An Overview of ANN based MPPT and an Example

Mehdi Rezaee*, Yusuf Gürcan Şahin**

* Master Student, Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Avcilar, Istanbul, Türkiye. ORCID: 0009-0001-4681-1170

** Asst. Prof. Dr., Department of Electrical and Electronics Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Avcilar, Istanbul, Türkiye. ORCID: 0000-0001-9640-1623

(mehdi.rezaee@ogr.gelisim.edu.tr, ygsahin@gelisim.edu.tr)

Corresponding Author, Istanbul Gelisim University, Avcilar, Istanbul, Türkiye. Tel: 0212 422 70 00, ygsahin@gelisim.edu.tr

Received: 09.06.2025 Accepted: 07.07.2025

Abstract - The study presents an overview and a simulation of maximum power point tracking (MPPT) for Photovoltaic (PV) systems that uses an artificial neural network (ANN) controller as proof of concept. Solar energy must be harvested with high efficiency as the world turns to renewables. The usual Perturb and Observe (P&O) and Incremental (InC) method loses power by oscillating around the Maximum Power Point (MPP) and reacts slowly to sudden weather changes. The work therefore tests an ANN as a better choice. The authors survey earlier ANN MPPT studies that cover many network types, training schemes and mixed strategies. They then build a MATLAB/Simulink model that runs an ANN controller and a P&O controller on the same PV array. The ANN learns from Istanbul 2020 weather data. The results show the ANN reaches 252 W and 87.9% of efficiency while P&O reaches 241 W and 84.26% of efficiency, and InC reaches 245 W and 78.1% of efficiency. The ANN also tracks the MPP faster and with steadier behaviour when irradiance varies. These outcomes confirm that ANN MPPT can raise the energy output of PV systems.

Keywords- Artificial Neural Network (ANN), Maximum Power Point Tracking (MPPT), Photovoltaics, Renewable energy

1. Introduction

Public concern for environmental stewardship has intensified owing to the ecological damage and finite nature of fossil fuel reserves. Consequently, researchers and policymakers are actively exploring alternative renewable energy technologies. Solar energy has attracted particular interest due to its accelerating global deployment. Solar energy comes from PV systems that use the photovoltaic effect. PV systems give clean and sustainable energy. They work well because sunlight is everywhere. PV modules are relatively straightforward to deploy in a modular fashion. PV systems are suitable for both grid-connected residential settings and off grid rural installations. PV technology already powers railway stations, standalone street lighting units and auxiliary vehicle loads. Urban architects integrate PV modules into rooftops, façades and glazing systems. The electricity from the panels can feed into the existing grid. There are two main types of systems, standalone and grid connected. Standalone systems suit dis-tant places where building a normal power plant is hard. Module output varies with meteorological conditions and attains its maximum at a unique IV operating point. Directly coupling a PV module to a load is therefore suboptimal. Engineers address this limitation by employing MPPT techniques to maximise energy extraction. An MPPT algorithm finds the point where the source gives its highest power. MPPT controllers often have simple design, low cost, small power swings, good work and quick response in changing conditions. Many MPPT methods exist, such as P&O, incremental conductance (Inc-Cond.), hill climbing (HC), neural network (NN), fuzzy logic and genetic search. P&O and Inc-Cond. remain popular owing to their computational simplicity and low hardware overhead. In P&O, the system measures power before and after a change and chooses the next change from the result. However, its tracking accuracy deteriorates under rapidly varying irradiance. So, designers try fuzzy logic and multilayer NN to gain better accuracy. The raw DC voltage produced by an individual PV cell typically requires step up conversion for practical utilisation. A boost converter sits between the panel and the load to raise the voltage. Simulation environments such as MATLAB/Simulink facilitate pre-deployment optimisation of the MPP, enabling informed design decisions.

2. Research Problem

Classical iterative algorithms oscillate about the MPP, incurring energy losses and exhibiting sluggish convergence. A trained ANN reads the panel voltage and current and, if given irradiance and temperature, gives the duty cycle for the best voltage. This network tracks with more than 98% success and almost no oscillation in a short time. When sunlight changes quickly in 100 to 300 milliseconds during cloud moves, classical methods fail because their step size is wrong. Because the network has seen these fast changes in training, it estimates the point in real time and keeps power loss low. Partial shading introduces multiple local maxima into the PV characteristic. Iterative methods can stop at the first peak they meet. A multilayer or hybrid NN learns the many peaks and chooses the highest one. Classical methods need model values like series resistance, shunt resistance, ideality factor, photo current or the slope of power with voltage. The NN works only with data, so even dirt or aging causes little error. A network that uses panel voltage and temperature can guess current and find the point, so costly current sensors are not needed. Adding new inputs such as cell temperature or dust makes classical code change a lot. In the NN case, you just feed the new inputs to the same code. After training, the network holds about ten to twenty weights and fits easily in a 32-bit microcontroller that runs fixed point multiply accumulate steps. Tests show the system answers in 0.02 seconds and keeps inverter distortion below 1.5%. The NN still needs enough and good data to learn well. If the data are poor, the network makes wrong guesses. Without online retraining, you must keep the data set up to date. Even with these limits, the neural approach beats P&O, Inc-Cond. and lone meta heuristic methods. It gives faster speed, better stability, the true global peak and lower cost, so it lifts the energy output of the solar system.

3. Literature Review

Md Tahmid Hussain et al. [1] benchmarks six ANN training algorithms for PV MPPT. The comparison is timely, heavy reliance on synthetic data limits real world transferability. Using a two-layer feed forward ANN (FFANN) with 20 hidden neurons, the authors train Levenberg–Marquardt (LM), Bayesian Regularisation (BR), resilient back propagation (RP), scaled conjugate gradient (SCG), gradient-descent momentum (GDM) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton optimisers on 1 000 Monte-Carlo G-T points (0–1 000 W m^{-2}, 15–35 °C) split 80/10/10 for training, validation and testing. Performance is rated via regression R, MSE, gradient norm and epochs. LM and BFGS achieve R = 1 with MSE 2.08×10^{-10} at 1 000 epochs and 9.98×10^{-17} at 10 epochs, respectively. BR and RP remain competitive (1.58×10^{-7} and 2.87×10^{-6}), whereas SCG and GDM lag (5.30×10^{-2} and 0.154). The authors recommend LM and BFGS for ANN MPPT and propose hybrid AI extensions. Provided that the research question - finding the quickest accurate optimiser is explicit, the six-way comparison is valuable. Yet generality is overstated: metrics derive from a single random seed, so order of magnitude gaps between LM and BFGS could disappear with repeated runs. Ideal MATLAB models omit sensor noise and converter losses; thus, BFGS's stellar 9.98×10^{-17} MSE after ten epochs may not persist in hardware. Still, the common network architecture ensures fair treatment, and contrasting gradient norms (GDM 11.485 vs. BFGS 1.02×10^{-7}) illustrates convergence trade-offs.

Ž. Zečević and M. Rolevski [2] propose a gradient based MPPT algorithm that exploits an identified feed forward neural network (FFNN) model of a PV module. By analytically differentiating the NN current equation, the authors derive an iterative voltage update and a two variant irradiance estimator of very low computational cost. The paper convincingly shows that model-based gradients can outperform classical heuristics at modest complexity. After reviewing single diode and NN PV modelling, the paper trains 4 neurons, one hidden layer NN to map irradiance (G), temperature (T) and voltage (V) to current (I). The analytic gradient of the NN output enables a Newton like MPPT rule with step size μ. Because G is usually unmeasured, two estimators are proposed: (i) an immersion and invariance (NI&I) integrator updated with the current residual, and (ii) a faster neural gradient estimator (NGE) using $\partial\hat{I}/\partial G$. Each MPPT iteration needs ≈5M multi-plications (M = neurons). Simulations on an 80 W module and on the NREL HIT05662 dataset compare the new method (NMPPT) with P&O, a Lambert W based model method and a cascaded two NN tracker. With two gradient steps per sample, NMPPT converges within eight iterations and keeps steady state power error <0.1 %, whereas P&O needs ≈75 iterations and oscillates ±2 %. Under ramped G/T profiles, NMPPT outperforms alternatives in both tracking speed and accuracy while consuming fewer arithmetic operations. The dual innovation of using the NN's derivative and embedding G estimation inside the same model is elegant. Training sets are synthetically generated from circuit equations and, in a second study, drawn from the NREL IV repository. However, hyperparameter tuning, data stratification and overfitting checks are only briefly mentioned. Converter dynamics are idealised; current loop and sampling delays are ignored, so closed loop stability is not guaranteed when ported to microcontrollers. Results focus on relative power error and convergence counts, omitting IEC EN 50530 dynamic efficiency, 24-h energy yield or computational latency on fixed-point cores. NI&I and NGE beat two existing irradiance estimators. Zečević and Rolevski deliver a neat, analytically grounded MPPT that achieves sub-0.1 % steady state error with only a handful of arithmetic operations which is an attractive option for low power converters.

