

Robustness Analysis of ELM-based Fault Detection in PV Systems

Halah Sabah Mutashar *[‡] , Amina Mahmoud Shakir * 

* Department of Electronic and Communications Engineering, Al-Nahrain University, Iraq

(St.hala.sabah.1@nahrainuniv.edu.iq, aminaalkafajiam@gmail.com,)

[‡]Halah Sabah Mutashar; Baghdad, Al-Jadria, Iraq

St.hala.sabah.1@nahrainuniv.edu.iq

Received: 06.09.2023 Accepted:08.10.2023

Abstract- Recently, the research emphasizes the importance of professional inspection and repair in case of suspected faults in Photovoltaic (PV) systems. The detection of faults in Photovoltaic (PV) systems is a critical aspect of maintenance, and machine learning models have emerged as valuable tools for this purpose. This study focuses on fault detection in a simulated 0.25 MW PV power system, utilizing various machine learning algorithms. The dataset comprises normal operation and different fault scenarios, spanning multiple fault types, with 26 predictor variables. Four machine learning algorithms, including Extreme Learning Machine (ELM) with different hidden node configurations (L=100 and L=200), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), and Neural Network, are evaluated through 10-fold cross-validation. The results demonstrate robust performance across all models. ELM (L=200) exhibits the highest accuracy, with 99.1663% during validation and 97.9727% on the test data, achieving a balance between accuracy and training time (0.1709 seconds). ELM (L=100) also shows strong performance with 98.3416% validation accuracy and 97.9592% on the test data, while LDA achieves 94.0298% (validation) and 90.8163% (test), SVM reaches 97.0149% (validation) and 96.4591% (test), and Neural Network attains 95.5224% (validation) and 94.9388% (test). ELM, especially with L=200, emerges as a superior choice for accurate and efficient fault detection in PV systems, highlighting its potential for real-world applications.

Keywords: Photovoltaic (PV), Fault detection, Extreme Learning Machine (ELM), Grid-connected system.

1. Introduction

Solar panels are intended to convert sunlight into power [1, 2]. Solar power plants are often made up of several solar panels stacked together. These panels may face a variety of faults throughout their operation, including electrical, mechanical, or thermal concerns. These problems have a detrimental influence on the performance of the panels and, in some cases, lead to their destruction or have an impact on other components engaged in the operation. The presence of faulty panels not only degrades the overall efficiency of the solar plant but also results in increased operational costs. To mitigate these costs and potential consequences, it is crucial to detect these malfunctions as early as possible. Typically, companies employ specialized teams to inspect the condition of the panels, analyzing the I-V curves or visually examining them for anomalies. Alternatively, thermal cameras mounted on drones can be utilized to identify hot spots on the surface of the panels [3-5].

However, many of the existing detection methods are expensive and time-consuming, often resulting in a delay of up to a year before faults are discovered. This delay can have a significant financial impact on the companies responsible for operating the solar plants. By promptly identifying anomalies using cost-effective methods, both the consequences of failures and the associated expenses can be minimized, thereby reducing the operating costs of power plants. Machine learning enhances fault detection and isolation by monitoring, detecting, identifying fault types, and reducing unnecessary procedures. It utilizes algorithms such as supervised learning, unsupervised learning, and reinforcement learning. These algorithms continuously analyze system data to detect faults, determine their types, and locate them, minimizing productivity loss and preventing unexpected incidents. Machine learning optimizes processes and promptly fixes errors via adaptation and improvement, leading to effective system performance. It's a potent instrument for improving workflow and preventing disruptions in system performance via early detection and correction of problems. Partial shading,

module degeneration, and cell fractures are just some of the problems that may be detected and categorized by the system, which makes use of data including voltage, current, and temperature readings [6-12]. Extreme Learning Machines (ELM), a machine learning method with a rapid training speed and strong generalization performance [13], has showed potential in the field of Photovoltaic (PV) defect detection.

In this work, for PV fault detection a different machine learning models were used. The training and testing datasets encompassed normal operation and various fault scenarios is used. The factors such as current, voltage, power, temperature was measured and exploited as features. In addition, four algorithms (Tree, LDA, SVM, and ANN) were tested using 10-fold cross-validation to identify errors in the PV system.

2. External Learning Machine

Extreme Learning Machine (ELM) is a feedforward neural network with a single hidden layer (SLFN) that randomly initializes the input weights and biases, reducing the need for time-consuming iterative training. The hidden layer of the network converts the input data into a high-dimensional feature space, and the output weights are calculated analytically using a least-squares solution. ELM is an appealing solution for PV fault detection applications due to its simplicity and computational efficiency [14]. ELM has been effectively used to a variety of applications in the area of PV fault detection. ELM has been used, for example, to discover and categorize many sorts of problems in PV systems, such as module deterioration, shading, and connection difficulties [15]. By training an ELM model on a dataset of labelled fault data, the algorithm can learn to distinguish between normal and faulty operating conditions with high accuracy. Furthermore, ELM can be employed for fault location estimation in PV arrays.

By utilizing measurements from multiple sensors or monitoring points within the PV system, an ELM-based model can estimate the location of a fault occurrence, facilitating targeted maintenance and repair actions [16-17].

Since, the application of ELM in PV fault detection offers several advantages, including fast training times, low computational requirements, and good generalization performance. Many ELM simulation results on artificial and real applications have rigorously proved that ELM can produce good generalization performance, most of these applications are on image object detection [18–20], time series analysis [21–24], fault detection [25, 26], and remote sensing [27], etc. These characteristics and various applications make ELM suitable for real-time fault detection applications in large-scale PV systems. As a result, the goal of this work is to investigate efficient ELM approach to detect the grid connected PV system.

3. ELM Theory and Feature Mapping

According to ELM theory [13], any nonlinear piecewise continuous activation function $G(w,b,x)$ may be utilized for feature mapping to approximate any continuous target function, as seen in Figure1. Sigmoid, tangential, Gaussian,

and other functions are examples of these functions. The output function of the hidden layer for N input and L hidden nodes is as follows:

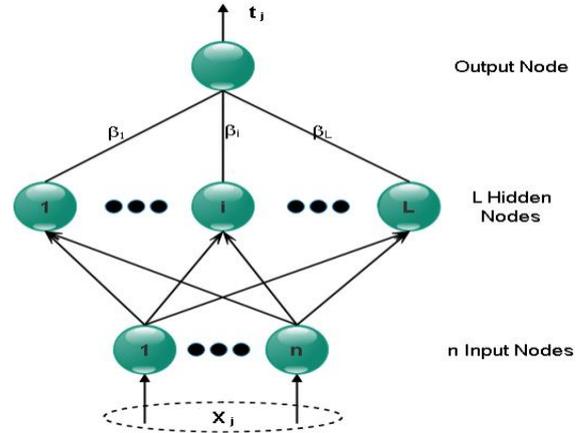


Figure 1: ELM/Single Hidden Layer Feedforward Network Architect.

$$h(x) = [G(w_1, b_1, x_1), \dots, G(w_L, b_L, x_L)] \tag{1}$$

SLFNs have the following output function:

$$t_j = \sum_{i=1}^L \beta_i \cdot G(w_i, b_i, x_j) \tag{2}$$

$j = 1, \dots, N$

ELM may be extended to generalize SLFN using a wide range of feature mapping functions on the hidden layer with a sufficient number of L hidden nodes, as shown in equation 3:

$$\|ft(x) - f(x)\| < \epsilon \tag{3}$$

Where ϵ is the error tolerance value which must be as small as possible.

