Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 6447434, 9 pages
https://doi.org/10.1155/2022/6447434

Research Article

Using Machine Learning for Performance Classification and Early Fault Detection in Solar Systems

Eshrag A. Refaee

Department of Information Technology & Security, Jazan University, Jazan 45142, Saudi Arabia

Correspondence should be addressed to Eshrag A. Refaee; eshragrefaee@gmail.com

Received 16 February 2022; Revised 10 March 2022; Accepted 21 March 2022; Published 9 April 2022

Academic Editor: Muhammad Faisal Nadeem

The steady increase in the world's population has directly influenced global climate change, resulting in catastrophic environmental consequences. This has created an immediate need for scientists from interdisciplinary domains like clean technology innovation in solar energy and computer science to join in the effort to save the world for future generations. As such, the United Nations has set a goal to ensure global access to affordable, sustainable, and clean energy. As a leading influential G20 economy, Saudi Arabia has recently established the Green Saudi initiative to align with the UN goal for enhancing the use of green energy. However, research in this area is sparse and greater effort is still required. This work is among the first to address the issue of enhancing and expanding the use of clean energy by means of studying the data collected from solar plants around Saudi. We used machine learning-based methods to assess the energy output performance of solar plants and employed the collected data to train the models to make early detection of faults. Our models achieved the highest performance at an accuracy score of 98.85% and 0.98 weighted *F*-score using the J48 model trained on a publicly available dataset of 874 instances collected from 26 different sites across Saudi. We anticipate that the findings of this work to serve as testbed to facilitate further research in this area and enhance the early fault detection in solar energy stations.

1. Introduction

The growing demand for clean energy requires increased utilization and improved efficiency of renewable technologies. Producing clean, sustainable, and efficient energy from wind and solar resources has become a high priority for ambitious countries like Saudi Arabia. To bring nations together to face the challenges associated with the steady increase in the world's population, the UN has created strategic goals to target the development and adaptation of clean energy resources, such as solar and wind. UN country members have created several initiatives to invest in developing energy-aware systems [1].

As a leading influential global economy among the G20 countries, Saudi Arabia has recently established the Green Saudi initiative. The initiative aims to minimise environmental impacts and improve quality of life by increasing the country's utilization of clean energy. In this context, the ambitious vision of the Kingdom is to make its cities smarter by investing in intelligent infrastructure. Specifically, cities will need to start utilising environment engineering techniques and use energy intelligent means to manage increasing populations while confronting the devastating consequences of climate change. As an example, smart cities supported by Internet of Things (IoT) applications have the potential to minimise the reliance on conventional infrastructure (e.g., environment friendly and energy-efficient bright lighting) [2, 3]. IoT is an emerging field in computer science that can be successfully advanced by engaging with other domains such as environmental engineering and climate sciences. This work aims to explore the extent to which IoT-based sensors, tools, and applications are currently being utilised in Saudi cities. We will also investigate the role of the existing rules and regulations in facilitating the rapid adoption of IoT-based technology in Saudi cities.

Accurately determining the efficiency of solar energy installations requires detailed analytics and information on each solar panel and array such as voltage, current, temperature, and irradiance. Conventional approaches involve monitoring utility-scale solar arrays which generate electricity that is fed into the grid, producing varying amounts of energy up to more than 50 megawatts. Recent research has shown that monitoring utility-scale solar arrays minimises the cost of maintenance and optimises the performance of the photovoltaic (photovoltaic is a clean, renewable source of energy that uses solar radiation to produce electricity) arrays under various conditions [4]. However, the monitoring of large-scale solar plants for human operators is not a trivial task. As such, there is demand for utilising machine learning-based techniques for automated performance monitoring. Specifically, machine learning algorithms have been successfully used in previous work [5] to detect solar systems' errors and faults [6].

There are several essential metrics used to determine the performance of solar systems. These metrics usually are based on the calculation of the solar panels' energy. In this work, we utilise several essential and well-established metrics: Direct Normal Irradiance (DNI), Diffuse Horizontal Irradiance (DHI), and Global Horizontal Irradiance (GHI). DNI is defined as the amount of solar radiation received per unit area by a surface that is always held perpendicular to the rays that arrive in a straight line from the direction of the sun at its current position in the sky [7]. DHI is defined as the terrestrial irradiance received by a horizontal surface that has been scattered or diffused by the atmosphere. GHI is defined as the total solar radiation incident on a horizontal surface. GHI is the sum of DNI, DHI, and ground-reflected radiation as follows:

$$\mathrm{GHI} = \mathrm{DHI} + \mathrm{DNI} * \cos(z). \tag{1}$$

As such, GHI is commonly used to compute the power output of solar flat panels. In this work, we utilise DNI and DHI as attributes, and we use GHI to calculate the classifying attribute GHI class to identify early occurrences of faults within solar systems based on the readings of these attributes. Figure 1 shows the GHI map and associated solar energy potential for Saudi from 1998 to 2018.

The main contributions of this work are the following: (1) utilisation for the first time of publicly available solar data collected from 26 different sites across Saudi; (2) exploration and analysis of the dataset and calculation of a class attribute with three possibilities; (3) use of the data to train machine learning classifiers to automatically detect early signs of the need for solar systems monitoring and inspection to reduce maintenance costs and prevent or minimise out-of-service periods; and (4) use of the output of our trained models as an input for another system or another element in the system that can detect the existence and type of faults.

2. Related Work

This section provides an overview of previous investigations of the utility of machine learning (ML) models for fault detection and/or performance improvement in solar systems, with a particular focus on Saudi.