S. D. Al-Majidi et al. [3] present an ANN MPPT scheme trained with one year of field data from a 925 W rooftop array at Brunel University, UK. ANN trackers have long promised faster convergence and lower ripple than P&O or InC methods, but many studies rely on synthetic data or small experiments. The paper's major merit is its large, real-world training set; however, its validation remains simulation only, and the performance gains over a well-tuned P&O con-troller are marginal. The authors collect 48500 five-minute samples of G, module T, and array power at the MPP. These data, spanning an entire year, form a 70%/15%/15% train / validate / test split for a feed forward multilayer perceptron with two inputs (G, T), one hidden layer of 10 neurones, and one output P_{MPP}. MSE after training falls to 7.9×10^{-3} with correlation R≈1. The ANN is embedded in a MATLAB/Simulink model that couples a five module Sharp NU-S5E3 PV array, a boost converter, and a PI controlled PWM stage. Under the rapid irradiance ramp (1000→200 W m^{-2} in 1 s, then back to 1000 W m^{-2})

at 25 °C, the ANN reaches the new MPP in 0.06 s versus 0.12 s for P&O. Steady state oscillation is visibly lower, and average output power rises from 922.50 W (P&O) to 923.25 W (ANN). The authors claim the ANN avoids the drift phenomenon that plagues P&O during rising irradiance. Large field datasets are uncommon, so the work fills a gap. Al-Majidi et al. demonstrate that a modest ANN trained on abundant real data can outpace classical P&O in simulation and mitigate drift during rapid irradiance changes. Strengths are the comprehensive dataset and clear comparative study.

L. M. Elobaid et al. [4] reviews three decades of research on ANN methods for MPPT in PV systems and proposes two taxonomies based on input variables and controller architect-ture. While the article offers a valuable, structured map of early ANN MPPT work. The authors first motivate ANN approaches as a remedy for the slow convergence and shading sensitivity of classical P&O or InC trackers. They then classify 38 ANN MPPT techniques into three input categories; electrical, non-electrical, and mixed; and two structural categories; stand-alone versus hybrid (combined with conventional or other AI controllers). Each paper is tabulated for network size, required sensors, training sets, experimental validation, converter type, power level, and dynamic / steady state results. Key observations include: (i) most early de-signs used feed forward multilayer perceptron trained off-line; (ii) hybrids with fuzzy logic or genetic algorithms (GA) tended to outperform stand-alone ANNs under partial shading; and (iii) sensor count and controller complexity scale roughly with tracking accuracy. The survey concludes that ANN trackers can achieve ≥98 % tracking efficiency, but stresses the need for standardized test protocols and real time hardware evaluations. Elobaid et al. offer a map of ANN MPPT research up to 2014, giving engineers a convenient taxonomy and extensive bibliographic entry point. Strengths are the dual classification scheme and comprehensive tabulation; A refreshed; data driven survey building on their framework would significantly benefit the rapidly evolving PV-MPPT community.

O. Veligorskyi, R. Chakirov, and Y. Vagapov [5] combined an ANN voltage predictor with the traditional P&O cycle, targeting the known limitations of the P&O method under partial shading. The paper convincingly demonstrates that the hybrid system in MATLAB can find global MPP faster than the classical P&O method. Three poly-Si modules (each with 7.2 V, 0.275 A) were connected in series and shaded according to two scenarios: moving cloud and fixed obstacle. Using these curves, the authors created a 4-8-7-7-1 feedforward ANN trained with Bayesian regularised back propagation on 10.510 input-output pairs and validated on 2.935 pairs. Prediction errors remain within ±0.01 V in 99.95% of cases and within ±0.2 V in 99.95% of cases. When deployed on a P&O converter, the ANN increases batch yield by approximately 70% during fixed partial shading and reduces fluctuations during dynamic shading; gains are smaller but positive during fast linear shadow expansion. Given that most global MPP studies still rely on metaheuristic methods, the data driven ANN layer is innovative and pragmatically appealing. However, the claim of 'significant' improvement would be more meaningful if supported by quantitative transitions (ms) and energy efficiency percentages. The network is reasonably compact, and the Bayesian regularisation prevents overfitting; however, the study is based on a single training run, and the voltage error of 0.94% is not accompanied by a confidence interval. The buck converter is ideal. Only one light intensity orbit was tested for each scenario. The figures show the P-V curves, error histograms, and shadow profiles. Subject to evaluation as proof of concept, the paper adds a compact ANN layer that shifts P&O away from local maximum values, reducing voltage error to ±0.2 V and increasing simulated energy under constant shading. Its strengths include a realistic training set and an open network topology.

J. C. Lima et al. [6] integrates a low-cost PIC microcontroller–driven NN MPPT with a single-stage boost converter and a three-phase inverter realised in an FPGA. The work offers an early proof-of-concept showing that ANN-assisted MPPT can be embedded in inexpensive hardware, yet its experimental scope and quantitative evidence remain limited. The authors model a 2-4-1 FFANN that maps a reference-cell open-circuit voltage (V_{oc}) and temperature T to the desired boost-converter output voltage V_{pm} . Training is performed off-line in MATLAB using back propagation with learning rate 0.2 and momentum 0.9 until the MSE falls below 5×10^{-3}; the convergence curve is shown in Fig. 8. The trained weights are ported to a PIC16 microcontroller that adjusts the duty cycle of a DCDC boost converter. DC output then feeds an FPGA-controlled three-phase inverter that synthesises an 8-bit PWM sinusoid at 24 kHz, Fig. 9. Field data of V_{oc} and V_{pm} collected on 30 April 1998, Fig. 7, underpin the training set. No direct efficiency metrics are reported. Provided that most late-1990s MPPT prototypes relied on look-up tables or DSPs, demonstrating an ANN on a PIC plus a low-cost FPGA inverter is timely. However, the claim that the neural tracker "always obtains the maximum power" lacks quantified energy-yield or tracking-time evidence. Strengths include real irradiance measurements and a clearly documented network topology. The boost converter and inverter are treated as ideal. Figures illustrate training convergence and inverter waveforms. The paper pioneers a cost-conscious PIC/FPGA platform for ANN MPPT and grid interface. Its primary strength is demonstrating feasibility with inexpensive hardware.

Messalti et al. [7] claim that by combining P&O logic with a three-layer FFANN, they have achieved faster and more accurate tracking under variable light intensity. The paper presents a clear design workflow and promising simulation results. The authors establish an offline/online framework: in offline mode, different network topologies are tested; the 2-4-10-4-1 network (logsig / purplelin activations) with the best performance is retained for online MPPT. Inputs are power and voltage derivatives (dP, dV), and the ANN extracts a task cycle increase signal derived from a P&O rule set (Table I). Matlab/Simulink tests conducted with the Solarex MSX-60 panel demonstrate that the ANN power traces align with the theoretical MPP curves for both constant (1000 W m^{-2}) and stepwise varying light intensity (1000→600→800 W m^{-2}). Among the reported dynamic metrics are a settling time of 0.035 seconds and an overshoot of 3 W when irradiance decreases. The use of dP/dV as input eliminates the need for additional sensors; however, the training set is synthetically generated from the same P&O algorithm that the ANN targets for overshoot, which introduces a potential circular dependency that could highlight weaknesses in the P&O algorithm. The figures validate the qualitative tracking. The study presents a concise ANN-P&O synthesis that reduces the response time to

0.035 seconds and limits overshoot to 3 W in simulation. Considering synthetic training and single-scenario validation, this approach could provide insights for low cost MPPT software.

M.-F. Tsai et al. [8] situates themselves in the continuing search for fast, low-oscillation MPPT schemes by embedding a six-neuron recurrent NN (RNN) and a GA tuned PI current loop on a TI TMS320F28335 DSP. The study offers a neat, hardware-demonstrated blend of neural compensation and classical control. The authors reformulate MPPT as a current-control problem. A two-input RNN (PV current I_{pv}, PV voltage V_{pv}) estimates an uncertainty term Δx that offsets nonlinear, temperature-dependent PV dynamics; the network contains only three hidden neurons and updates every 10 ms. A PI regulator, whose gains ($kp \approx 1, ki \approx 125$) are optimised off-line via a MATLAB genetic algorithm, drives the boost-converter duty cycle. MATLAB/Simulink tests on a 310 W array show rapid convergence (<0.2 s) under 200 → 1000 W m^{-2} irradiance steps and sinusoidal 25 ± 20 °C swings, with the tracking error decaying to zero. A laboratory prototype using two 155 W modules delivers 267 W at 64.5 V and 4.14 A, and retains regulation when one panel is briefly shaded. Compared with baseline P&O or InC algorithms the authors cite, the proposed scheme exhibits smaller ripples and faster settling, though no side-by-side metrics are given. The paper asks whether a minimalist RNN plus PI can outperform perturbative MPPT without irradiance or temperature sensors. The formulation is clear and the DSP implementation commendable. Using only V_{pv} and I_{pv} as network inputs simplifies wiring. The GA search space is bounded analytically, but the GA settings (population, crossover, mutation) are chosen heuristically. All training data are acquired online from the actual plant. Simulation plots demonstrate qualitative convergence. Hardware validation uses a 267 W bench; scaling to kW-class strings is merely simulated. Tsai et al. showcase a compact, DSP friendly neural compensated MPPT that achieves smooth tracking and partial-shade resilience with only two sensors. Strengths are the elegant control architecture, clear derivations, and hardware proof-of-concept.

