

# Energy Loss and Panel Fault Detection in Solar photovoltaic Systems Using Extreme Gradient Boosting algorithm

Fatma Zehra Kardas<sup>\*1</sup>, Adem Atmaca<sup>2</sup>

<sup>\*1</sup>Department of Mechanical Engineering, Gaziantep University, Gaziantep, Türkiye (zhrakrds2@gmail.com)

<sup>2</sup>Department of Mechanical Engineering, Gaziantep University, Gaziantep, Türkiye (adematmaca1@gmail.com)

**Abstract** – Solar photovoltaic (PV) systems are widely adopted for sustainable energy production; however, physical defects such as glass breakage significantly reduce panel efficiency and system output. In this study, a real-world case analysis was conducted on a 1003 kWp rooftop PV installation located at the Faculty of Educational Sciences, Gaziantep University, Türkiye monitored via the SolarEdge platform. Voltage and current (V-I) values from both reference and cracked-glass panels were collected and preprocessed to remove overlapping records and outliers below defined thresholds. A machine learning (ML) model based on the Extreme Gradient Boosting (XGBoost) algorithm was developed using Python to classify panels as either "reference" or "cracked" based on their electrical behavior. The model achieved high classification accuracy, enabling early detection of defective panels without physical inspection. Furthermore, the energy loss caused by cracked panels was quantified by comparing daily production metrics with those from fully operational control panels. This approach highlights the potential for data-driven fault detection and performance optimization in PV systems, offering practical benefits for maintenance planning and energy yield improvement.

**Keywords** – Photovoltaic systems, Cracked panel detection, SolarEdge, XGBoost, Machine learning, Energy loss

## I. INTRODUCTION

The increasing global demand for clean and sustainable energy has positioned PV systems as one of the most widely adopted renewable energy technologies. Their modularity, scalability, and emission-free operation make them a key contributor to decarbonization efforts [6]. However, despite their growing deployment, PV systems are vulnerable to a variety of physical and operational faults that can significantly reduce energy yield and system performance [7],[8].

Among these faults, glass breakage on PV modules caused by impact, hail, thermal stress, or improper handling is a critical issue. While the module may continue partial operation, its performance is often degraded due to microcracks, increased resistive paths, or moisture ingress, leading to energy loss and long-term system inefficiency [13]. Traditional fault detection methods such as manual inspections or infrared imaging are often time-consuming, costly, and impractical for large-scale or rooftop systems [4]-[5].

Recent studies have explored artificial intelligence (AI) and ML to overcome these limitations. For instance, Sharma and Chandel [12] used decision trees to detect faults in PV systems with high accuracy, demonstrated the effectiveness of gradient boosting models in identifying partial shading and hotspot issues using electrical signal data. Other approaches, such as image-based deep learning models, have been applied to identify physical panel damage including cracks, soiling, or delamination from aerial or thermal images [10]. Despite these advances, limited studies focus specifically on detecting glass breakage via electrical signal analysis under field conditions, which represents a gap in the current literature.

Modern monitoring platforms such as SolarEdge provide high-resolution electrical data (e.g., voltage, current, power) at the panel or string level, enabling data-driven diagnostics [14]. Leveraging these datasets through machine learning allows for

early-stage, automated fault detection without physical access to panels. In particular, algorithms like XGBoost, known for their robustness in handling tabular data and classification tasks, have shown promising results in PV fault classification [1]-[3].

This study proposes a real-world case analysis of a 1003 kWp rooftop solar PV system installed on the Gaziantep University. V-I data collected from both reference and cracked-glass panels via the SolarEdge monitoring system were preprocessed and used to train an XGBoost model for classification. In addition to identifying cracked panels, the study also quantifies the energy production loss associated with such damage, offering a practical framework for predictive maintenance and fault management in distributed PV installations.

