

*International Journal of*

**ADVANCES IN  
ARTIFICIAL  
INTELLIGENCE  
RESEARCH**

**AIR**

**Volume 4, Issue 1, 2024**

**ISSN: 2757-7422**





**Advances in Artificial Intelligence Research**

<https://dergipark.org.tr/aair>

**Owner**

Osman ÖZKARACA - Muğla Sıtkı Koçman University

**Editor-in-Chief**

Ali KEÇEBAŞ - Muğla Sıtkı Koçman University  
Osman ÖZKARACA - Muğla Sıtkı Koçman University

**Editors**

Dr. Hüseyin ŞEKER	Staffordshire University, School of Computing and Digital Tech, England
Dr. Tuncay YİĞİT	Süleyman Demirel University, Computer Engineering Department, Turkey
Dr. Uğur Güvenç	Electric-Electronic Engineering, Düzce University, Düzce, Turkey
Dr. Jude HEMANTH	Karunya University, Electronics and Communication Engineering, India
Dr. Yusuf SÖNMEZ	Gazi University, Vocational College of Technical Sciences, Turkey
Dr. Ender ÖZCAN	Nottingham University, Computer Science and Operational Research, England
Dr. Hamdi Tolga KAHRAMAN	Karadeniz Technical University, Software Engineering, Turkey
Dr. Bogdan PATRUT	Alexandru Ioan Cuza University, Faculty of Computer Science, Romania
Dr. Ali Hakan IŞIK	Mehmet Akif Ersoy University, Computer Engineering, Turkey
Dr. İsmail Serkan ÜNCÜ	Isparta Applied Sciences University, Electrical-Electronics Engineering Turkey
Dr. Gürcan Çetin	Information Systems Engineering, Muğla Sıtkı Koçman University, Turkey
Dr. İsmail Yabanova	Mechatronics Engineering, Afyon Kocatepe University, Turkey

---

<b>Date of Publication</b>	August 2024
<b>Language</b>	English
<b>Frequency</b>	Published twice in a year
<b>Graphic designer</b>	Özden Işıktaş

---

<b>Correspondence Address</b>	Muğla Sıtkı Koçman University, Faculty of Technology, Information Systems Engineering, 48000 Kötekli/MUĞLA
<b>Phone</b>	0 506 4887818 – 0252 211 5526
<b>Correspondence Mail</b>	osmanozkaraca@mu.edu.tr



---



**Table of Contents**

<b>Pages</b>	<b>Articles</b>
1 – 9	COMPARISON OF PERFORMANCE OF SOME CLASSIFICATION METHODS TO EVALUATE THE QUALITY OF VEGETABLES FROM ITS MORPHOLOGY <b>Joy Deb, Dibyojyoti Bhattacharjee</b>
10 - 17	EFFICIENT AND ACCURATE DATE EXTRACTION FROM INVOICES: A COMPREHENSIVE THREE-STEP METHODOLOGY INTEGRATING CUSTOM OBJECT DETECTION, OCR, AND REFINED REGULAR EXPRESSIONS <b>Mehmet Hilmi Emel, Murat Terzioğlu, Ramazan Özkan</b>
18 - 35	GENERATIVE ADVERSARIAL NETWORKS IN ANOMALY DETECTION AND MALWARE DETECTION: A COMPREHENSIVE SURVEY <b>Bishal KC, Shushant Sapkota, Ashish Adhikari</b>
36 - 52	RECENT PROGRESS ON APPLICATIONS OF ARTIFICIAL INTELLIGENCE FOR SUSTAINABILITY OF SOLAR ENERGY TECHNOLOGIES: AN EXTENSIVE REVIEW <b>Jamilu Ya’u Muhammad, Abubakar Abdulkarim, Nafi’u Muhammad Saleh, Israel Ehile, Nuraini Sunusi Ma’aji, Audu Taofeek Olaniyi</b>
53 - 61	ARTIFICIAL NEURAL NETWORK PARAMETER OPTIMIZATION: IMPROVING METEOROLOGICAL DATA PREDICTIONS THROUGH MACHINE LEARNING <b>Ceyhun Kapucu, Oğuz Akpolat</b>

# Comparison of Performance of Some Classification Methods to Evaluate the Quality of Vegetables from its Morphology

Joy Deb <sup>1</sup> \* , Dibyojyoti Bhattacharjee <sup>1</sup> 

<sup>1</sup> Department of Statistics, Assam University

## Abstract

One important aspect of Data Science is its ability to classify subjects into non-overlapping groups based on one or several input variables. Several methods and algorithms are available in the literature for classifying subjects based on the values of multiple observed variables. Such classification tools are Naive Bayesian Classifiers, Logistic Regression, Discriminant Analysis, k-nearest neighborhood etc. This paper attempts to recognize if the morphological variables, identified either through literature review or from expert opinion, can be utilized to understand the quality of vegetables. Consequently, the current researchers obtained primary data about the morphology of the vegetables through experimentation. The outcome variable is the quality of the vegetables classified as eatable or not-eatable because of worm attack. Several classification methods are then compared for the classification exercise by building the model based on the training sample and testing the performance of the models in the holdout sample. Methods of classification performance statistics like sensitivity, specificity, precision etc. are used for their comparison. The study finds that Naive Bayes and Logistic Regression models perform better for this classification exercise. For example, only eggplant (brinjal) is considered for the study.

**Keywords:** Machine learning; classification; morphology of Vegetables; data science.

## 1. Introduction

Vegetables plays a significant role in individuals' daily diet. Vegetables offer essential nutrients, contain minimum fat and carbohydrates, but are rich in vitamins, minerals, and dietary fibre. Vegetables are excellent minerals providers, particularly calcium, iron, vitamins A and C [1]. In the financial year 2022-23 India produced an estimated 200 million metric tons of vegetables. These vegetables include, among others, potatoes, onions, eggplants and cabbage.

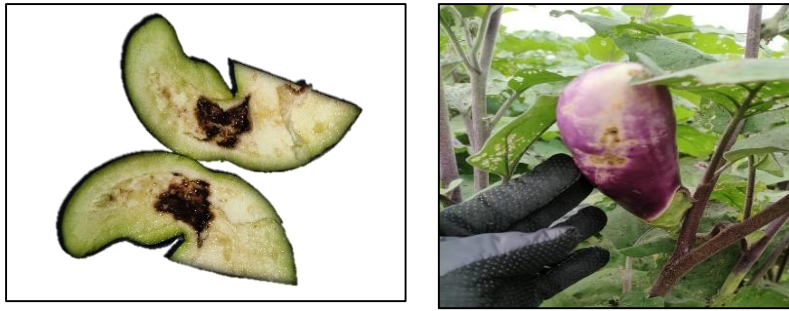
The eggplant (*Solanum melongena* L.) is called Brinjal in South Asia, particularly in India, Pakistan and Bangladesh. The name "eggplant" was given by the USA and Canada because some varieties are shaped like eggs [3]. Eggplant is a versatile crop suitable for cultivation in different agro-climatic regions, with annual cultivation. It is India's second most popular vegetable, after potatoes, tomato, and onion [3]. India has various eggplants with differing tastes, shapes, colours and sizes. Cultivated varieties of eggplants exhibit a wide range of sizes, including small to large, and various shapes, from oblong to round, and oval to club-shaped. They come in diverse colors, including green, white, yellow, and a spectrum of purple hues, spanning from nearly black to striped patterns and gradients [4].

Eggplants are a great provider of many vitamins and nutrients, including protein, fat, Carbohydrates, sugar, vitamin A, vitamin B complex, etc. Additionally, contains the full complement of vitamins, minerals, nutritious fiber, antioxidants, and phytochemicals with scavenging properties [5] [6]. According to the production volume of vegetables in India FY, 2008-2022, on average, 12.98 million metric tons of brinjal are produced in India [2].