M. Yilmaz, R. Celikel, and A. Gundogdu [9] propose an ANN algorithm that generates an adaptive reference voltage (ARV) for MPPT and regulates it with a particle-swarm-optimised (PSO) PI controller equipped with a clamping anti-windup loop. The study targets the well-known trade-off between fast tracking and low steady state oscillation, especially under EN 50530 dynamic-test profiles. While the paper offers a neat integration of sensor-light ANN estimation and anti-windup PI control. The 10-kW test system in MATLAB/Simulink comprises a boost converter, a single diode PV model, and three EN 50530-based irradiance scenarios plus a custom fast-ramp case. A 2-10-1 feed forward ANN uses only temperature and PV voltage to predict the reference voltage V_{ref} eliminating current sensors. The error $e = V_{ref} - V_{pv}$ feeds a PI controller whose gains are tuned via PSO; a clamping anti-windup block freezes the integrator when the duty cycle D reaches the 0–1 limits. Performance is benchmarked against conventional P&O and InC algorithms and against the same ANN with a classical PI. Reported dynamic efficiencies are 99.4 % (high irradiance), 95.9 % (medium), and 96 % (fast-ramp) versus 99 %, 85.8 %, and 68 % for P&O and still lower for INC. Voltage and power plots show visibly reduced ripple and quicker settling with the proposed controller. Converter parameters, ANN topology, PSO settings, and EN 50530 waveforms are fully disclosed. The ANN is trained on 350 000 synthetic samples originating from the same single diode equations used for testing. Efficiency figures are computed once per scenario. Yilmaz et al. demonstrate that coupling a lightweight ANN ARV estimator with a clamping anti-windup PI can outperform classical P&O and InC algorithms under dynamic-test standards. Strengths include sensor reduction, simple control logic, and consistent simulated gains.

J. Morgoš, P. Klčo et al. [10] proposes a FFNN controller that maximises the energy harvested by rooftop PV panels dedicated to charging an electric vehicle battery. Within the fast-growing literature on AI assisted MPPT, the paper's novelty lies in tailoring the algorithm to a household-scale PV/EV architecture and eliminating irradiance sensors to cut hardware cost. The study shows encouraging simulation accuracy by purely offline validation. The authors first model an 8-string array of 125 W CIS panels and a unidirectional DCDC converter that charges an intermediate home-storage battery before feeding the EV charger. An FFNN with three inputs (PV current, voltage, temperature) and one hidden layer of 25 neurons outputs estimates of I_{mpp}, V_{mpp}, and P_{mpp}. Training data (≈200 000 samples) are synthetically generated in MATLAB/Simulink by sweeping 100 irradiance levels (10-1000 W m^{-2}) and 96 temperature points (–20 °C to 75 °C) across the single diode PV model. The LM algorithm trains the network until MSE < 1 × 10^{-4}; ten random restarts yield MSE values between 0.015 and 0.10 with R ≈ 1.0. When driven by satellite-derived weather data for 12 Sep 2020, the NN tracker maintains the predicted power within 0.5 % of the analytical MPP all day and <0.1 % for most daylight hours. Response time is reported as "limited chiefly by input-sensor speed,". The synthetic dataset covers a wide G/T space. Regression plots and day-long power traces support high correlation. The method is benchmarked only against the analytical MPP, not against conventional P&O, InC. Morgoš et al. provide a concise proof-of-concept that an inexpensive FFNN can approximate the PV MPP without irradiance sensors and track rapid irradiance changes with <0.5 % error in simulation. Strengths include a clear household-EV use case and a large synthetic training set.

U. Younas et al. [11] propose a long short-term memory (LSTM)–driven (MPPT) algorithm for a high power (100 kW) grid connected PV plant. Within the accelerating shift from heuristic or shallow-learning MPPT to data driven controllers, the paper argues that a two-layer stacked-LSTM can capture complex irradiance/temperature dynamics with lower complexity than deep feed forward networks or reinforcement-learning schemes. The work convincingly demonstrates simulation-level efficiency gains. The authors collect a one-million-sample dataset of irradiance G (0–1000 W m^{-2}), cell temperature T (20–80 °C) and MPP voltage V_{mp} using empirically derived equations. Data are z score normalized, visualized, and split 80/20. A stacked LSTM with 64 and 32 units, tanh activation, 20 % dropout, Adam optimiser ($\eta = 0.05$), 32-sample batches and 50 epochs are trained to regress V_{mp}. Reported metrics are MSE = 2.3 × 10^{-3}, RMSE = 0.048, R^2 = 0.998 on test data. In a MATLAB/Simulink dual-stage topology the LSTM drives a 5 kHz boost converter, regulates an 800 V DC link, and is compared with P&O and a two-layer FFANN (64-

32 neurons). Under nine composite G/T scenarios the LSTM harvests 98.2 kW from a 100-kW array, versus 96.1 kW (DNN) and 94.3 kW (P&O); 98.4 kW is delivered to the grid through a three-level NPC inverter controlled by PI loops. Positioning against recent LSTM or CNN MPPT studies is adequate. One previous bidirectional LSTM with 600 neurons delivered similar gains on a 230 W system (cited by the authors). The main contribution is scaling LSTM control to 100 kW while keeping only 96 neurons. Using synthetic equations to generate one million points ensures broad coverage but risks bias if empirical constants differ from real modules or shading conditions. Hyperparameters are chosen by manual sweep; automated search or ablation could strengthen optimisation claims. All results are open loop simulations. Performance metrics (MSE, R^2) are strong. Younas et al. advance LSTM based MPPT by scaling to a 100-kW dual-stage topology and demonstrating simulation efficiency above 98 %. Strengths include a clear data pipeline, modest network size, and integration into full inverter control.

M. T. Makhloufi et al. [12] targets the longstanding problem of extracting maximum power from PV arrays when irradiance and temperature vary, proposing an online multilayer-perceptron (MLP) tracker coupled to a modified Cúk converter. The study demonstrates that the neural tracker improves simulated efficiency and transient response., Its evidence base is confined to MATLAB/Simulink. The authors (i) model a 150 W PV panel, an energy storage battery, and a modified Cúk converter, (ii) design an MLP (inputs: PV power P_{pv}; hidden layers: 5-2 neurons; output: duty cycle D) trained online with back propagation, and (iii) blend the network with P&O logic P&O governs under slow irradiance change, while the MLP takes over during fast transients. Three simulation campaigns are presented: standard test conditions (1 000 W m^{-2}, 25 °C), two abrupt irradiance steps (±1 000 W m^{-2}), and a fast multi-step irradiance ramp. In all cases the neural tracker achieves 97.5 % energy efficiency versus <95 % for earlier ANN MPPT reports, reaches the new MPP within 10–30 ms, and keeps battery state of charge (SOC) within tight limits. Graphs of IV, PV, converter waveforms, and SOC trajectories substantiate these claims. Waveforms convincingly show reduced oscillation. Makhloufi et al. deliver a clear proof-of-concept that an on-line MLP combined with P&O can raise MPPT efficiency and speed in simulation. Strengths include a well explained converter model and insightful disturbance tests.

J. J. Khanam and S. Y. Foo [13] present a Simulink based PV model and compares P&O MPPT with an ANN tracker. The paper offers a clear didactic walkthrough and shows that an ANN can outperform classical P&O in simulation. The authors first derive a single diode 60 W PV model (MSX-60) and verify its IV/PV curves against datasheet values under varied irradiance and temperature. A standard P&O algorithm, implemented with a buck boost converter, reaches 53.94 W at 14.33 V in 0.072 min, versus 37.31 W without MPPT. The proposed ANN uses two inputs (G, T), one hidden layer, and two outputs (V_{mp}, P_{mp}). From 104 simulated operating points, 94 train the network (LM); 10 are held out for testing. After 1000 epochs the mean square error falls to 6.8×10^{-6} and regression plots approach unity. On the unseen test set, ANN predictions of P_{mp} and V_{mp} differ from Simulink ground truth by <1 % (e.g., 51.18 W vs 51.18 W at 1000 W m^{-2}, 300 K). Given many hybrid MPPT studies, the paper's value lies mainly in its pedagogical modelling steps rather than algorithmic novelty. Strengths include transparent equations, clear Simulink blocks and explicit neuron counts. Nevertheless, the training data are synthetically generated by the very model used for testing. Only one hidden layer size is tried. Converter losses, sensor noise and microcontroller latency are ignored. Figures document training curves and test scatter. The comparison with P&O uses a single irradiance step; partial shading, temperature ramps or real outdoor traces are absent. Provided that its findings are interpreted as proof-of-concept, the paper shows an ANN can approximate the simulated MPP within 1 % and suggests a quicker convergence than P&O. Its chief strengths are pedagogical clarity and reproducible Simulink models.

In conclusion; standard back propagation updates the weights until the predicted MPP matches the actual one [1,5,6,7]. Large, well-prepared datasets help the network to learn and predict more quickly and accurately. Adaptive learning rates and momentum speed up convergence without overfitting. Selecting appropriate inputs, such as voltage, current, irradiance and temperature, is crucial for accurate MPP estimation [5,13]. Filtering, scaling and other cleaning steps reduce noise and maintain model stability. Small feed forward networks keep real time processing demands low. Quantisation and pruning reduce memory usage while maintaining accuracy [5,6]. Early work involved running ANN MPPT on microcontrollers, DSPs, and field programmable gate arrays (FPGAs) to demonstrate real time control [6,8]. Deeper networks require a joint hardware-software design with fast links and robust signal paths. Research has progressed from simple feed forward networks to recurrent and deep networks that can recall past data [2,5,11,12]. Lima et al. adjusted network size and training to reduce computation and prevent steady oscillations [1,5,6,7]. New designs save energy and resist sudden changes [8,1]. RNN and LSTM cells is used to track rapid grid changes [11,2]. Deep learning can handle long patterns and requires less retraining than older ANNs. Training with real PV data enables the model to predict the MPP under many conditions [3,13]. FFANNs on Cúk and boost converters quickly reach the MPP with minimal ripple [1,5,12,18]. ANN trackers adapt quickly to changing weather conditions and maintain high power output [1,14]. Hybrid ANN-PI controllers prevent wind-up and improve the response [9,10]. Filtering and normalising inputs helps ANNs to handle noisy sensors [2,16]. Adaptive learning enables ANNs to remain on the MPP when conditions change rapidly [1,17].

PV models help engineers predict the current and voltage of solar cells and modules, plan MPPT, and estimate energy yield for grids and micro-grids [1]. Accurate models cut the cost of solar power because they let designers size converters, batteries, and cooling systems with less error [1].