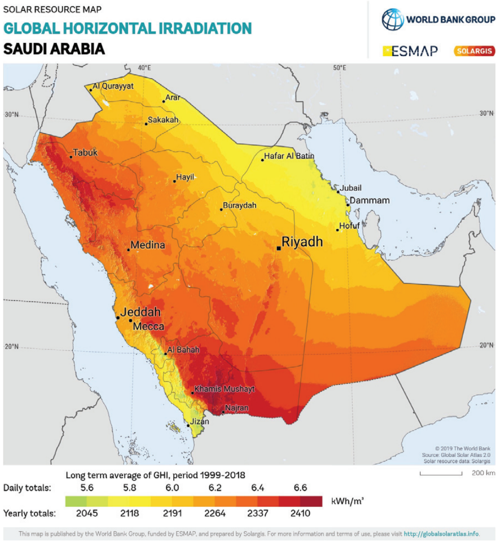

SOLAR RESOURCE MAP
GLOBAL HORIZONTAL IRRADIATION
SAUDI ARABIA
WORLD BANK GROUP
ESMAP
SOLARGIS
40°E
50°E
30°N
20°N
Al Qurayyat
Arar
Sakakah
Tabuk
Hafar Al Batin
Hayil
Jubail
Dammam
Buraydah
Hofuf
Riyadh
Medina
Jeddah
Mecca
Al Bahah
Khamis Mushayt
Najran
Jizan
© 2019 The World Bank
Source: Global Solar Atlas 2.0
Solar resource data: Solargis
200 km
Long term average of GHI, period 1999-2018
Daily totals: 5.6 5.8 6.0 6.2 6.4 6.6
kWh/m²
Yearly totals: 2045 2118 2191 2264 2337 2410
This map is published by the World Bank Group, funded by ESMAP, and prepared by Solargis. For more information and terms of use, please visit http://globalsolaratlas.info.

Figure 1: The Global Horizontal Irradiation (GHI) map of Saudi Arabia. Source: SolarGIS.

Rao et al. [8] described a proposed Cyber-Physical system approach to fault detection in photovoltaic arrays. The system used feed forward neural network algorithms for fault detection from monitoring devices that sense data from a set of individual panels. The authors reported that their proposed system improved overall efficiency by detecting and identifying eight different faults and conditions related to power output in utility-scale photovoltaic arrays.

Fazai et al. [9] considered a ML approach with a statistical testing hypothesis for enhanced fault detection performance in photovoltaic systems. Their method makes use of a ML-based Gaussian process regression technique as a modelling framework in addition to a generalised likelihood ratio test chart to detect photovoltaic system faults. The authors used both simulated and actual photovoltaic systems data in their assessment. They monitored the key system variables, namely, current, voltage, and power. In addition, the authors reported the computation time, missed detection rate, and false alarm rate to evaluate the fault detection performance of the proposed approach. Overall, they reported their best score of the goodness of fit rate at 98.24, considering factors such as relative humidity, wind speed, and rainfall.

Alajmi et al. [10] presented a study using ML with voltage and current sensors to detect, locate, and classify common faults. The authors considered including open circuit, short circuit, and hot-spot and reported a perfect accuracy score of 100%. However, the authors used simulated data as no real-life data were utilised in their proposed approach.

Another work by Zubair et al. [11] reported their efforts to optimise a parabolic trough (PT)-based concentrated solar power system. The study analysed solar energy data collected from Saudi Arabia as well as some European and Asian countries. Among the countries, different cities were compared based on their peak load. The purpose of their analysis was to sell electricity generated locally to the customers during their peak load hours to reduce their load factor and minimise the capital cost. However, the study did not consider the investigation of fault detection factors and causes.

Recent work by Benavides et al. [12] used ML to predict the energy produced from three different photovoltaic systems and the supervision of measurement sensors. The authors investigated the energy production in response to changes in the climatic variables of the site under study. They provided an implementation of several indicators in order to allow the solar plant operators to actively manage the electricity grid. The authors also claimed that their system can provide real-time predictions of photovoltaic systems and measurement sensors.

In summary, the increasing demand for clean energy requires increased utilisation and improved efficiency of renewable technologies. The task of monitoring solar systems is becoming more difficult due to this growth and the continuous need for performance enhancement. As such, there has been a recent spike in research in this field to address the increasing demand with the limited number of energy systems currently operating. More importantly, there is a growing demand for solar data to be made available for research purposes.

3. Experimental Framework

This section presents the experimental setup and configuration used in this work, including the dataset and the ML models utilised.

3.1. Dataset. We use a publicly available dataset of solar systems measured values. Specifically, we utilise a dataset collected and owned by the King Abdullah City for Atomic and Renewable Energy from 2013 to 2016. The dataset was made publicly available via the OpenData platform in 2020 and was further updated in 2021 (OpenData is a government-based publicly accessible platform for open data. The data are available via OpenData at https://data.gov.sa/Data/en/dataset/kacare%20andrratlas.energy.gov.sa. Accessed on 26/10/2021). The dataset contains 874 data instances and 26 features as shown in Table 1. The data were collected via 26 solar power facilities around Saudi (Specifically, the solar stations were located at Makkah Umm Al-Qura University, Shaqra University, Hagl SWCC, Farasan SWCC, Al Khafji SWCC, Rania, Najran University, Riyadh King Saud University, Al Ahsa King Faisal University, Thuwal King Abdullah University for Science and Technology, Osfan, Jeddah King Abdulaziz University, Hada Al Sham, Riyadh K.A.CARE City T2, Riyadh—K.A.CARE HQ, Jazan University, Hail, Hafar Al Batin, Duba, Arar, Al Wajh, Riyadh Al Uyaynah, Al Qunfudhah, Al Hanakiyah, Al Dawadmi, and Al Baha University). Figure 2 shows the data structure in JSON files retrieved from OpenData and used in this work.