Consider a data set containing N - training samples $(x_i, t_j) \in R^n \times R^m, j=1, 2, \dots, N$ with n = number of input attributes, m = number of output classes, and $G(x)$ = activation function, the estimated output $y_i = \hat{t}_i$ is mathematically characterized as:

$$f_L(x) = y_i = \hat{t}_j = \sum_{i=1}^L \beta_i \cdot G(w_i x_j + b_i) = t_j - \epsilon_j = h(x) \cdot \beta \tag{4}$$

Minimizing the relative error between \hat{t}_i and t_i is the ELM goal, i.e., $\min \sum_{j=1}^1 \|t_j - \hat{t}_j\|$,

Where β is the output weight vector connecting the i^{th} hidden nodes and the output nodes.

w_i is the input weight vector that was arbitrarily selected to link the i^{th} hidden node to the input nodes of the i^{th} hidden node.

b_i is the randomly selected bias of the i^{th} hidden nodes; x_j the input attributes; y_j the actual output.

In matrix form, equation 2 is equivalent to:

$$H \cdot \beta = T \rightarrow \min_{\beta} \|H \cdot \beta - T\|, \text{ where } \|\cdot\| \text{ stands for the norm in Euclidean distance, then } \beta = H^{\dagger} T \text{ where}$$

$H^\dagger = (H^T H)^{-1} H^T$ is Moore-Penrose generalized pseudo inverse of hidden layer output matrix .

Equation 5 may be used to represent the desired output weights and objectives.

$$\beta = H^\dagger T = (H^T H)^{-1} H^T T \hat{T} = H\beta = H(H^T H + \lambda I)^{-1} H^T T = HAT. T \tag{5}$$

$$H(w_1, \dots, w_L; b_1, \dots, b_L; x_1, \dots, x_N) = \begin{bmatrix} G(w_1, b_1, x_1) & \dots & G(w_L, b_L, x_1) \\ \vdots & \dots & \vdots \\ G(w_1, b_1, x_N) & \dots & G(w_L, b_L, x_N) \end{bmatrix} \tag{6}$$

where,

$$\beta = \begin{bmatrix} \beta_1^1 \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad \& T = \begin{bmatrix} t_1^1 \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}, T_{ij} = \begin{cases} 1 \text{ for vectors of class } i=j \\ 0 \text{ for vectors of class } i \neq j \end{cases} \tag{7}$$

The i^{th} column of T is the output target of the i th hidden nodes with respect to inputs x_1, \dots, x_N . As a result, the ELM algorithm is a learning approach formed by two processes. The first is the mapping process that converts the input space R^n or reduced input space R^d to a usually high-dimensional feature space R^L (noted as ELM space) with preserving some training data properties, i.e, ($X \in R^{N \times n}$ or $X \in R^{N \times d} \Rightarrow H \in R^{N \times L}$). The second is the optimization scheme process that projects high-dimensional feature space R^L to a low projection linear space R^m , i.e, ($H \in R^{N \times L} \Rightarrow T \in R^{N \times m}$).

The following steps describe the typical ELM approach for classification purpose:

ELM Algorithm

Given a training set containing of N -samples as $\mathfrak{X} = \{(x_i, t_i) \mid x_i \in R^n, t_i \in R^m, i = 1, \dots, N\}$, activation function $G(x)$, and hidden nodes number L ;

1. Assign randomly input weight vectors w_i and hidden nodes bias $b_i, i = 1, \dots, L$.
2. Using equation 6, calculate the hidden layer output matrix $H(w_1, \dots, w_L, b_1, \dots, b_L, x_1, \dots, x_N)$.
3. Calculate the output weight β as: $\beta = H^\dagger T$, where $H^\dagger = (H^T H)^{-1} H^T$ or $H^\dagger = H(H^T H + \lambda I)^{-1} H^T$ is the Moore-Penrose generalized inverse of hidden layer output matrix H , and λ is a suitable regularization parameter value.

According to the perspective of evaluation the samples are divided into the training and the testing

sets. The training sets are used firstly for getting the value of the output weight (β), then

4. Employing the pre-calculated β in step 3 to approximate or classify the test patterns using the following equation $y_{test} = h(x) \cdot \beta = H_{test} \beta$ for regression, then apply $L_{test} = \text{argmax}^{row}(y_{test})$ for classification, the arg function returns the index of the maximum value for each row of y_{test} . Where, $L_{test}(m \times 1)$ is the output label of m testing instances.

4. PV System Model

Figure 2 gives the principle of grid-tied photovoltaic power generation (PVPGs).

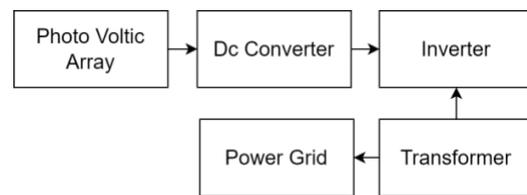


Fig. 2. Structure diagram of grid-tied photovoltaic power generation.

A grid-connected PV system, as illustrated in Figure 3, is a system that generates electricity using solar panels and is connected to the power grid. It operates in conjunction with the grid and can feed excess electricity back into the grid when it generates more power than the connected load requires [28]. The primary components of a grid-connected PV system include PV modules, an inverter, and the grid connection. In photovoltaic modules (PV), solar cells are used to convert sunlight into energy. The inverter is essential because it transforms the DC power produced by the PV modules into grid-friendly AC power.

The system's ability to interchange energy with the power grid is made possible by the grid connection. Any of these parts might fail, decreasing the PV system's efficiency and shortening its lifespan. In order to keep the system running smoothly and error-free, it is crucial that it be regularly inspected and maintained.

The suggested microgrid structure as seen in Figure 3 includes a photovoltaic (PV) system linked to the DC connection through a unidirectional boost converter. In addition to the LC filter, the Point of Common Coupling (PCC) variable load, and the inverter linking the PCC to the main load make up the rest of the system.

The PV system uses sun power SPR-400E model panels, each of which has a capacity of 400 W and 90 cells, and is comprised of 12 panels linked in series and 52 panels connected in parallel. The plant has a total output of 250 kW of electricity.