Ensuring optimal nutrition requires selecting high-quality vegetables from the market. Unfortunately, pest infestations pose a challenge to consistently obtaining such vegetables. Insects contaminate the vegetables, degrading their quality and making them unsuitable for human consumption. A significant risk to brinjal crops is the brinjal fruit and shoot borer, "Leucinodes orbonalis Guenee," which is responsible for suspectable damages and losses in production. As per AVRDC's research on eggplant entomology, *Leucinodes orbonalis* stands out as the most prominent and severely impacting pest across several Asian nations, including but not limited to India, Pakistan, Sri Lanka, Nepal, Bangladesh, Thailand, and the Philippines [7]. *Leucinodes orbonalis* has four distinct growth stages such as egg, larva, pupa, and adult. Usually, eggs are typically laid one by one on the undersides of young leaves, green stems, flower buds, or the calyces of fruit. Afterwards, the larvae infiltrate flower buds and also gain entry in the same way. During the reproductive period, they invade susceptible fruits through the calyx [8].

\*Corresponding author

E-mail address: jdeb48389@gmail.com



**Figure 1.** *The picture of infected eggplant.*

Brinjal is a popular vegetable that is frequently consumed. However, the quality of brinjal compromised due to insect infestations. This problem in brinjal is a multifaceted issue having profound implications for consumers health and financial losses. Consumers often lack the expertise in accurately identifying such infested products, potentially exposing themselves to health risks and economic losses. Furthermore, infestation may not be easily detectable at the time of purchasing but become evident only during the initial stage of preparation for consumption. This situation poses a significant challenge for consumers in identifying infested brinjal eventually leading to potential financial losses. Addressing this issue is essential to ensure that consumers can make informed decisions about the quality of the vegetables they purchase, promoting food safety and reducing economic losses.

In the field of agriculture, several research studies have utilised classification techniques. For instance, Ajaz and Hussain researched seed classification based on morphological features [9]. At the same time, Malyadri, M.S., and J. focused on developing a classification model to identify suitable crops for specific soil and weather conditions, along with the corresponding fertilizer recommendations [10]. Padoa and Maravillas employed the Naïve Bayesian method for plant classification [11], while Bishnoi et al. utilized classification tools to identify cotton genotypes [12]. Gawande and Dhande reviewed the application of classification techniques in grading fruits based on quality before packaging [13].

Additionally, Lauguico et al. performed lettuce life stage classification using texture attributes [14]. Iroliam et al. employed different machine learning techniques to predict okra's shelf life using various parameters. They found that SVM, Naïve Bayes, and decision tree algorithms effectively predicted okra's shelf life [15]. Davis et al. predicted added sugar content in packaged foods using machine learning, with k-NN showing similar capability to explain variation in added sugar compared to existing approaches [16]. Despite brinjal (eggplant) being India's second most popular vegetable, previous research in agriculture utilising machine learning has yet to address infested brinjals' identification adequately. This research gap motivates our study, where we aim to identify infested eggplants based on their morphological traits using machine learning techniques. Through an assessment of different machine learning techniques, our objective is to identify the most efficient approach for identifying pest infestations in eggplants.

## **2. Variable Selection and Different Classification Tool**

This section presents the variable selection process and the different classification tools employed in this study.

### **2.1. Variable selection**

Here we investigated six morphological characteristics related to fruit and shoot borer resistance. These characteristics were fruit hardness, deformity, surface spots, density (Weight/Volume), stem colour distance, and fruit colour distance. Some of these characteristics were identified based on a literature review. For instance, Krishnaiah & Vijay reported that lower susceptibility to borer injury in specific cultivars might be attributed to the hardness of the fruit skin [17]. Garewal & Singh observed that long-fruited varieties were more susceptible to borer infestations [18]. Additionally, Hazra et al. found a correlation between eggplant weight and susceptibility to fruit infestation by the pest [19]. Furthermore, some morphological characteristics were identified through consultations with farmers and experts in the field.

To evaluate fruit hardness, we manually assessed the fruit's texture using tactile perception. Fruit deformities and surface spots were identified through careful visual examination, analysing their visual attributes.

We used an electric weighing machine to measure the fruit's weight to determine the fruit density. Concurrently, the volume of the fruit was measured using the water displacement method, enabling us to calculate the density as weight per volume accurately.

We used the "Color Identification" application to measure the RGB values of individual stems to assess stem colour distance. We obtained the stem colour distance by subtracting its RGB value from its

corresponding standard RGB value. The standard RGB value for the stem was calculated by averaging the RGB values obtained from various stems. Similarly, the fruit colour distance was estimated using the same approach. The Euclidean distance formula was employed to calculate the distance measurements.

Let  $x$  and  $y$  be two colour surfaces, the Euclidean distance of RGB value of and is given by,

$$d(x, y) = \sqrt{(r_x - r_y)^2 + (g_x - g_y)^2 + (b_x - b_y)^2} \quad (1)$$

Where  $r_x, g_x$  and  $b_x$  represents the RGB value as its individual red, green, and blue components of  $x$ -surface respectively while  $r_y, g_y$  and  $b_y$  represent the red, green and blue of RGB value of  $y$ -surface respectively.

## 2.2. Different Classification Tools

### 2.2.1 k-Nearest Neighbour (k-NN) Classifier

The k-Nearest Neighbour (k-NN) algorithm is a nonparametric and supervised classification method. It was originally developed by Fix & Hodges [20] and subsequently expanded upon by Cover [21]. It works by categorizing data points based on their proximity to nearby neighbours. To improve validity, multiple neighbours are considered, hence the name k-Nearest Neighbour (k-NN) classifier. In this approach, the class of a new observation is determined by analysing the  $k$  nearest neighbours in the dataset.

This algorithm classified as Lazy Learning, as it only stores and memorizes the training data without performing significant computation during the time of training. As a results, it doesn't generalize the training dataset. Consequently, during the testing phase the entire training dataset is needed [22]. The Nearest Neighbourhood classifier is used to classify a new, unlabeled observation based on the positions of its neighbouring data points.

### 2.2.2 Naïve Bayesian Classifier

The Naïve Bayesian classifier stands out as one of the most frequently employed machine learning techniques for classification tasks. It operates on probabilistic principles rooted in the Bayes theorem and offers flexibility in handling an arbitrary number of independent variables, whether they are continuous or categorical [23]. The Naïve Bayes classifier assumes that all the predictor values are independent. In other words, the presence of one particular feature in a class does not affect the presence of another one. This concept is known as class conditional independence [24]. A significant advantage of this classifier is its ability to perform well with small training datasets and it can also handle incomplete data, including instances with missing values.

### 2.2.3 Logistic Regression

Logistic Regression (LR) is a widely used supervised classification technique that derive from the field of Statistics. Its main goal is to discover the appropriate model to explain how a categorical dependent variable is related to one or more independent variables.

### 2.2.4 Linear Discriminant Analysis (LDA)

Another effective classification technique is Linear Discriminant Analysis. It was developed by Ronald A. Fisher in the year 1936. LDA is commonly used for dimension reduction techniques, i.e., separating two or more classes. According to Manage et al. [25], LDA creates the axes that effectively distinguish or separate different classes in the data. The LDA algorithm divides and uses linear boundaries to categorize data based on the predictor variables.

### 2.2.5 Decision Tree

A decision tree (DT) was built by Hunt et al. [26]. It is an essential nonparametric supervised machine learning algorithm. A tree structure with a root node, branches, internal nodes, and leaf nodes characterizes it. It is often drawn from left to right, starting at the root and moving downward. The root node is the node from which the tree begins. Leaf nodes are the nodes at the endpoints of chains. Internal nodes are not leaf nodes and may stretch across two or more branches. While branches show a range of values, nodes reflect a specific attribute. Different feature selection methods exist for nodes, and by those methods, many decision tree construction algorithms exist. CART is the most widely used algorithm for creating decision trees. Depending on whether the dependent variable is categorical or continuous, the nonparametric decision tree algorithm CART generates classification or regression trees [27].