4. Solar Cell Model

In 1980 Rauschenbach gave the first circuit model of a solar cell. The model joins a current source with one or more diodes to copy the PN junction in the device and shows how the cell behaves in many conditions [14]. The idea stays important because it set the base for all later work in solar modelling. Researchers still start from his frame to make single diode and two diode models that fit real cells better [15,16]. Modern software and hardware for PV systems still use the same core ideas

[17]. A PV cell can be shown by an equivalent circuit in Fig. (1). The circuit has a light generated current source, one diode, and a series resistance. A small parallel resistance stands for leakage that is tiny in one module. The current source marks the photons that create charge. The output of the model is found at fixed temperature and fixed sunlight.

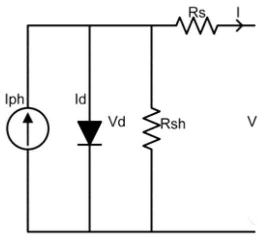

Fig. 1. Equivalent Solar Cell Circuit

Single diode model terminal current is given as in Eq. (1)

$$I = I_{\text{ph}} - I_0\left(e^{\frac{q\,(V+I\,R_s)}{n\,k\,T}} - 1\right) - \frac{V + I\,R_s}{R_{sh}} \tag{1}$$

Light generated current given as in Eq. (2).

$$I_{\text{ph}} = \left(G/G_{ref}\right)\left[I_{\text{sc,ref}} + \alpha_I\,(T - T_{\text{ref}})\right] \tag{2}$$

Diode saturation current given as in Eq. (3).

$$I_0 = I_{0,\text{ref}}(T/T_{ref})^3 exp\left\{\frac{q\,E_g}{n\,k}\left[(1/T_{ref}) - (1/T)\right]\right\} \tag{3}$$

Thermal voltage per cell given as in Eq. (4).

$$V_T = k\,T/q \tag{4}$$

Module voltage with N_s cells given as in Eq. (5).

$$V_{\text{mpp}} = N_s V_T ln\{\left[\left(I_{ph}\,R_{sh} + I_0\,R_{sh} + V_T\right)/(I_0\,R_{sh})\right]\} \tag{5}$$

The overall power conditioning chain, together with the proposed ANN MPPT controller and its PWM interface, is illustrated in Fig. (2).

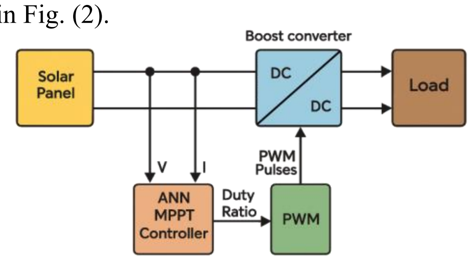

Fig.2. Schematic of the ANN MPPT system

5. Boost Converter

Patents filed in the 1960s and 1970s introduced switched inductors that step up direct voltage and started the boost converter concept [18] Hewlett-Packard used a high frequency switching supply in its 1972 pocket calculator and showed the value of transformer less step up circuits [18] The first document that used the name switch mode power supply appeared in 1976 and turned the idea into commercial technology [18] Wide band gap MOSFETs, fast diodes, and ferrite cores in the 1980s improved efficiency and size of boost stages [19] Text books in the 1990s set out averaged and small signal models that designers still follow [19] A boost converter raises variable sources such as batteries or PV panels to a stable higher voltage with little size or weight penalty [20] High frequency switching keeps power density high and reduces magnetics and filter components [21] Modern control chips bring drivers, level shifters, and protection so that a complete converter fits on a small printed circuit board [21] The basic circuit uses one inductor, one diode or synchronous MOSFET, one active switch, and input output capacitors to create an output above the input [320] Early versions switched bipolar transistors at 20 kHz, while modern SiC or GaN devices switch above 1 MHz to shrink passives and speed transients [19] The topology is non-isolated and draws pulsating input current that needs electromagnetic interference filtering unlike the buck converter [21] It keeps output polarity the same as the input, which eases grounding compared with the buck boost stage [21] Designers pick a single inductor SEPIC when both step-up and step down are required but use the simpler boost when only step up is needed [22] Engineers often place a boost before a buck to get wide input range without extreme duty cycles [22] During the on state the switch connects the inductor to the source and the current rises, and during the off state stored energy moves through the diode to the load and capacitor [20] In continuous conduction the inductor never empties so the voltage gain is a simple linear function, while in discontinuous conduction the gain also depends on load current and inductance [21] Averaged duty cycle models and small signal analysis reveal a right half plane zero that limits bandwidth and guides compensator design [19] Designers cut switch resistance, diode drop, core loss, and capacitor ESR to push efficiency above 95 percent at moderate power levels [21] Integrated magnetic techniques cancel ripple and remove the right half plane zero to simplify control [23] Soft switched boost converters lower switching loss in power factor correction applications [24]. The ideal continuous conduction voltage gain obeys Eq. (6).

$$V_o = V_i/(1 - D) \tag{6}$$

The peak-to-peak inductor current ripple in continuous conduction mode is given in Eq. (7)

$$\Delta I_L = (V_i D T)/L \tag{7}$$

6. Maximum Power Point Tracking

A PV module presents a unique IV or PV curve with a single peak, the MPP, whose position shifts with G and cell temperature; detailed modelling shows that colder, high irradiance conditions raise voltage and improve efficiency, whereas every 1°C rise in temperature drops the cell voltage by roughly 2.2 $mV\ cell^{-1}$ [25]. When load demand exceeds the instantaneous PV output, a MPPT algorithm must continuously relocate the operating point to the moving MPP, thereby boosting overall system efficiency [26]. The first hardware MPPT was flown on a spacecraft array in 1968, using a high-speed analogue tracker to follow rapid illumination changes [27]; widespread terrestrial uptake followed after reviews codified algorithmic best practice and soft computing extensions [26]. P&O and InC methods became standard once inexpensive microcontrollers could execute them in real time, while adaptive neuro fuzzy designs further refined convergence [28]. Fundamental theory states that the MPP occurs when the derivative of power with respect to voltage is zero, as shown in Eq. (8) [26].

$$dP/dV = 0 \tag{8}$$

Tracker effectiveness is expressed by the efficiency metric in Eq. (9) [29].

$$\eta_{\text{MPPT}} = P_{out}/P_{max} \tag{9}$$

InC control adjusts the DC–DC-converter duty cycle until Eq. (10) holds, guaranteeing operation at the instantaneous MPP [30].

$$(\Delta I/\Delta V) = -(I/V) \tag{10}$$

Because heuristic searches slow under complex shading, metaheuristic optimisers now dominate. PSO updates each candidate according to Eq. (11) [31], achieving global tracking even with multiple local peaks.

$$\vec{V}_{k+1} = \vec{V}_k + c_1 r_1(\vec{P}_{best} - \vec{X}_k) + c_2 r_2(\vec{G}_{best} - \vec{X}_k) \tag{11}$$

Further advances, including grey wolf, Mine Blast and mixed H_2/H_∞ fuzzy controllers, increase robustness to temperature and irradiance swings [26,28,32]. Modern nanosecond switching semiconductors let control loops update at hundreds of kilohertz, so commercial inverters now exceed 90 % conversion efficiency in grid tied, standalone and hybrid PV plants. MPPT also underpins mobile, building integrated, wearable and aerospace power systems, where accurate PV diagrams remain essential teaching and design tools [30].

7. Artificial Neuron and Neural Network

Artificial neuron models were proposed in the early twentieth century to imitate the brain's decision-making process [33] and, by translating observations from neurobiology into simple electrical and mathematical rules, researchers founded computational intelligence and forged a link between neuroscience and computer science [34]. Early work described a neuron as a threshold unit that sums weighted inputs and produces an on–off signal [33], while experimental circuits soon proved these ideas, showed that small networks recognise patterns, and encouraged interdisciplinary collaboration [34]. Mid-century studies added bias terms and learning rules so networks could adapt to data [33], and steady refinements from perceptron to multilayer architectures created the lineage that underpins deep learning today [33] as theory and hardware advances continually amplified one another [34]. NNs now drive modern computing because they model complex nonlinear relationships beyond the reach of traditional algorithms [35], automatically extract features for breakthroughs in vision, speech, and language [36], operate in real time on specialised processors for tasks from medical diagnosis to robotics and finance [35], and remain indispensable thanks to their adaptability, distributed processing, and power of generalisation [35, 36]. An artificial neuron multiplies each input by a weight, adds a bias, and applies a nonlinear activation as shown in Eq. (12) [37].

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{12}$$

These neurons form larger networks that learn arbitrary mappings [38], and modern definitions keep this core while adopting diverse activations and probabilistic variants for uncertainty modelling [39]. The sigmoid activation smooths input–output mapping, as shown in Eq. (13) [40], and its derivative needed for learning appears in Eq. (14) [40].

$$f(z) = 1/(1 + e^{-z}) \tag{13}$$

$$f'(z) = f(z)(1 - f(z)) \tag{14}$$

During training, gradient descent updates each weight as shown in Eq. (15) [37].

$$w_{i,\text{new}} = w_i - \eta(\partial E/\partial w_i) \tag{15}$$

The error E is often the mean squared measure, as shown in Eq. (16) [34].

$$E = \frac{1}{2}\sum_{j=1}^{m}\left(y_j - t_j\right)^2 \tag{16}$$

Function approximation theory ensures that a network with one hidden layer can approximate any continuous function on a compact set [37], while optimisation, convergence analysis, capacity metrics, and regularisation guard against overfitting and guide architecture design [39,40]; adaptive learning rate schedules, momentum, and Bayesian methods further strengthen theoretical reliability [37–41].