The suggested microgrid structure includes a photovoltaic (PV) system linked to the DC connection through a unidirectional boost converter. In addition to the LC filter, the PCC variable load, and the inverter linking the PCC to the

main load make up the rest of the system. The PV system uses SunPower SPR-400E model panels, each of which has a capacity of 400 W and 90 cells, and is comprised of 12 panels linked in series and 52 panels connected in parallel. The plant has a total output of 250 kW of electricity. Tables 1 includes a listing of the characteristics and measurements of the PV panels that were used for this study. All simulation investigations were carried out with the help of the MATLAB program.

Table 1 provides the technical specifications of the proposed PV system, including open circuit voltage (V_{OC}), short circuit current (I_{SC}), current at the maximum power point (I_{mpp}), voltage at the maximum power point (V_{mpp}), and the number of series panels (N_s) and parallel panels (N_p) used in the configuration. The total power of the solar power plant is stated as 250 kW.

Table 2 presents the determined parameters of the suggested system, such as rated power, grid line voltage, DC voltage, grid frequency, switching frequency, and various component specifications such as boost capacitor, boost inductance, DC-link capacitor, filter capacitor, and bi-directional inductance.

The production of electricity by the solar power plant ranges from 200 kW to 250 kW to meet the demand load, and MATLAB was used for all simulation investigations.

Table 1. The proposed PV system technical specifications

Symbol	Description	Value
V_{oc}	Open circuit Voltage	85.3V
I_{sc}	Short circuit current	5.87A
I_{mpp}	Current at maximum power point	5.49A
V_{mpp}	Voltage at maximum power point	72.9V
M_{pp}	Max Power point	400W
N_s	Number of series panels	12
N_p	Number of parallel panels	52
P_{total}	Total power of solar power plant	250KW

5. Dataset Preparation

To create training and testing datasets for PV defect scenarios, a simulated 250kW PV power system was created. The system was designed to replicate normal operation without any faults (referred to as F0). Additionally, three distinct types of faults were introduced, as illustrated in Figure 4. These faults include string-to-string faults (F1), on-string faults (F2), and string-to-ground faults (F3). Each fault category represents a predefined set of specific fault conditions within the PV system. The training and testing datasets were built on the DC side of the PV system, including both normal operation and different forms of PV failures. The following are the individual instances contained in the datasets:

➤ Fault-free (F0): This scenario illustrates the PV system operating normally with no defects or irregularities.

All components are operating at peak efficiency, resulting in the projected power output.

➤ String-to-string fault (F1): This problem arises when the wiring between two strings of PV modules fails. It may be caused by defective connectors, loose connections, or broken wiring. As a result, the afflicted string's power output may be diminished or possibly lost entirely.

➤ On-string fault (F2): An on-string fault is defined as a malfunction in one or more PV modules inside a string. This might be due to a faulty module, a faulty bypass diode, or shading difficulties. The afflicted module(s) may produce much less or no power at all, affecting the string's overall performance.

➤ String-to-ground fault (F3): This problem occurs when the wiring connecting a string of PV modules to the ground fails. It can be caused by a broken grounding wire or an inadequate grounding electrode. In addition to potential safety concerns, such a fault can result in limited or no power production from the affected string. These represent some of the possible PV defects that can occur, and they can have a significant impact on the efficiency and performance of a PV system.

The dataset consists of 701 instances, each with 26 features and one column for the fault classes; training data consists of 603×27, while testing data consists of the remaining 98×27. The simulations will account for most of the I-V characteristics curve of the PV array under varying environmental conditions.

The distribution of PV simulation datasets with all failure types is shown in Table 3. The dataset used for defect identification includes various electrical features such as minimum, maximum, average, and range values of currents, voltages, and powers.

Additionally, environmental features such as temperature and irradiation are included. The temperature (T) ranges from 5°C to 45°C with a resolution of 5°C, while light irradiation (G) ranges from 50 W/m² to 1000 W/m² with a step size of 50 W/m². To evaluate the accuracy of the suggested models in defect identification, a 10-fold cross-validation approach is employed [29-34]. The distribution of the PV simulation datasets, including all types of failures, is presented in Table 3.

Table 2. The functional parameters of the microgrid PV system.

Parameters	Value	Parameters	Value
Rated power	200kW	Boost capacitor	3mF
Grid line voltage	400V	Boost inductance	0.8mH
DC-Voltage	800 V	DC-link capacitor	5mF
Grid frequency	50 Hz	Filter capacitor	40µF
Switching frequency	10kHz	Bi-directional inductance	0.3mH

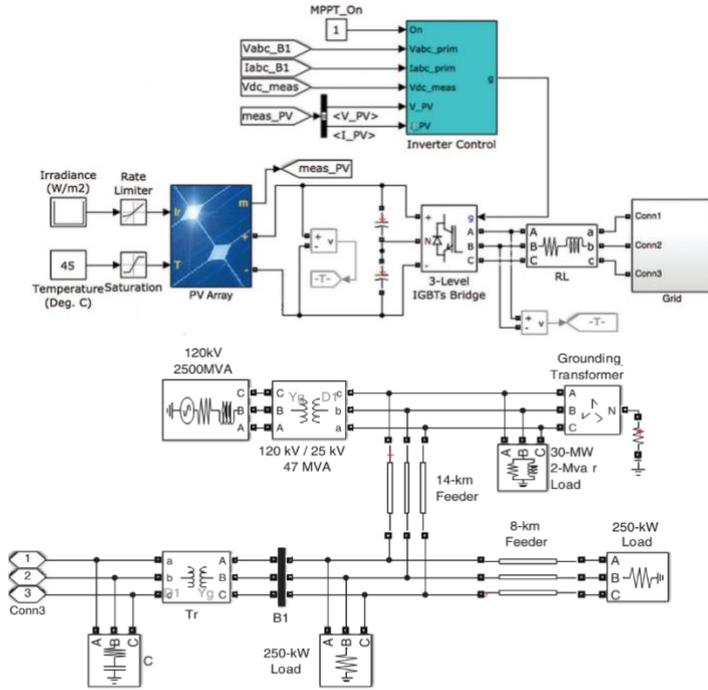


Fig.3. The grid PV system block diagram.

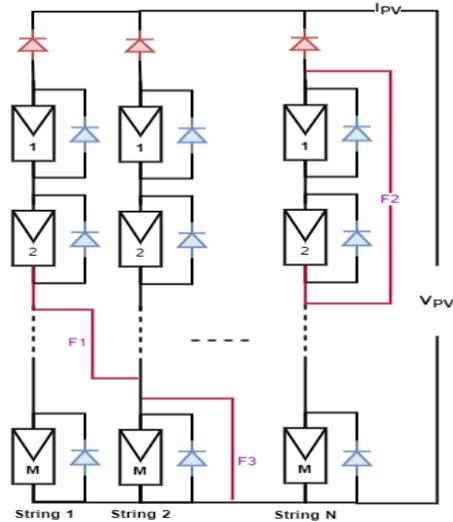


Fig. 4. Three different faults in MxN PV array.

