} \in \{\boldsymbol{pbest_1}, \boldsymbol{pbest_2}, \boldsymbol{pbest_3}, \boldsymbol{pbest_4} \dots, \boldsymbol{pbest_N} | \boldsymbol{f}(\boldsymbol{cbest})\}$$
$$= \mathbf{min}\ \{\{\boldsymbol{f}(\boldsymbol{pbest_1}), \quad \boldsymbol{f}(\boldsymbol{pbest_2}), \dots, \boldsymbol{f}(\boldsymbol{pbest_N}) |\}$$

Where $N = total\ number\ of\ male\ mayflies\ in\ the\ swarm,$
Step 11: If t< $\boldsymbol{Max_{iter}}$ then t= $\boldsymbol{t+1}$ and GOTO step 1 else GOTO step l2

Step 12: Output: optimum parameters of CNN are selected solution as $\boldsymbol{Gbest_{bD}}$.

$$\boldsymbol{Gbest_{bD}} = x_b$$

3.1 The optimization of hyper-parameters

PMA algorithm is adopted in the CNN architecture model's classification section as optimization technique in the batch size and dropout-layer section. The hyper-parameters of CNN optimized by PMA are: numbers of convolutional layers, size of the filters in each layer, the number of filters, and the batch size. Figure 1 showed the block diagram of optimization process of the CNN-PMA model. Each mayfly acted as a configuration of CNN with its hyper parameters. The general methodology of CNN-PMA is shown in Figure 2 with the flowchart of CNN-PMA, as the "training and optimization" block is the most important part of the whole process, where the CNN was initialized to integrate the parameter optimization by applying the PMA algorithm. In this process, the PMA was initialized in accordance with parameter given for the execution in Algorithm 1 and this generated males and females' mayflies. Each mayfly is a likely solution and its location has parameters to be optimized, hence, a complete CNN training.

Training process begins with an iterative cycle and ends with evaluation of all the mayflies generated using the PMA for each generation. The database size, size of mayflies, number of iterations of the PMA as well as number of males and females' mayflies in each iteration determine the computational cost of the model. That is, if the PMA is executed with 10 male and female mayflies and 10 iterations, the training of CNN is carried out in 100 times. The step by step algorithm for optimization of CNN using the PMA algorithm are clarified in Figure 2.

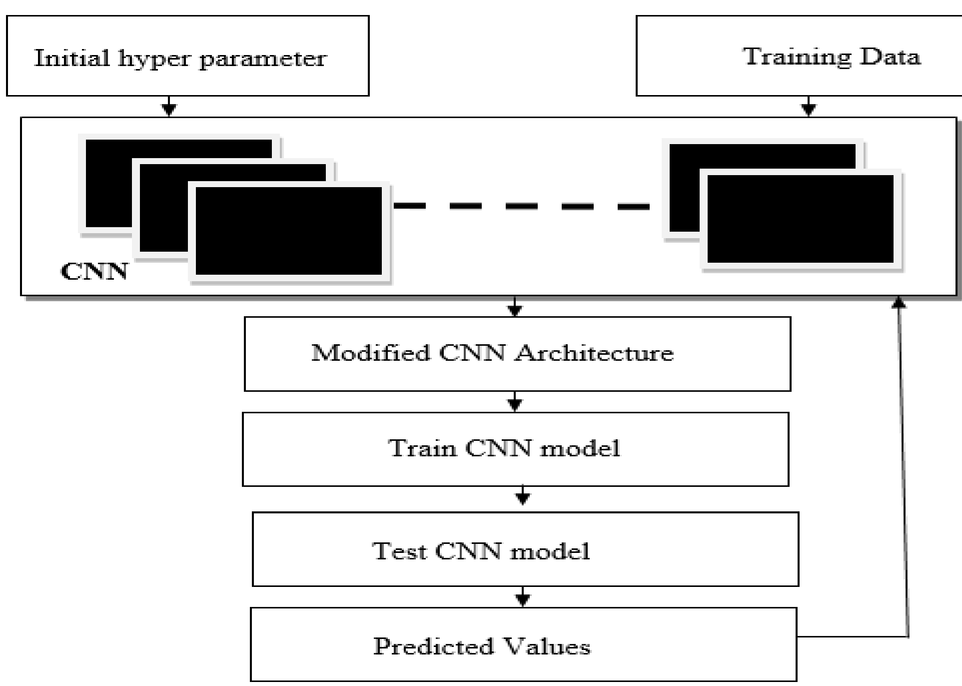


Figure 1 Block Diagram of the Proposed Hybridized CNN-PMA Model

3.2 Application CNN-PMA as Fault Detection and Classification Models

CNN-PMA model is designed and simulated for detection and classification of faults in 330kV electrical line in SWN, its flow diagram is shown in Figure 3 and the architectural drawing is shown in Figure 4. The encoded Gramian angular field (GAF) images of the three voltages and currents for 330kV lines is fed to the models of CNN- PMA. An interactive GUI application is developed with electrical faults on SWN 330kV network data. The GUI is designed using deep learning and optimization toolboxes in MATLAB 2020a.

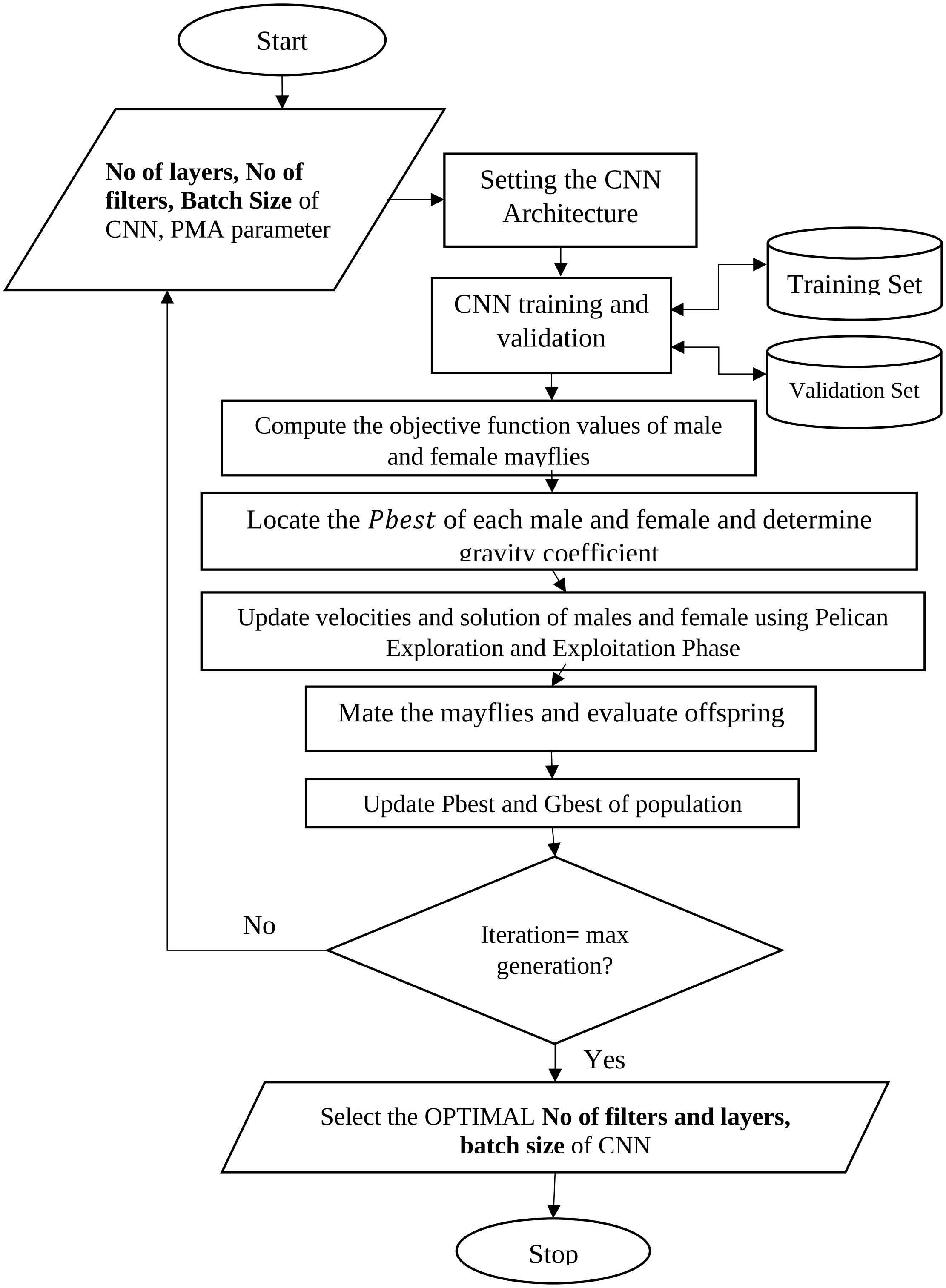


Figure 2 Flowchart of Optimization of Convolutional Neural Network with Pelican Mayfly Algorithm (CNN-PMA)

