Article

Machine Learning Schemes for Anomaly Detection in Solar Power Plants

Mariam Ibrahim [1], Ahmad Alsheikh [2], Feras M. Awaysheh [3,*], and Mohammad Dahman Alshehri[4]

[1] Department of Mechatronics Eng., German Jordanian University, Amman 11180, Jordan; mariam.wajdi@gju.edu.jo
[2] Department of Natural Science & Industrial engineering, Deggendorf Institute of Technology, Deggendorf, Germany
[3] Institute of Computer Science, Delta Center, University of Tartu, Estonia; feras.awaysheh@ut.ee, 0000-0002-9561-6099
[4] Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; alshehri@tu.edu.sa
* Correspondence author: feras.awaysheh@ut.ee;

Abstract: The rapid industrial growth in solar energy is gaining increasing interest in renewable power from smart grids and plants. Anomaly detection in photovoltaic (PV) systems is a demanding task. In this sense, it is vital to utilize recent advances in machine learning to accurately and timely detect different anomalies and condition monitoring. This paper addresses this issue by evaluating different machine learning techniques and schemes and showing how to apply these approaches to solve anomaly detection and detect faults on photovoltaic components. For this, we apply distinct state-of-the-art machine learning techniques (AutoEncoder Long Short-Term Memory (AE-LSTM), Facebook-Prophet, and Isolation Forest) to detect faults/anomalies and evaluate their performance. These models shall identify the PV system's healthy and abnormal actual behaviors. Our results provide clear insights to make an informed decision, especially with experimental *trade-offs* for such complex solution space.

Keywords: anomaly detection; machine learning; time series analysis; correlation

1. Introduction

For the past decade, rapid development and expansion of renewable energy have been explored, including power plants. Such development is expected to advance our abilities to produce clean and affordable energy, creating economic growth. As so, solar power generation challenges have attracted significant attention recently. A leading concern is detecting and localizing anomalous patterns within the solar systems. Big data [4], and data-driven techniques highly assist in detecting and preventing such anomalies and detect faults on photovoltaic (PV) components.

The scalable and coherent functionality of PV systems needs advanced tools to monitor the system parameters' dynamic evolution and release alerts about anomalies to decision-makers. Online monitoring of PV systems is technically beneficial to assist operators in managing their plants and establishing economic assimilation into smart grids [1]. Disastrous faults in photovoltaic (PV) arrays, if not appropriately identified, will accordingly diminish the generated power and indeed introduce fire hazards [2]. After anomalies appear on the surface of solar panels, if panel holders know the existence of the anomalies in time, they can eliminate the anomalies to prevent more energy loss [3]. Thus, quick and precise anomaly detection methods are significant to enhance the performance, reliability, and safety of PV plants.

PV systems usually run trivial as a result of various forms of anomalies. These anomalies are either internal or external [5]. Internal PV system faults arise from the PV system itself causing daytime zero-production. Common faults are failure in a component, system isolation, inverter shutdown, shading, and inverter maximum power point [22]. External factors do not emerge from the PV system and still undermine its electricity generation. Shading, humidity, dust, and temperature are considered the major external anomalies affecting the PV system production [5].

In this work, we compare three machine learning techniques.

The significant contributions of the paper are summarized as follows.

1. The investigation of the anomaly's detection accuracy and performance of three popular models: Facebook prophet, Autoencoder LSTM (AE-LSTM), and Isolation Forest. This model is done by designing comparison tests by optimizing all their possible hyperparameters.
2. Define and classify the internal and external factors that induce anomalies in the PV power plant, and investigate their effect on the model's accuracy, as well the study of the correlation effect and its impact on detecting anomalies.

The remainder of this paper is assembled as follows. Section 2 discuss the paper background and related work. Section 3 describes the used machine learning algorithms. Section 4 describes the collected data sets. Section 5 presents the experimental results and parameters optimization. Finally, we draw our conclusions and presents some future directions at Section 6.

2. Related Work

Several works have investigated anomaly detection techniques in Photovoltaic (PV) power systems. For instance, [3] compared multiple methods to detect and classify anomalies containing the auto-regressive integrated moving average model (ARIMA), neural networks, support vector machines, and k-nearest-neighbors classification. They established that anomaly classification using the k-nearest-neighbors could precisely detect and classify 97% of the anomalies in their test set. In [6] authors implemented an anomaly detection system and predictive maintenance model in PV systems. The model is implemented to anticipate the AC power generation built on an Artificial Neural Network (ANN), which determines the AC power generation utilizing solar irradiance and temperature of PV panel data. The model had a validation error of 2.3%, and the predictive anomaly detection rate was better than 90

A model-based anomaly detection technique is proposed by [7] for inspecting the DC part of PV plants and momentary shading. Initially, a model based on the one-diode model is composed to outline the ordinary nature of the supervised PV system and form residuals for fault detection. Next, a one-class Support Vector Machine (1-SVM) process is implemented to residuals starting with the running model for fault disclosure. [8] presented SunDown, a sensorless method for detecting per-panel faults in solar arrays. SunDown's model-driven method influences interactions among the power generated by adjoining panels to detect disparities from anticipated nature. The model can manage simultaneous faults in many panels and classify anomalies to decide possible sources counting from snow, leaves, and debris, and electrical failures.

A new tool (called ISDIPV) is presented by [9], capable of detecting anomalies and diagnosing them in a PV solar power plant. It includes three fundamental operational items for data acquisition, anomaly detection, and diagnosis of the detected disparities regarding regular performance. Two forms of modeling methods were implemented to describe the ordinary performance anticipated: Linear Transfer Functions (LTF) and neural networks models based on multi-layer perceptrons (MLP). [10] presented a data-driven answer for adequate anomaly detection and classification, which applied PV string currents as signs to detect and classify PV systems anomalies. The proposed anomaly detection approach used unsupervised machine learning techniques. The approach included two stages, particularly local context-aware detection (LCAD) and global context-aware anomaly detection (GCAD).

Anomalies related to TeleInfra base stations' fuel consumption were detected by [11] in the registered data utilizing the generator as an origin of power. Anomalies were detected by learning the patterns of the fuel consumption applying four classification methods, particularly: Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Multilayer Perceptron (MLP). The outcomes of this study illustrated that MLP has the best performance in the interpretation measurement, registering a score of 96

A new technique is presented by [1] for monitoring PV systems by detecting anomalies using "kNearest-Neighbours (kNN)" and "one-class support vector machine (OCSVM)". The Self-learning algorithms markedly decreased the measuring exertion and supported reliable monitoring of faults. The authors of [12] used a k-Nearest-Neighbours algorithm and a Multi-layer Perceptron to process the data from a DC sensor and detect differing attributes of the electrical current. A sensorless detection method is presented by [2] controlled by the rapid current decline enclosed by two maximum power point tracking (MPPT) sampling moments in PV plants. Simulations have been executed to validate its possibility to determine anomalies against fluctuating environments, regardless of the discrepancy and irradiance ranks.