Table 3. The PV simulation dataset distribution with various fault types.

Fault Type	Nominal or class	size
Fault free		
string-to-string fault	F0	133x27
on-string fault	F1	172x27
string-to-ground	F2	174x27
	F3	222x27
Total data=701x27, training part=603x27, testing part=98x27		

6. Data Preprocessing

Many cells of our dataset are missing, needing to use imputation technique to estimate and fill these missing values in the dataset. One common approach is using the k-nearest neighbors (KNN) algorithm for imputation. In KNN imputation, the missing values are replaced with estimated values based on the values of their nearest neighbors. To perform missing imputation using KNN, the following steps are typically followed [35]:

- Identify the features with missing values in the dataset.
- For each instance with missing values, find the k-nearest neighbors based on the available features. The choice of k depends on the dataset and can be determined through experimentation or domain knowledge.
- Calculate a distance metric (e.g., Euclidean distance) between the instance with missing values and its k nearest neighbors.
- Weight the values of the nearest neighbors based on their distances. Typically, the closer neighbors have higher weights.
- Calculate the imputed value by taking the weighted average or majority vote of the values from the k nearest neighbors.
- Replace the missing values in the dataset with the imputed values. In the context of ELM, normalization serves multiple purposes.

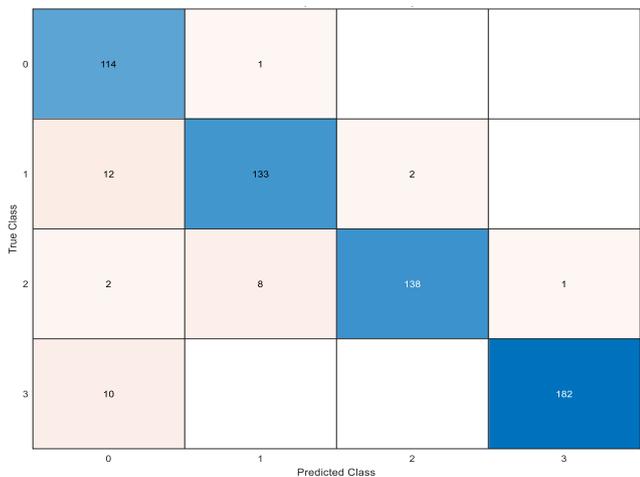
Firstly, it helps to avoid features with larger values dominating the learning process, ensuring that all features are treated equally. Secondly, it prevents numerical instability and improves the convergence of the ELM algorithm by scaling the input data. Finally, normalization enhances the interpretability of the model and reduces the influence of outliers or extreme values [36].

By applying min-max normalization to the input data in ELM, the features are brought to a consistent scale between 0 and 1, enabling fair comparisons and effective analysis. This normalization step contributes to the overall performance and stability of the ELM model.

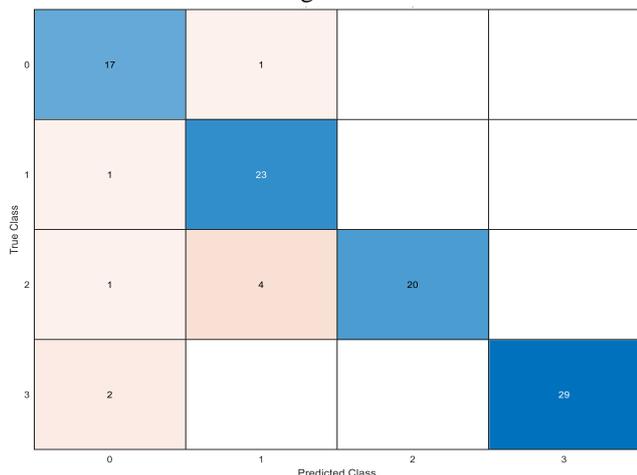
7. Experiential Results

The all proposed models of the PV systems and all machine learning approaches were designed using MATLAB software environment version 2021b. The training and testing confusion resultant matrices obtained from the four models (LDA, SVM, ANN, and ELM) are detailed in Figures (5-8), Table 4 explains the performance metrics including training accuracy, training time, and testing accuracy. The hyper-parameter assumptions for the ELM classifier are: the first layer is of size 10, hidden layer is of size L=100 or L=200 with sigmoid activation function, and the output layer is of size

4. While the ANN hyper-parameter with the first layer is of size 25, first hidden layer is of size 40, second hidden layer is of size 20 with sigmoid activation functions, and the output layer is of size 4, and the iteration limit with 1000 epochs.

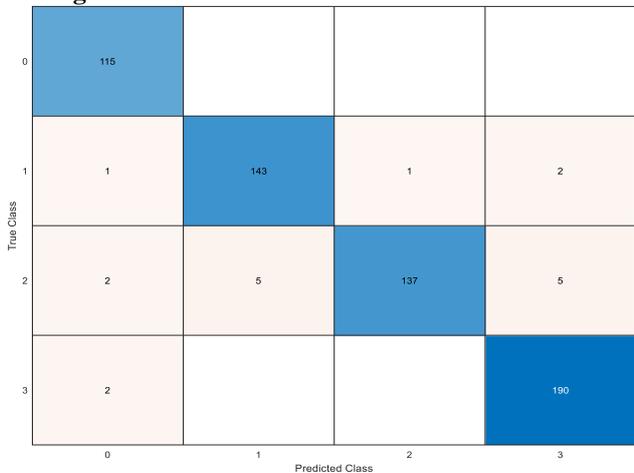


Training confusion matrix

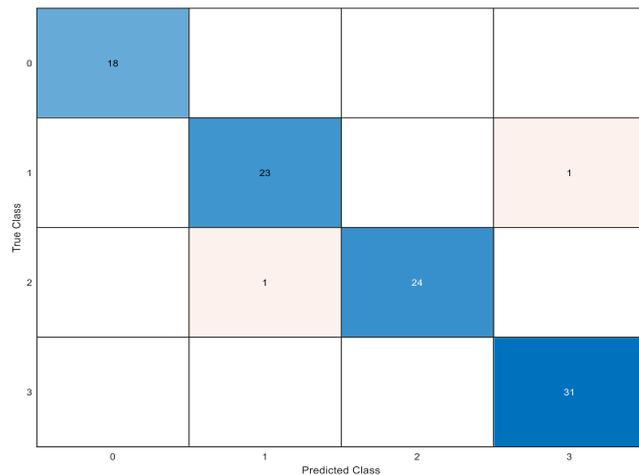


Testing confusion matrix

Fig. 5. Confusion matrix for the LDA classifier.