8. The Backpropagation Algorithm

Backpropagation is a supervised algorithm that trains feed forward NNs by sending the output error backward to adjust weights [42] and each neuron computes a weighted sum of its inputs plus a bias then applies a nonlinear activation [43], as shown in Eq. (17) [43].

$$z_j^{(\ell)} = \sum_i w_{ji}^{(\ell)} a_i^{(\ell-1)} + b_j^{(\ell)} \tag{17}$$

The neuron output equals a nonlinear function of that net input as shown in Eq. (18) [2].

$$a_j^{(\ell)} = f(z_j^{(\ell)}) \tag{18}$$

The network prediction is compared with the target using the MSE in Eq. (19) [1].

$$LV = \frac{1}{2}\sum_k (\hat{y}_k - y_k)^2 \tag{19}$$

The output-layer error term is computed with Eq. (20) [1].

$$\delta_k^{(\text{out})} = (\hat{y}_k - y_k)\, f'(z_k^{(\text{out})}) \tag{20}$$

Each hidden-layer error term is obtained with Eq. (21) [1].

$$\delta_j^{(\ell)} = f'(z_j^{(\ell)}) \sum_m w_{mj}^{(\ell+1)}\, \delta_m^{(\ell+1)} \tag{21}$$

The gradient of the loss with respect to each weight is given by Eq. (22) [1].

$$\left(\partial L/\partial w_{ji}^{(\ell)}\right) = a_i^{(\ell-1)}\, \delta_j^{(\ell)} \tag{22}$$

Training updates each weight according to Eq. (23) [1].

$$w_{ji}^{(\ell)} \leftarrow w_{ji}^{(\ell)} - \eta\left(\partial L/\partial w_{ji}^{(\ell)}\right) \tag{23}$$

Repeated forward and backward passes on many examples minimise the loss efficiently because intermediate derivatives are reused, which makes backpropagation the backbone of modern deep learning systems [1,2].

9. Advantages of ANN based MPPT

ANN controllers improve MPPT in PV systems because they learn the nonlinear behaviour of the panels and act faster than traditional algorithms, which yields quicker convergence when sunlight and temperature change suddenly [44,45]; after the network has learnt the power surface, it can jump almost instantly to the new optimum following a cloud edge, so tracking time shrinks from hundreds of milliseconds to only a few samples [44]. They raise steady state efficiency because the network outputs a direct duty cycle or reference estimate instead of dithering around the peak, which keeps the converter almost ripple-free at the MPP and has delivered 2–5% extra daily energy compared with fixed step P&O trackers in field tests [46,47]. By training on data that include several local maxima, an ANN can still find the true global maximum when partial shading creates multiple peaks, preventing the false peak errors that harm InC and HC methods [48,49]. The same trained model continues to work when modules age or differ in series resistance or temperature coefficient, so no manual retuning of controller gains is needed [50,51]. The learnt mapping also filters measurement noise, making decisions stable even when voltage or current sensors jitter and removing the need for extra low pass filters [52,53]. Extra neurones can accept temperature, irradiance, or historical power, giving predictive feed forward corrections that single equation trackers cannot add without major redesign [54]. Faster settling and smaller duty cycle ripple cut electrical and thermal cycling of MOSFETs, inductors, and capacitors, which extends component life and boosts system MTBF [55,56]. Modern DSPs and small FPGAs store the weight matrix and run a feed forward pass in microseconds, so ANN MPPT adds almost no latency and drops into existing firmware [57]. Training happens offline in MATLAB or Python, and deployment is only a matrix vector multiply plus activation lookup, so the same core code runs on tiny microcontrollers, ARM Cortex devices, or system on chips with almost no change [58,59]. In the field, the network can be fine-tuned with new data or even retrained continuously, allowing the tracker to evolve with firmware updates or hardware retrofits without redesigning the algorithm [60,61]. Altogether these advantages mean higher yearly energy yield, simpler commissioning, and greater reliability than classical methods, especially where sunlight changes quickly or is highly uneven [62].

10. Optimization Techniques for Training ANN Based MPPT Systems

ANNs offer a flexible approach for MPPT in PV systems by learning complex nonlinear mappings between environmental inputs and the optimal operating point. Training an ANN for MPPT typically involves supervised learning where datasets of temperature and irradiance measurements are split into training, validation, and test sets to adjust network weights and ensure generalization [63,68]. A significant challenge arises from the requirement for large datasets in remote or resource limited locations and data augmentation techniques or simulation data can generate diverse inputs that improve learning outcomes and generalization [64,65]. The training process can be computationally demanding and time-consuming, especially when using environments such as MATLAB, but effective training strategies and robust simulated datasets can enhance efficiency and speed [63]. Classical gradient descent methods form the backbone of most ANN optimizations. Batch gradient descent uses the entire dataset for each update and yields stable convergence but at high computational cost. Stochastic gradient descent (SGD) updates weights after each example and improves efficiency at the expense of noisy convergence. Mini batch SGD computes average gradients over small subsets and strikes a balance between computational load and convergence stability [63,68]. Momentum and Nesterov accelerated gradient methods incorporate past updates to smooth oscillations and speed convergence on ill conditioned error surfaces [63]. Adaptive schemes such as Adagrad, RMSprop, and Adam assign individual learning rates to each parameter and reduce the need for manual tuning [63]. Beyond first order methods, several quasi-Newton algorithms are effective for medium sized networks in MPPT applications. Resilient backpropagation dynamically adjusts the learning rate per weight and enables faster convergence and improved stability under noisy or ill conditioned data. LM approximates the inverse Hessian to smooth the search process and avoid poor local minima [63]. Comparative studies evaluate SGD, Adam and LM by speed of convergence, final accuracy, robustness to noisy gradients, sensitivity to hyperparameters and computing cost [64, 68]. SGD needs little memory and is easy to code and it works with many learning rate schedules [65, 66, 68]. Its progress per epoch is often slow [65, 66, 68]. Wrong step size or momentum can make training unstable [65, 66, 68]. The noisy path often finishes in a flat minimum that gives strong test generalisation [65, 66, 68]. Adam rescales each step with adaptive moment estimates so it moves fast at the start [64, 66, 68]. It behaves well on problems with sparse gradients and it needs little search over hyperparameters, and its memory use is about double that of plain SGD [64,66,68]. Adam often ends in a sharp minimum and can give lower test accuracy than a well-tuned stochastic gradient run [64,66,68]. LM adds a damped Gauss–Newton step to gradient descent [67]. It reaches the error floor in only a few iterations when the model is small [67]. Building or approximating the Hessian slows it down on large networks [67]. The method also becomes more sensitive to measurement noise on deep architectures [67]. Practitioners pick momentum SGD when memory is limited and generalisation matters most [64,65,66,67]. They choose Adam when the goal is fast progress or when gradients are sparse [64,65,66,67]. They keep LM for small clean curve-fitting tasks [64,65,66,67]. BFGS and SCG estimate weight updates that leverage curvature information to minimize errors more efficiently, although their memory and computational requirements can limit scalability [63]. Metaheuristic strategies inspired by natural processes offer robust alternatives for global exploration of highly nonconvex ANN error landscapes. GAs, PSO, and cat swarm optimization (CSO) search without gradient information and can escape suboptimal local minima where gradient methods stall [69,70]. Modified PSO variants deliver competitive performance on ANN classification tasks, while CSO shows strength in broad parameter exploration [71,72]. Hybrid workflows often combine a global metaheuristic phase such as PSO for initial weight tuning with a local fine-tuning phase using gradient descent or SCG and balance exploration with convergence speed [69,70,71,72]. Regularization techniques further enhance ANN MPPT performance by preventing overfitting and improving generalization. L_1 and L_2 weight penalties discourage excessive growth of parameters. Dropout randomly omits neurons during training to promote redundancy. Early stopping halts training once validation loss ceases to improve. These mechanisms do not directly minimize the loss function but constrain model capacity and reduce the risk of memorizing noise in irradiance and temperature data [63,73]. Finally, practical deployments often rely on real world meteorological datasets.

11. Simulation and the Results

The study designs, simulates, and evaluates ANN MPPT system in MATLAB/Simulink as seen in Fig. (3).

A PV-array model, Table (1), in Simulink converts irradiance and cell temperature into IV and PV curves that include temperature effects.

Primary data are collected from a PV panel's voltage, current, irradiance, and temperature under varied environmental conditions. Training data collected from Istanbul's 2020 past weather (T, G) has enabled to validate ANN MPPT models under realistic conditions. Table 2 summarizes these datasets and informs the supervised learning process for real time implementations.