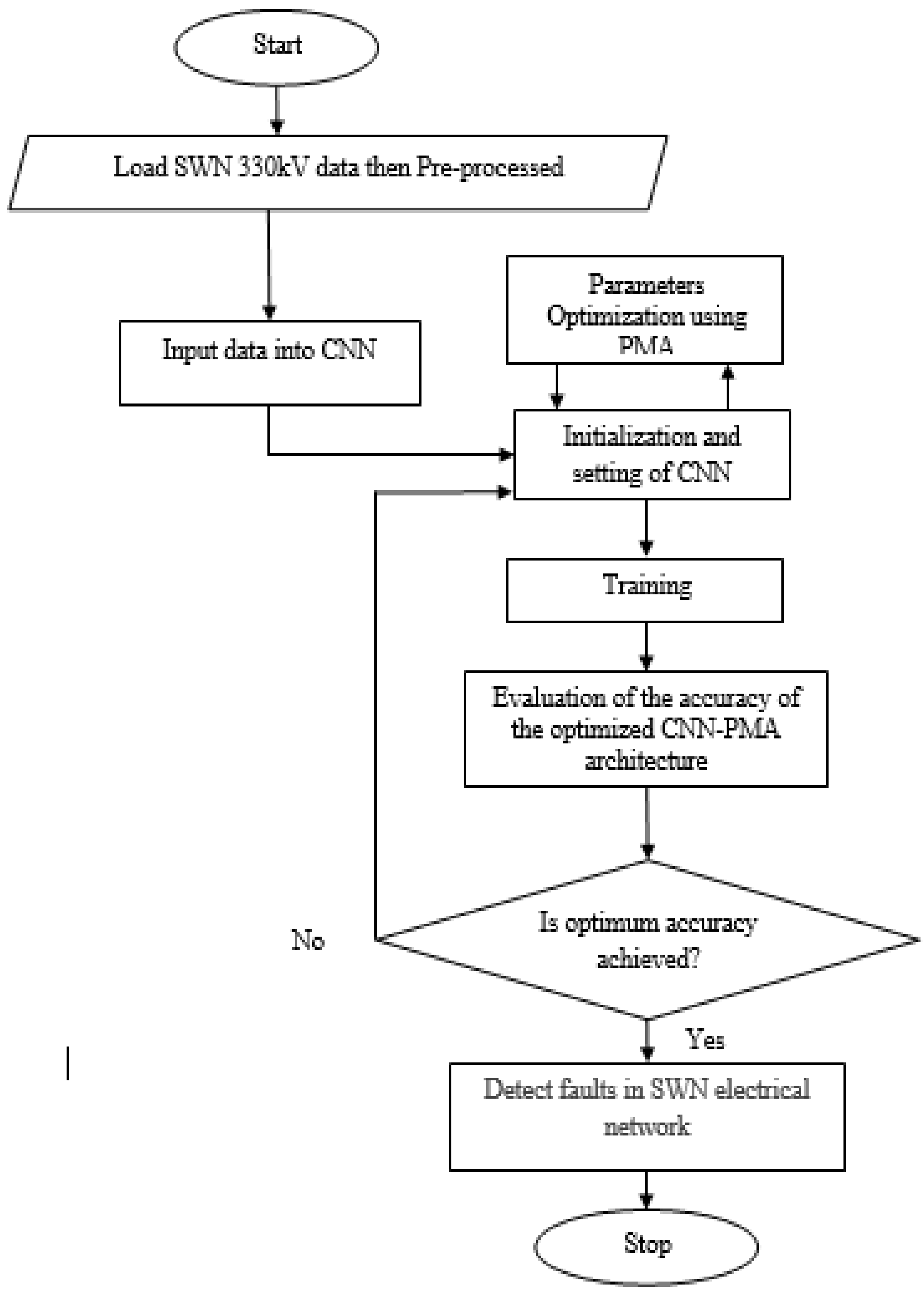
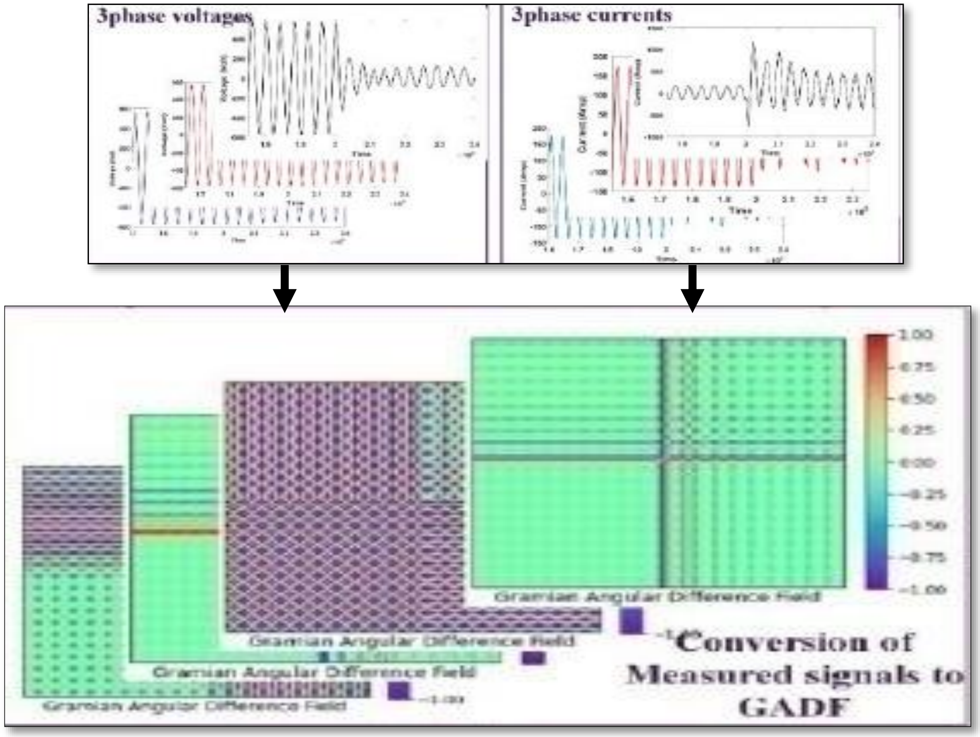


Figure 3 Flow Diagram of Implementation of CNN-PMA Model


3phase voltages
3phase currents
Gramian Angular Difference Field
Gramian Angular Difference Field
Gramian Angular Difference Field
Gramian Angular Difference Field
Conversion of Measured signals to GADF