### 2.2.6 Random Forest

Leo Breiman developed the Random Forest (RF) algorithm in 2001 [28]. It is a supervised ensemble machine learning algorithm. The fundamental idea behind the ensemble technique is that it's like having a team of different experts who aren't super accurate on their own, but when they all work together, they become really good at making predictions. In random forests, decision trees serve as the fundamental classifier. A random

subsample of data from the available sample generates each decision tree. In each decision split, features are chosen at random in Random Forest. The association between the trees and the prediction accuracy decreases as the features are randomly chosen. The random forest is appropriate for high-dimensional data sets because it can handle categorical, continuous, and missing values.

The Random Forest algorithm is executed through the following steps:

- If the training group consists of  $N$  cases selected with replacement, then  $N$  cases are randomly taken from the original data to serve as the training group for the tree's growth.
- For an  $M$ -variable input, the variable  $m$  is selected in such a way that  $m \ll M$  for each node.  $m$  variables are then randomly selected from  $M$ , and used to split the node. The value of  $m$  remains constant throughout the forest's widening.
- Each tree in the Random Forest is permit to grow its full potential without any pruning or trimming [29].

### 2.3 Performance Evaluation Method

In this study, we evaluate the performance of the classification models by assessing their accuracy and error rate. These metrics are commonly used to measure the effectiveness of classifiers and are derived from the confusion matrix. The confusion matrix comprehensively summarises the model's predictions and the actual class labels. The accuracy is determined by dividing the total number of correct predictions by the total number of observations. By examining these measures and analysing the confusion matrix, we gain insights into the performance of the classification models in our study.

$$\text{Accuracy} = \frac{tp + tn}{tp + fp + fn + tn} \tag{2}$$

Where,  $tp$  = true positive,  $tn$  = true negative,  $fp$  = false positive and  $fn$  = false negative. When there is a significant imbalance in the number of observations between classes, accuracy and error rate may not be reliable performance indicators [30]. Hence, additional metrics are employed to evaluate the classification model's performance comprehensively. Sensitivity and G-mean are positive measures, ranging from 0 to 1, with higher values indicating better performance. Specificity, precision, and F1 score are also positive measures, with values ranging from 0 to 1, where higher values indicate better performance. These metrics, derived from the confusion matrix of each respective classification model, provide a more comprehensive assessment of the model's performance beyond accuracy.

$$\text{Sensitivity} = \frac{tp}{tp + fn} \tag{3}$$

$$\text{Specificity} = \frac{tn}{tn + fp} \tag{4}$$

$$\text{Precision} = \frac{tp}{tp + fp} \tag{5}$$

$$\text{F1 score} = \frac{2 \times \text{sens} \times \text{prec}}{\text{sens} + \text{prec}} \tag{6}$$

$$\text{G-mean} = \sqrt{\text{sens} \times \text{prec}} \tag{7}$$

### 3. Findings and Deliberations

This segment presents the findings and deliberation of the data analysis phase, where the dataset was partitioned into training and holdout samples in an 80:20 ratio. To ensure the robustness and reliability of the findings, the analysis was conducted through ten iterations with replacement. This approach accounts for the variations in model performance due to specific data instances and allows for a comprehensive evaluation.

**Table 1** presents a comprehensive overview of the results obtained from the various methods utilized, encompassing metrics such as true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN). Additionally, it includes the average accuracy percentage computed over ten iterations. These findings offer insights into how well each method can detect and categorize the target variable, offering valuable information about their performance and appropriateness for the task at hand.

**Table 1.** Confusion matrix results for 10 iterations of sample size 80:20

Method	Correct Prediction		Wrong Prediction		Average Accuracy per cent
	TP	TN	FP	FN	
<b>k-Nearest Neighbour</b>	10,12,9,9,11, 8,7,7,10,12	4,9,9,11,12 13,11,6,10,7	1,5,3,3,1 2,5,2,3,1	6,2,5,5,2 6,2,2,6,10	72.4
<b>Naïve Bayes</b>	8,4,8,13,13,8, 10,7,10,10	14,7,6,8,7,6,7, 13,12,13	3,4,2,6,2,8,4,7,0,2	0,5,0,2,1, 3,4,2,3,3	75.30
<b>Logistic Regression</b>	12,7,6,11,9,8, 16,10,5,13	8,15,13,11,12, 11,17,8,9,14	2,4,5,0,0,2,5,2,3,4	3,2,2,3,3, 2,2,4,2,0	80.81
<b>Linear Discriminant Analysis</b>	7,5,9,9,5,9, 9,9,10,9	9,12,10,5,5,8, 10,11,13,15	2,3,9,4,2,2,4,2,4,2	4,5,0,4,3, 5,3,6,1,1	72.52
<b>CRAT</b>	10,11,8,12,7, 6,7,5,3,8	18,12,6,12,9, 10,10,15,11,9	6,4,3,1,4,1,2,5,4,3	1,1,1,5,3, 3,2,0,3,6	76.25
<b>Random Forest</b>	4,6,10,10,11, 8,8,9,8,9	11,9,8,11,9, 13,8,14,6,10	4,7,2,4,4,7,3,2,4,4	0,2,6,6,2, 1,1,3,4,5	72.14

A critical aspect of implementing the *k*-NN classifier is choosing the optimal value for *k*. In this study, the square root of the total of observations in the train sample determined *k* = 11. However, it is worth noting that the *k*-NN classifier may introduce bias due to using the Euclidean distance, which tends to favour variables with larger values. A subsequent step to address this potential bias involved normalizing the data using the min-max normalization function. The *k*-NN model was then constructed using the training data, and the testing data was utilised for cross-validation purposes. For the *k*-NN model, due to multiple iterations, the accuracy shows fluctuations, making it challenging to pinpoint precise results. However, when averaging the outcomes over all samples, the model achieved an accuracy rate of 72.4 per cent.

The Naïve Bayes model also exhibited varying accuracy across ten iterations. On average, the model demonstrated an overall accuracy of 75.3 per cent, indicating its competitive performance.

In contrast, the Logistic Regression model outperformed the others with an average accuracy of 80.80 per cent, showcasing its effectiveness in classifying the data.

The Linear Discriminant Analysis (LDA) model achieved an average accuracy of 72.58 per cent across ten iterations, showing its capability to perform reasonably well on the given task.

For the decision tree constructed using the CRAT algorithm, the model attained an average accuracy of 76.25 per cent after ten iterations, making it a viable choice for classification tasks.

In the case of the Random Forest model, a crucial aspect was determining the appropriate number of trees. In this study, 1000 decision trees were utilized to achieve better performance and accuracy. On average, across ten iterations, the Random Forest model achieved an accuracy of 72.14 per cent, demonstrating its suitability for the task.

Overall, the Logistic Regression model showed the highest average accuracy, followed closely by the decision tree constructed using the CRAT algorithm. These findings provide valuable insights into the effectiveness of different machine learning models and can be helpful to guide the selection of the most fruitful approach for similar classification tasks.

In an effort to explore how these models perform with smaller training samples, we conducted an additional test using a 60:40 samples split, as presented in **Table 2**. It is evident from **Table 2**, that the *k*- Nearest Neighbour (*k*-NN) model exhibited variability in accuracy across ten iterations, ultimately yielding an average accuracy of 74.68 per cent. Notably, Naïve Bayes consistently delivered a competitive performance, demonstrating an average accuracy of 75.03 per cent, reaffirming its reliability in the context of pest infestation identification. Likewise, the Logistic Regression model remain effective, with an average accuracy of 74.67 per cent, albeit slightly lower than the previous results, underscoring its robust classification capabilities.