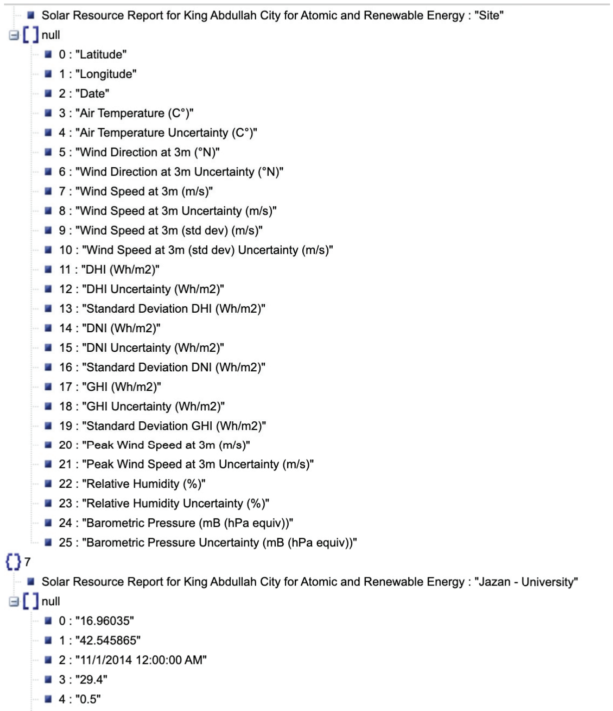
3.1.1. Data Preprocessing and Identifying the Class Attribute. Unlike most other available solar energy datasets, this dataset includes only raw readings from several solar stations around Saudi. Other datasets typically include a class attribute for fault categorisation, such as in the PVWatts dataset used by Rao et al. [4]. An additional attribute named GHI class was calculated with three possible values. To obtain GHI class, we first normalised all the GHI numerical values to be in range $[-1, +1]$. Normalising a numerical attribute needed to address the variation in amount of power produced between different solar stations. Moreover, numerical attribute normalisation is a particularly useful practise with ML models [13]. After normalising the GHI values, three classes were assigned to the GHI values such as Running when GHI values are greater than $+0.4$, Monitoring when GHI values are greater than or equal to -0.4 or less than or equal to $+0.4$, and Inspecting when GHI values are between -0.4 and -1.0. The three classes cover three different possibilities for early default detection in the solar station using the identified threshold. Each class corresponds to one of three zones, the green, orange, and red zones. The first class, Running, denotes the green zone when the system is operational, demonstrated by the stable performance of the power produced. The second class, Monitoring, denotes the orange zone when the system is not performing optimally, indicating an early need for system monitoring. The third class, Inspecting, denotes the red zone when the performance of the solar station is below the expected level and requiring urgent inspection for the solar panels at the site, as shown in Figure 3.

Figure 4 shows the class distribution in the dataset we used across all the attributes. It can be seen that classes are unbalanced, with Monitoring representing the majority class with 399 instances. The Inspecting class is the minority class with 186 instances, while the Running class contains 289 instances. Although the classes are not balanced, for two reasons, we opted not to employ any class balancing technique that primarily relies on creating artificial synthetic instances to force class balancing. Firstly, we believe it is essential to train the models with real-life datasets with their actual distribution as this will reflect the trained model's more realistic performance. Secondly, the class distribution in our solar dataset does not skew significantly towards a specific class; therefore, we expect that it will not lead to overfitting. The processed dataset will be made publicly available to reproduce the results and facilitate future research.

3.2. Machine Learning Models. This section presents the ML models we utilise in this work, including several models known to perform well on classification tasks [13]. We employ WEKA's implementation of the ML models we use in this work (WEKA is a well-known Java-based open-source package that incorporates implementations for a collection of machine learning algorithms for data mining. WEKA is developed by the Machine Learning Group at the University of Waikato; accessing and downloading WEKA is available at: http://www.cs.waikato.ac.nz/ml/weka/. In this work we use version 3.8.5 of WEKA).

Table 1: Attributes of the solar energy dataset retrieved from OpenData platform.