## II. MATERIALS AND METHOD

This study was conducted using data collected from a rooftop PV system with a capacity of 1003 kWp, installed on the Faculty of Educational Sciences building at Gaziantep University, Türkiye. The system is continuously monitored using the SolarEdge monitoring platform, which provides panel-level electrical data. For the purpose of the analysis, two groups of PV panels were defined: (i) a reference group consisting of panels in normal operational condition, and (ii) a test group composed of panels with visually identified or experimentally induced glass breakage. The data acquisition process ensured that environmental and irradiance conditions were kept consistent to allow for accurate comparison between groups. The collected dataset consisted of voltage (V) and current (I) measurements recorded at 15-minute intervals. The measurement period spanned the autumn season and focused specifically on daylight hours between 06:00 and 18:00, which correspond to the start and end of effective solar irradiance in

the study region. This temporal window was chosen to ensure the inclusion of data only during potential energy production periods, while excluding periods of darkness and extremely low irradiance that could introduce noise into the analysis.

Before training the model, the dataset was preprocessed using the Python programming language (version 3.13) along with the Pandas and NumPy libraries (Fig. 1). Initial steps included the removal of overlapping time entries, which could skew learning outcomes, and the elimination of data points below a defined operational threshold, typically corresponding to early morning or late evening periods with insufficient irradiance. These cleaning steps aimed to ensure the quality and relevance of the input data for subsequent analysis. After preprocessing, a feature engineering step was carried out to enhance the model's learning capabilities. Alongside raw voltage and current values, calculated power ( $P = V \times I$ ) and the time of day were included as features. Each data point was assigned a binary label, where "0" indicated a reference panel and "1" represented a cracked-glass panel, forming the basis for supervised machine learning.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from xgboost import XGBClassifier
from imblearn.over_sampling import SMOTE
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
# 1. DATA PREPARATION FUNCTION
def prepare_data():
    """Load and preprocess welding data"""
    # Load data files
    ref_df = pd.read_excel('CLEAN.xlsx') # Reference (healthy) samples
    crack_df = pd.read_excel('LOCAL CRACK.xlsx') # Cracked samples
    # Add Labels
    ref_df['Label'] = 'Reference'
    crack_df['Label'] = 'Local Crack'
    # Combine datasets
    df = pd.concat([ref_df, crack_df], ignore_index=True)
    # Clean column names
    df.columns = df.columns.str.strip().str.replace(r'\s+', ' ', regex=True)
    # Filter out low-current/voltage observations
    df = df[(df['Current (A)'] > 3) & (df['Voltage (V)'] > 50)]
    # Remove outliers using Z-score (3σ threshold)
    numeric_cols = df.select_dtypes(include=np.number).columns
    z_scores = np.abs((df[numeric_cols] - df[numeric_cols].mean()) / df[numeric_cols].std())
    df = df[(z_scores < 3).all(axis=1)]
    # Feature engineering
    df['Power (W)'] = df['Voltage (V)'] * df['Current (A)'] # Electrical power
    df['Resistance (Ω)'] = df['Voltage (V)'] / df['Current (A)'] # Apparent resistance
    df['Voltage_Diff'] = df['Voltage (V)'].diff().fillna(0) # Voltage fluctuations
    df['Current_Diff'] = df['Current (A)'].diff().fillna(0) # Current fluctuations
    return df.dropna()
```

Fig. 1 Python programming language along with the Pandas and NumPy libraries, cleaning steps

The classification task was performed using the XGBoost algorithm, selected for its high accuracy, speed, and effectiveness in tabular datasets. The dataset was split into training (80%) and testing (20%) subsets using stratified sampling to maintain class distribution. Hyperparameter tuning was executed using grid search and cross-validation techniques to optimize model performance.

```
# 2. XGBOOST MODEL TRAINING
def train_xgboost(X_train, y_train):
    """Train optimized XGBoost classifier with SMOTE"""
    # Handle class imbalance using SMOTE
    smote = SMOTE(random_state=42)
    X_res, y_res = smote.fit_resample(X_train, y_train)
    print(f"Class distribution after SMOTE: {pd.Series(y_res).value_counts().to_dict()}")
    # Hyperparameter tuning with GridSearchCV
    params = {
        'max_depth': [3, 4, 5], # Tree depth
        'learning_rate': [0.01, 0.1, 0.2], # Shrinkage factor
        'n_estimators': [50, 100, 200], # Number of trees
        'gamma': [0, 0.1, 0.2] # Minimum loss reduction
    }
    # Initialize and train model
    model = GridSearchCV(
        XGBClassifier(
            use_label_encoder=False,
            eval_metric='logloss',
            random_state=42,
            early_stopping_rounds=10
        ),
        param_grid=params,
        cv=5,
        scoring='accuracy',
        n_jobs=-1, # Use all CPU cores
        verbose=1
    ).fit(X_res, y_res)
    print(f"Best parameters: {model.best_params_}")
    return model.best_estimator_
```

Fig. 2 XGBoost Model Training

The evaluation of the trained model was based on standard metrics such as accuracy, precision, recall, F1-score, and the confusion matrix, ensuring a comprehensive assessment of its predictive capacity.