An anomaly detection framework of monocrystalline solar cells was proposed by [13]. The framework is branched into two stages: In the first stage, an anomaly detection model based on a Generative Adversarial Network (GAN) is used. This model permits the detection of anomalous patterns using only non-defective samples for training. In a second stage, as defective samples appear, the detected anomalies will be employed as automatically produced annotations for the supervised training of a Fully Convolutional Network.

An analytical scheme is presented by [14] for online investigation of the raw video streams of aerial thermography. This scheme combined image processing and statistical machine learning methods. The presented scheme depended on Robust Principal Component Analysis (RPCA), which is utilized on PV images for concurrent detection and confinement of anomalies. In addition to RPCA, post-processing procedures are proposed for image noise reduction and segmentation. Distinct models were chosen by [15] for the energy yield data examination counting: linear models, proximity-based models, probabilistic models, anomaly ensembles, and neural networks. All models were optimized with an empirical parameter adjusted to get the correct outcomes desirable for the data Locally. The models based on neural networks were at the head of the other models in the detection rate.

SolarClique, a data-driven method, is considered by [16] to detect anomalies in the power generation of a solar installation. The method doesn't need any sensor apparatus for fault/anomaly detection. Instead, it exclusively needs the assembly output of the array and those of close arrays for operating anomaly detection. An anomaly detection technique utilizing a semi-supervision learning model is suggested by [17] to pre-determine solar panel conditions for bypassing the circumstance that the solar panel cannot produce power precisely due to equipment damages. This method utilized the clustering model for regular actions filtration and then followed the neuron network model, Autoencoder, to establish the classification.

A general, unsupervised and scalable scheme is presented by [18] to detect anomalies in time series data that can run offline and online. The scheme composed of a rebuilding model following a variational autoencoder. Both encoder and decoder were parametrized with recurrent neural networks to recognize the temporal reliance of time series data. The outcomes illustrated that the model could detect anomalous arrangements by utilizing probabilistic restoration metrics like anomaly scores. [19] proposed a new ensemble model anomaly detection approach with non-linear regression models and anomaly scores following correlation analysis used for cyber-physical intrusion detection in smart grids.

The unsupervised contextual and collective detection approach is utilized by [20] to data streams from a large energy distributor in the Czech Republic. The approach examined distinctive forms of potential anomalies (e.g., over-voltages, under-voltages).

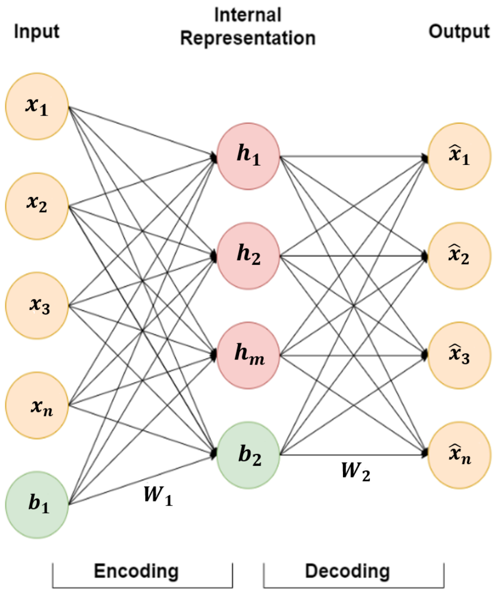


Figure 1. The AutoEncoder (AE) model.

Common item-set mining and categorical clustering techniques were used along with clustering silhouette thresholding to identify anomalies. A recent survey is presented by [21] of distinct Anomaly detection methods. These techniques consist of classification, nearest neighbor, clustering, statistical, spectral, information-theoretic, and graph. Selecting the convenient AD algorithm relies on input data, the form of anomalies, output data, and domain knowledge.

3. Materials & Methods: ML Algorithms

This section discusses the different approaches and methods used in this paper. Namely, we shed more light on the used ML algorithms (i) AutoEncoder Long Short-Term Memory (AE-LSTM), (ii) Facebook-Prophet, and (iii) Isolation Forest. These algorithms architectures and modelling are intensively discussed, creating a solid understanding of this research methodology.

3.1. AutoEncoder Long Short-Term Memory (AE-LSTM)

AutoEncoder (AE) is an unsupervised ANN. It has the same structure of three symmetrical layers that include an input, hidden (interval description), and an output layer (remodeling) [23]. It has internal encoding and decoding processes. The encoding operates starts from the input to the hidden layer, whereas decoding handles the hidden layer to the output layer. AE has the merit of learning unlabelled data efficiently to predict from the input vector. Figure 1 illustrates the construction of AE.

The encoding process is described by:

$$H = f_1(W_i \cdot X + b_i) \tag{1}$$

Where W_i and b_i are the weights and bias parameters between the input and the hidden layer. X is the primary input, H is the intermediate representation of the primary data, and f_1 is the activation function (e.g., ReLU, Logistic (Sigmoid) and TanH). Likewise, the decoding process is expressed as:

$$\widehat{X} = f_2(W_h \cdot H + b_h) \tag{2}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2021 doi:10.20944/preprints202112.0481.v1

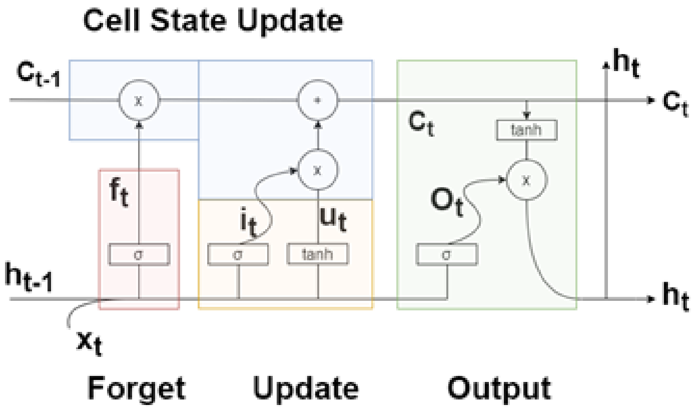


Figure 2. The AutoEncoder (AE) model.

where W_h and b_h are the weights and bias parameters between the hidden and the output layer, respectively. $\widehat{X}$ is the output that is reconstructed from the input data. AE is trained with the objective of minimizing the difference between the output $\widehat{X}$ and the input vector X through squared error [24], also called the reconstruction error [23] that is represented by:

$$\mathcal{L}(X, \hat{X}) = \|\hat{X} - X\|^2 \tag{3}$$

Long Short-Term Memory (LSTM) is part of Recurrent Neural Networks (RNNs). It employs an enclosed state (memory) to handle time-series inputs to capture the sequence relation of the input vector X [25]. It also uses the backpropagation through time (BPTT) model [26], but this causes a gradient vanishing. Therefore, LSTM uses three controlling gates: the input, forget, and output gates and the memory cell that memorizes a temporal state. The gates can reduce the gradient vanishing intensively by renewing and controlling the data flow [25]. Figure 2 illustrates the LSTM unit.