Attribute name	Attribute type	Description
Site	Nominal	Total of 26 distinct sites around Saudi
Latitude	Numeric	Geographical latitude of a specific solar system site
Altitude	Numeric	Geographical altitude of a specific solar system site
Date	Date	The time and date at which data was collected
Air temperature (C)	Numeric	Air temperature at the site
Air temperature uncertainty (C)	Numeric	Uncertainty window
Wind direction at 3 m (N)	Numeric	Wind direction
Wind direction at 3 m uncertainty (N)	Numeric	Uncertainty window
Wind speed at 3 m (m/s)	Numeric	Wind speed
Wind speed at 3 m uncertainty (m/s)	Numeric	Uncertainty window
Wind speed at 3 m (std dev) (m/s)	Numeric	Wind speed standard deviation
DHI (Wh/m^2)	Numeric	Diffuse horizontal irradiance
DHI uncertainty (Wh/m^2)	Numeric	Uncertainty window
Standard deviation DHI (Wh/m^2)	Numeric	DHI SD
DNI (Wh/m^2)	Numeric	Direct normal irradiance
DNI uncertainty (Wh/m^2)	Numeric	Uncertainty window
Standard deviation DNI (Wh/m^2)	Numeric	DNI SD
GHI (Wh/m^2)	Numeric	Global horizontal irradiance
GHI uncertainty (Wh/m^2)	Numeric	Uncertainty window
Standard deviation GHI (Wh/m^2)	Numeric	GHI SD
Peak wind speed at 3 m (m/s)	Numeric	Peak wind speed at site
Peak wind speed at 3 m uncertainty (m/s)	Numeric	Uncertainty window
Relative humidity (%)	Numeric	Humidity level at site
Relative humidity uncertainty (%)	Numeric	Uncertainty window
Barometric pressure (mB (hPa equiv))	Numeric	Pressure at site
Barometric pressure uncertainty (mB (hPa equiv))	Numeric	Uncertainty window
GHI class	Nominal	Calculated attribute with three possible values: running, monitoring, inspecting



```
■ Solar Resource Report for King Abdullah City for Atomic and Renewable Energy : "Site"
[ ] null
   ■ 0 : "Latitude"
   ■ 1 : "Longitude"
   ■ 2 : "Date"
   ■ 3 : "Air Temperature (C°)"
   ■ 4 : "Air Temperature Uncertainty (C°)"
   ■ 5 : "Wind Direction at 3m (°N)"
   ■ 6 : "Wind Direction at 3m Uncertainty (°N)"
   ■ 7 : "Wind Speed at 3m (m/s)"
   ■ 8 : "Wind Speed at 3m Uncertainty (m/s)"
   ■ 9 : "Wind Speed at 3m (std dev) (m/s)"
   ■ 10 : "Wind Speed at 3m (std dev) Uncertainty (m/s)"
   ■ 11 : "DHI (Wh/m2)"
   ■ 12 : "DHI Uncertainty (Wh/m2)"
   ■ 13 : "Standard Deviation DHI (Wh/m2)"
   ■ 14 : "DNI (Wh/m2)"
   ■ 15 : "DNI Uncertainty (Wh/m2)"
   ■ 16 : "Standard Deviation DNI (Wh/m2)"
   ■ 17 : "GHI (Wh/m2)"
   ■ 18 : "GHI Uncertainty (Wh/m2)"
   ■ 19 : "Standard Deviation GHI (Wh/m2)"
   ■ 20 : "Peak Wind Speed at 3m (m/s)"
   ■ 21 : "Peak Wind Speed at 3m Uncertainty (m/s)"
   ■ 22 : "Relative Humidity (%)"
   ■ 23 : "Relative Humidity Uncertainty (%)"
   ■ 24 : "Barometric Pressure (mB (hPa equiv))"
   ■ 25 : "Barometric Pressure Uncertainty (mB (hPa equiv))"
{ } 7
 ■ Solar Resource Report for King Abdullah City for Atomic and Renewable Energy : "Jazan - University"
[ ] null
   ■ 0 : "16.96035"
   ■ 1 : "42.545865"
   ■ 2 : "11/1/2014 12:00:00 AM"
   ■ 3 : "29.4"
   ■ 4 : "0.5"
```

Figure 2: Screenshot of the JSON files of the solar systems dataset used.

3.2.1. Majority Baseline ZeroR. We compare our results against a majority baseline that always predict the most frequent class in the dataset. ZeroR is a useful classifier to provide a lower bound on the performance of the dataset [14].

3.2.2. Random Forests. Random Forest (RF) is a ML-based model that utilises an ensemble learning method for classification and regression. RF works by constructing many decision trees using a given set of training data. For classification tasks, the output of the RF is the class selected by most trees, i.e., the trained ensemble.

3.2.3. Decision Tree J48. J48 is often referred to as a statistical classifier. The J48 algorithm generates a decision tree Witten et al. [14]. As a classification algorithm, J48 is used to produce decision trees based on information theory.

3.2.4. LibLinear SVM. We use Support Vector Machines (SVMs) as an ML scheme that is particularly successful for classification problems [15]. This is due to their ability to handle many features in high dimensional feature space (i.e., text classification problems). A trained SVM will attempt to classify a new instance to one of the predefined classes on

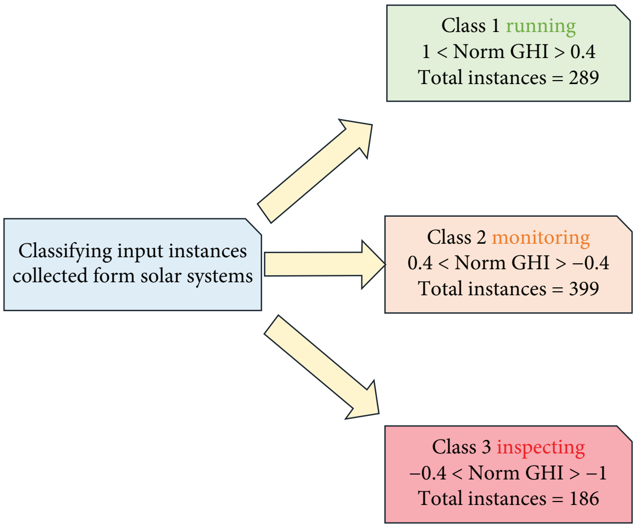


Figure 3: Class distribution in the solar dataset.