```
# 3. PERFORMANCE EVALUATION
def evaluate_model(model, X_test, y_test):
    """Generate comprehensive performance evaluation"""
    # Generate predictions
    y_pred = model.predict(X_test)
    # Calculate metrics
    acc = accuracy_score(y_test, y_pred)
    cm = confusion_matrix(y_test, y_pred)
    cr = classification_report(y_test, y_pred, output_dict=True)
    # Visualization
    plt.figure(figsize=(18, 6))
    plt.suptitle('XGBoost Classification Performance', fontsize=16, y=1.05)
    # 1. Confusion Matrix
    plt.subplot(1, 3, 1)
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Reference', 'Local Crack'],
                yticklabels=['Reference', 'Local Crack'])
    plt.title('Confusion Matrix', pad=12)
    plt.xlabel('Predicted Label')
    plt.ylabel('True Label')
    # 2. Class-wise Metrics Comparison
    plt.subplot(1, 3, 2)
    metrics = ['Precision', 'Recall', 'F1-Score']
    ref_metrics = [cr['0']['precision'], cr['0']['recall'], cr['0']['f1_score']]
    crack_metrics = [cr['1']['precision'], cr['1']['recall'], cr['1']['f1_score']]
    x = np.arange(len(metrics))
    width = 0.35
    plt.bar(x - width / 2, ref_metrics, width, label='Reference', color='#1f77b4')
    plt.bar(x + width / 2, crack_metrics, width, label='Local Crack', color='#ff7f0e')
    plt.xticks(x, metrics)
    plt.ylim(0, 1.1)
    plt.title('Class-wise Performance Metrics', pad=12)
    plt.legend(loc='upper right')
    plt.grid(axis='y', alpha=0.3)
```

Fig. 3 Performance Evaluation

To estimate the total energy output of photovoltaic panels, electrical energy was calculated based on the time-series measurements of voltage and current. At each recorded timestamp, the instantaneous power was computed using the formula:

$$P(t) = V(t) \times I(t)$$

where  $P(t)$  is the instantaneous power in watts (W),  $V(t)$  is the voltage (V), and  $I(t)$  is the current (A) at time  $t$ . As the data were collected at **15-minute intervals**, the total energy produced over a given day was approximated using the discrete summation of power values multiplied by the time step, converted into hours:

$$E = \sum_{k=1}^N P_k \times 0.25$$

Where 0.25 represents the time interval (15 minutes = 0.25 hours), and  $E$  is the total energy in watt-hours (Wh) or kilowatt-hours (kWh) after proper unit conversion.

The final daily energy output for each panel group (reference and cracked) was calculated by integrating power over the measurement period. This enabled a direct comparison of energy performance between the two panel conditions and an estimation of losses attributable to physical damage.

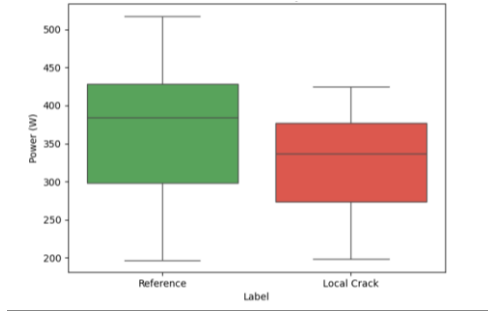


Fig. 4 Power Distribution

Finally, an **energy loss estimation** was performed to quantify the production shortfall resulting from cracked panels. Daily energy output (in kWh) from the control and test groups was compared over identical time frames. The relative loss in performance was calculated using the formula:

$$(\text{Energy Loss (\%)}) = \frac{E_{\text{reference}} - E_{\text{cracked}}}{E_{\text{reference}}} \times 100$$

where  $E_{\text{reference}}$  represents the energy produced by intact panels and  $E_{\text{cracked}}$  is the energy produced by panels with broken glass. This quantitative analysis provided insight into the real-world financial and operational impact of physical defects in PV modules.