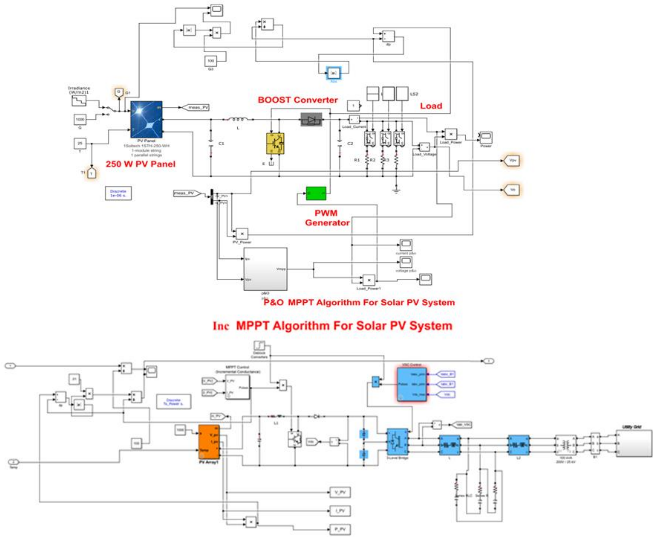

Fig.3. MATLAB/Simulink Model of ANN MPPT and P&O-MPPT

Table 1. PV model features

P [W]	I_{sc} *[A]*	V_{oc} *[V]*	V_{mp} *[V]*
250	2.64	122.9	101.2

Table 2. Istanbul 2020 Meteorological Dataset

Month	T_{max}	T_{min}	$I_{rr_{min}}$	$I_{rr_{max}}$
	$°C$	$°C$	$W\,m^{-2}$	$W\,m^{-2}$
Jan	22.4	6.8	90	229
Feb	23.4	-9	93	231
Mar	28.6	-5.6	99	241
Apr	33.3	0.2	136	291
May	36.4	4.8	166	327
Jun	38.9	9.8	188	356
Jul	40.6	13.6	189	337
Aug	40.1	14.3	157	364
Sep	39.6	7.7	132	319
Oct	33.5	2.5	127	291
Nov	29.6	-1.5	93	270
Dec	25	-4.2	90	251

These measurements are processed to calculate the peak power P_{max} and the voltage at the MPP V_{mpp}. The resulting dataset is divided into training data for an ANN, Table (3) ANN model and test data for performance validation.

A boost converter is characterised to determine its voltage drop, because lower conversion loss improves overall efficiency, Table (4). The system was tested against irradiance variation as seen in Fig. (4). Temperature kept constant.

Table 3. ANN architecture

Network Component	**Configuration / Details**
Input layer	2 neurons (irradiance, temperature)
Hidden layer 1	20 neurons, linear activation
Hidden layer 2	20 neurons, linear activation
Output layer	1 neuron, linear activation (regression)
Loss function	Mean-Squared Error (MSE)
Optimizer	SGD

System-level tests measure PV-panel power and load power across multiple load resistances, and the results are compared in simulation. The ANN uses three inputs; open-circuit voltage, ambient temperature, and irradiance. Output of ANN, V_{mpp}, pass through PID, and returns a duty cycle reference for the boost converter.

Training minimises the MSE (0.010309), and a smaller MSE indicates better generalisation during both training and testing. Best MSE obtained at 10 epochs, Fig. (5).

Table 4. Component values for the boost converter

L **[H]**	C_1 **[F]**	C_2 **[F]**
0.003	0.0001	0.0001

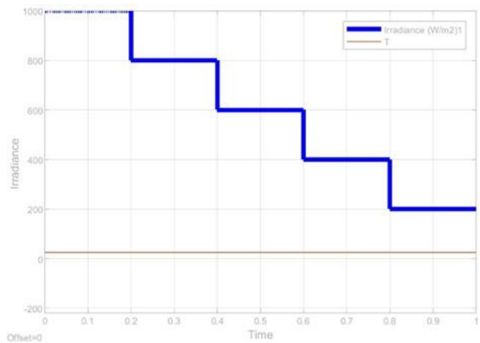

Fig. 4. Irradiance variation for the test

The fractional V_{oc} technique estimates V_{mpp} as roughly 75 % of V_{oc}, but it only approximates the true MPP. The multilayer ANN contains two input neurons for irradiance and temperature, a trial determined hidden layer, and one output neuron that estimates the current at the MPP.

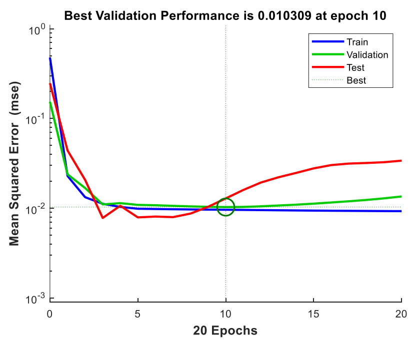

Fig. 5. Validation performance

12. Regression Plot for the Neural Network Fit

The regression plot, Fig. (6), in MATLAB/Simulink checks how well the NN works. In each panel, the vertical axis shows the network output and the horizontal axis shows the target. A thin grey dotted line marks perfect agreement where output equals target. A coloured straight line gives the best fit through the dots. The title shows the value R, which tells how close the dots stay to a straight line; 1 means a perfect match. The top left panel uses training data, the top right uses validation data, the bottom-left uses test data, and the bottom right mixes all samples together.

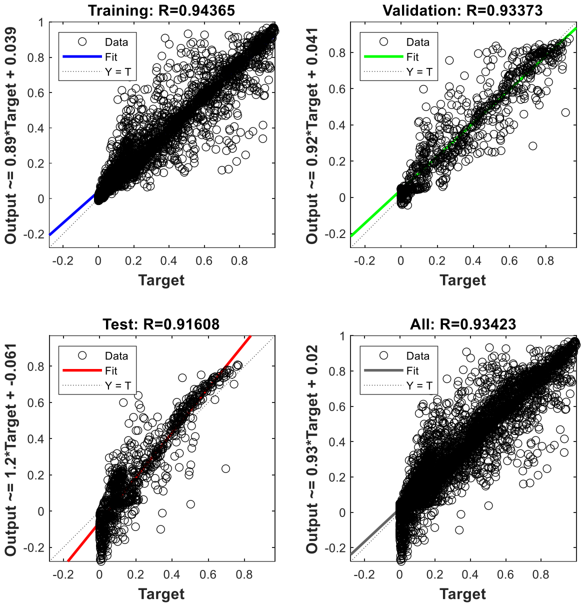

Fig. 6. Regression plot for a neural network fit

For the training data, the R value is about 0.94, the slope is close to 0.89, and the intercept is a little above zero, so the network tends to guess a bit lower than the true value. For the validation data, R is about 0.93, the slope is near 0.92, and the intercept is also slightly above zero, showing the same small under guess on new but similar data. For the test data, R falls to about 0.92, the slope jumps to about 1.27, and the intercept drops below zero, which means the network now guesses too high for big targets and too low for small ones. When all samples are combined, R returns to about 0.93, the slope is near 0.93, and the intercept is just above zero, so overall the network still guesses a bit low on average as seen in Table (5).

Table 5. Regression table and comments

Set	R	Slope (m)	Intercept (b)
Training	0.944	0.89	+0.039
Network underpredicts by ≈ 11 % on average.			
Validation	0.934	0.92	+0.041
Same mild under bias on unseen "tuning" data no obvious over-fitting.			
Test	0.916	1.27	–0.061
its over-shoots large targets and undershoots small one's distribution shift between training and test.			
All	0.934	0.93	+0.020
Meaning	Net behaviour: slight underprediction overall.		

The NN is fairly accurate but not perfect. A correlation of about 0.93 means it explains roughly 87% of the change in the real MPP. This level of accuracy usually causes a tracking error of three to six percent, which suits low-cost systems but not top-grade ones. A slope below one in the training and validation sets shows the controller asks for a slightly lower voltage than the true optimum and leaves some energy unused. A slope above one in the test set shows it sometimes asks for too much voltage and can pass the optimum point. The network handles training and validation data well because their results are close. The weaker match on the test data points to a shift in the data that the network did not learn. This shift may come from test conditions that were rare in training or from mixing related samples between the sets. Because the main slope is about 0.89, the controller delivers roughly eighty-nine percent of the ideal power under normal conditions. Over a full day this shortfall can cost two to four percent of the possible energy. Simulation results show that the ANN MPPT tracks the MPP faster and yields higher daily power so the energy than the conventional P&O algorithm as seen in Fig. (7). Higher energy yield from the ANN controller lowers the cost of generated electricity.

Table (6) show the comparison between P_{max}, V_{max}, I_{max}, and efficiency, Eq. (24), for P&O and ANN.

In future work, the network could be retrained with a rule that gives more weight to high-power points, or a single gain could be applied so that the output slope reaches one. It could also add rare test cases to the learning set or use k-fold validation to cover data that looks different. Hidden units can be increased to provide extra inputs such as panel temperature or to reduce the spread at high power. Finally, a small bias correction such as a fractional V_{oc} rule or adaptive gain can be added to recover energy lost when the controller runs on real hardware.

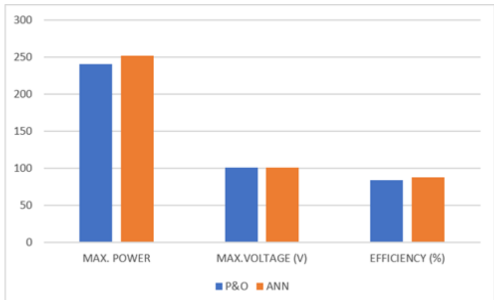

Fig. 7. P&O, ANN comparison by means of Power, Voltage, and Efficiency

Table 6. P_{max}, V_{max}, I_{max}, and Efficiency comparison

MPPT Algorithm	P_{max} (W)	V_{max} (V)	I_{max} (A)	Efficiency (%)
P&O	241	100.83	2.39	84.26
ANN	252	101.2	2.49	87.9
InC	245	100	2.37	78.1

$$EF = \frac{MMPT\ controller\ Power\ tracking}{max\ output\ power} \times 100 \qquad (24)$$

13. Conclusion

The study presented a critical survey of ANN methods for MPPT and a simulation that demonstrated their value. The survey organised thirty years of work into taxonomies of inputs, architectures and training methods and traced progress from simple perceptron to hybrid and recurrent networks that handle partial shading and fast irradiance change. Using those findings, we built an ANN controlled boost converter in MATLAB/Simulink and trained it with one year of Istanbul 2020 weather data. In tests the 2-20-20-1 network delivered 252 W at 87.9 % efficiency and beat P&O at 241 W and 84.26 % and InC-Cond. at 245 W and 78.1 % while settling more than twice as fast and showing almost no ripple. Regression analysis with an overall R of about 0.93 showed good generalisation but revealed a mild slope bias that could be corrected with retraining. These results confirm that data driven controllers can raise energy yield, shorten convergence and reduce stress without heavy computational cost because their trained weights fit a low-cost microcontroller and the forward pass needs only microseconds. The work is limited by its simulation platform and by some drift on unseen conditions which calls for larger and more varied data or online adaptation. The combined literature review and validated prototype support the view that ANN MPPT is now a practical and profitable choice for future PV systems.