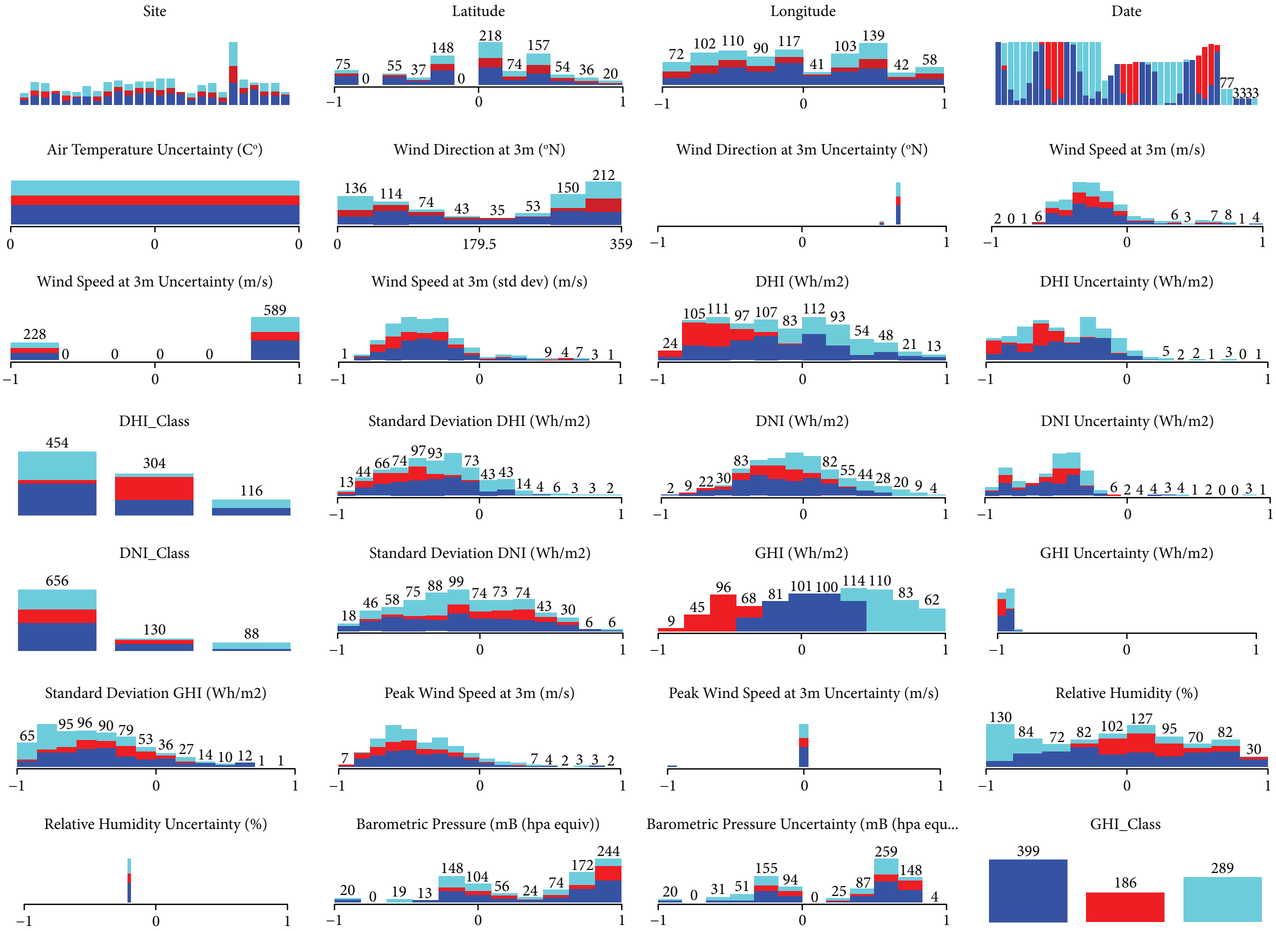


Figure 4: Distribution of all attributes with respect to the classifying class GHI. The colours that denote the three classes are dark blue for Monitoring class, red for Inspecting class, and light blue for the Running class.

which the model is initially trained by finding a hyperplane or a decision-surface that separates the instances of classes [14]. Two more hyperplanes parallel to the separating hyperplane are created, called support hyperplanes, as shown in Figure 5. The support hyperplanes cut through the closest training instances, called support vectors, on either side.

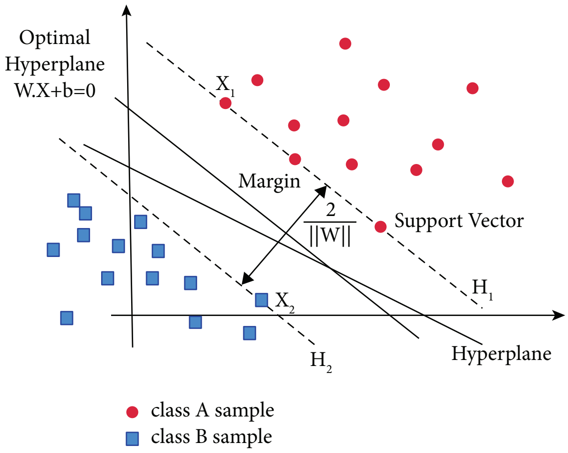


Figure 5: Hyperplanes in SVM.

3.2.5. Naïve Bayes (NB). Naïve Bayes is a classification algorithm based on Bayes' theorem, assuming independence among predictors. In simple terms, an NB classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

3.2.6. Logistic Regression (Simple Logistic). Logistic Regression (LR) is a classification algorithm that predicts a categorical output. It is used to model the relationship between two variables by fitting a linear equation to observed data.