LSTM controls the information flow through the gates using the following equations:

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \tag{4}$$

$$f_t = \sigma\left(W_f[x_t, h_{t-1}] + b_f\right) \tag{5}$$

$$u_t = \tanh(W_u[x_t, h_{t-1}] + b_u) \tag{6}$$

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \tag{7}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot u_t \tag{8}$$

$$h_t = o_t \odot \tanh(c_t) \tag{9}$$

Where h_t is the present final output, c_t is current cell state, x_t is the present input, f_t is the forget gate, i_t is the input gate, u_t is the input to the cell c that is gated by the input gate, o_t is the output control signal, and $\odot$ is an element-wise multiplication [25]. The AE-LSTM neural network learns the correlation between input variables and the correlation in the time series. The LSTM unit also avoids the issue of long-term memory reliance.

3.2. Facebook-Prophet

A prophet is a time series forecasting algorithm; it extends Twitter's Anomaly Detection (TAD) by replacing the residual component with holidays to detect changepoints [28]. Prophet separates a time series into three elements, seasonal, trend, and holidays as follows:

$$y(t) = g(t) + s(t) + h(t) + \epsilon t \tag{10}$$

Where g(t) is the trend function that captures non-seasonal changes, s(t) is the seasonal changes function, and h(t) is the holiday's function. t is a function that seizes any other changes that do not fit the three main functions. g(t) has both saturating growth, and piecewise linear models [28]. g(t) defines the logistic growth model as follows:

$$g(t) = \frac{c}{1 + \exp(-(\mathrm{k}(\mathrm{t} - (\mathrm{m}))} \tag{11}$$

Where C is the carrying capacity, k is the growth rate, and m is an offset specification. g(t) then incorporates trend updates in the growth model by describing changepoints s_j where the growth rate is permitted to update at time t. Suppose there are S changepoints at times s_j, where j= 1,..., S. Define a vector of rate adjustments $\delta \in R^S$, where δ_j is the change in rate that occurs at time s_j. The rate at time t is defined as follows:

$$t = k + a(t)^T \delta \tag{12}$$

where $a(t)^T$ is the cumulative growth till changepoints s_j [29] and $a(t) \in \{0,1\}^S$ is a vector that can be computed as follows [28]:

$$a_j(t) = \begin{cases} 1, & \text{if } t \geq s_j \\ 0, & \text{Otherwise} \end{cases} \tag{13}$$

The prophet then modifies the primary logistic growth model to include trend updates for non-linear, saturating growth as follows:

$$g(t) = \frac{c(t)}{1 + \exp(-(\mathrm{k} + \mathrm{a(t)}^T\delta)(\mathrm{t} - (\mathrm{m} + \mathrm{a(t)}^T\gamma)))} \tag{14}$$

and the linear growth can be draft as follows:

$$g(t) = \left(k + a(t)^T \delta\right) t + \left(m + a(t)^T \gamma\right) \tag{15}$$

Let $\delta \in R^S$ such that points in δ are rate of modifications in $g(t)$. The allocation of change points would be by assigning δ using Laplacian distribution $(\delta_j \sim$ Laplace $(0, \tau))$, where τ controls the compliance of growth rate [29] and γ_j is set to $-s_j\delta_j$ to make the function continuous [28].

3.3. Isolation Forest

Isolation Forest is an unsupervised anomaly detection algorithm based on decision trees. It defines anomalies as data points that are limited, and abnormal [30]. Isolation Forest works by defining a tree structure based on randomly selected features and then processing a sample of the data picked randomly into the tree [30]. The branching structure process is done with a random threshold selected in the range of the selected feature's minimum and maximum values. If a sample goes deeper into the tree, then it is unlikely to be an anomaly. On the contrary, if the sample is positioned in shorter branches, it is more probable to be an anomaly [30].

The algorithm can be described as follow: Let T be a node in the tree, q is a sample of selected features, p is the threshold value, and $X = \{x_1, x_2, x_3, x_4, \ldots, x_n\}$ is the dataset with n samples where each sample has d features. T can be a leaf node or can be an inside node (with two sub-nodes T_{left}, T_{right}). If the threshold $p > q$, then the sample will be maintained to the T_{left}, otherwise the sample will be assigned to T_{right}. This process keeps repeating until either all data at the node have similar values, or the node has one sample only, or the tree reaches the maximum possible depth (length). The length of path $h(x)$ can be measured by counting the number of edges which connects the tree from the root node to an outside node. The smaller $h(x)$ means that sample x is more likely to be defined as an anomaly. The anomaly score s of the sample x can be calculated as:

$$s(x,n) = 2^{\frac{E(h(x))}{c(n)}} \tag{16}$$

where c(n) is the evaluation of average h(x) for outside node and can be computed as:

$$c(n) = \begin{cases} 2\mathrm{H(n-1)} - \frac{2(\mathrm{n}-1)}{\mathrm{n}} & \text{for } n > 2 \\ 1 & \text{for } n = 2 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

where $H(i)$ expresses the harmonic which that can be evaluated by $\ln(i) + \gamma(\gamma$ represents Euler's constant) [30].

4. Collected Data

The data used were collected at two solar power plants in India (Plant 1 is near Gandikotta, Andhra, and Plant 2 is near Nasik, Maharashtra) over 34 days, each with 15 minutes intervals. Every plant includes 22 inverter sensors connected at both the inverter and the plant levels to measure the generation rate (an internal factor that can cause anomalies), such as AC and DC powers. At the plant level, the inverter measures the irradiation, the ambient, and the module temperatures (they represent the external factors that can cause anomalies) for weather measurements. The data are published, licensed, and accessed under [31].