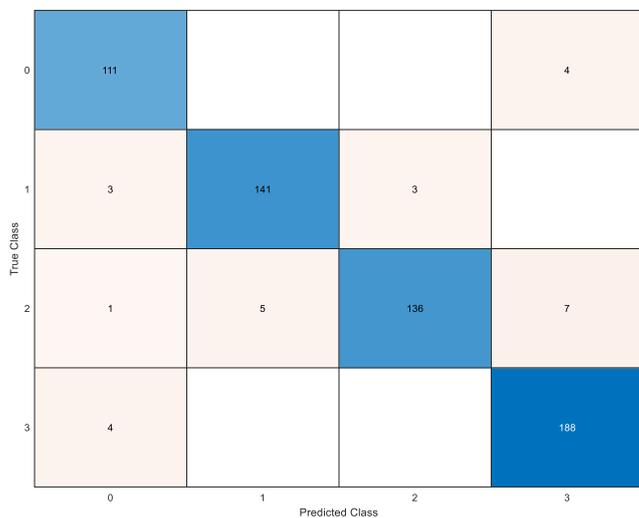


Training confusion matrix

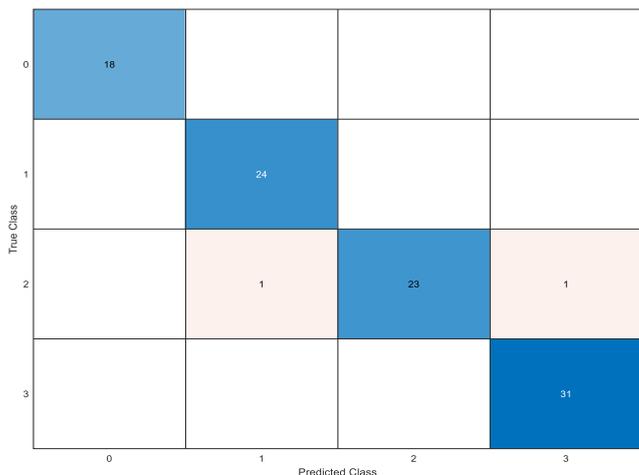


Testing confusion matrix

Fig. 6: Confusion matrix for the SVM classifier

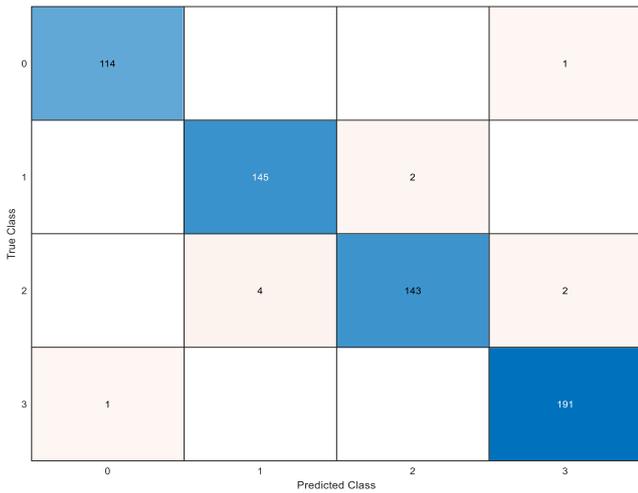


Training confusion matrix

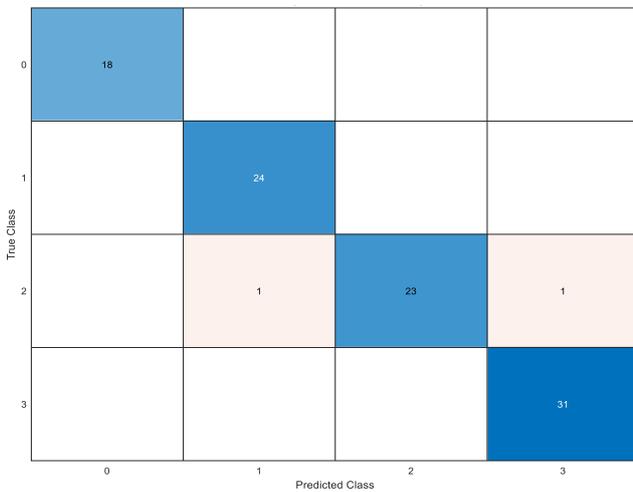


Testing confusion matrix

Fig. 7. Confusion matrix for the ANN classifier



Training confusion matrix



Testing confusion matrix

Fig. 8. Confusion matrix for the ELM (L=100) classifier

Table 4 shows each model's accuracy without any feature selection on the validation and test sets, as well as the training time for each model.

Table 4. All classifiers performances

Session: Classification normalization without Feature Selection				
Training Data: x Observations: 603				
Predictors: 26				
Response Name: column 27 Response Classes: 4				
Response Class Names: 0, 1, 2, 3				
Validation: 10-fold cross validation				
Test Data: x				
Observations: 98				
Model No.	Model Type	Accuracy % (Validation)	Accuracy % (Test)	Training Time (sec)

1	ELM (L=100)	98.3416	97.959 2	0.106 3
1	ELM (L=200)	99.1663	97.972 7	0.170 9
2	LDA	94.0298	90.816 3	4.985 4
3	SVM	97.0149	96.459 1	3.887 1
4	Neural Network	95.5224	94.938 8	2.306 3

The training and testing accuracies are illustrated as in Figure 9, while the training time is as shown in Figure 10. The classification results indicate the performance of various models on a dataset consisting of 603 observations and 26 predictors, with the goal of predicting a response variable with four classes. The ELM models showed exceptional accuracy, with the ELM (L=200) model achieving a perfect accuracy of 99.1663% on the validation data and 97.9727% on the test data, trained in 0.1709 seconds. The LDA, SVM, and neural network models also demonstrated good accuracies, although slightly lower than the ELM models. Notably, the LDA model had the longest training time at 4.9854 seconds, while the ELM models trained the fastest.

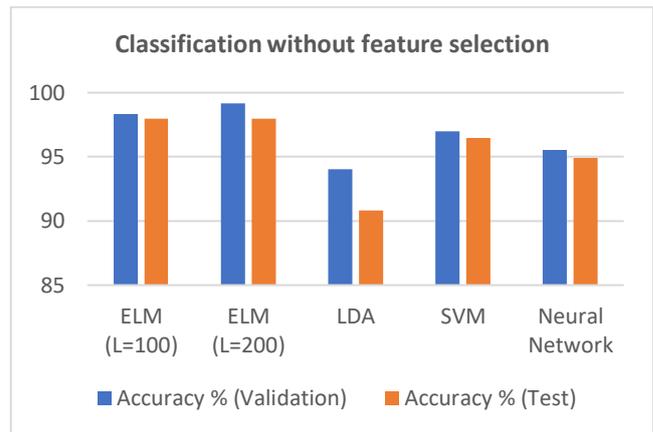


Fig. 9. The training and testing accuracies of all proposed classifiers without any feature selections.