3.2.7. Deep Learning (DL) with Convolutional Neural Network (CNN). CNN is a class of deep neural networks that uses a special technique called convolution, which is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other [16]. We use the following parameters in the model: CNN with five epochs, network configuration as dropout = disabled, Adam optimiser, learning rate = 0.001, epsilon = 1.0$E-8$, optimisation Algo = Stochastic Gradient Descent (SGD), gradient normalisation threshold = 1.0, and total trainable parameters = 29 with zero frozen parameters. The time required to build and train the model was 14.9 seconds.

3.3. Evaluation Metrics. In classification problems, the overall performance is typically measured by the success rate, which is the proportion of the correctly classified instances over the entire set of instances. Here, we report the results using two metrics: weighted *F*-score and accuracy [13]. The weighted *F*-score is the average of all *F*-scores attained for each class (i.e., *F*-running, *F*-monitor, and *F*-inspect). Each *F*-score is weighted according to the number of instances with that particular class.

Accuracy is one of the most widely reported metrics in the literature and is calculated as follows:

$$\text{accuracy} = \frac{\text{number of correctly classified instances}}{\text{total number of instances}}. \tag{2}$$

The *F*-score is defined as the harmonic average of precision and recall (A control parameter β can be used to decide how much emphasis to put on precision vs. recall. $F1$, or by convention F, is where β's value is 1 denoting an equal/balanced emphasis on both metrics) and is calculated as follows:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \tag{3}$$

where precision is calculated as follows:

$$\text{precision} = \frac{A}{A + C}, \tag{4}$$

where A is the number of correct/relevant instances classified/retrieved and C is the number of incorrect/irrelevant instances classified/retrieved Moreover, recall is calculated as follows:

$$\text{recall} = \frac{A}{A + B}, \tag{5}$$

where B is the number of correct/relevant instances not classified/retrieved.

3.4. Evaluation Methods. Assessing the success rate of a classifier on previously unseen instances that have played no role in building the classifier should provide a reliable indicator of the classifiers' future performance [14]. We use cross-validation (CV) for evaluating the trained models. CV uses a fixed number of data proportions, namely, folds to split the data into test and training sets. The dataset is randomly reordered before being split into n folds of equal size. In each fold, every class is represented by approximately the same fraction as the entire dataset, also called stratified CV. Previous work has shown that 10 is the number of folds for obtaining the best estimate of error [14]. Each fold is then held-out to be used in turn for testing. This results in the learning process being run 10 times on different combinations of the training set. In the end, the resultant 10 error rates are averaged to yield the overall score. As an enhancement for the reliability of the results and as suggested by Witten et al. [14], we ran 10 experiments for different 10-fold CV for each dataset, resulting in 100 invokes of each learning algorithm on each dataset with scores averaged over 10 repetitions.

3.5. Problem Formulation. We experiment with single-level problem formulations for fault detection and classification in solar systems flat three-way classification. Figure 3 shows the structure of the classification problem we use.

Figure 6 shows the process flow of our investigations. The first step in the process is the readings of the solar panels being collected and made publicly available via OpenData. The next step is to use the dataset to train ML models to classify the performance of the solar system based on the amount of energy produced. The output of classified instances can then be used as an input for another system or by domain experts for further inspections.

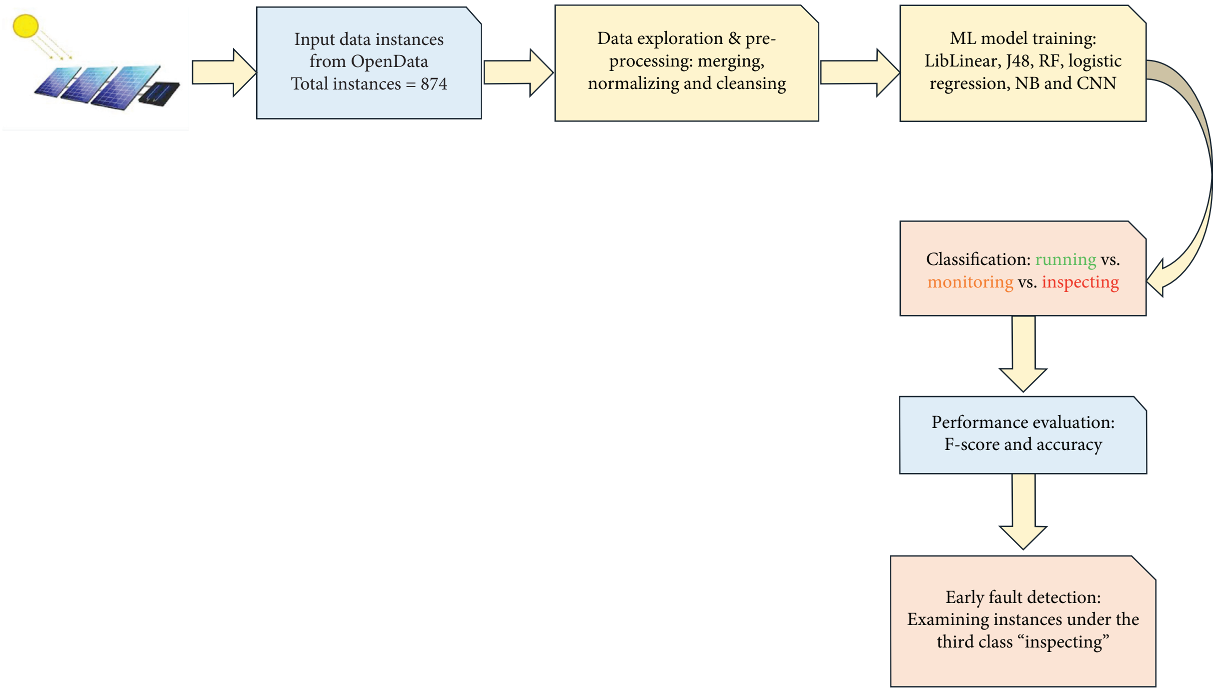

Input data instances from OpenData Total instances = 874
Data exploration & pre-processing: merging, normalizing and cleansing
ML model training: LibLinear, J48, RF, logistic regression, NB and CNN
Classification: running vs. monitoring vs. inspecting
Performance evaluation: F-score and accuracy
Early fault detection: Examining instances under the third class "inspecting"

Figure 6: Process flow for fault detection and classification in solar systems.