Figure 3 is a correlation matrix displaying the correlation coefficients between the feature parameters. Correlation is a normalized covariance with its values varying between -1 to 1. The matrix measures a linear relationship among variables with -1 indicating that the related variables have a strong negative relationship, that is, as one variable increases, the other one decreases, and 1 indicates a strong positive relation, that is, an increase in one variable results in an increase in the other one. The diagonal values indicate the correlation of a variable with itself (known as autocorrelation). Spearman's rank correlation [32] is used to determine the correlation rank between features as follows:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{18}$$

Where is the Spearman's rank correlation coefficient, d_i is the difference among the two ranks of each observation, and n is the number of observations. The figure shows that both internal and external factors are highly correlated except for the daily and the total yields, The daily yield represents all the generated power in KW for this particular inverter until the recorded time t. The total yield on the other hand is the summation of all the generated power from the 22 inverters at this particular plant. In the future we will also consider allocating data in a federated architectures [38].

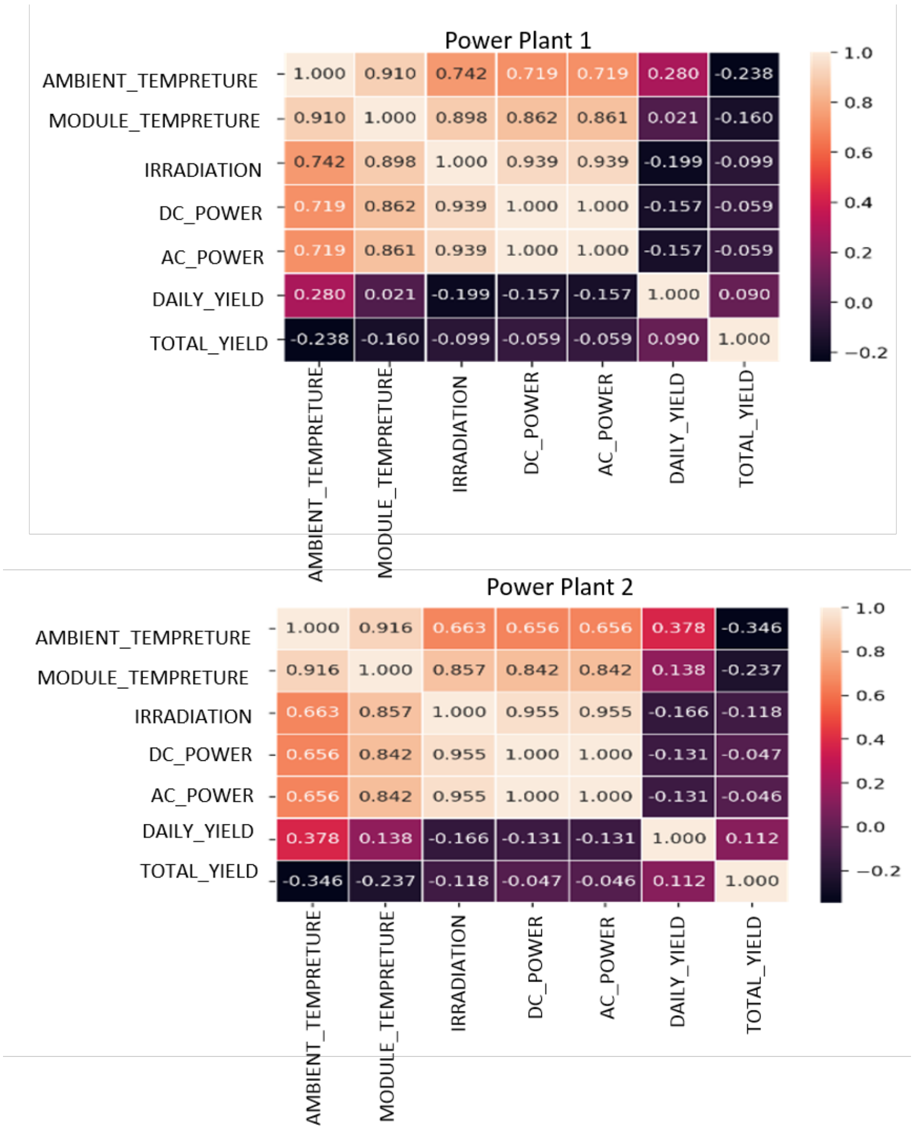


Figure 3. Correlation matrix measuring the linear dependence between the feature parameters for Power Plant 1 and 2.

5. Results and Discussion

This section discusses the experiment evaluation carried to validate and evaluate the paper claims. A complete description of the experimental setup is provided. Following, we analyze our findings and results in detail.

PV systems may have many types of anomalies. In order to make a fair comparison between the used anomaly detection algorithms, tests were conducted to investigate the effect of both internal and external factors as well the correlation effect on the data of all inverter sensors for the two plants. For instance, a test was done to compare the generated AC power and the irradiation for inverter number 1 of power plant 1 as illustrated in Figure 4, and it can be noticed that in the periods of Jun 7 and Jun 14, respectively, there was a drop in the AC power. This notice can indicate a failure at the inverter level.

The number of anomalies in the signal is 13, distributed on Jun 7 and Jun 14. On the contrary, for other inverters such as inverter number 12, there was no drop in the AC power generation, as illustrated in Figure 5.

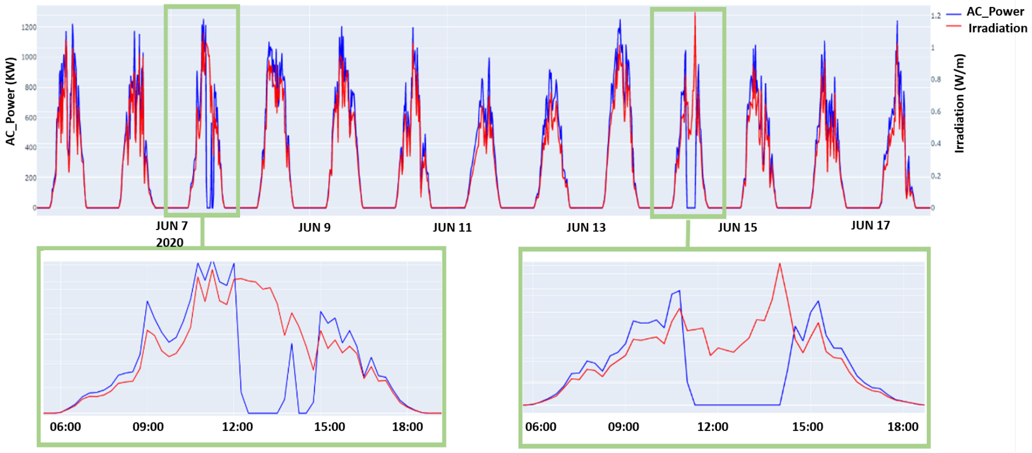


Figure 4. Signal comparison between AC, DC Power, Irradiation and the Module Temperature signals from Inverter number 12.