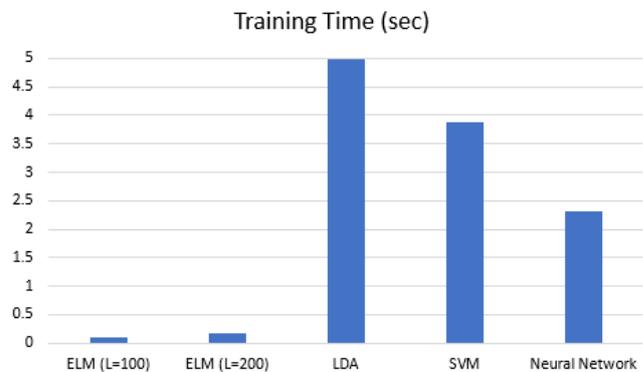


Fig. 10. The training time of all proposed classifiers without any feature selections.

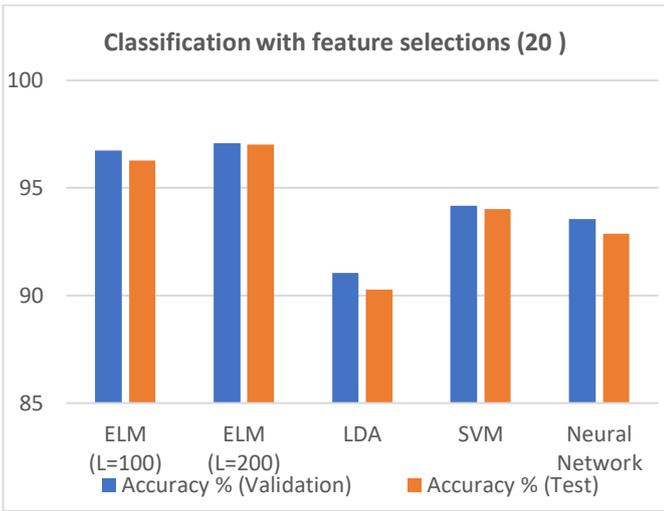


Fig.11. The training and testing accuracies of all proposed classifiers with 20 feature selections.

When performing feature selection using the Analysis of Variance (ANOVA) [37] methodology as an example to choose the top 20 features to lowering the dimensionality from 26 features to 20 or 10 features, the accuracy of a classifier method might fluctuate depending on the dataset and context. Table 5 and Figures 11&12 show each model's accuracy with feature selection with 20 features on the validation and test sets, as well as the training time for each model.

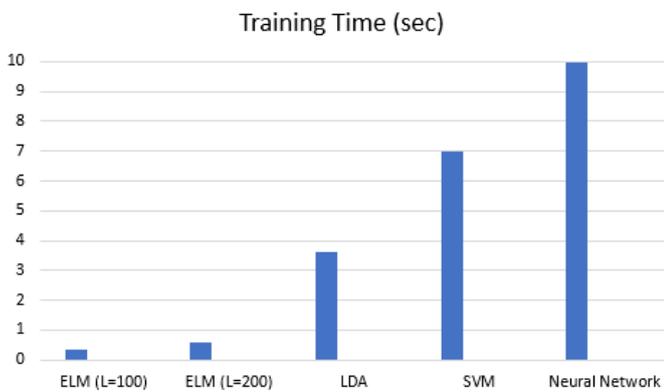


Fig. 12. The training time of all proposed classifiers with 20 feature selections.

Table 5. All classifiers' performances with feature selection of 20 features.

Session: Classification_normalization with ANOVA (20 features)				
Training Data: x Observations: 603				
Predictors: 20				
Response Name: Column 21 Response Classes: 4				
Response Class Names: 0, 1, 2, 3				
Validation: 10-fold cross-validation				

Test Data: x				
Observations: 98				
Model No.	Model Type	Accuracy % (Validation)	Accuracy % (Test)	Training Time (sec)
1	ELM (L=100)	96.7538	96.2791	0.3285
1	ELM (L=200)	97.0803	97.0206	0.5832
2	LDA	91.0447	90.2775	3.6017
3	SVM	94.1907	94.0081	7.0069
4	Neural Network	93.5441	92.8751	9.9881

The findings reveal that various categorization models have varied accuracies and training timeframes. The ELM models (with 100 and 200 hidden nodes) maintained excellent accuracy, with the ELM (L=200) model reaching near-perfect validation accuracy of 97.0803% and test data accuracy of 97.0206%. However, when compared to the prior findings, the accuracy of the LDA model was lower. The accuracy of the SVM and neural network models ranged from moderate to high, with the neural network taking the most training time. These results emphasize the necessity of addressing dataset-specific criteria while choosing and assessing features, as well as the influence of feature selection on classifier performance. Table 6 and Figures 13&14 show each model's accuracy with feature selection with 10 features on the validation and test sets, as well as the training time for each model.

Table 6. All classifiers performances with feature selection of 10 features.

Session: Classification normalization with ANOVA (10 features)				
Training Data: x Observations: 603				
Predictors: 10				
Response Name: column_21 Response Classes: 4				
Response Class Names: 0, 1, 2, 3				
Validation: 10-fold cross-validation				
Test Data: x Observations: 98				
Model No.	Model Type	Accuracy % (Validation)	Accuracy % (Test)	Training Time (sec)
1	ELM (L=100)	92.1483	92.1154	0.5644
1	ELM (L=200)	93.6504	93.2432	0.7390
2	LDA	50	49.5854	3.4078

3	SVM	72.1393	70.5306	18.548
4	Neural Network	91.6981	90.8979	20.937

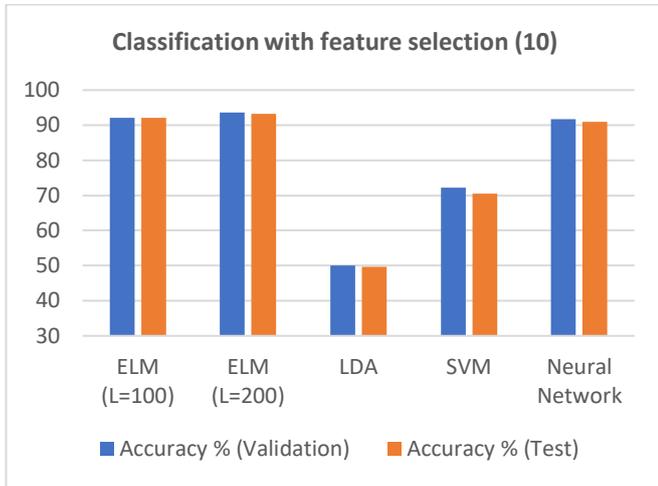


Fig. 13. The training and testing accuracies of all proposed classifiers with 10 feature selections.