4. Results

We ran a series of experiments with the configurations discussed previously.

In this section, we present and discuss the main findings of our experimental work. Table 2 presents a summary of the results.

As can be seen in Table 2, all models significantly outperformed a simple majority baseline at an accuracy score of 45.65% (The majority class is Inspecting with a total of 399 instances. A T-test (χ^2) is conducted on accuracy and F-scores and at a confidence interval of 95% ($p < 0.05$) [14]). Interestingly, the tree-based models, i.e., RF and J48, achieved the highest performance. Specifically, J48 outperformed all models with an accuracy score of 98.85% and weighted F-score at 0.978, which is very similar to the performance achieved by the RF model. In other studies, J48 has been reported to perform well on classification tasks [17], especially with small-to-medium size datasets (e.g., nearly 1k instances). Here, the size of the tree (pruning) is only five, with the time required to construct the model being nearly one second. The next best performance was achieved by the linear regression model at an accuracy score of 97.25% and weighted F-score of 0.973. This performance is very close to the tree-based models, J48 and RF.

With respect to LibLinear (SVM), we can see that the highest accuracy score is 82.95% and weighted F-score is 0.801; this is considered to be reasonable performance for a three-way classification task. However, LibLinear's performance is significantly lower than the scores attained by J48. This is possible because the size of the dataset as LibLinear might require more training examples to be able to capture more distinctive patterns between the three classes. The confusion matrix of the model shows that 17% of the Inspecting class was wrongly classified as Monitoring. In comparison, 25% of the Monitoring class was mistakenly classified by the model as either Running or Inspecting.

We can see that both the NB and CNN models achieved nearly identical performance of approximately 91% accuracy. It is somewhat surprising to see good performance with a deep learning-based model like CNN with a relatively small dataset (i.e., less than 1k instances). A possible explanation for this is the quality of the dataset used with no outliers or noise. It has been argued that despite the well-known belief that deep learning models require large sets of training data containing millions or even billions of features, the quality of the data is a key factor [18]. As such, we believe that the quality of the dataset used allowed the model to capture valuable distinctive features. However, considering the computational power and model training time required, we believe that a decision tree model is more efficient and able to outperform other models.

To gain a better understanding of the informative nature of the attributes in our dataset, Table 3 lists the most informative attributes ranked according to their chi-squared (χ^2) values. χ^2 evaluates features by computing the chi-square value with respect to the class [14]. As a result, we used chi-square to obtain ranked lists of the most informative features for error analysis purposes and to gain insight into the subset of attributes that are beneficial and discriminative. It is interesting that attributes like humidity, wind speed, and barometric pressure are considered informative for the ML models in detecting one of the three classes that we use to determine signs of diminished performance of the solar systems' power production. These attributes have been shown to be essential factors in

Table 2: Experimental results on the solar dataset using different ML models.

ML algorithm	Accuracy	*F*-monitoring	*F*-inspecting	*F*-running	Weighted avg. *F*-score	Weighted avg. precision	Weighted avg. recall
ZeroR	45.65%	0.627	0	0	0.217	0.457	0.545
Random Forest	98.28%	0.981	0.971	0.993	0.983	0.983	0.983
J48	**98.85%**	**0.988**	**0.978**	0.997	**0.989**	**0.989**	**0.989**
LibLinear	82.95%	0.801	0.832	0.863	0.828	0.833	0.830
Naïve Bayes	91.18%	0.904	0.908	0.926	0.912	0.915	0.912
Linear Regression	97.25	0.970	0.939	**0.998**	0.973	0.973	0.973
CNN	91.42%	0.903	0.920	0.925	0.914	0.915	0.914

Values highlighted in bold indicate the highest value across all models with respect to a certain metric.

Table 3: List of the most informative attributes associated with their calculated chi-squared values.

X^2	Attribute
1233.848	Date
383.1912	Relative humidity (%)
352.7815	DHI (Wh/m^2)
234.4035	Barometric pressure (mB (hPa equiv))
187.6781	DNI (Wh/m^2)
131.0724	Standard deviation DHI (Wh/m^2)
122.4529	Wind direction at 3 m (°N)
102.0796	Barometric pressure uncertainty (mB (hPa equiv))
88.7592	Site
87.8772	Standard deviation GHI (Wh/m^2)
64.5802	Latitude
28.5757	Wind speed at 3 m (std dev) (m/s)
27.8873	Standard deviation DNI (Wh/m^2)
18.614	Peak wind speed at 3 m (m/s)

significantly influencing the overall performance of solar panels and systems [8].