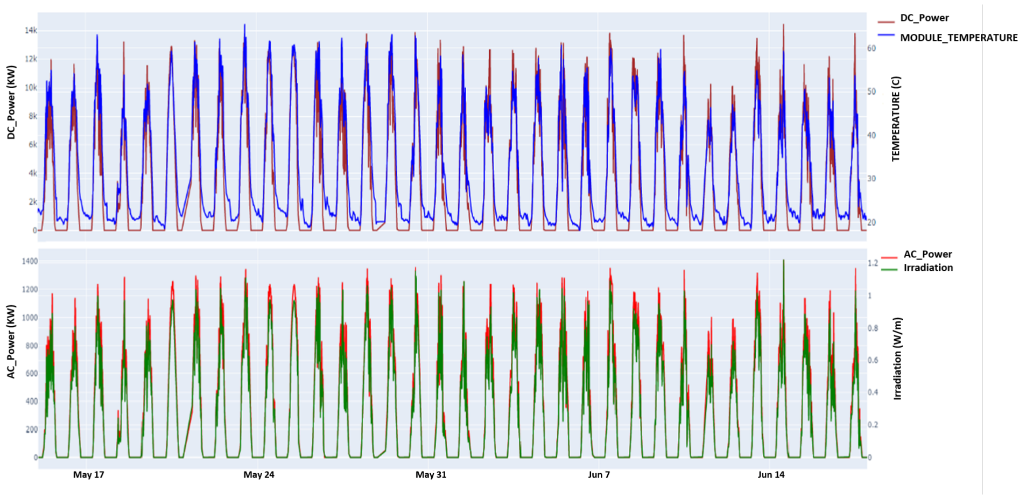


Figure 5. Signal comparison between AC, DC Power, Irradiation and the Module Temperature signals from Inverter number 12.

Before testing the candidate algorithms, the Grid parameters search optimizer, supported by Scikit learn [33], was used to tune each algorithm's hyperparameters. The optimizer explores all possible combinations of a defined range of values for each parameter until the best accuracy is obtained. This measure means that an appropriate objective function can be defined for each algorithm to select the optimal parameters. The algorithms were tested on the AC Power signal from inverter number 1 in power plant 1 to detect the 13 true anomalies. The algorithm's parameters and optimizer results are stated in the following subsections.

5.1. Facebook-Prophet Optimized Parameters

An essential parameter in Facebook-Prophet is the number of Changepoints (*n-changepoints*) in the dataset. Its usual value is 25. Changepoints are uniformly distributed on the first 80% of the time-series signal. The *changepoint_prior_scale* indicates how flexible the changepoints are allowed to be, which means how much the changepoints can fit the data. Its usual value is 0.05. The *seasonality_mode* parameter defines two modes. The Additive and Multiplicative modes. The default mode is additive, which signifies that the seasonality's impact is combined with the forecast trend. Table 1 shows the parameters grid for Prophet with a total of 162 possible models.

Table 1: Grid Parameters of Facebook-Prophet

Parameter	Grid
n_changepoints	[10,25,50,75,100,150,200,300,400,500]
changepoint_prior_scale	[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
seasonality_mode	['multiplicative', 'additive']

Due to the working principle of Prophet of predicting a time series signal, the objective function was selected to be R-squared (R^2) which can be computed as follow:

$$R^2 = \frac{\sum_{i=1}^{N_i} (y_i - \hat{Y}_i)}{\sum_{i=1}^{N_i} (y_i - \bar{Y}_i)} \tag{19}$$

Where y, $\hat{Y}$ are the actual and the predicted data, respectively, while $\bar{Y}$ is the mean value of actual data. The optimization results found the best R^2 to be 87.448 for the following optimal parameters:

- *n_changepoints* = 0.9
- *changepoint_prior_scale* = 200
- *seasonality_mode* = multiplicative

5.2. AE-LSTM Optimized Parameters

AE-LSTM, on the contrary, shares the same parameters that any other neural network model have, which are the number of hidden neurons, the number of layers, activation function, epochs, and batch size. For simplification, the number of hidden layers was chosen to be 4 layers with Rectifier (ReLU) activation function, the number of hidden neurons was optimized for each layer separately. Table 2, shows the parameters grid for AE-LSTM with a total of 54432 possible models:

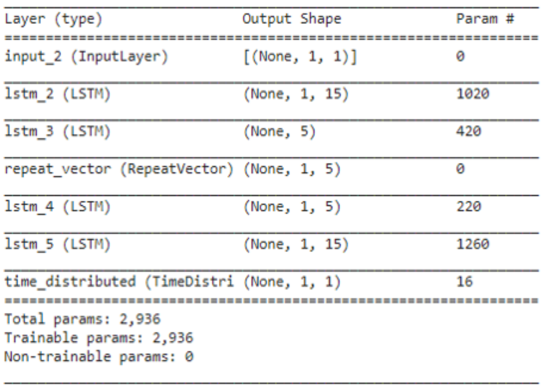
Table 2: Grid Parameters of AE-LSTM.

Parameter	Grid
Number_hidden_neurons L1	[5,10,15,20,25,30]
Number_hidden_neurons L2	[5,10,15,20,25,30]
Number_hidden_neurons L3	[5,10,15,20,25,30]
Number_hidden_neurons L4	[5,10,15,20,25,30]
batch	[5,10,15,20,25,30]
epochs	[200,250,300,350,400,450,500]

The AE-LSTM, also learns/trains on time series signal, and then tries to predict/forecast this signal characteristics in the future. Therefore, same as in Prophet, the R^2 was used as an objective function. The optimization results found the best R^2 to be 98.1749 for the optimal parameters epochs (200) and batch size of 20. The number of hidden neurons and the complete AE-LSTM model are illustrated in Figure 6.

5.3. Isolation Forest Optimized Parameters

Isolation forest was also optimized for the number of estimators (n_estimators) or trees in the ensemble. In other words, it is the number of trees that will construct the forest. It has a default value of 100. Another parameter is the contamination which describes the expected proportion or rate of outliers/abnormality in the data set. Table 3 shows the parameters grid for AE-LSTM with a total of 338 possible models. The bootstrap is a parameter that controls the sampling process. If it is set to True, then the



```
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 1, 1)]            0
_________________________________________________________________
lstm_2 (LSTM)                (None, 1, 15)             1020
_________________________________________________________________
lstm_3 (LSTM)                (None, 5)                 420
_________________________________________________________________
repeat_vector (RepeatVector) (None, 1, 5)              0
_________________________________________________________________
lstm_4 (LSTM)                (None, 1, 5)              220
_________________________________________________________________
lstm_5 (LSTM)                (None, 1, 15)             1260
_________________________________________________________________
time_distributed (TimeDistri (None, 1, 1)              16
=================================================================
Total params: 2,936
Trainable params: 2,936
Non-trainable params: 0
_________________________________________________________________
```

Figure 6. The AE-LSTM model.