In this amended analysis, the number of predictors was reduced from 26 to 10 by feature selection using ANOVA. The results indicate that classification models have varying degrees of precision and training periods.

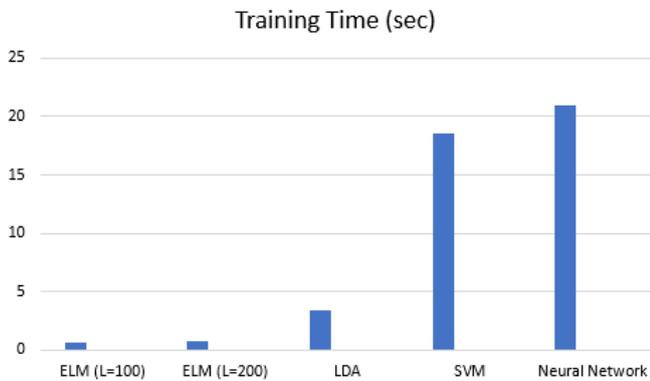


Fig 14. The training time of all proposed classifiers with 10 feature selections.

The ELM models with 100 and 200 concealed nodes demonstrated relatively high accuracies, with the ELM (L=200) model achieving validation and test accuracies of 93.654% and 93.2434%, respectively. However, the LDA model's accuracy decreased significantly compared to the previous results. The SVM and neural network models exhibited moderate to high levels of accuracy, with the neural network requiring more time to train. These results underscore the impact of feature selection on classifier performance and emphasize the need to take dataset-specific factors into account when selecting and evaluating features.

The accuracy percentages indicate the efficacy of the models on training and assessment datasets, with ELM models attaining high accuracy on average. However, after feature selection, the accuracy of the majority of models decreases. Training durations for all models are brief. These results underscore the importance of feature selection and the need for additional analysis and optimization to improve the performance of the models.

8. Conclusion

This study contributes to the field of fault detection in grid-connected photovoltaic systems by examining the efficacy of Extreme Learning Machines (ELM) as a fault detection technique. The results emphasize the significant benefits of ELM, such as its quick training periods, minimal computational requirements, and excellent generalization performance. ELM's application in identifying and classifying various categories of defects in PV systems, such as module degradation, shading, and connection problems, demonstrates its potential as a cost-effective and efficient method for prompt fault detection.

Solar power plant operators can minimize the financial impact of failures and optimize system performance by promptly detecting and classifying defects. ELM's enhanced utility in large-scale PV systems is a result of its capacity to facilitate targeted maintenance and repair actions. By utilizing ELM, solar power plant operators are able to detect and resolve defects in real-time, thereby reducing operational expenses and ensuring the system's seamless operation. The findings of this study highlight the importance of machine learning approaches, such as ELM, for increasing fault detection and operational efficacy in renewable energy systems. Future research can investigate the application of ELM in conjunction with other defect detection techniques to further improve its performance and its scalability in larger PV systems. Overall, the use of ELM as a defect detection method offers promising prospects for the dependable and cost-effective operation of photovoltaic systems that are grid-connected. Future research directions in the field of defect detection in grid-connected photovoltaic (PV) systems may include the following: Integration of Multiple Defect Detection Methods: Investigate integrating ELM with other defect detection methods to enhance precision and reliability.

In order to enhance the efficacy of the defect detection system, it is important to explore sophisticated feature selection approaches such as genetic algorithms and particle swarm optimization. It is important to assess the efficacy of ELM-based fault detection in the context of bigger solar systems. Furthermore, it is advisable to include real-time monitoring and adaptive learning capabilities in order to improve responsiveness and flexibility [38]. Finally, it is important to confirm the effectiveness of the system by subjecting it to rigorous testing using varied data sets across a range of operational situations like involving PID controller with adaptive algorithms [39], or fractional order PI [40], or adopting fast fuzzy-neural systems [41], etc.