I

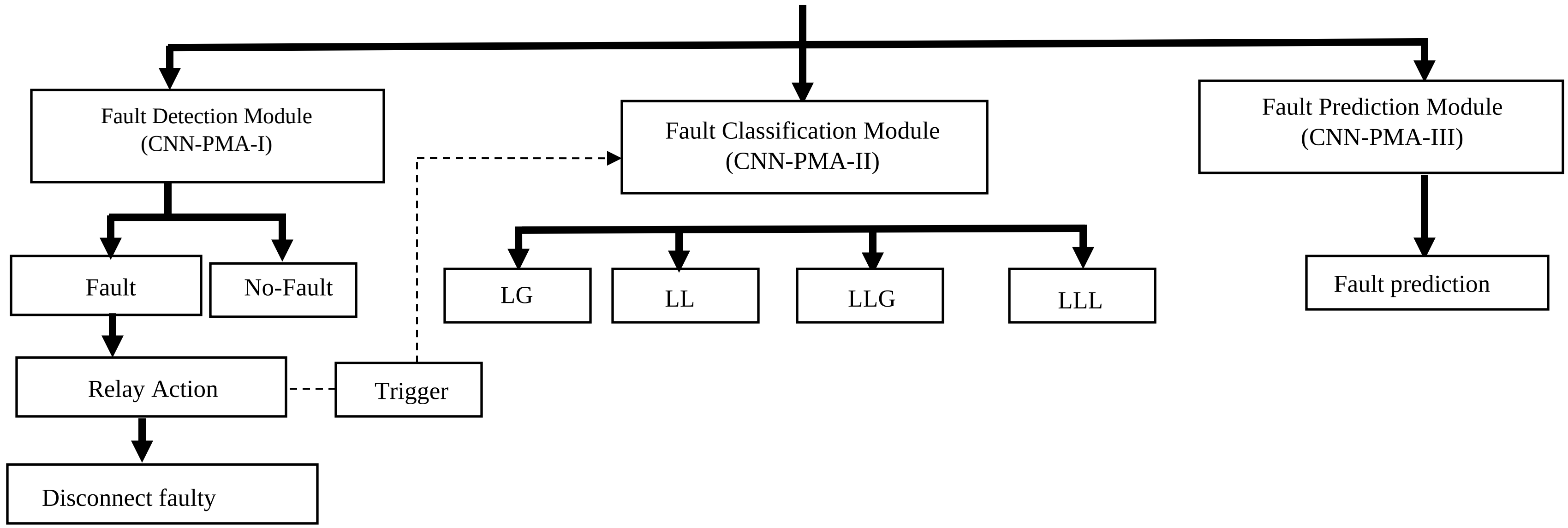


Figure 4 Architecture of the Proposed CNN-PMA Model for SWN 330kV Network

4. RESULTS AND DISCUSSION

Results obtained from optimization of CNN using MA and PMA algorithms are discussed in this section. The developed algorithm was tested and evaluated using the following performance metrics: MAPE, MSE, RMSE, Corr-Coeff., SNR and PSNR. Tables 2 and 3 showed the optimization results of MA on CNN and PMA on CNN at 30 iterations with different numbers of layers, filters, filter size and batch size. The best recognition rates were achieved at 99.27% (iteration: 6) and 99.53% (iteration: 30) for MA and PMA respectively. In addition, Figures 5 and 6 showed graphical representation of CNN Hyper-parameter selection using MA and PMA. Based on the results in Table 4, which is graphically shown in Figure 7, comparing MA and PMA performances, PMA achieved the optimum CNN architecture as follows: 1convolutional layer, 128 number of filters per layer and filter size of 6 x 6, the batch size is 256 which guaranteed convergence of CNN-PMA to global optimal. Furthermore, Tables 5(a, b and c), showed the results obtained by CNN, CNN-MA and CNN-PMA respectively at different threshold with respect to the performance metrics, their graphical representation were displayed in Figures 8(a, b and c) respectively.

The results obtained from Table 5(c) at test sample percentage of 20% clearly revealed that CNN-PMA had the least MAPE of 8.576531, least MSE of 0.011512, least RMSE of 0.107293, highest SNR of 8.813529 and highest PSNR of 8.930958. These implied that PMA has higher accuracy and efficiency compared to CNN and CNN-MA. Hence, CNN-PMA has better performance compared to CNN and CNN-MA with accuracy of 99.53%.

4.1 CNN-PMA as Fault Detection model

To perform fault detection, the Transmission line (TL) configuration comprises two generating units and three RLC loads was used. Irregular flow of voltage and current are termed as the TL fault. Likewise, all forms of faults were activated based on a set program using a fault generator block. Fault detection was performed with TL as shown by confusion matrix in Figure 9(a), (b) and (c) for CNN, CNN-MA and CNN-PMA respectively. Two classes: faulty and no-faulty were considered. Based on these confusion matrixes, it is shown that CNN-PMA model detected electrical fault accurately. Table 6 showed summary of performance evaluation of CNN, CNN-MA and CNN-PMA as fault detection model with the use of 80% of augmented (8832) data as testing data.

Table 2: Selection of CNN optimal parameters using MA optimization technique

S/N	Number of Layers	Number of Filters	Filter Size	Batch Size	Recognition Rate (%)
1	3	104	3	236	97.31
2	3	109	3	136	98.47
3	3	112	5	150	96.51
4	1	83	6	149	98.15
5	1	57	7	238	95.55
6	**1**	**128**	**6**	**155**	**99.27**
7	3	128	6	203	97.81

8	2	128	5	228	95.62
9	3	128	5	256	95.98
10	1	128	6	256	95.38
11	3	128	6	256	96.78
12	3	128	5	256	98.87
13	2	128	3	256	99.15
14	1	128	5	256	98.60
15	2	128	3	256	97.96
16	1	128	6	256	96.42
17	2	128	3	256	98.76
18	3	128	5	256	96.02
19	2	128	6	256	96.74
20	3	128	4	256	98.57
21	3	128	7	256	98.70
22	2	128	7	256	96.95
23	2	128	3	256	98.19
24	3	128	7	256	98.33
25	3	128	3	256	96.09
26	1	128	6	256	95.28
27	2	128	6	256	96.44
28	1	128	5	256	95.69
29	3	128	7	256	98.73
30	2	128	3	256	97.08