Linear Discriminant Analysis (LDA), consistently achieved an average accuracy of 74.73 per cent, reflecting its reliability in delivering consistent and reasonable performance. Moreover, both the CRAT and Random Forest models showcased their suitability for the task, achieving average accuracies of 73.88 per cent and 72.72 per cent respectively. The findings from the 60:40 sample split indicate that the performance patterns of these models remain consistent with the 80:20 results.

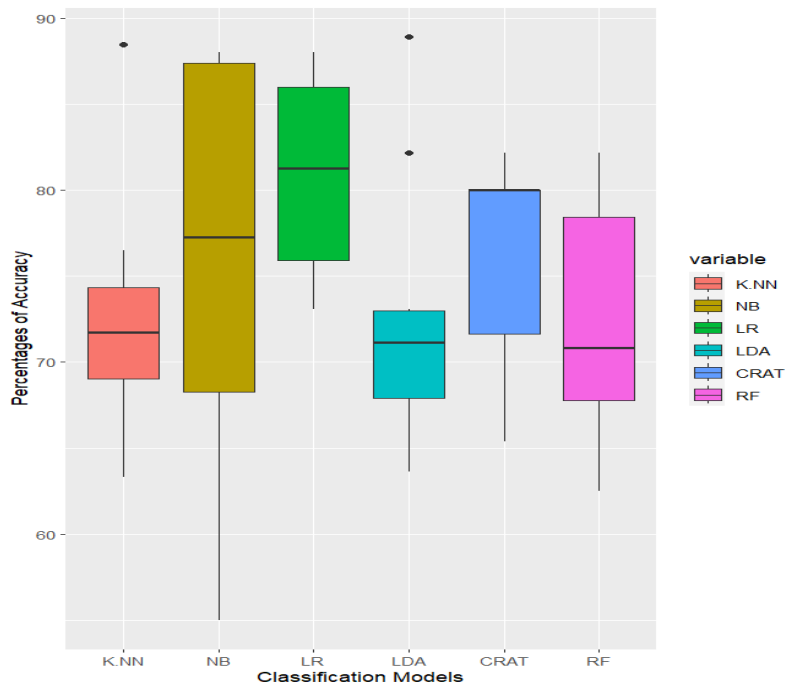


**Table 2.** Confusion matrix results for 10 iterations of sample size 60:40

Method	Correct Prediction		Wrong Prediction		Average Accuracy per cent
	TP	TN	FP	FN	
<b>k-Nearest Neighbour</b>	28,19,19,11,14, 15,15,17,21,16	18,23,22,17,18, 15,16,21,21,26	4,7,2,9,9, 1,2,5,4,8	13,6,9,5,4, 12,7,4,8,6	74.68
<b>Naïve Bayes</b>	13,15,19,17,19, 19,20,24,19,18	26,22,20,23,18, 22,24,21,16,18	9,4,4,4,6, 4,7,11,6,8	6,6,8,4,5, 15,4,4,10,7	75.03
<b>Logistic Regression</b>	17,15,19,17,16, 14,20,12,14,18	25,22,25,16,18, 25,23,22,22,17	8,8,8,7,10, 5,3,7,9,6	4,5,3,5,6, 6,10,7,4,6	74.67
<b>Linear Discriminant Analysis</b>	18,20,17,7,18, 17,19,14,15,18	18,21,17,24,17, 16,24,26,22,21	7,10,8,3,10, 8,4,5,10,6	4,2,4,7,2, 5,11,11,3,5	74.73
<b>CRAT</b>	15,10,19,9,14, 12,16,17,8,10	15,18,17,21,18, 23,23,25,16,23	6,10,5,8,7, 7,8,8,6,4	5,10,2,2,6, 4,2,4,5,6	73.88
<b>Random Forest</b>	13,19,17,13,14, 18,18,14,18,19	21,21,19,20,16, 19,23,19,25,21	11,5,7,7,14, 3,6,10,4,2	4,8,8,7,6, 7,8,3,7,10	72.72

**3.7 Comparison of different classification models**

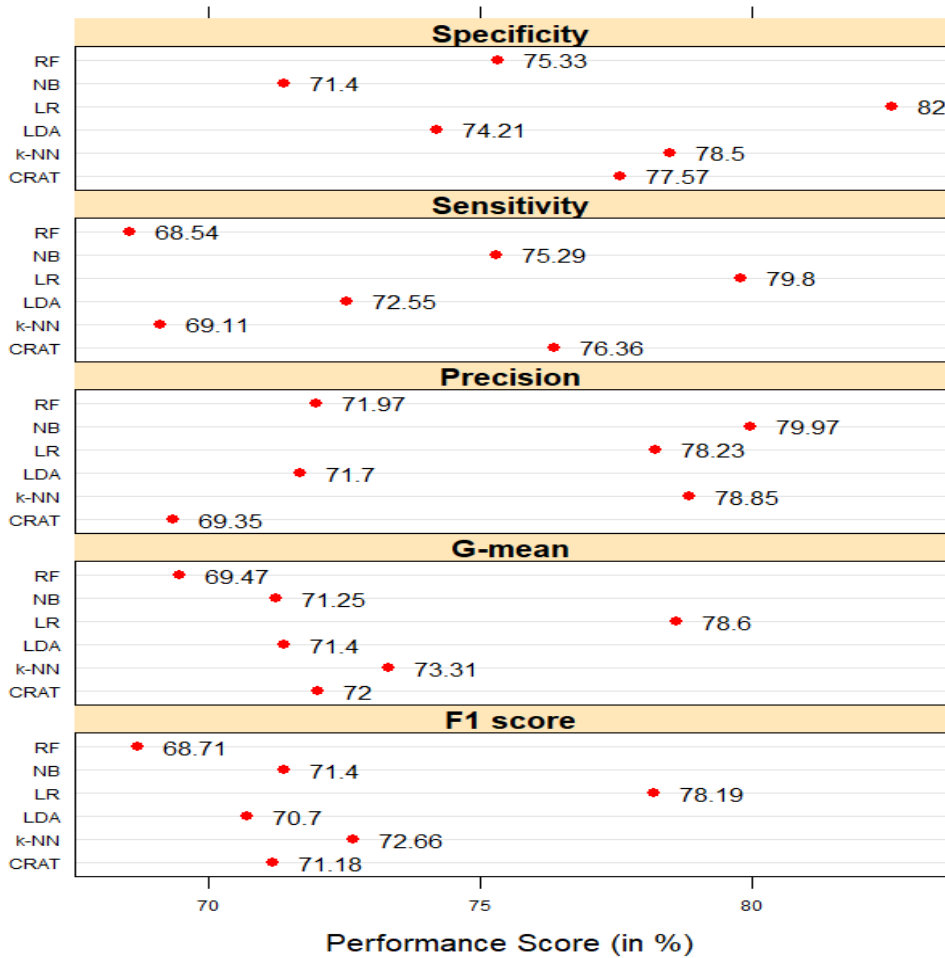
**Figure 2.** illustrates the results of the various classification models analysed in this study. According to the findings, the logistic regression model exhibited the highest accuracy rate of 80.80 per cent, indicating success.



**Figure 2.** Boxplot of the accuracy for the different classification model

When considering the F1 score and G-mean metrics, the Naive Bayes and Logistic Regression models consistently rank at the forefront. These models consistently exhibit strong performance in both F1 score and G-mean, showcasing their effectiveness in accurately classifying the data.

Based on the results, the Naive Bayes and Logistic Regression models consistently outperform the other models in detecting infected eggplants. These two models exhibit superior performance across multiple evaluation metrics, indicating their effectiveness in accurately identifying infested samples.