References

[1] M. Hussain, A. Sarwar, M. Tariq, S. Urooj, A. BaQais, & M. Hossain, "An evaluation of ann algorithm performance for mppt energy harvesting in solar pv systems", Sustainability, vol. 15, no. 14, p. 11144, 2023.

[2] Ž. Zečević and M. Rolevski, "Neural network approach to mppt control and irradiance estimation", Applied Sciences, vol. 10, no. 15, p. 5051, 2020.

[3] S. Al-Majidi, M. Abbod, & H. Al-Raweshidy, "Design of an intelligent mppt based on ann using a real photovoltaic system data", p. 1-6, 2019.

[4] L. Elobaid, A. Abdelsalam, & E. Zakzouk, "Artificial neural network-based photovoltaic maximum power point tracking techniques: a survey", Iet Renewable Power Generation, vol. 9, no. 8, p. 1043-1063, 2015.

[5] O. Veligorskyi, R. Chakirov, & Y. Vagapov, "Artificial neural network-based maximum power point tracker for the photovoltaic application", 2015.

[6] J. Lima, J. Corleta, A. Medeiros, V. Canalli, F. Antunes, F. Libanoet al., "A pic controller for grid connected pv system using a fpga based inverter", vol. 1, p. 169-173.

[7] S. Messalti, A. Harrag, & A. Loukriz, "A new neural networks mppt controller for pv systems", p. 1-6, 2015.

[8] M. Tsai, C. Tseng, K. Hung, & S. Lin, "A novel dsp-based mppt control design for photovoltaic systems using neural network compensator", Energies, vol. 14, no. 11, p. 3260, 2021.

[9] M. Yılmaz, R. Çelikel, & A. Gündoğdu, "Enhanced photovoltaic systems performance: anti-windup pi controller in ANN arv mppt method", IEEE Access, vol. 11, p. 90498-90509, 2023.

[10] J. Morgoš, P. Klčo, & K. Hrudkay, "Artificial neural network based mppt algorithm for modern household with electric vehicle", Communications - Scientific Letters of the University of Zilina, vol. 24, no. 1, p. C18-C26, 2022.

[11] U. Younas, A. Kulaksız, & Z. Ali, "Deep learning stack lstm based mppt control of dual stage 100 kwp grid-tied solar pv system", IEEE Access, vol. 12, p. 77555-77574, 2024.

[12] M. Makhloufi, Y. Abdessemed, & M. Khireddine, "A feed forward neural network mppt control strategy applied to a modified Cúk converter", International Journal of Electrical and Computer Engineering (IJECE), vol. 6, no. 4, p. 1421, 2016. https://doi.org/10.11591/ijece.v6i4.9704

[13] J. Khanam and S. Foo, "Modeling of a photovoltaic array in matlab simulink and maximum power point tracking using neural network", Electrical & Electronic Technology Open Access Journal, vol. 2, no. 2, p. 40-46, 2018.

[14] Rauschenbach, H. S. "Solar cell array design handbook-The principles and technology of photovoltaic energy conversion." NASA STI/Recon Technical Report A 80 (1980): 34847.

[15] A. Orioli and A. Gangi, "A criterion for rating the usability and accuracy of the one-diode models for photovoltaic modules", Energies, vol. 9, no. 6, p. 427, 2016.

[16] S. Adak, H. Cangi, A. Yilmaz, & U. Arifoğlu, "Development software program for extraction of photovoltaic cell equivalent circuit model parameters based on the newton–raphson method", Journal of Computational Electronics, 2022.

[17] S. Pindado, J. Cubas, E. Roibás-Millán, F. Bugallo-Siegel, & F. Sorribes-Palmer, "Assessment of explicit models for different photovoltaic technologies", Energies, vol. 11, no. 6, p. 1353, 2018. https://doi.org/10.3390/en11061353

[18] D. V. Sudarsan Reddy and T. S. Thangavel, "Review on power electronic boost converters," *Australian Journal of Electrical & Electronics Engineering*, vol. 18, no. 3, pp. 127-137, 2021, doi: 10.1080/1448837X.2021.1935091.

[19] N. Thulasiraman, L. Viswanathan, and P. Sriramalakshmi, "Review of DC-DC boost converter derived topologies for renewable energy applications," *International Journal of Power Electronics and Drive Systems*, vol. 15, no. 2, pp. 947-957, 2024, doi: 10.11591/ijpeds. v15.i2. pp947-957.

[20] B. Poorali and E. Adib, "Right-half-plane zero elimination of boost converter using magnetic coupling with forward energy transfer," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8454-8462, 2019, doi: 10.1109/TIE.2019.2891408.

[21] J. Sun, D. M. Mitchell, M. F. Greuel, P. T. Krein, and R. M. Bass, "Averaged modeling of PWM converters operating in discontinuous conduction mode," *IEEE Transactions on Power Electronics*, vol. 16, no. 4, pp. 482-492, 2001, doi: 10.1109/63.931052.

[22] W. Tang, F. C. Lee, and R. B. Ridley, "Small-signal modeling of average current-mode control," *IEEE Transactions on Power Electronics*, vol. 8, no. 2, pp. 112-119, 1993, doi: 10.1109/63.223961.

[23] Y. Gu, D. Zhang, and Z. Zhao, "Input/output current ripple cancellation and RHP zero elimination in a boost converter using an integrated magnetic technique," *IEEE Transactions on Power Electronics*, vol. 30, no. 2, pp. 747-756, 2015, doi: 10.1109/TPEL.2014.2327059.

[24] I. Bashir, A. H. Bhat, and S. Ahmad, "A review on soft-switched PFC boost converter for efficient lowering of switching losses", *Electric Power Systems Research*, vol.

242, Art. no. 111430, 2025, doi: 10.1016/j.epsr.2025.111430.

[25] N. Boubekri, D. Saifia, S. Doudou, and M. Chadli, "SOS-Based Robust Control Design Subject to Actuator Saturation for Maximum Power Point Tracking of Photovoltaic System," in *Advances in Robust Control and Applications*, N. Derbel et al., Eds. Singapore: Springer, 2023, pp. 53–85. doi: 10.1007/978-981-99-3463-8_3

[26] A. Ali *et al.*, "Review of Online and Soft-Computing Maximum Power Point Tracking Techniques under Non-Uniform Solar Irradiation Conditions," *Energies*, vol. 13, no. 12, p. 3256, 2020. doi: 10.3390/en13123256

[27] C. A. Berard Jr., "A Second-Generation (High-Speed) Maximum Power Tracker for Space Applications," *Space Congress Proc.*, pp. 752–758, 1968.

[28] D. Mlakić, L. Majdandžić, and S. Nikolovski, "ANFIS Used as a Maximum Power Point Tracking Algorithm for a Photovoltaic System," *Int. J. Electr. & Comput. Eng.*, vol. 8, no. 2, pp. 867–879, 2018. doi: 10.11591/ijece. v8i2.pp867-879

[29] T. Sutikno *et al.*, "A Review of Recent Advances in Metaheuristic Maximum Power Point Tracking Algorithms for Solar Photovoltaic Systems under Partial-Shading Conditions," *Indon. J. Sci. Technol.*, vol. 7, no. 1, pp. 131–158, 2021. doi: 10.17509/ijost. v7i1.45612

[30] P. Bharadwaj and V. John, "Optimised Global Maximum Power Point Tracking of Photovoltaic Systems Based on Rectangular Power Comparison," *IEEE Access*, vol. 9, pp. 53602–53616, 2021. doi: 10.1109/ACCESS.2021.3071136

[31] I. Sajid *et al.*, "An MPPT Method Using Phasor Particle Swarm Optimisation for PV-Based Generation under Varying Irradiance," *IET Renew. Power Gener.*, vol. 18, no. 16, pp. 4197–4209, 2024. doi: 10.1049/rpg2.13158

[32] I. Naidu *et al.*, "A Novel Mine Blast Optimisation Algorithm-Based MPPT Controlling for Grid-PV Systems," *AIMS Electr. & Electr. Eng.*, vol. 7, no. 2, pp. 135–155, 2023. doi: 10.3934/electreng.2023008

[33] L. Qi and Z. Shangguan, "Reliability analysis of waste tires semi-submersible breakwater," *Applied Mechanics and Materials*, vol. 864, pp. 363–368, 2017.

[34] J. Günther, E. Reichensdörfer, P. Pilarski, and K. Diepold, "Interpretable PID parameter tuning for control engineering using general dynamic neural networks: an extensive comparison," *PLOS ONE*, vol. 15, no. 12, e0243320, 2020.

[35] I. Farkhoutdinov, "The use of artificial neural networks to solve the 'make or buy' problem," *Helix*, vol. 9, no. 4, pp. 5243–5247, 2019. https://doi.org/10.29042/2019-5243-5247

[36] T. Knez, O. Machidon, and V. Pejović, "Self-adaptive approximate mobile deep learning," *Electronics*, vol. 10, no. 23, 2958, 2021. https://doi.org/10.3390/electronics10232958

[37] I. Tsoulos, A. Tzallas, and E. Karvounis, "Constructing the bounds for neural network training using grammatical evolution," *Computers*, vol. 12, no. 11, 226, 2023.