Table 3: Grid Parameters of Isolation Forest

Parameter	Grid
bootstrap	[False, True]
n_estimators	[50,100,200,300,400,500,600,700,800,900,1000,1500,2000]
contamination	[0,0.01,0.03,0.06,0.09,0.12,0.15,0.2,0.25,0.3,0.4,0.45,0.5]

individual trees fit random subsets of the training data sampled with replacement. If it is set to False, then sampling without replacement is performed.

The Isolation forest does not predict any time-series signal compared to Prophet and AE-LSTM. Instead, it classifies the data points into normal and abnormal, with the same concept as random forest. Therefore, the objective function would focus on the number of true and false anomalies. The optimization results for 338 possible models found only one value which is 25 anomalies, where 12 points are true anomalies, and the other 13 points are false anomalies.

5.4. Anomaly Detection Performance

The Isolation forest, AE-LSTM and Prophet algorithms were implemented to evaluate their performance in detecting the AC generated power signal abnormalities. The outcomes are illustrated in Figure 7. It can be noticed that even though Prophet detected the anomalies on Jun 7 and 14, it failed to determine the healthy signal by labelling it as an anomaly with a total of 53 anomalies (false anomalies). Isolation forest also detected the true positive anomalies but marked all the signal peaks as anomalies with 25 outliers. The AE-LSTM was able to detect the correct 13 anomalies and successfully identified the healthy signal. Also, the models were tested on a healthy AC Power signal from inverter number 12. Prophet and isolation Forest found anomalies within this signal. It is worth mentioning that the isolation forest detected false anomalies on the peaks while AE-LSTM determined that there are no anomalies, as shown in Figure 8.

The second test investigated the external correlated factor of module temperature as shown in Figure 9. It can be seen that the signal is healthy. However, the Prophet found anomalies within a complete healthy signal. The isolation forest detected false anomalies on the peaks, and the AE-LSTM determined that it is a healthy signal and thus, detected no anomalies.

The third test examined the effect of the uncorrelated internal factors, which are expressed in the daily yield. The daily yield signal is a healthy signal with no apparent anomalies that could be determined. This signal means that even if there is a failure in one of the 22 inverters, it did not affect the signal. The Prophet and Isolation forest also failed this test, while AE-LSTM succeeded in determining no anomalies in a healthy signal as shown in Figure 10.

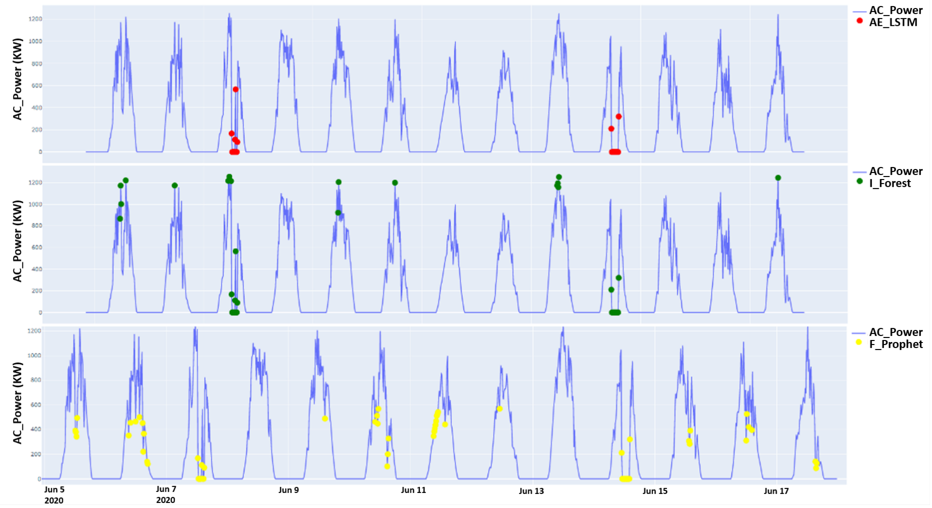


Figure 7. Anomaly Detection Results from the three models on a Faulty AC Power signal.

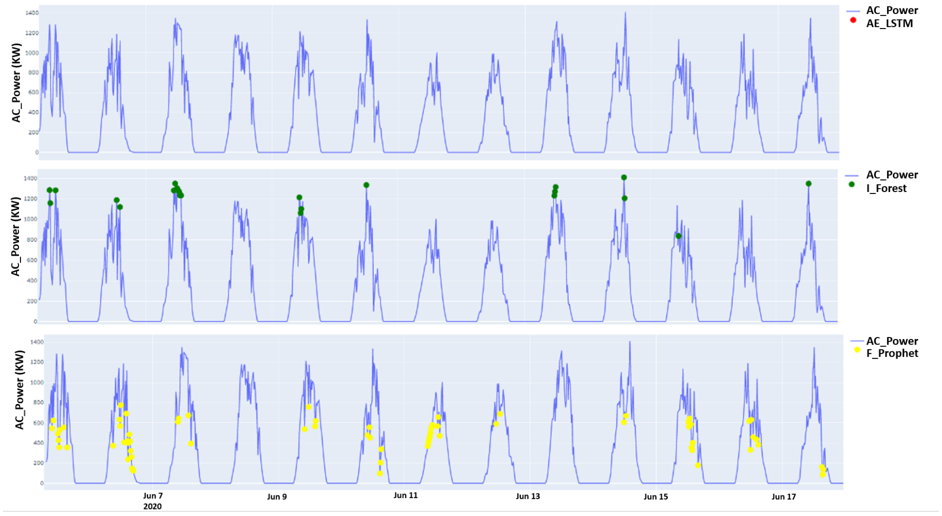


Figure 8. Anomaly Detection Results from the three models on a Healthy AC Power signal.