**Figure 3.** Sensitivity, Specificity, Precision, F1 score and G-mean for different classification models classification models.

**4. Conclusion**

The goal of this research is to assess the effectiveness different supervised classification models, namely *k*-Nearest Neighbour (*k*-NN), Naïve Bayes (NB), Logistic Regression (LR), Linear Discriminant Analysis (LDA), CRAT of decision tree, and Random Forest. The aim is to evaluate the efficiency of these models in classification.

The study considers six morphological characteristics of the eggplants, including fruit hardness, deformity, surface spots, density (Weight/Volume), stem colour distance, and fruit colour distance. The database is categorized into two parts: training data for build-up models and testing data for cross-validation. The accuracy of the models is assessed using a confusion matrix that recorded accurate and inaccurate classifications for all classes. The analysis revealed that Naive Bayes, logistic regression, and CRAT consistently demonstrated favorable performance, particularly when considering the average accuracy over ten iterations.

The *k*-NN model achieves an average accuracy of 72.4 per cent in classifying infested eggplants. Comparatively, the NB model exhibits a relatively higher accuracy of 75.3 per cent. The LR model demonstrates the highest accuracy among all classification models, with a rate of 80.8 per cent. Following LR, the accuracy scores for LDA, CRAT, and RF were 72.52 per cent, 76.25 per cent, and 72.14 per cent, respectively. When considering the classification of infested eggplants, the Logistic Regression model emerges as the most accurate among all other models, with CRAT and Naive Bayes occupying the second and third positions, respectively. However, Random Forest has poor accuracy compared to classification models in this dataset.

To facilitate a comprehensive comparison, we computed additional performance measures, including sensitivity, specificity, accuracy, F1 score, and G-mean. Our analysis revealed that logistic regression achieved the highest performance value among the evaluated metrics. Following closely, Naive Bayes and CRAT decision trees ranked second and third, respectively, for the classification of infected eggplants. This assessment allowed for a more detailed evaluation of the models' effectiveness in accurately classifying the target samples.

Based on the findings, it can be concluded that the logistic regression, Naive Bayes, and CRAT decision tree models demonstrate sufficient capability to categorize the infected eggplants effectively. These models exhibit promising performance across various evaluation metrics, suggesting their suitability for accurate classification in this context.

In agriculture, there is a significant potential for future research utilizing machine learning techniques. Currently, our focus is primarily on one specific vegetable, namely eggplant. However, the study can encompass a broader range of vegetables. Also, utilizing machine learning to assess the quality attributes of vegetables, such as ripeness, freshness, and nutritional content can be studied. This information can aid in quality control and consumer satisfaction.

Furthermore, using regression-based machine learning tools, one can evaluate the nitrogen content in vegetables, as elevated levels of nitrates have been associated with endogenous nitrosation. This process has been implicated in developing thyroid conditions, several types of human cancers, neural tube defects, and diabetes. Assessing the nitrogen content levels in eggplant or in any other horticultural crop can disclose their potential effect on health.

## References

- [1] Ulger TG, Songur AN, Cirak O and Cakiroglu FP, "Role of Vegetables in Human Nutrition and Disease Prevention", in *Vegetables-Importance of Quality Vegetables to Human Health*, Intechopen, 2018, pp. 7-32; doi: 10.5772/intechopen.77038.
- [2] Gowda LR, "Genetically Modified Aubergine(Also Called Brinjal or Solanum melongena)" in *Genetically Modified Organisms in Food*, (2016), 27-37; doi: 10.1016/B978012802259700004-X.
- [3] S. Herbst, "The New Food Lover's Companion: Comprehensive Definitions of Nearly 6,000 Food, Drink, and Culinary Terms. Barron's Cooking Guide," *Hauppauge, NY : Barron's Educational Series. ISBN 0764112589*.
- [4] Y. Noda , T. Kaneyuki, K. Igarashi and A. Mori, "Antioxidant activity of nasunin, an anthocyanin in eggplant peels," *Toxicology*, pp. 119-123, 2000.
- [5] B. Whitaker and J. Stommel, "Distribution of Hydroxycinnamic Acid Conjugates in Fruit of Commercial Eggplant (*Solanum melongena* L.) Cultivars," *Journal of Agricultural Food Chemistry*, vol. 51, pp. 3448-3454, 2003.
- [6] A. Minhas, "Production volume of vegetables India FY 2008-2022," 22 03 2023. [Online]. Available: <http://www.statista.com>.
- [7] AVRDC Eggplant entomology, "Control of eggplant fruit and shoot borer. Progress Report," Asian Vegetable Research and Development Center,(AVRDC), Shanhua,Taiwan, 1994.
- [8] E. A. Netam M ., "Screening of shoot and Fruit Borer(*Leucinodes orbonalis* Guenee) for Resistance in Brinjal (*Solanum melongena* L.) Germplasm Lines," *International Journal of current Microbiology and Applied Sciences* 7.2, pp. 3700-3706, 2018.
- [9] R. H. Ajaz and L. Hussain, "Seed Classification using Machine Learning Techniques," *Journal of Multidisciplinary Engineering Science and Technology(JMEST)*, pp. 1098-1102, 2015.
- [10] Manoj Kumar D P, Malyadri N, Srikanth MS, Ananda Babu J, "A Machine Learning model for Crop and Fertilizer recommendation", *Natural Volatiles & Essential Oils*, 8(5) (2021) 10531-10539.
- [11] Padoa F.R.F. & Maravillas E. A. "Using Naive Bayesian method for plant Leaf classification based on shape and texture features" *2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment Management(HNICEM)*, 2015.
- [12] S. Bishnoi, N. A. Ansari, M. Khan, S. Heddami and A. Malik, "Classification of Cotton Genotypes with Mixed Continuous and Categorical Variables: Application of Machine Learning Models," *Sustainability*, pp. 14,13685, 2022.
- [13] A. P. & D. S. S. Gawande, "Implementation of fruit grading system by image processing and data classifier-a review," *International Journal of Engineering Research and General Science*, vol. 2, no. 6, pp. 411-413, 2014.
- [14] S. C. Lauguico, R. I. S. Cocepacion, J. D. Alejandrino, R. R. Tobias and E. P. Dadios, "Lettuce life stage classification from texture attributes using machine learning estimators and feature selection processes," *International Journal of Advances in Intelligent Informatics*, pp. 173-184, 2020.
- [15] I. ., Iorliam, " Application of Machine Learning Techniques for Okra Shelf Life Prediction," *Journal of Data Analysis and Information Processing*, pp. 136-150, 2021.