References

- [1] Kenneth E. Okedu, Kenneth E. Okedu, A. Al Senaidi, Al Hajr, "Real Time Dynamic Analysis of Solar PV Integration for Energy Optimization," *International Journal of Smart grid*, vol.4, June, 2020.
- [2] L. Hernández-Callejo, S. Gallardo-Saavedra, and V. Alonso-Gómez, "A review of photovoltaic systems: Design, operation and maintenance," *Solar Energy*, vol. 188, 2019.
- [3] L. Tightiz, S. Mansouri, F. Zishan, J. Yoo, and N. Shafaghatian, "Maximum Power Point Tracking for Photovoltaic Systems Operating under Partially Shaded Conditions Using SALP Swarm Algorithm," *Energies (Basel)*, vol. 15, no. 21, 2022.
- [4] L. Hocine, K. M. Samira, M. Tarek, N. Salah, and K. Samia, "Automatic detection of faults in a photovoltaic power plant based on the observation of degradation indicators," *Renew Energy*, vol. 164, 2021.
- [5] K. Abdulmawjood, S. S. Refaat, W. G. Morsi, "Detection and prediction of faults in photovoltaic arrays: A review", In *Proceedings - 2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG2018)*, pp. 1–8, June 2018.
- [6] R. Fazai, K. Abodayeh, M. Mansouri, M. Trabelsi, H. Nounou, M. Nounou, G. E. Georghiou, "Machine learning-based statistical testing hypothesis for fault detection in photovoltaic systems", *Solar Energy*, Vol. 190, pp. 405-413, 2019.
- [7] M. Hajji, M. F. Harkat, A. Kouadri, K. Abodayeh, M. Mansouri, H. Nounou, M. Nounou, "Multivariate feature extraction based supervised machine learning for fault detection and diagnosis in photovoltaic systems", *European Journal of Control*, Vol. 59, pp. 313-321, 2021.
- [8] K. Pahwa, M. Sharma, M. S. Saggu, A. K. Mandpura, "Performance evaluation of machine learning techniques for fault detection and classification in PV array systems", In *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 791-796, February 2020, IEEE.
- [9] R. T. Cabeza, A. S. Potts, "Fault diagnosis and isolation based on Neuro-Fuzzy models applied to a photovoltaic system", *IFAC-Papers On Line*, Vol. 54, No. 14, pp. 358-363, 2021.
- [10] K. Pahwa, M. Sharma, M. S. Saggu, A. K. Mandpura, "Performance evaluation of machine learning techniques for fault detection and classification in PV array systems", In *2020 7th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 791-796, February 2020, IEEE.
- [11] A. Mellit, S. Kalogirou, "Artificial intelligence and internet of things to improve efficacy of diagnosis and remote sensing of solar photovoltaic systems: Challenges, recommendations and future directions", *Renewable and Sustainable Energy Reviews*, Vol. 143, Article No. 110889, 2021.
- [12] Z. Mustafa, A. S. Awad, M. Azzouz, A. Azab, "Fault identification for photovoltaic systems using a multi-output deep learning approach", *Expert Systems with Applications*, Vol. 211, no. 118551, 2023.
- [13] G. B. Huang et al., "Extreme learning machine: Theory and applications", *Neurocomputing*, Vol. 70, No. 1-3, pp. 489-501, 2006.
- [14] Y. Wu, Z. Chen, L. Wu, P. Lin, S. Cheng, P. Lu, "An intelligent fault diagnosis approach for PV array based on SA-RBF kernel extreme learning machine", *Energy Procedia*, Vol. 105, pp. 1070-1076, 2017.
- [15] Z. Chen, L. Wu, S. Cheng, P. Lin, Y. Wu, W. Lin, "Intelligent fault diagnosis of photovoltaic arrays based on optimized kernel extreme learning machine and IV characteristics", *Applied Energy*, Vol. 204, pp. 912-931, 2017..
- [16] Y. Sun, J. Wang, Q. Yang, X. Li, W. Yan, "Fault Diagnosis of Photovoltaic Module Based on Extreme Learning Machine Technique", *Proceedings of the IMCIC*, 2018.
- [17] M. Ahmadipour, M. M. Othman, M. Alrifayy, R. Bo, C. K. Ang, "Classification of faults in grid-connected photovoltaic system based on wavelet packet transform and an equilibrium optimization algorithm-extreme learning machine", *Measurement*, Vol. 197, Article No. 111338, 2022. (Article)
- [18] Y. Bazi, N. Alajlan, F. Melgani, H. AlHichri, S. Malek, R. R. Yager, "Differential Evolution Extreme Learning Machine for the Classification of Hyperspectral Images", *IEEE Geoscience and Remote Sensing Letters*, Vol. 11, No. 6, pp. 1066–1070, 2014.
- [19] M. Abd Shehab, N. Kahraman, "Optimum, projected, and regularized extreme learning machine methods with singular value decomposition and Tikhonov regularization", *Turkish Journal of Electrical Engineering and Computer Science*, Vol. 26, No. 4, pp. 1685-1697, 2018.
- M. Abd Shehab, N. Kahraman, "A weighted voting ensemble of efficient regularized extreme learning machine", *Computers & Electrical Engineering*, Vol. 85, Article No. 106639, 2020.
- [20] D. Sovilj, A. Sorjamaa, Q. Yu, Y. Miche, E. Séverin, "OPELM and OPKNN in Long-Term Prediction of Time Series using Projected Input Data", *Neurocomputing*, Vol. 73, No. 10–12, pp. 1976–1986, 2010.
- [21] J. B. Butcher, D. Verstraeten, B. Schrauwen, C. R. Day, P. W. Haycock, "Reservoir Computing and Extreme Learning Machines for Non-Linear Time-

- Series Data Analysis", *Neural Networks*, Vol. 38, pp. 76–89, 2013.
- [22] X. Wang, M. Han, "Multivariate Time Series Prediction Based on Multiple Kernel Extreme Learning Machine", In *Neural Networks (IJCNN)*, 2014 International Joint Conference on IEEE, pp. 198–201, 2014.
- [23] Z. Hong-Li, "Incremental Regularized Extreme Learning Machine Based on Cholesky Factorization and Its Application to Time Series Prediction [J]", *Acta Physica Sinica*, Vol. 11, 2011.
- [24] H. Yang, W. Xu, J. Zhao, D. Wang, Z. Dong, "Predicting the Probability of Ice Storm Damages to Electricity Transmission Facilities Based on ELM and Copula Function", *Neurocomputing*, Vol. 74, No. 16, pp. 2573–2581, 2011.
- [25] D. Martínez-Rego, O. Fontenla-Romero, B. Pérez-Sánchez, A. Alonso-Betanzos, "Fault Prognosis of Mechanical Components Using on-Line Learning Neural Networks", In *International Conference on Artificial Neural Networks*, Springer, pp. 60–66, 2010.
- [26] M. Pal, A. E. Maxwell, T. A. Warner, "Kernel-Based Extreme Learning Machine for Remote-Sensing Image Classification", *Remote Sensing Letters*, Vol. 4, No. 9, pp. 853–862, 2013.
- [27] O. A. Ahmed, W. H. Habeeb, D. Y. Mahmood, K. A. Jalal, H. K. Sayed, "Design and performance analysis of 250 kw grid-connected photovoltaic system in iraqi environment using pvsyst software", *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, Vol. 7, No. 3, pp. 415-421, 2019.
- [28] S. R. Madeti, S. N. Singh, "A comprehensive study on different types of faults and detection techniques for solar photovoltaic system", *Solar Energy*, Vol. 158, pp. 161–185, Dec. 2017.
- [29] S. S. Ghoneim, A. E. Rashed, N. I. Elkalashy, "Fault Detection Algorithms for Achieving Service Continuity in Photovoltaic Farms", *Intelligent Automation & Soft Computing*, Vol. 30, No. 2, 2021.
- [30] A. Triki-Lahiani, A. Bennani-Ben Abdelghani, I. Slama-Belkhodja, "Fault detection and monitoring systems for photovoltaic installations: A review", *Renewable and Sustainable Energy Reviews*, Vol. 82, pp. 2680–2692, Feb. 2018, DOI: <https://doi.org/10.1016/j.rser.2017.09.101>. (Article)
- [31] Z. Yi, A. H. Etemadi, "Fault Detection for Photovoltaic Systems Based on Multi-Resolution Signal Decomposition and Fuzzy Inference Systems", *IEEE Transactions on Smart Grid*, Vol. 8, No. 3, pp. 1274–1283, May 2017, DOI: <https://doi.org/10.1109/tsg.2016.2587244>.
- [32] M. H. Ali, A. Rabhi, A. E. Hajjaji, G. M. Tina, "Real Time Fault Detection in Photovoltaic Systems", *Energy Procedia*, Vol. 111, pp. 914–923, Mar. 2017, DOI: <https://doi.org/10.1016/j.egypro.2017.03.254>.
- [33] R. F. Colmenares-Quintero, E. Rojas, F. Macho-Hernantes, K. E. Stansfield, J. C. Colmenares, "Methodology for automatic fault detection in photovoltaic arrays from artificial neural networks", Vol. 8, No. 1, Sep. 2021, DOI: <https://doi.org/10.1080/23311916.2021.1981520>.
- [34] C. Jiang, Z. Yang, "CKNNI: an improved knn-based missing value handling technique", In *Advanced Intelligent Computing Theories and Applications: 11th International Conference, ICIC 2015, Fuzhou, China, August 20-23, 2015*