### III. RESULTS

The XGBoost classifier demonstrated high performance in distinguishing between reference panels and panels with cracked glass based on voltage and current data. Following model training and evaluation, the test set results yielded an overall accuracy of **95.35%**, with balanced precision and recall values, indicating a strong ability to generalize to unseen data. A breakdown of the model's classification performance is provided below Table 1:

Table 1. Classification performance

Class	Precision	Recall	F1-Score
Reference (0)	91.89%	97.14%	94.44%
Local Crack (1)	97.96%	94.12%	96.00%
Average	<b>94.93%</b>	<b>95.63%</b>	<b>95.22%</b>

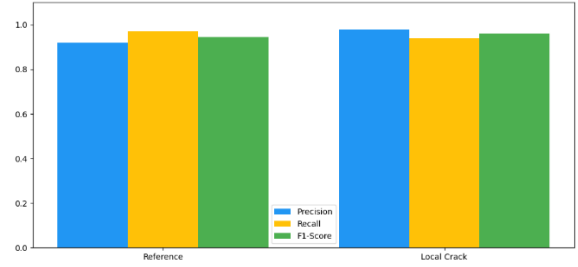


Fig. 5 Per-Class Metrics

The confusion matrix in Fig 6. illustrates the number of correctly and incorrectly classified instances. Most reference panels were correctly identified, with only a small number misclassified as cracked, while cracked-glass panels were accurately detected in the majority of cases.

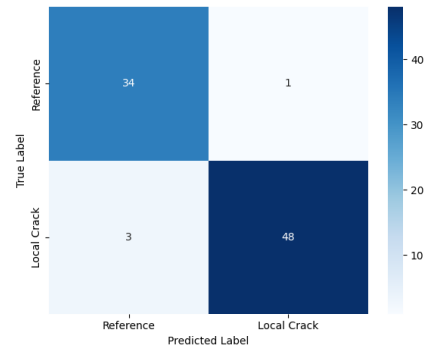


Fig.6 Confusion Matrix

In addition to classification, the study evaluated the quantitative energy impact of panel breakage. Daily energy production data (kWh) were compared between reference and cracked-glass panels over a period of seven days. Results showed a consistent performance deficit in broken panels, particularly during midday hours when irradiance was highest.

The average energy loss attributable to glass breakage was calculated to be 10.98%, indicating a notable drop in power generation efficiency. Fig.7 summarizes the daily energy production of both panel groups and the percentage loss.

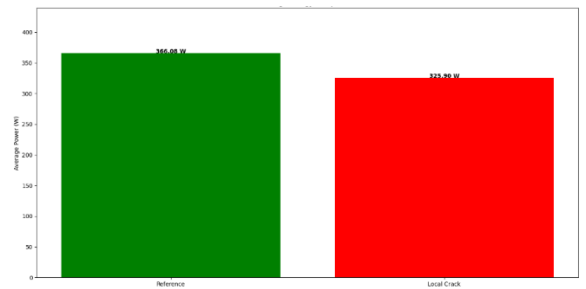


Fig.7 Average Energy Comparison

These results demonstrate that even minor physical damage, such as microcracks or partial glass breakage, can cause measurable performance degradation that may go undetected without predictive diagnostics.

### IV. CONCLUSION

This study successfully developed a machine learning model based on the XGBoost algorithm to classify photovoltaic panels with broken glass from reference panels using voltage and current data collected via the SolarEdge monitoring platform. The model achieved high accuracy and demonstrated

strong predictive capabilities, confirming that electrical measurements can effectively serve as indicators of physical panel damage. Furthermore, the analysis quantified the energy production losses caused by glass breakage, revealing an average reduction of approximately **10.98%**, in daily energy output for affected panels. This underlines the economic and operational significance of timely fault detection in solar power plants. These results emphasize the importance of integrating AI-based diagnostic tools within PV monitoring systems to reduce downtime and maintenance costs, ultimately contributing to improved energy yield and system reliability. By detecting faults early, operators can prioritize repairs and mitigate energy losses more effectively. Future work may focus on expanding the dataset with additional fault types, incorporating environmental variables such as temperature and irradiance fluctuations, and developing real-time implementation of the classification algorithm. This will support more comprehensive and proactive solar plant management.