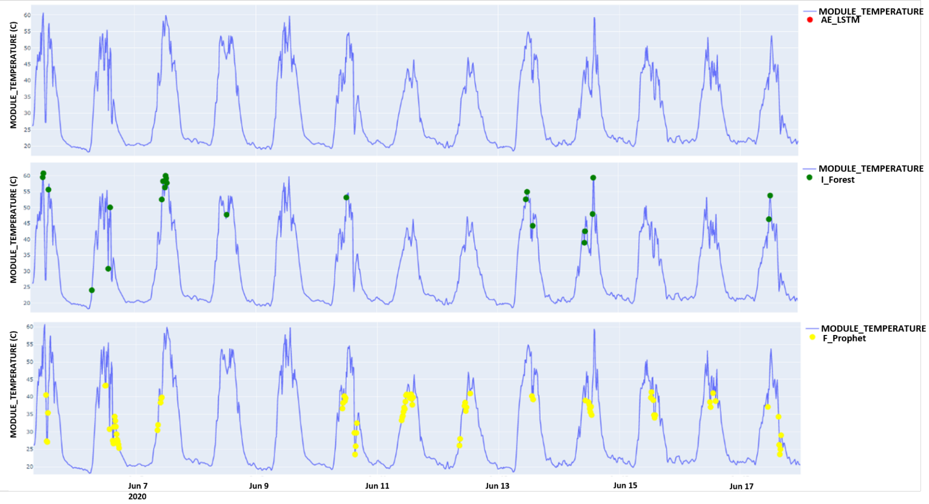


Figure 9. Anomaly Detection Results from the three models on a Healthy Module Temperature signal.

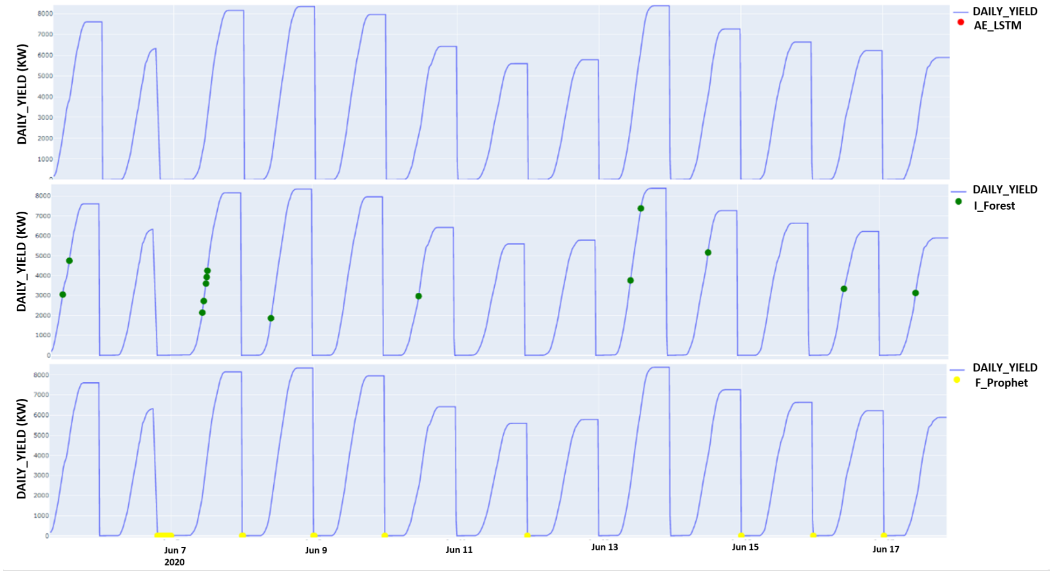


Figure 10. Anomaly Detection Results from the three models on a Healthy Daily Yield signal.

The results above showed that the AE-LSTM was more accurate in detecting the true anomalies without tagging false positive points (normal) as anomalies. Also, we prove that the two optimized models did not accurately distinguish the correct and the false anomalies. On the contrary, although Prophet and Isolation Forest found the true anomalies, both models labeled healthy/normal points as anomalies. Furthermore, results demonstrate that Prophet and Isolation Forest are more sensitive to noisy signals and need more datasets to generalize and capture signal characteristics to distinguish a false from a true anomaly. An interesting future work would be investigating the blockchain technology for the large-scale Solar Power Plants networks [37]. Also, examining recent trends in machine learning like active machine learning [36].

6. Conclusions

In this paper, a comparative analysis of the performance of three machine learning models was conducted to determine the best model that can accurately detect the anomalies in the photovoltaic (PV) system dataset. The correlation coefficients between the plants' internal and external feature parameters were determined and used to analyze the efficiency of machine learning models in detecting anomalies. The AE-LSTM detected anomalies and successfully identified the healthy signal. Future work would include the investigation of intelligent anomaly mitigation techniques. Also, an interesting open question would be on employing the recent distributed machine learning trend, i.e., federated learning, in large-scale intelliegent solar power grids.

Acknowledgments: Dr Feras M. Awaysheh acknowledges support via IT Academy program and the European Regional Development Funds via the Mobilitas Plus programme (grant MOBTT75). The authors would like to acknowledge Deanship of Graduate Studies and Scientific Research at the German Jordanian University for the Seed fund SATS 03/2020. Also, Taif University Research supporting, project number (TURSP-2020/126), Taif university, Taif Saudi Arabia.