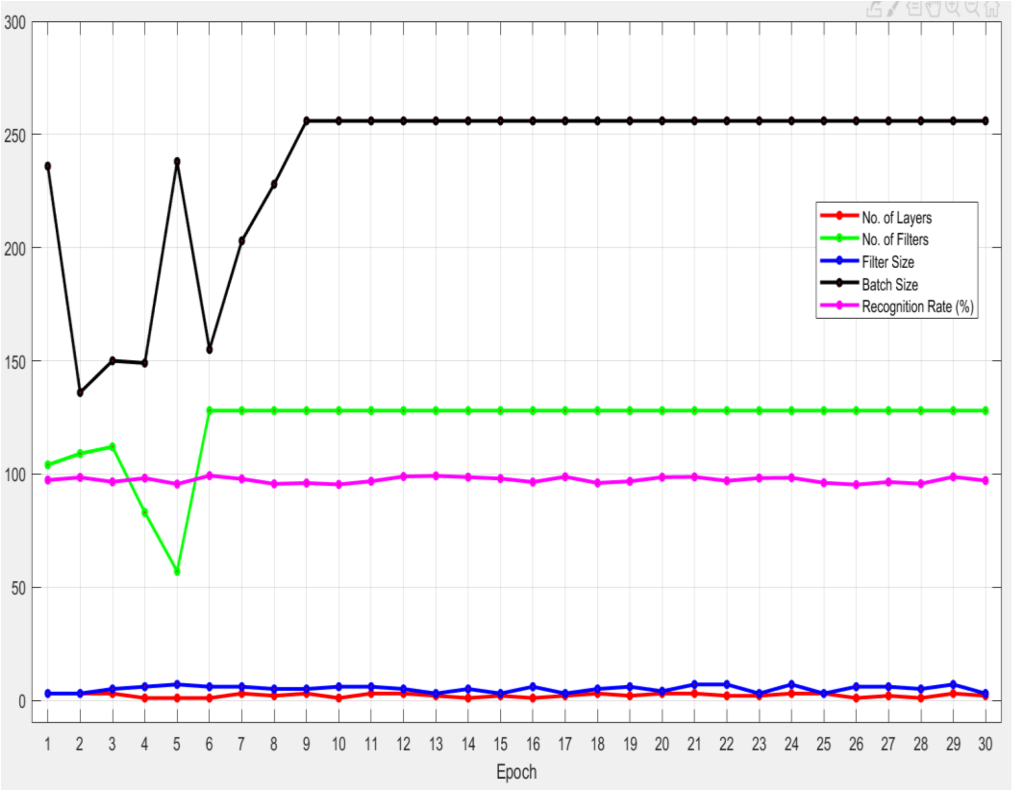


Figure 5: CNN Optimal Hyper-parameter selection process using MA

Table 3: Selection of CNN optimal parameters using PMA optimization technique

S/N	Number of Layers	Number of Filters	Filter Size	Batch Size	Recognition Rate (%)
1	3	58	7	230	95.31
2	1	67	3	196	99.16
3	1	97	7	137	95.27
4	2	60	4	115	95.69
5	1	61	7	230	95.50
6	1	128	6	107	98.10
7	2	128	7	109	98.12
8	2	128	4	221	98.60
9	3	128	6	256	95.86
10	3	128	5	256	97.30
11	3	128	4	256	95.95
12	3	128	7	256	96.00
13	2	128	7	256	95.90
14	1	128	3	256	96.50
15	3	128	7	256	95.39
16	2	128	3	256	98.49
17	3	128	5	256	97.42
18	2	128	7	256	98.39
19	2	128	4	256	96.70
20	2	128	7	256	98.22
21	2	128	3	256	98.26
22	1	128	7	256	95.78
23	1	128	4	256	95.66
24	1	128	7	256	95.79
25	1	128	5	256	95.84
26	3	128	3	256	98.17
27	1	128	4	256	95.87
28	2	128	7	256	97.66
29	2	128	3	256	95.43
30	**1**	**128**	**6**	**256**	**99.53**

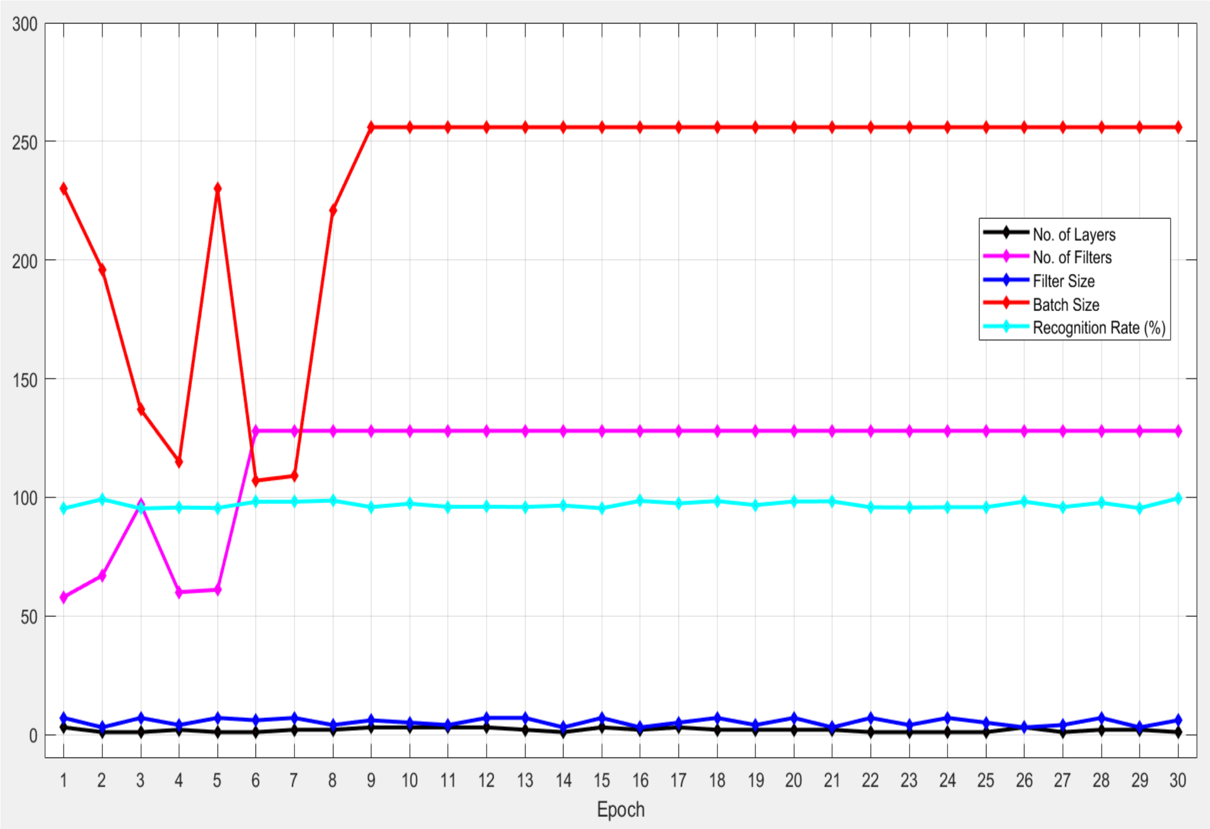


Figure 6: CNN Optimal Hyper-parameter selection process using PMA

Table 4: Comparison of Selected best Optimal Hyper-parameters of CNN using MA and PMA

Epoch	No. Layers	No. Filters	Filter Size	Batch Size	Recognition Rate (%)	Optimization Methods
6	1	128	6	155	99.26983	MA
30	1	128	6	256	99.53374	PMA