- [16] Davies T, Yu Louie JC, Ndanuko R, Barbieri S, Perez-Concha O, H Y Wu J, "A Machine Learning Approach to Predict the Added-Sugar Content of Packaged Foods", *The Journal of Nutrition* 152(1), (2022) 343-349.
- [17] K. Krishnaiah and O. Vijay, "Evaluation of brinjal varieties for resistance to shoot and fruit borer," *Indian J Hort*, pp. 84-86, 1975.
- [18] R. Garewal and D. Singh, "Fruit characters of brinjal in relation to infestation by *Leucinodes orbonalis* Guen," *Indian J Ent*, vol. 57, pp. 336-343, 1995.
- [19] P. Hazra, R. Dutta and T. Maity, "Morphological and Biochemical characters associated with field tolerance of brinjal (*Solanum melongena* L.) to shoot and fruit borer and their implication in breed for tolerance," *Indian Journal Genet*, pp. 255-256, 2004.
- [20] E. Fix and J. L. Hodges, "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties," *USAF School of Aviation Medicine, Randolph Field, Texas*, 1951.
- [21] T. M. Cover and P. E. Hart, " "Nearest neighbor pattern classification", " *IEEE Transactions on Information Theory*, p. 13 (1): 21–27, 1967.
- [22] N. a. N. D. a. T. P. Ali, "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets," *SN Applied Sciences*, 2019.
- [23] H. Zhang, "The Optimality of Naive Bayes," *Proceedings of the Seventeenth International Florida Artificial Intelligence* .
- [24] P. A. a. L. N. Flac, "Naive Bayesian Classification of Structured Data," *Machine Learning, Boson: Kluwer Academic Publisher*, pp. 1-37, 2004.
- [25] A. W. Manage, " classification of all rounders in limited over cricket - a machine learning approach," *Journal of Sports analytics*, pp. 6(4),295-306, 2020.
- [26] E. Hunt, J. Marin and P. Stone, "Experiment in induction academic press," *N.Y.*, p. 247, 1966.
- [27] L. Breiman, J. Friedman, R. Olshen and C. Stone , "Classification and Regression Trees," *Chapman Hall/ CRC Press: New York, NY, USA*, 1984.
- [28] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5-32, 2001.
- [29] J. Ali, R. Khan and N. Ahmad, "Random Forests and Decision Trees," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 5, pp. 272-278, 2012.
- [30] M. Kubat, R. Holte and S. Matwin, "Machine learning for the detection of oil spoils in satellite radar images.," *Mach. Learn*, vol. 30, pp. 195-215, 1998.

# Efficient and Accurate Date Extraction from Invoices: A Comprehensive Three-Step Methodology Integrating Custom Object Detection, OCR, and Refined Regular Expressions

Mehmet Hilmi Emel<sup>1,\*</sup> , Murat Terzioğlu<sup>1</sup> , Ramazan Özkan<sup>2</sup> 

<sup>1</sup> Egaranti Teknoloji A.Ş., Istanbul, Turkey;

<sup>2</sup> Haliç University, Faculty of Engineering, Department of Computer Engineering, Istanbul, Turkey;

## Abstract

In the realm of contemporary document processing, the challenge of extracting crucial information from diverse invoices necessitates innovative solutions. This article presents a comprehensive three-step methodology to address the complexity of date extraction from invoices. Leveraging LabelStudio, Python, and OpenCV, we constitute a dataset and train a custom object detection model using Ultralytics YOLOv8. Optical Character Recognition (OCR) provides us to convert the image data to string data that is enable to be processed. Regular expressions refine the extracted text, achieving precise date formats. The developed system significantly enhance the time efficiency, marking a noteworthy advancement in date extraction from invoices.

**Keywords:** *Custom Object Detection; OCR; Invoice Processing; YOLOv8.*

## 1. Introduction

In the dynamic landscape of contemporary document processing, the challenge of extracting crucial information from a multitude of invoices underscores the necessity for innovative and adaptable solutions. Invoices, being pivotal financial documents, exhibit a remarkable diversity in terms of structure, formatting, and content, rendering the creation of a one-size-fits-all algorithm for date extraction a formidable task. Conventional rule-based methodologies, while effective in specific contexts, struggle to accommodate the intricate variations presented by invoices globally. Recognizing this complexity, our research endeavors to overcome these limitations by proposing a comprehensive three-step methodology: Custom Object Detection, OCR, and Regular Expression. To annotate our dataset and train our object detection model, we harness the capabilities of LabelStudio [1], an efficient data labeling tool. This tool, coupled with the versatility of Python and OpenCV libraries, empowers us to meticulously delineate regions of interest within our invoice images. Moreover, to streamline our object detection endeavors, we use the Ultralytics YOLOv8 [2] framework, enhancing the efficiency and accuracy of our model training process. The usage of these technologies allows us to craft a dataset that not only captures the nuances of invoice layouts but also serves as a robust foundation for training an object detection model. Subsequently, we mention the transformative realm of Optical Character Recognition (OCR) using the Paddle OCR [3], converting the visually encoded text within identified regions into machine-readable formats. As we continue to study on this research, we produce regular expressions to refine and pinpoint the date formats within the extracted text. In presenting our comprehensive approach, we not only showcase the technical intricacies but also underscore the practical implications and advancements in the realm of document processing, marking a significant step forward in the quest for efficiency and accuracy in date extraction from diverse invoices.

## 2. Related Works

Satav et al. [4]: Addressing the challenges of invoice management in small enterprises, their web-based application offers a streamlined solution. Recognizing the prevalent use of traditional manual systems, their focus is on accessibility and data extraction. The proposed system employs OpenCV for preprocessing, OCR for data extraction, and RegEx for refining the results. Specifically tailored for small businesses, this application aims to enhance efficiency and accessibility in managing invoice data.

Kosiba et al. [5]: A method for analyzing invoice document structure is proposed to extract essential information. The approach combines textual and graphical processing, examining line and line intersection features, and searching for keywords like item number and total. Valid keyword search regions are identified using connected-component analysis before OCR. Results from keyword search and line analysis are combined to extract relevant data. This analysis will be integrated into a larger invoice interpretation system currently in development.

\*Corresponding author

E-mail address: mehmethilmi81@gmail.com

Received: 6/Dec/2023; Accepted: 30/Aug/2024.

Sidhwa et al. [6]: This study aims to develop a methodology for extracting data from everyday printed bills and invoices, which can be utilized for tasks like machine learning and statistical analysis. The focus is on extracting key information such as the final bill amount, itinerary, and date from these documents, which contain valuable insights into user preferences. Optical Character Recognition (OCR) technology, specifically the Tesseract OCR engine [7],[8], is employed. Initially, OpenCV is utilized for bill or invoice detection and noise reduction in the image. The Tesseract OCR engine, incorporating Text Segmentation, is then applied to extract written text in various fonts and languages. The methodology demonstrates high accuracy when tested on diverse input images of bills and invoices.

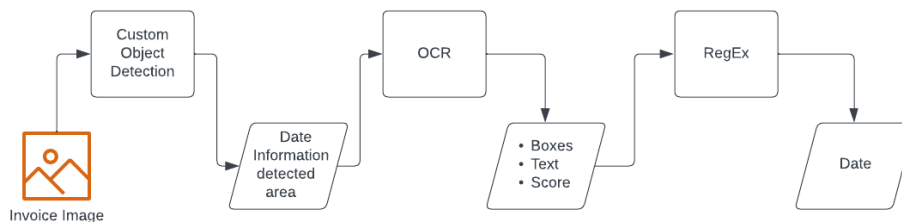
Karsligil et al. [9]: This conference paper introduces a novel texture-based method for identifying text areas in complex document images. The approach utilizes Gabor filter, inspired by the multi-channel filtering approach of the Human Visual System (HVS), to generate an energy map of the document. Text areas, presumed to be rich in high-frequency components, are extracted as connected components through binarization of the energy map employing Otsu's adaptive threshold method. Initial removal of non-text components like pictures and lines is achieved through Gabor filtering. A distinctive aspect of the method involves further elimination of remaining non-text components using character component interval tracing. This two-stage elimination process enhances the accuracy of detecting text areas in various types of digital documents.

Zhang et al. [10]: This paper introduces a method for extracting text from historical Tibetan document images, treating the task as a problem of detecting and locating text areas. The process involves preprocessing the images to address illumination imbalances, tilt, and noise, resulting in a binary image. The regions of interest in historical Tibetan documents are categorized using connected components and grid-based filtering. The remaining grids are utilized to compute vertical and horizontal projections, enabling the detection of the approximate text area location. The final step involves accurate extraction of the text area by correcting the bounding box of the approximate text area. Experimental results on a historical Tibetan document image dataset illustrate the effectiveness of the proposed method.