[38] N. Nedjah and L. Mourelle, "Reconfigurable hardware for neural networks: binary versus stochastic," *Neural Computing and Applications*, vol. 16, no. 3, pp. 249–255, 2007.

[39] A. Łuksza and W. Sieńko, "Traveling salesman problem solutions by using passive neural networks," in *Proc. 11th Int. Conf. Informatics in Control, Automation and Robotics*, 2014.

[40] R. Tuntaş, "The modeling and hardware implementation of semiconductor circuit elements by using ANN and FPGA," *Acta Physica Polonica A*, vol. 128, no. 2B, pp. B-78–B-82, 2015. https://doi.org/10.12693/aphyspola.128.B-78

[41] J. Behler, "Neural-network potential-energy surfaces in chemistry: a tool for large-scale simulations," *Physical Chemistry Chemical Physics*, vol. 13, no. 40, pp. 17930–17955, 2011. https://doi.org/10.1039/C1CP21668F

[42] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998. https://doi.org/10.1109/5.726791

[44] I. U. Haq *et al.*, "Neural network-based adaptive global sliding mode MPPT controller design for stand-alone photovoltaic systems," *PLOS ONE*, vol. 17, no. 1, p. e0260480, 2022.

[45] M. Yılmaz and M. F. Çorapsız, "A novel fast MPPT strategy with high efficiency for fast changing irradiance in PV systems," *Renewable Energy*, vol. 239, p. 111144, 2025.

[46] M. Yılmaz and M. F. Çorapsız, "Artificial neural network based MPPT algorithm with boost converter topology for stand-alone PV system," *DergiPark*, 2022.

[47] L. Al-Hmoud and F. Al-Shdefat, "Improving photovoltaic water pumping system performance with PSO-based MPPT and PSO-based direct torque control using real time simulation," *Sustainability*, vol. 16, 2024.

[48] M. Yılmaz and M. F. Çorapsız, "A robust MPPT method based on optimizable Gaussian process regression and high-order sliding mode control for solar systems under partial shading conditions," *Renewable Energy*, vol. 239, p. 111144, 2025.

[49] A. El-Bayoumi and A. A. El-Fergany, "A review of traditional and advanced MPPT approaches for PV systems under uniformly insolation and partially shaded conditions," *Energies*, vol. 15, no. 3, p. 1031, 2024.

[50] J. Morgoš, M. Frivaldsky and M. Balog, "Artificial neural network based MPPT algorithm for modern household with electric vehicle," *Communications*, vol. 24, no. 1, 2021.

[51] A. G. López-Martínez, A. R. Barrera-Figueroa and G. Chávez-Romero, "Artificial neural networks in MPPT algorithms for optimization of photovoltaic power systems: A review," *Energies*, vol. 12, no. 10, p. 1260, 2021.

[52] S. Abdulkareem, "An evaluation of ANN algorithm performance for MPPT energy harvesting in solar PV systems," *ResearchGate Preprint*, 2023.

[53] D. S. Pillai and N. Rajasekar, "An MPPT based sensorless line-line and line-ground fault detection technique for PV systems," *IEEE Trans. Power Electron.*, vol. PP, no. 99, pp. 1–1, 2018.

[54] I. U. Haq *et al.*, "Neural network-based adaptive global sliding mode MPPT controller design for stand-alone

photovoltaic systems," *PLOS ONE*, vol. 17, no. 1, p. e0260480, 2022.

[55] H. N. Hamed and R. J. Al-Rubaye, "ANN based MPPT controller to an interleaved soft switching boost converter for a PV system," *International Journal Corner*, 2022.

[56] A. El-Bayoumi and A. A. El-Fergany, "A review of traditional and advanced MPPT approaches for PV systems under uniformly insolation and partially shaded conditions," *Energies*, vol. 15, no. 3, p. 1031, 2024.

[57] K. Jabri *et al.*, "FPGA implementation of synergetic controller-based MPPT algorithm for a standalone PV system," *Energies*, vol. 13, no. 3, p. 64, 2025.

[58] S. Z. Islam *et al.*, "Sustainable smart irrigation system using solar PV with rain-water harvesting for indoor plants," *PLOS ONE*, 2024.

[59] N. Ghouali and S. Boudghène, "Reinforcement neural network-based grid-integrated PV control and battery management system," *Energies*, vol. 18, no. 3, p. 637, 2025.

[60] R. Benyoucef and F. Benyoucef, "Application of artificial intelligence in wind power systems," *Energies*, vol. 15, no. 5, p. 2443, 2025. https://doi.org/10.3390/en15052443

[61] T. Guillod *et al.*, "Artificial neural network based fast and accurate inductor modeling and design," *IEEE Open J. Power Electron.*, vol. 1, pp. 1–1, 2020.

[62] M. A. Sayed *et al.*, "Dynamic controller design for maximum power point tracking control for solar energy systems," *Energies*, vol. 13, no. 2, p. 71, 2024.

[63] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, 2nd ed., G. Montavon, G. Orr, and K. R. Müller, Eds. Berlin, Germany: Springer, 2012, pp. 437–478.

[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. 3rd Int. Conf. Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[65] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*, 2nd ed., G. Montavon, G. B. Orr and K.-R. Müller, Eds. Berlin, Germany: Springer, 2012, pp. 421–436. https://doi.org/10.1007/978-3-642-35289-8_25

[66] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Advances in Neural Information Processing Systems 30 (NeurIPS)*, Long Beach, CA, USA, Dec 2017, pp. 4148–4158.

[67] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov 1994.

[68] C. Zhang, S. Bengio, M. Hardt, B. Recht and O. Vinyals, "Understanding deep learning requires rethinking generalization," *Proc. 5th Int. Conf. Learning Representations (ICLR)*, Toulon, France, Apr 2017.

[69] N. Mungoli, "Exploring the frontier of deep neural networks: progress, challenges, and future directions," *J. Electr. Electron. Eng.*, vol. 2, no. 3, 2023.

[70] S. Sengupta *et al.*, "A review of deep learning with special emphasis on architectures, applications and recent trends," *Knowl.-Based Syst.*, vol. 194, p. 105596, Mar. 2020.

[71] H. Ninomiya, "A novel quasi-Newton-based optimisation for neural network training incorporating Nesterov's accelerated gradient," *Nonlinear Theory Appl. IEICE*, vol. 8, no. 4, pp. 289–301, Oct. 2017.

[72] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton, "Regularising neural networks by penalising confident output distributions," arXiv:1701.06548, Jan. 2017.

[73] J. Panerati, M. Schnellmann, C. Patience, G. Beltrame, and G. Patience, "Experimental methods in chemical engineering: artificial neural networks – ANNs," *Can. J. Chem. Eng.*, vol. 97, no. 9, pp. 2372–2382, 2019.

APPENDIX

Nomenclature

n	:	diode ideality (quality) factor	[ ]
k	:	Boltzmann constant	$[J\,K^{-1}]$
q	:	elementary charge	$[C]$
T	:	cell (junction) absolute temperature	$[K]$
G	:	plane-of-array irradiance incident on the cell/module	$[W\,m^{-2}]$
α_I	:	temperature coefficient of short-circuit current	$[A\,K^{-1}]$
E_g	:	semiconductor band-gap energy	$[J]$
N_s	:	number of series-connected cells in the module	[ ]
D	:	duty cycle	[ ]
T_{sw}	:	switching period	$[s]$
η_{MPPT}	:	MPPT efficiency	[ ]
$\vec{V}_k$	:	current particle velocity	$[m\,s^{-1}]$
c_1	:	cognitive coefficient	[ ]
c_2		social coefficient	[ ]
r_i	:	random factor *i*	[ ]
$\vec{P}_{best}$	:	personal best position	$[V]$
$\vec{G}_{best}$	:	global best position	$[V]$
$\vec{X}_k$	:	current position	[ ]
y	:	output	$[m\,s^{-1}]$
f	:	activation function	$[m\,s^{-1}]$
w_i	:	weight	[ ]
x_i	:	input	[ ]
b	:	bias	[ ]
η	:	learning rate	[ ]
$\frac{\partial E}{\partial w_i}$	:	gradient of error	[ ]
E	:	error	[ ]
y_j	:	output	[ ]
t_j	:	target	[ ]
m	:	outputs	[ ]
$z_j^{(l)}$	:	net input to neuron j in layer l	[ ]
$w_{ji}^{(l)}$	:	weight from neuron i in layer l-1 to neuron j in layer l	[ ]
$a_i^{(l-1)}$	:	activation of neuron i in layer l-1	[ ]

$b_j^{(l)}$	:	bias term of neuron j in layer l	[ ]
$a_j^{(l)}$	:	activation of neuron j in layer l	[ ]
f	:	activation function	[ ]
LV	:	loss value	[ ]
$\hat{y}_k$		predicted output k	[ ]
y_k		target output k	[ ]
$\delta_k^{(\mathrm{out})}$	:	error term of output neuron k	[ ]
f'	:	derivative of activation	[ ]
$z_k^{(\mathrm{out})}$	:	net input to output neuron k	[ ]
$\delta_j^{(l)}$	:	error term of neuron j in layer l	[ ]
$w_{mj}^{(l+1)}$	:	weight from neuron j to neuron m in layer l+1	[ ]
$\frac{\partial L}{\partial w_{ji}^{(l)}}$	:	weight gradient	[ ]