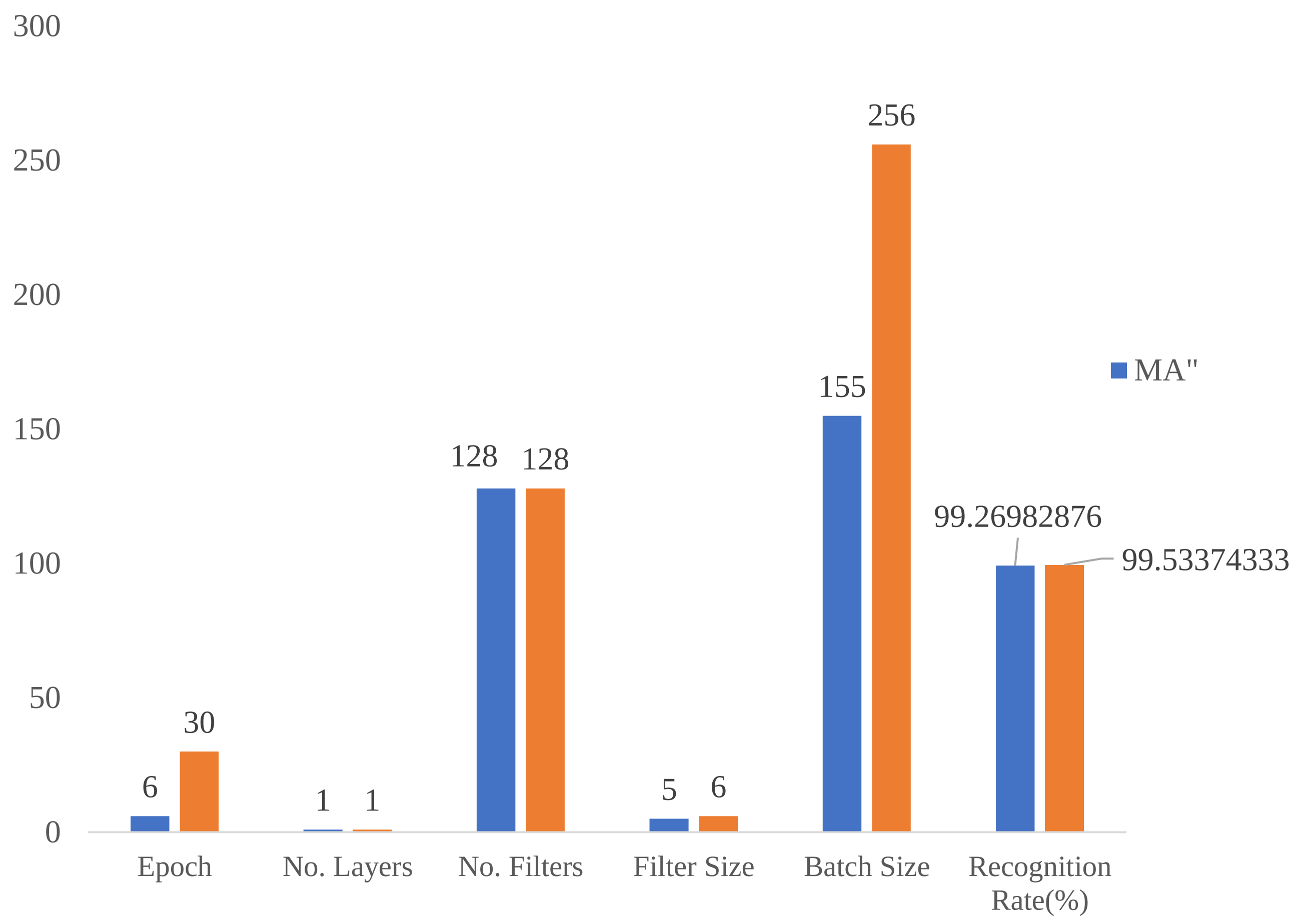


Figure 7: Comparison of Selected best Optimal Hyper-parameters of CNN using MA and PMA

Table 5a: Validation of CNN using MAPE, MSE, RMSE, CorrCoeff, SNR and PSNR

MAPE	MSE	RMSE	CorrCoeff	SNR	PSNR	Technique	Sample Percentage
18.80112	1.171857	1.082524	0.008102	8.740738	8.853678	CNN	0.4
18.71146	1.161948	1.077937	0.019925	8.788822	8.890559	CNN	0.3
18.58248	1.155719	1.075044	0.044140	8.791979	8.913902	CNN	0.2

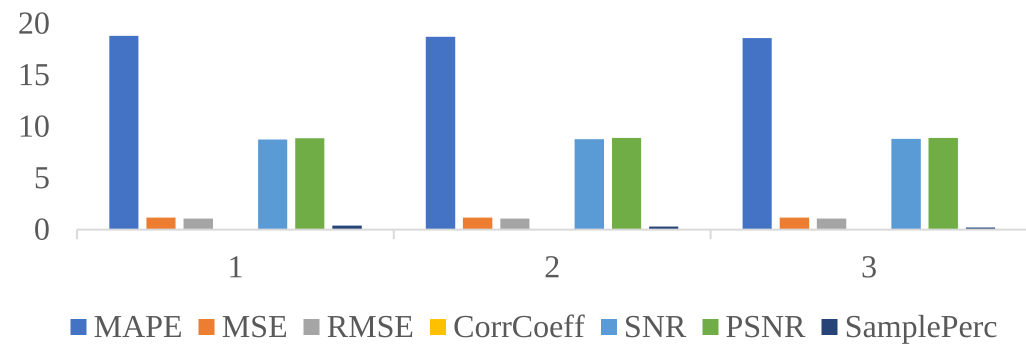
Table 5b: Validation of CNN-MA using MAPE, MSE, RMSE, CorrCoeff, SNR and PSNR

MAPE	MSE	RMSE	CorrCoeff	SNR	PSNR	Technique	Sample Percentage
12.43761	0.046195	0.21493	0.033427	8.815359	8.917095	MA-CNN	0.4
12.50786	0.046357	0.215307	0.019991	8.810555	8.901861	MA-CNN	0.3
12.51416	0.046319	0.215219	0.026586	8.812605	8.905399	MA-CNN	0.2

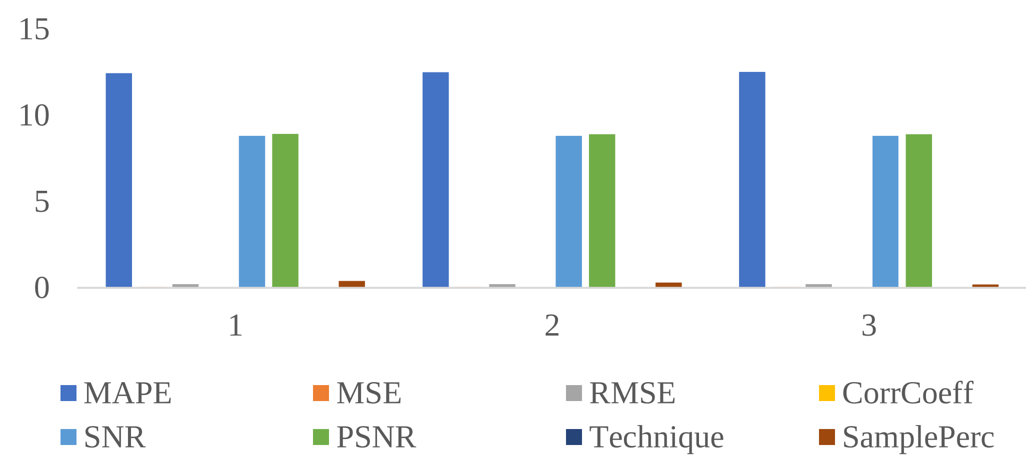
Table 5c: Validation of CNN-PMA using MAPE, MSE, RMSE, CorrCoeff, SNR and PSNR

MAPE	MSE	RMSE	CorrCoeff	SNR	PSNR	Technique	Sample Percentage
8.651305	0.011662	0.10799	0.013993	8.76962	8.874714	PMA-CNN	0.4
8.682522	0.01168	0.108073	0.005817	8.770779	8.868042	PMA-CNN	0.3
8.576531	0.011512	0.107293	0.046538	8.813529	8.930958	PMA-CNN	0.2

(a)



(b)



(c)

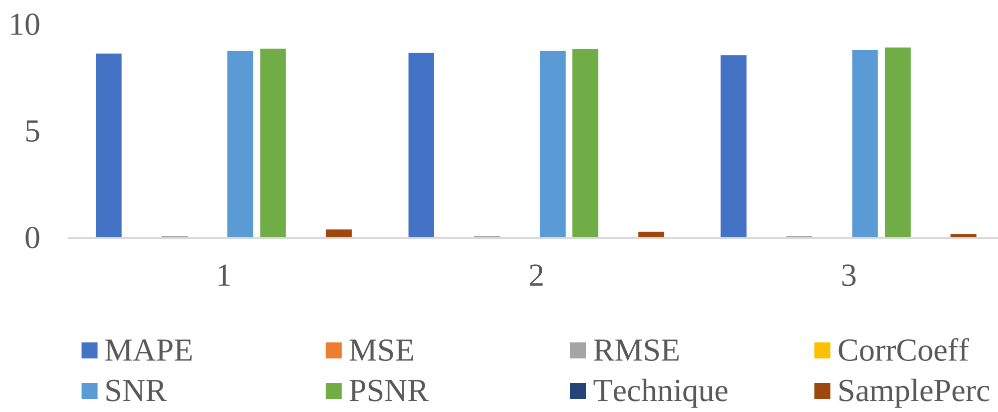
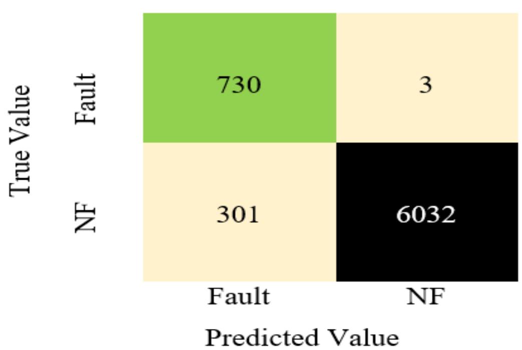
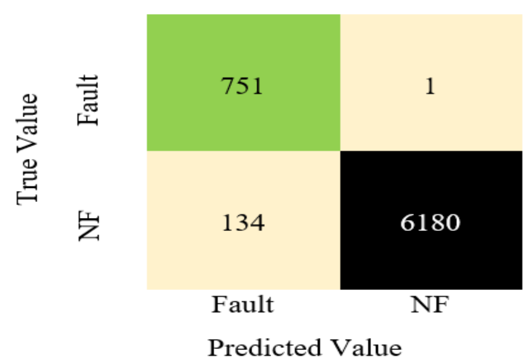


Figure 8: Evaluation parameters of (a) CNN, (b) CNN-MA, (c) CNN-PMA.