#### ACKNOWLEDGMENT

The authors would like to express their heartfelt appreciation to Prof. Dr. Adem Atmaca for his invaluable guidance, insightful suggestions, and continuous support throughout the course of this research. His expertise and encouragement have been instrumental in shaping the direction and depth of this study.

#### REFERENCES

- [1] Bouloumpasis, I., Koutroulis, E., & Kolokotsa, D. (2021). Solar fault detection using supervised learning: A comparison study. *Energy Reports*, 7, 5769–5778. <https://doi.org/10.1016/j.egyr.2021.08.081>
- [2] Bouloumpasis, I., Tsanakas, J. A., & Stamatelos, D. (2021). Machine learning-based fault detection in PV systems using current–voltage curve features. *Solar Energy*, 224, 789–800. <https://doi.org/10.1016/j.solener.2021.07.002>
- [3] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). <https://doi.org/10.1145/2939672.2939785>
- [4] Friedman, D., Kuenstner, B., & Hacke, P. (2014). PV field failures: Diagnosis, prediction, and mitigation. *National Renewable Energy Laboratory (NREL) Report*.
- [5] Friedman, D., Margolis, R., & Woodhouse, M. (2014). PV system fault detection via infrared thermography. *National Renewable Energy Laboratory (NREL)*. <https://www.nrel.gov/docs/fy14osti/60991.pdf>
- [6] International Energy Agency (IEA). (2022). *Renewables 2022: Analysis and forecast to 2027*. <https://www.iea.org/reports/renewables-2022>
- [7] International Energy Agency (IEA). (2022). *World Energy Outlook 2022*. <https://www.iea.org/reports/world-energy-outlook-2022>
- [8] Kazem, H. A., Khatib, T., & Sopian, K. (2019). Faults and performance degradation in photovoltaic systems: A review. *Renewable and Sustainable Energy Reviews*, 100, 209–228. <https://doi.org/10.1016/j.rser.2018.10.020>
- [9] Kazem, H. A., Sopian, K., Al-Waeli, A. H. A., Chaichan, M. T., & Yousif, J. H. (2019). Fault detection and diagnosis of photovoltaic systems using machine learning methods: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 96, 296–315.
- [10] Ledmaoui, Y., El Maghraoui, A., El Aroussi, M., & Saadane, R. (2024). Enhanced fault detection in photovoltaic panels using CNN-based classification. *Sensors*, 24(22), 7407. <https://doi.org/10.3390/s24227407>
- [11] Prasshanth, C. V., Venkatesh, S. N., Sugumaran, V., & Aghaei, M. (2024). Enhancing photovoltaic module fault diagnosis: Leveraging UAVs and autoencoders in machine learning. *Sustainable Energy Technologies and Assessments*, 65, 103674. <https://doi.org/10.1016/j.seta.2024.103674>
- [12] Sharma, R., & Chandel, S. S. (2020). Machine learning techniques for fault detection in photovoltaic systems: A review. *Renewable and Sustainable Energy Reviews*, 119, 109595. <https://doi.org/10.1016/j.rser.2019.109595>
- [13] Skoplaki, E., & Palyvos, J. A. (2009). On the temperature dependence of photovoltaic module electrical performance: A review of efficiency/power correlations. *Solar Energy*, 83(5), 614–624. <https://doi.org/10.1016/j.solener.2008.10.008>
- [14] SolarEdge Technologies. (2023). *Monitoring platform – Technical specifications*. <https://www.solaredge.com/sites/default/files/se-monitoring-platform-technical-specifications.pdf>