Dachengwang et al. [11]: Analyzing images of filled-out forms is a significant problem with practical applications in office automation and theoretical challenges for document image analysis. The method described in this work focuses on extracting both typed and handwritten text from scanned and digitized images of completed forms. Without relying on prior knowledge of form structure, the approach decomposes a filled-out form into three fundamental components: boxes, line segments, and the remainder (including handwritten and typed characters, words, and logos). The input binary image undergoes segmentation into small and large connected components. Complex boxes are further broken down into elementary regions using key-point analysis. To separate handwritten and machine-printed text that intersects guide lines and boxes, lines are removed. Characters fragmented by line removal are then reconnected using a character patching method. The effectiveness of the method is demonstrated through experimental results with filled-out forms from various domains, including insurance, banking, tax, retail, and postal sectors.

### 3. Methodology

We proceed through three steps for this purpose: Custom Object Detection, OCR, and RegEx rules. Initially, we employ our custom object detection model to identify the area containing historical information within our proprietary dataset. Subsequently, within the identified region, we employ Optical Character Recognition (OCR) to transform text from a photographic format to a textual format. In the third step, we utilize regular expressions to extract the desired date format from the textual data. The flowchart depicting the methodology employed in this study is illustrated in **Figure 1**.



**Figure 1.** The Flowchart of the methodology

There were two main reasons for developing our object detection model. The first reason was that when applying OCR technology directly to the entire invoice image, the scanning process of the entire invoice by OCR was time-consuming. Therefore, when turning our developed system into a product, using OCR directly on the entire invoice would make the system inefficient in terms of time. By utilizing our Custom Object Detection model, we could identify a smaller region within the invoice image, specifically containing the date information, and converting the text within that area into string format using OCR proved to be faster. This

approach enhanced the efficiency of the system in terms of processing time. The second reason was evident in **Figure 2**, where the sections containing date information on invoices varied. Developing a rule-based algorithm seemed insufficient in this regard, as it would require examining all invoice images and determining a rule for each invoice example. This approach would be impractical due to the diversity in the location of date information across different invoices.

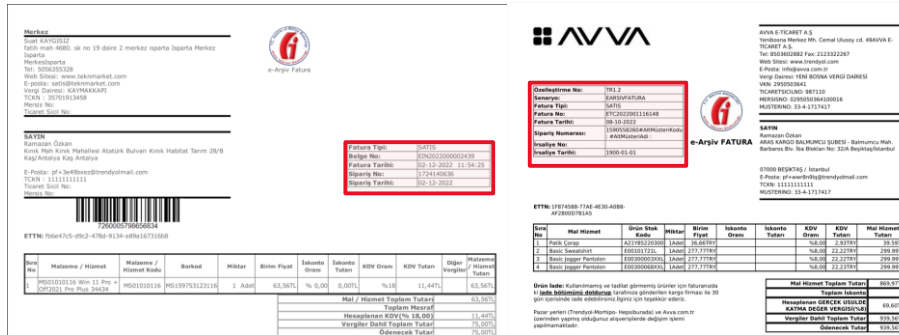


Figure 2. Location of date information in different invoice samples

### 3.1 Custom Object Detection Model

One of the primary reasons for incorporating object detection into our workflow is the extended length of date information obtained when applying OCR alone. As OCR is required to scan the entire invoice image, locating the date information naturally incurred a longer processing time. By employing object detection to specifically scan the region containing the date information, the response time has been significantly reduced. We conducted the development of the object detection in three stages. The first stage involved preparing the dataset, the second stage focused on model development, and the third stage centered around model evaluation.

#### 3.1.1 Preparing Dataset

To construct our dataset, we initiated the data labeling process employing the Label Studio tool. In the process of dataset creation, we judiciously labeled the region containing date information with the tag "invoice\_data." Subsequently, to encompass the details of the individuals acquiring the product in our subsequent endeavors, we introduced the label "personal\_data" (The area highlighted within the red rectangle in **Figure 3**). In our pursuit of enhancing the reliability and efficacy of the Object Detection model under development, deliberate efforts were made to diversify our dataset. This involved the augmentation of data variety by engaging in the collection of product acquisition information from diverse users representing various companies. As a result of these initiatives, we amassed a total of 161 invoice examples. The foundational step in our dataset construction involved the utilization of the Label Studio tool, a pivotal instrument for data labeling processes. Specifically, we employed the "invoice\_data" (The area highlighted within the blue rectangle in **Figure 3**) label to demarcate the regions within our dataset housing pertinent date information. Subsequent refinement of our project objectives necessitated the incorporation of the "personal\_data" label to address the additional dimension of identifying and processing the information pertaining to individuals involved in the product acquisition process.

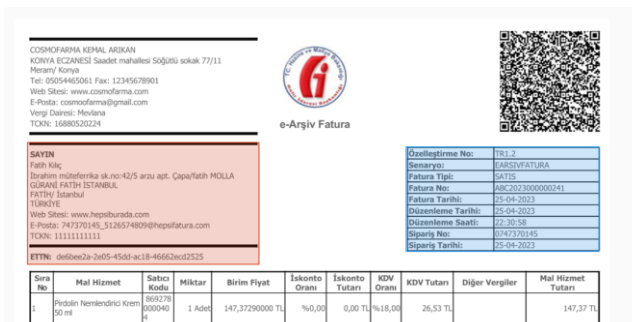
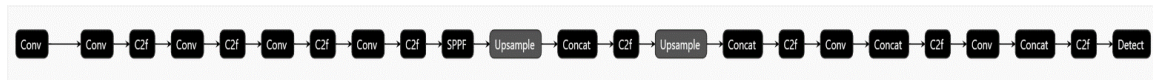


Figure 3. Labelling the invoices

### 3.1.2 Creating Custom Object Detection

For the construction of our model, we opted to utilize the Ultralytics YOLOv8 model. The Ultralytics YOLOv8 represents a state-of-the-art model within the field of object detection and tracking, instance segmentation, image classification, and pose estimation. This cutting-edge iteration builds upon the achievements of its YOLO predecessors, incorporating novel features and enhancements to augment both performance and adaptability. YOLOv8 is engineered to deliver high-speed processing, precision, and user-friendly operation, rendering it a highly commendable option for diverse applications in computer vision tasks. The model's advancements are poised to contribute significantly to the evolution of methodologies employed in tasks such as object recognition, tracking, and image analysis. The architecture of the YOLOv8 object detection model is illustrated in **Figure 4**.



**Figure 4.** The architecture of the custom object detection model.

We conducted experiments on this architecture using various optimizers and learning rate values to determine the optimal performance of our model.

### 3.1.3 Evaluating the Object Detection Model

To attain optimal results in our custom object detection model, we conducted a comprehensive study by systematically exploring various epoch values, learning rate parameters, and optimizer configurations. Our objective was to identify the most effective combination of these hyperparameters to enhance the model's performance. Through a series of experiments, we calculated the Mean Average Precision at 50 (mAP50) for each model, aiming to ascertain the impact of different settings on the precision and recall of the object detection framework. This iterative exploration allowed us to fine-tune our model by considering the interplay between epoch duration, learning rate adjustments, and optimizer choices, providing valuable insights into the nuanced dynamics of our custom object detection architecture. Deep learning models heavily rely on optimization algorithms to enhance their training process and achieve optimal performance. In this study, we conducted a detailed examination of various optimization algorithms, including Stochastic Gradient Descent (SGD), Adam, Adamax, AdamW, Nadam, and RAdam, across different learning rates (lr) and epochs. The performance of these algorithms was evaluated based on their impact on the accuracy of a deep learning model. **Table 1** illustrates the performance of the optimization algorithms.