(a)



(b)



(c)

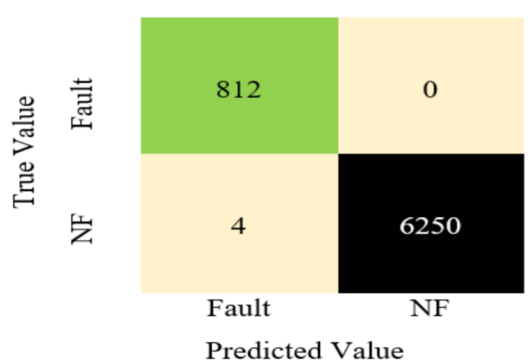


Figure 9 Confusion Matrix for fault detection: (a) CNN, (b) CNN-MA and (c) CNN-PMA

Table 6 Summary of Evaluation Standard for fault detection

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	No-Fault	Fault	Mis-match	Count
CNN	95.70	99.59	70.80	82.76	730	6032	304	7066
CNN-MA	98.09	99.86	84.86	91.75	751	6180	135	7066
CNN-PMA	99.94	100	99.51	99.75	812	6250	4	7066

4.2 CNN-PMA as Fault Classification model

A total of 3541fault data of SWN transmission lines were collected for the period of twenty-three years. K-fold cross validation is used for training and for testing where K=10. Fault classification was performed on the training data using CNN, CNN-MA and CNN-PMA. Figures 10, 11 and 12 represented corresponding confusion matrixes for CNN, CNN-MA and CNN-PMA respectively. Table 7 presented different performance evaluation criteria for CNN, CNN-MA and CNN-PMA. Considering Table 7, CNN-PMA results showed a better performance compared to CNN and CNN-MA in term of accuracy, precision, recall and F1-score. Hence, CNN-PMA displayed excellent performance in classification of electrical faults in SWN electrical network.

Figure 10: Confusion Matrix for fault classification using CNN model in 330kV network

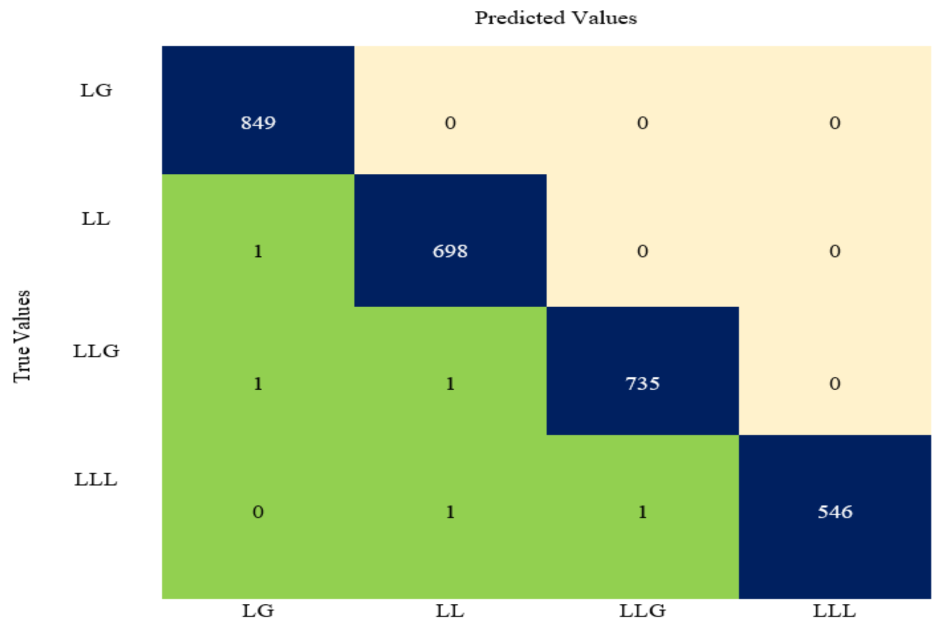


Figure 11: Confusion Matrix for fault classification using CNN-MA in 330kV network

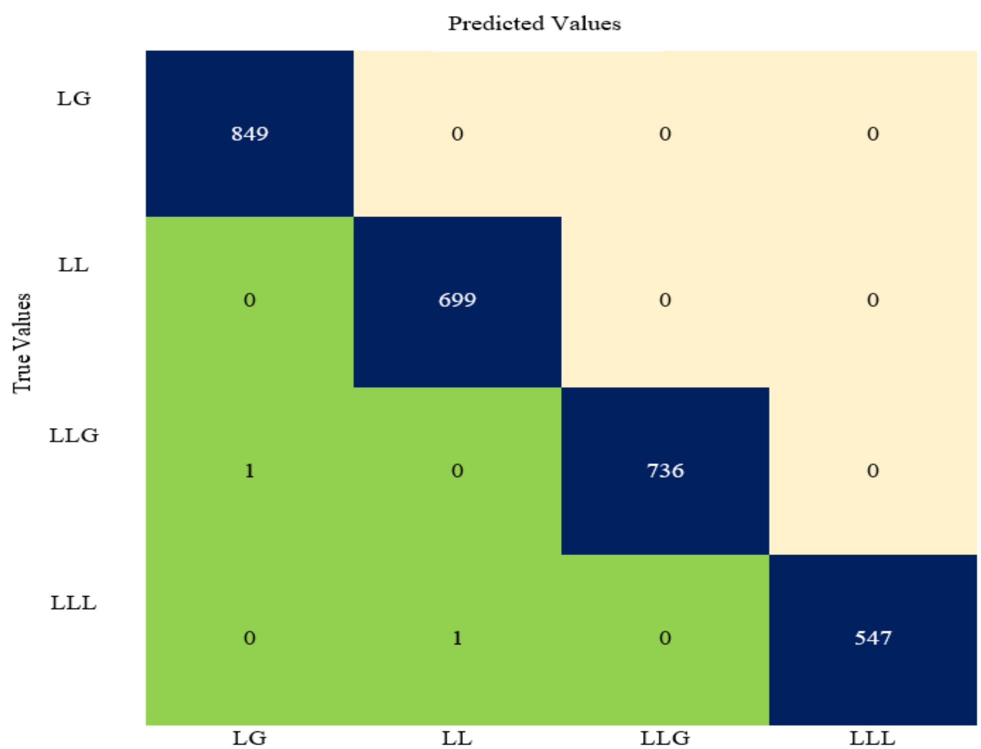


Figure 12: Confusion Matrix for fault classification using CNN-PMA in 330kV network

Table 7: Summary of Evaluation Standard for Fault Classification in 330kV network