References

1. Benninger, M., Hofmann, M., Liebschner, M. (2019, September). Online Monitoring System for Photovoltaic Systems Using Anomaly Detection with Machine Learning. In NEIS 2019; Conference on Sustainable Energy Supply and Energy Storage Systems (pp. 1-6). VDE.
2. Li, C., Yang, Y., Zhang, K., Zhu, C., Wei, H. (2021). A fast MPPT-based anomaly detection and accurate fault diagnosis technique for PV arrays. Energy Conversion and Management, 234, 113950.
3. Hu, B. (2012). Solar panel anomaly detection and classification (Master's thesis, University of Waterloo).
4. Awaysheh, F. M., Alazab, M., Garg, S., Niyato, D., & Verikoukis, C. (2021). Big data resource management & networks: Taxonomy, survey, and future directions. IEEE Communications Surveys Tutorials.
5. Branco, P., Gonçalves, F., Costa, A. C. (2020). Tailored algorithms for anomaly detection in photovoltaic systems. Energies, 13(1), 225.
6. De Benedetti, M., Leonardi, F., Messina, F., Santoro, C., Vasilakos, A. (2018). Anomaly detection and predictive maintenance for photovoltaic systems. Neurocomputing, 310, 59-68.
7. Harrou, F., Dairi, A., Taghezouit, B., Sun, Y. (2019). An unsupervised monitoring procedure for detecting anomalies in photovoltaic systems using a one-class Support Vector Machine. Solar Energy, 179, 48-58.
8. Feng, M., Bashir, N., Shenoy, P., Irwin, D., Kosanovic, D. (2020, June). SunDown: Model-driven Per-Panel Solar Anomaly Detection for Residential Arrays. In Proceedings of the 3rd ACM SIGCAS Conference on Computing and Sustainable Societies (pp. 291-295).
9. Sanz-Bobi, M. A., San Roque, A. M., De Marcos, A., Bada, M. (2012, May). Intelligent system for a remote diagnosis of a photovoltaic solar power plant. In Journal of Physics: Conference Series (Vol. 364, No. 1, p. 012119). IOP Publishing.
10. Zhao, Y., Liu, Q., Li, D., Kang, D., Lv, Q., Shang, L. (2018). Hierarchical anomaly detection and multimodal classification in large-scale photovoltaic systems. IEEE Transactions on Sustainable Energy, 10(3), 1351-1361.
11. Mulongo, J., Atemkeng, M., Ansah-Narh, T., Rockefeller, R., Nguegnang, G. M., Garuti, M. A. (2020). Anomaly detection in power generation plants using machine learning and neural networks. Applied Artificial Intelligence, 34(1), 64-79.
12. Benninger, M., Hofmann, M., Liebschner, M. (2020, September). Anomaly detection by comparing photovoltaic systems with machine learning methods. In NEIS 2020; Conference on Sustainable Energy Supply and Energy Storage Systems (pp. 1-6). VDE.
13. Balzategui, J., Eciolaza, L., Maestro-Watson, D. (2021). Anomaly detection and automatic labeling for solar cell quality inspection based on Generative Adversarial Network. arXiv preprint arXiv:2103.03518.
14. Wang, Q., Paynabar, K., Pacella, M. (2021). Online automatic anomaly detection for photovoltaic systems using thermography imaging and low rank matrix decomposition. Journal of Quality Technology, 1-14.
15. Hempelmann, S., Feng, L., Basoglu, C., Behrens, G., Diehl, M., Friedrich, W., ... Pfeil, T. (2020, June). Evaluation of unsupervised anomaly detection approaches on photovoltaic monitoring data. In 2020 47th IEEE Photovoltaic Specialists Conference (PVSC) (pp. 2671-2674). IEEE.
16. Iyengar, S., Lee, S., Sheldon, D., Shenoy, P. (2018, June). Solarclique: Detecting anomalies in residential solar arrays. In Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies (pp. 1-10).
17. Tsai, C. W., Yang, C. W., Hsu, F. L., Tang, H. M., Fan, N. C., Lin, C. Y. (2020, February). Anomaly Detection Mechanism for Solar Generation using Semi-supervision Learning Model. In 2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN) (pp. 9-13). IEEE.
18. Pereira, J., Silveira, M. (2018, December). Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention. In 2018 17th IEEE international conference on machine learning and applications (ICMLA) (pp. 1275-1282). IEEE.
19. Kosek, A. M., Gehrke, O. (2016, October). Ensemble regression model-based anomaly detection for cyber-physical intrusion detection in smart grids. In 2016 IEEE Electrical Power and Energy Conference (EPEC) (pp. 1-7). IEEE.

20. Rossi, B., Chren, S., Buhnova, B., Pitner, T. (2016, October). Anomaly detection in smart grid data: An experience report. In 2016 ieee international conference on systems, man, and cybernetics (smc) (pp. 002313-002318). IEEE.
21. Toshniwal, A., Mahesh, K., Jayashree, R. (2020, October). Overview of anomaly detection techniques in machine learning. In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 808-815). IEEE.
22. Firth, S.K.; Lomas, K.J.; Rees, S.J. A simple model of PV system performance and its use in fault detection.Sol. Energy 2010, 84, 624–635.
23. Hu, D., Zhang, C., Yang, T., Chen, G. (2020). Anomaly Detection of Power Plant Equipment Using Long Short-Term Memory Based Autoencoder Neural Network. Sensors, 20(21), 6164
24. Que, Z., Liu, Y., Guo, C., Niu, X., Zhu, Y., Luk, W. (2019, December). Real-time Anomaly Detection for Flight Testing using AutoEncoder and LSTM. In 2019 International Conference on Field-Programmable Technology (ICFPT) (pp. 379-382). IEEE
25. Ibrahim, M., Alsheikh, A., Al-Hindawi, Q., Al-Dahidi, S., ElMoaqet, H. (2020). Short-time wind speed forecast using artificial learning-based algorithms. Computational intelligence and neuroscience, 2020.
26. Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. Proceedings of the IEEE, 78(10), 1550-1560
27. Shashank Shanbhag and Tilman Wolf. Accurate anomaly detection through parallelism. Network, IEEE, 23(1):22–28, 2009.
28. Taylor, S. J., Letham, B. (2018). Forecasting at scale. The American Statistician, 72(1), 37-45.
29. Srivastava, S. (2019). Benchmarking Facebook's Prophet, PELT and Twitter's Anomaly detection and automated de ployment to cloud (Master's thesis, University of Twente
30. Hariri, S., Kind, M. C., Brunner, R. J. (2019). Extended isolation forest. IEEE Transactions on Knowledge and Data Engineering.
31. A. Kannal, "Solar Power Generation Data," Kaggle.com, [Online]. Available: https://www.kaggle.com/anikannal/solar-power-generation-data.
32. Corder, G. W., Foreman, D. I. (2014). Nonparametric statistics: A step-by-step approach. John Wiley Sons.
33. ParameterGrid. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.ParameterGrid.html
34. Agarwal, A., Sharma, P., Alshehri, M., Mohamed, A. A., & Alfarraj, O. (2021). Classification model for accuracy and intrusion detection using machine learning approach. PeerJ Computer Science, 7, e437.
35. Alshehri, M., Kumar, M., Bhardwaj, A., Mishra, S., & Gyani, J. (2021). Deep Learning Based Approach to Classify Saline Particles in Sea Water. Water, 13(9), 1251.
36. Kebande, V. R., Alawadi, S., Awaysheh, F. M., Persson, J. A. (2021). Active Machine Learning Adversarial Attack Detection in the User Feedback Process. IEEE Access, 9, 36908-36923.
37. Kebande, V. R., Awaysheh, F. M., Ikuesan, R. A., Alawadi, S. A., Alshehri, M. D. (2021). A Blockchain-Based Multi-Factor Authentication Model for a Cloud-Enabled Internet of Vehicles. Sensors, 21(18), 6018.
38. Awaysheh, F. M., Alazab, M., Gupta, M., Pena, T. F., Cabaleiro, J. C. (2020). Next-generation big data federation access control: A reference model. Future Generation Computer Systems, 108, 726-741.