5. Conclusions and Future Work

Solar systems provide a reliable source of clean and sustainable energy that can enable smart cities to participate in the international efforts to reduce emissions and enhance quality of life. Saudi Arabia recently introduced the Green Saudi initiative, which aims to increase the use of clean energy technologies (e.g., solar and wind). However, research on the current solar systems in Saudi for addressing existing issues (e.g., early fault detection) is limited. An in-depth investigation is required to enable the automatic detection and classification of solar systems' issues, with the goal of realising valuable benefits such as reduced maintenance costs and improved risk detection. Given the size and diversity of Saudi's geographical regions, early and automatic fault detection using ML-based methods and data obtained from distributed solar systems is crucial (e.g., to prevent permanent panel damage resulting from dust storms).

This work involves the analysis of performance measurements from solar systems installed in different geographical locations around Saudi. Specifically, we utilised a publicly available dataset collected from solar systems installed in 26 different locations. We trained multiple ML models, including Random Forest, LibLinear, Naive Bayes, linear regression, and CNN. We found that the tree-based models, J48 and RF, and linear regression models were able to outperform other models. J48 achieved the best performance at an accuracy score of 98.85% and weighted average *F*-score of 0.988. To the best of our knowledge, this is the first work to address the issue of early fault detection in solar systems in Saudi. As such, we believe our findings will benefit the research community by serving as a testbed and facilitating future research.

Opportunities for future research include expanding the proposed work pipeline by adding elements that can benefit from the output of the trained models. For example, domain experts can annotate sample datasets for types of faults (e.g., caused by dust, outdated units, and humidity). These datasets can also be used to train ML models to not only detect the early existence of faults but also to distinguish between the different types of faults. Future work can also involve obtaining larger and more diverse datasets spanning an extended period so that the model can predict possible issues that might occur within an extensive network of solar panels.

Data Availability

Data used in the research are obtained from the OpenData. OpenData is a government-based publicly accessible platform for open data. The data are available via OpenData at: https://data.gov.sa/Data/en/dataset/kacare andrratlas.energy.gov.sa. accessed on 26/10/2021.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] E. A. Refaee and S. Shamsudheen, "Trust-and energy-aware cluster head selection in a UAV-based wireless sensor network using FIT-FCM," *The Journal of Supercomputing*, pp. 1–16, 2021.

[2] P. Chithaluru, F. Al-Turjman, M. Kumar, and T. Stephan, *I-areor: An Energy-Balanced Clustering Protocol for Implementing green IoT in Smart Cities*, Sustainable cities and society, 2020.

[3] C. K. Metallidou, K. E. Psannis, and E. A. Egyptiadou, "Energy efficiency in smart buildings: iot approaches," *IEEE Access*, vol. 8, pp. 63679–63699, 2020.

[4] S. Rao, S. Katoch, V. Narayanaswamy et al., "Machine learning for solar array monitoring, optimisation, and control," *Synthesis Lectures on Power Electronics*, vol. 7, no. 1, pp. 1–91, 2020.

[5] U. Nations, "UN goal 7: affordable and clean energy," 2021, https://www.un.org/sustainabledevelopment/energy/.

[6] M. David, H. Diagne, and P. Lauret, "Outputs and error indicators for solar forecasting models," *Proceedings of the World Renewable Energy Forum (WREF)*, pp. 13–17, 2012.

[7] C. J. Cleveland and C. Morris, "Section 10—solar," in *Handbook of Energy*, C. J. Cleveland and C. Morris, Eds., pp. 405–450, Elsevier, Amsterdam, Netherlands, 2013, https://www.sciencedirect.com/science/article/pii/%20B9780080464053000103.

[8] S. Rao, A. Spanias, and C. Tepedelenlioglu, "Solar array fault detection using neural networks," in *2019 IEEE International Conference on Industrial Cyber Physical Systems (Icps)*, pp. 196–200, 2019.

[9] R. Fazai, K. Abodayeh, M. Mansouri et al., "Machine learning-based statistical testing hypothesis for fault detection in photovoltaic systems," *Solar Energy*, vol. 190, pp. 405–413, 2019.

[10] M. Alajmi, S. Aljahdali, S. Alsaheel, M. Fattah, and M. Alshehri, "Machine learning as an efficient diagnostic tool for fault detection and localisation in solar photovoltaic arrays," *Proceedings of 32nd international conference on Computer Applications in Industry and Engineering*, vol. 63, pp. 21–33, 2019.

[11] M. Zubair, A. B. Awan, M. A. Baseer, M. N. Khan, and G. Abbas, "Optimisation of parabolic trough based concentrated solar power plant for energy export from Saudi Arabia," *Energy Reports*, vol. 7, pp. 4540–4554, 2021.

[12] D. J. Benavides, P. Arévalo-Cordero, L. G. Gonzalez, L. Hernández-Callejo, F. Jurado, and J. A. Aguado, *Method of Monitoring and Detection of Failures in PV System Based on Machine Learning*, Revista Facultad de Ingeniería Universidad de Antioquia, no. 102, , pp. 26–43, 2022.

[13] I. H. Witten and E. Frank, "Data mining: practical machine learning tools and techniques with java implementations," *Acm Sigmod Record*, vol. 31, no. 1, pp. 76-77, 2002.

[14] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2013.

[15] T. Joachims, *Text Categorisation with Support Vector Machines: Learning with many Relevant Features*, Springer, 1998.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016, http://www.deeplearningbook.org.

[17] A. Priyam, G. Abhijeeta, A. Rathee, and S. Srivastava, "Comparative analysis of decision tree classification algorithms," *International Journal of current engineering and technology*, vol. 3, no. 2, pp. 334–337, 2013.

[18] A. Ng, "Machine learning yearning," vol. 139, 2017, https://www.mlyearning.org/(96).