		Accuracy	Precision	Recall	F-1 score	Mis-match	Count
CNN							
	LG	97.83	96.54	98.22	97.34	1	849
	LL	97.83	96.93	98.63	97.71	2	699
	LLG	97.83	98.23	98.93	98.57	3	737
	LLL	97.83	98.44	98.84	98.63	3	548
Average value		**97.83**	**97.54**	**98.66**	**98.06**		**2833**
CNN-MA							
	LG	98.80	99.54	99.26	98.94	0	849
	LL	98.72	99.34	98.57	98.94	1	699
	LLG	98.72	98.44	99.56	99.13	2	737
	LLL	98.72	98.63	98.56	98.86	2	548
Average value		**98.72**	**98.74**	**98.99**	**98.97**		**2833**
CNN-PMA							
	LG	99.96	99.73	100	99.92	0	849
	LL	99.96	100	99.86	99.93	0	699
	LLG	99.96	100	100	100	1	737
	LLL	99.96	100	100	100	1	548
Average value		**99.96**	**99.93**	**99.97**	**99.96**		**2833**

4.3 Comparison of CNN-PMA with other Methods

To show the superiority of CNN-PMA model, few existing methods for fault detection and classification were compared as shown in Table 8.: Amiruddin *et al*. (2018) and Leh *et al.* (2020) used ANN model to detect and classify electrical faults. Whereas Goni *et al.* (2023) used ELM model to detect and classify electrical fault in power system. Guo *et al*. (2019) and Moradzadeh (2022) employed HTT-CNN and CNN-LSTM models respectively for detection and classification of electrical faults. Their percentage accuracies when compared showed that CNN-PMA performed better in fault detection and classification than others. In addition, others had their number of layers between three and nine whereas CNN-PMA has one layer, this made it faster in operation when compared with others. Moreover, CNN-PMA has high learning rate as a result of its high batch size of 256 when compared with others.

Table 8: Comparison of CNN-PMA with other faults diagnosis models in transmission lines

Author(s)	Algorithm	Data (Training and Testing)	No. of Class Considered	No. of Layers	Accuracy (%)
Amiruddin *et al*. (2018)	ANN	Detection (190:41)	2	2	78
Fahim *et al*. (2019)	ANN	Detection (44) Classification (208)	3	3	84.40
Guo *et al*. (2019)	HTT-CNN	Detection (1672) Classification (1752)	2 10	6	99.92
Leh *et al*. (2020)	ANN	Not stated	11	3	70.00
Moradzadeh (2022)	CNN-LSTM	Not stated	11	9	98.60
Goni *et al*. (2023)	ELM	Detection (1000:4001) Classification (9909: 1102)	2 11	2	99.09
	CNN	Detection (8832: 7066) Classification (3541: 2833)	2 4	3	97.83
	CNN-MA	Detection (8832: 7066) Classification (3541: 2833)	2 4	1	98.72
Proposed	CNN-PMA	Detection (8832: 7066) Classification (3541: 2833)	2 4	1	99.96

5. CONCLUSION

This work has successfully carried out hybridization of MA and POA. PMA was developed by applying Pelican Exploration Model to model the attraction process as a deterministic process in order to assist the standard MA. Pelican Exploitation Model was applied to establish a balance between exploration and exploitation process in standard MA. The PMA was applied to detect the optimal hyper-parameters of CNN, such as: number of layers, filter size used in each convolutional layer, number of filters and the batch size. Developed CNN-PMA was simulated using deep learning and optimization tools boxes of MATLAB 2022a and in turn used to detect and classify electrical faults on SWN electrical network. The proposed model detected and classified electrical fault accurately and timely compared to standard CNN, CNN-MA and few other existing models. The results obtained were examined using MAPE, RMSE, CorrCoeff, PSNR, MSE, SNR and confusion matrix as performance metrics.

REFERENCES

Abedinia, O., Amjady, N. and Ghasemi, A. (2016). A New Metaheuristic Algorithm Based on Shark Smell Optimization. *Complexity*, **21**(5):97-116.

Afrasiabi, M., Mohammadi, M., Rastegar, M. and Kargarian, A. (2019). Probabilistic Deep Neural Network Price Forecasting Based on Residential Load and Wind Speed Predictions, *IET Renewable Power Generation*, **13**(11):1840-1848.

Allan, J. D., and Flecker, A. S. (1989). The Mating Biology of a Mass-Swarming Mayfly. *Animal Behavior*, ***37***, 361-374.

Amiruddin, A. A. A. M., Zabiri, H., Taqvi, S. A. A. and Tufa, L.D. (2020). Neural Network Applications in Fault Diagnosis and Detection: An Overview of Implementations in Engineering Related Systems. *Neural Computer Applications* **32** (2):447-472.

Anderson, J. G. (1991). Foraging Behavior of the American White Pelican (Pelecanusery-throrhyncos) in Western Nevada. *Colonial Water Birds,* **14**, 166-172.

Askarzadeh, A. (2016). A Novel Metaheuristic Method for Solving Constrained Engineering Optimization Problems: Crow Search Algorithm. *Computers and Structures*, **169**, 1-12.

Baykasoglu, A. and Akpinar, S. (2017). Weighted Superposition Attraction (WSA): A Swarm Intelligence Algorithm for Optimization Problems-Part1: Unconstrained Optimization. *Applied Soft Computing*, **56**, 520-540.

Bracale, A., Caramia, P., Carpinelli, G. and Fazio, A.R.D. (2017) Modeling the Three-phase Short Circuit Contribution of Photovoltaic Systems in Balanced Power Systems. *Electrical Power Energy Systems*. **93**, 204-215.

Bukhari, S. B. A., Kim, C., Mehmood, K. K., Haider, R. and Zaman, M. S. U. (2020). Convolutional Neural Network-Based Intelligent Protection Strategy for Micro Grids, *IET Generation Transmission. Distribution*, **14**(7):1177-1185.

Chen, K., Hu, J., and He, J. (2018). Detection and Classification of Transmission Line Faults Based on Unsupervised Feature Learning and Convolutional Sparse Auto Encoder. *IEEE Transaction. Smart Grid*, ***9***(3):1748-1758.

Dhiman, G. and Kumar, V. (2017). Spotted Hyena Optimizer: A Novel Bio-Inspired Based Metaheuristic Technique for Engineering Applications. *Advances in Engineering Software*, **114**, 48-70.

Fahim, S. R., Sarker, Y., Islam, O. K., Sarker, S. K., Ishraque, M. F. and Das, S. K. (2019). An Intelligent Approach of Fault Classification and Localization of a Power Transmission Line. *2019 IEEE International Conference Power, Electrical Electronic Industrial Applications PEEIACON*, 53-56

Fausto, F., Cuevas, E., Valdivia, A. and González, A. (2017). A Global Optimization Algorithm Inspired in the Behavior of Selfish Herds. *Biosystems*, **160,** 39-55.

Geem, Z. W. and Kim, J. H. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Research Gate*, **3,** 34-47.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. *Addison Wesley*. 1-6.

Goni, M. F., Nahiduzzaman, M., Anower, M.S., Rahman, M.M., Islam, M.R., Ahsan, M., Haider, J. and Shahjalal, M. (2023). Fast and Accurate Fault Detection and Classification in Transmission Lines using Extreme Learning Machine. *Advances in Electrical Engineering, Electronics and Energy*, **3**(100107):1-12.

Guo, M. F., Yang, N. C. and Chen, W. F. (2019). Deep Learning-based Fault Classification using Hilbert–Huang Transform and Convolutional Neural Network in Power Distribution

Systems. *IEEE Sensors*, **19**(16):6905-6913.

Hatata, A. Y., Essa, M. A. and Sedhom, B. E. (2022). Adaptive Protection Scheme for FREEDM Microgrid Based on Convolutional Neural Network and Gorilla Troops Optimization Technique. *IEEE Access*, **10**, 55583-55595.

Holland, J. H. (1960). Genetic Algorithms: Compare Programs that 'Evolve' in Ways the Resemble Natural Selection can Solve Complex Problems even their Creators do not fully understand. http//www.econ.lastate.edu/tesfatsi/Holland.G.AIntro. htm. 07/02/2023.

Husseinzadeh-Kashan, A., Tavakkoli-Moghaddam, R., and Gen, M. (2019). Find-Fix-Finish-xploit-Analyze (F3EA) Meta-heuristic Algorithm: An Effective Algorithm with New Evolutionary Operators for Global Optimization. *Computers and Industrial Engineering*, **128**, 192-218.

Illias, H. A., Chai, X. R., Abu Bakar, A. H., Mokhlis, H. (2015). Transformer Incipient Fault Prediction Using Combined Artificial Neural Network and Various Particle Swarm Optimisation Techniques. *Plosone*, **10**(6):1-16.

Jahani, E. and Chizari, M. (2018). Tackling Global Optimization Problems with a Novel Algorithm: Mouth Brooding Fish Algorithm. *Applied Soft Computing*, **62**, 987-1002.

Jing, L., Zhao, M., Li, P. and Xu, X. (2017). A Convolutional Neural Network-Based Feature Learning and Fault Diagnosis Method for the Condition Monitoring of Gearbox. *Measurement*, **111**, 1-10.

Kaveh, A., and Dadras, A. (2017). A Novel Meta-Heuristic Optimization Algorithm: Thermal Exchange Optimization. *Advances in Engineering Software*, **110**, 69-84.

Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimization. *Proceedings of ICNN'95*-International Conference on *Neural Networks*, 1942-1948.

Leh, N. A. M., Zain, F. M., Muhammad, Z., Abd Hamid, S. and Rosli, A. D. (2020). Fault Detection Method using ANN for Power Transmission Line, in: *2020 10th IEEE International Conference on Control System, Computing and Engineering* (ICCSCE), 79-84.

Li, M. D., Zhao, H., Weng, X. W., and Han, T. (2016). A Novel Nature-Inspired Algorithm for Optimization: Virus Colony Search. *Advances in Engineering Software*, **92**, 65-88.

Lu, J., Ye, Y., Xu, X. and Li, Q. (2019). Application Research of Convolution Neural Network in Image Classification of Icing Monitoring in Power Grid. *EURASIP Journal on Image and Video Processing*, **49**, 1-11.

Marchant, S. (1990). *Handbook of Australian*, New Zealand and Antarctic Birds: Australian Pelican to Ducks; Oxford University Press: Melbourne, Australia.

Mehrabian, A. R. and Lucas, C. (2006). A Novel Numerical Optimization Algorithm Inspired from Weed Colonization. *Ecological Informatics*, **1**(4):355–366.

Mirjalili, S. and Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**(C):51-67.

Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., and Mirjalili, S. M. (2017). Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. *Advances in Engineering Software*, **114**, 163-191.

Mirjalili, S., Mirjalili, S. M., and Hatamlou, A. (2016). Multi-Verse Optimizer: A Nature Inspired Algorithm for Global Optimization. *Neural Computing and Applications*, **27**(2):495-513.

Moradzadeh, A., Teimourzadeh, H., Mohammadi-Ivatloo, B. and Pourhossein, K. (2022). Hybrid CNN-LSTM Approaches for Identification of Type and Locations of Transmission Line Faults, *International Journal of Electrical Power Energy System*, 135(107563):117-131.

Mozo, A., Ordozgoiti, B. and GoÂmez-Canaval, S. (2018). Forecasting Short-Term Data Center Network Traffic Load with Convolutional Neural Networks. *PLOS ONE*, **13**(2):1-31.

Nematollahi, A.F., Rahiminejad, A. and Vahidi, B. (2017). A Novel Physical Based Meta-Heuristic Optimization Method known as Lightning Attachment Procedure Optimization. *Applied Soft Computing Journal*, **59,** 596-621.

Ogundoyin, S. O. and Kamil, I. A. (2021). Optimization Techniques and Applications in Fog Computing: An Exhaustive Survey. *Elsevier: Swarm and Evolutionary Computation*, **66**(100937):1-55.

Pakzad-Moghaddam, S. H., Mina, H., and Mostafazadeh, P. (2019). A Novel Optimization Booster Algorithm. *Computers and Industrial Engineering*, **136**, 591-613.

Pan, C., Lu, M., Biao Xu, B. and Gao, H. (2019). An Improved CNN Model for Within Project Software Defect Prediction. *Applied Sciences.* **9**(2138):1-27.

Peckarsky, B. L., McIntosh, A. R., Caudill, C. C. and Dahl, J. (2002). Swarming and Mating Behavior of a Mayfly Baetisbicaudatus suggest Stabilizing Selection for Male Body Size. *Behavioral Ecology and Sociobiology*, **51**(6):530-537.

Perrins, C. M., Middleton, A. L. (1985). *The Encyclopedia of Birds*; Guild Publishing: London, UK, 53-54.

Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S. and Zaidi, M. (2006). The Bees Algorithm-A Novel Tool for Complex Optimization Problems. Intelligent Production Machines and Systems. *Proceedings of 2nd IPROMS Virtual International Conference* 3-14 July 2006, 454-459.

Qi, X., Zhu, Y. and Zhang, H. (2017). A New Meta-Heuristic Butterfly-Inspired Algorithm. *Journal of Computational Science*, **23**, 226-239.

Samet, H., Ketabipour, S., Afrasiabi, S., Afrasiabi, M. and Mohammadi, M. (2021). Prediction of Wind Farm Reactive Power Fast Variations by Adaptive One-Dimensional Convolutional Neural Network, *Computer Electronics Engineering*, **96**(107480):1-16.

Saremi, S., Mirjalili, S. and Lewis, A. (2017). Grasshopper Optimization Algorithm: Theory and Application. *Advances in Engineering Software*, **105**, 30-47.

Spieth, H. T. (1940). Studies on the Biology of the Ephemeroptera II. The Nuptial Flight. *Journal of the New York Entomological Society*, **48**(4):379-390.

Storn, R., and Price, K. (1997). Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, **11**, 341-359.

Tabari, A. and Ahmad, A. (2017). A New Optimization Method: Electro-Search Algorithm. *Computers and Chemical Engineering*, **103**, 1-11.

Tang, S., Yuan, S. and Zhu, Y. (2020). Data Preprocessing Techniques in Convolutional Neural Network Based on Fault Diagnosis Towards Rotating Machinery. *IEEE Access*, **8**, 149487-149496.

Trojovský, P. and Dehghani, M. (2022). Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications. *Sensors*, **22**(855):1-31.

Uymaz, S. A., Tezel, G., and Yel, E. (2015). Artificial Algae Algorithm (AAA) for Non-linear Global Optimization. *Applied Soft Computing*, **31**, 153-171.

Wolpert, D. H. and Macready, W.G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions Evolutionary Computation.***1**, 67-82.

Yang, X. S. (2008). *Nature-inspired Metaheuristic Algorithms*. Luniver Press. 45-53.

Yang, X. S. and Karamanoglu, M. (2016). Swarm Intelligence and Bio-Inspired Computation. *Elsevier*, 41-58.

Yang, X. S. (2009). Firefly Algorithms for Multimodal Optimization. *Springer (Edition: Stochastic Algorithms Foundations and Applications*, 169-178.

Yang, X. S. and He, X. (2013). Firefly Algorithm: Recent Advances and Applications. *International Journal of Swarm Intelligence*, **1**, 36-50.

Yong, W., Tao, W., Cheng-Zhi, Z. and Hua-Juan, H. (2016). A New Stochastic Optimization Approach-Dolphin Swarm Optimization Algorithm. *International Journal of Computational Intelligence and Applications*, **15**(2):16500-16511.

Zervoudakis, K. and Tsafarakis, S. A. (2020). Mayfly Optimization Algorithm. *Computers and Industrial Engineering*, 1-80.

Zhao, C., Huang, X., Li, Y. and YousafIqbal, M. (2020). A Double-Channel Hybrid Deep Neural Network Based on CNN and BiLSTM for Remaining Useful Life Prediction. *Sensors*, **20**(7109)1-15.