**Table 1.** The experimental results obtained with different epochs, learning rates, and optimizer values

Epoch=10	SGD	Adam	Adamax	AdamW	NAdam	RAdam
Lr=0.1	0.00075	0.00963	0.00682	0.00178	0.00021	0.00127
Lr=0.01	0.93539	0.00293	0.81918	0.0027	4.00E-05	0.67992
Lr=0.001	0.68521	0.96554	0.98308	0.96151	0.89302	0.90717
Epoch=20	SGD	Adam	Adamax	AdamW	NAdam	RAdam
Lr=0.1	0.06638	0.00164	0.03064	0.0087	0.00147	0.00082
Lr=0.01	0.98723	0.8449	0.97058	0.26271	0.00212	0.97742
Lr=0.001	0.93794	0.98947	0.99483	0.99106	0.98192	0.95569
Epoch=50	SGD	Adam	Adamax	AdamW	NAdam	RAdam
Lr=0.1	0.98467	0.13369	0.40084	0.15401	0.00322	0.9861
Lr=0.01	0.995	0.98432	0.99226	0.98604	0.70449	0.99484
Lr=0.001	0.98942	0.995	0.99492	0.995	0.99145	0.995

For the Stochastic Gradient Descent (SGD) algorithm, higher learning rates (e.g., lr=0.1) initially resulted in a significant error rate. Lower learning rates (e.g., lr=0.001) produced a more stable performance, albeit requiring an extended training period. Notably, with increased epochs, SGD demonstrated convergence, indicating improved accuracy, particularly at lower learning rates.

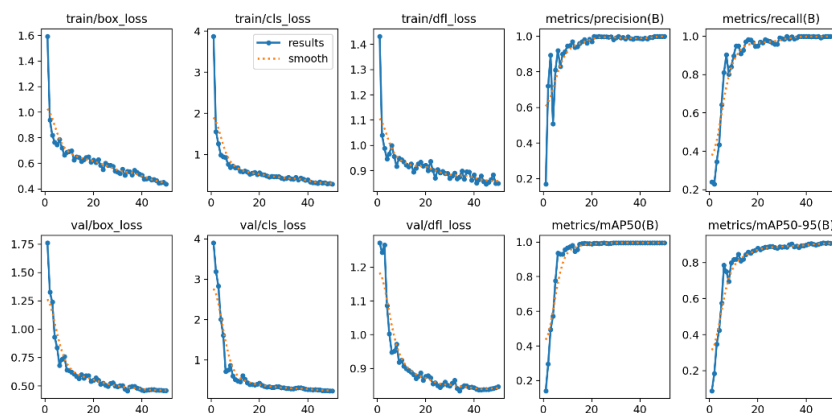
The Adam optimization algorithm, incorporating adaptive momentum, exhibited superior performance at lower learning rates (e.g., lr=0.001), while displaying a potential decrease in accuracy at higher rates (e.g., lr=0.1). Longer training periods (epochs=20 and epochs=50) generally favored Adam, indicating a progressive improvement in accuracy over time. Adamax, a variant of Adam that considers larger gradient



values, demonstrated consistent performance, particularly at lower learning rates ( $lr=0.001$ ), compared to other algorithms. Across different epochs, Adamax maintained a stable accuracy profile, showcasing its reliability in various training scenarios. AdamW, an extension of Adam with weight decay, consistently outperformed other algorithms, especially at lower learning rates ( $lr=0.001$ ). As epochs increased, AdamW showcased sustained improvement in accuracy, suggesting its efficacy in capturing complex patterns over extended training periods. Nadam, incorporating Nesterov's momentum into Adam, showcased effective performance, particularly at lower learning rates (e.g.,  $lr=0.001$ ). The convergence of Nadam was evident across epochs, with a steady increase in accuracy. Rectified Adam (RAdam), an adaptive learning rate variant of Adam, proved to be effective, particularly at lower learning rates (e.g.,  $lr=0.001$ ). RAdam demonstrated a consistent and notable increase in accuracy as epochs progressed, underlining its suitability for prolonged training durations. As a result of our extensive investigations, we identified the optimal parameters for our custom object detection model as follows: setting the epoch value to 50, establishing a learning rate of 0.001, and utilizing the Adam optimizer.

When evaluating our model with an epoch value of 50, a learning rate of 0.001, and an Adam optimizer, the train/box\_loss graph in **Figure 5** illustrates a decreasing trend in the loss value as the epoch increases. A similar pattern is observed in the val/box\_loss graph. Notably, the train/cls\_loss graph exhibits a dramatic reduction in cls\_loss during the initial epoch, with a subsequent gradual decrease as the epoch progresses. This pattern is mirrored in the val/cls\_loss graph. The train/dfl\_loss graph shows an initial sharp decline followed by a somewhat fluctuating but overall decreasing trend. A comparable movement is observed in the Val/dfl\_loss graph.

Examining the metrics/precision(B) graph in **Figure 5**, despite initial fluctuations in the ratios during the first epochs, precision stabilizes and approaches a value close to 1 in subsequent epochs. The metrics/recall(B), metrics/mAP50(B), and metrics/mAP50-95(B) graphs demonstrate similarities among each other. They all exhibit a sudden increase in the early epochs, followed by a slower ascent in subsequent values. It is worth noting that, despite the occasional fluctuations, the metrics/precision(B) curve attains stability, emphasizing the model's precision as it progresses through epochs. Additionally, the metrics/recall(B), metrics/mAP50(B), and metrics/mAP50-95(B) graphs collectively indicate the model's ability to maintain a consistent performance in terms of recall and mean average precision across varying confidence thresholds.



**Figure 5.** The graphical representation of the values during the training phase of our model.

When examining the confusion matrix, we observed that during training, our model correctly identified 59 instances of invoice data and predicted 59 instances of personal data accurately, as shown in **Figure 6**. However, we also observed that the model made an error by predicting one instance of background as invoice data, i.e., misclassifying a region without invoice data.

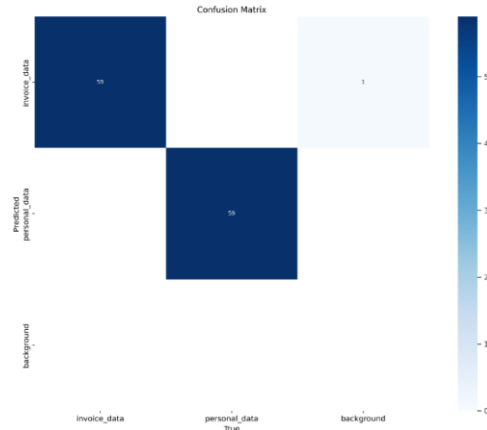


Figure 6. The Confusion Matrix.

As illustrated in **Figure 7**, when presented with an invoice image that our model had not encountered before, it accurately identified the 'personal\_data' region with a confidence rate of 95%. Additionally, it predicted the location of the date information with a confidence rate of 93%.

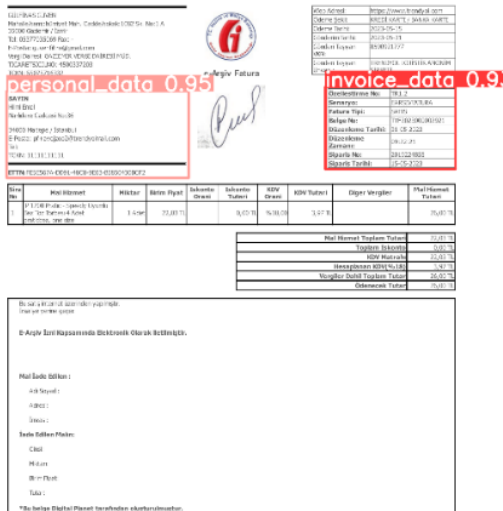


Figure 7. The predictions of our model on the test image.

### 3.2 OCR

To perform the extraction of dates from invoice images, we needed to convert the data from an invoice image into a string format that we could process. For this purpose, we utilized the Paddle OCR model. When we fed an invoice image into the OCR model, it provided us with the box coordinates of the area containing the text, the text within that area, and the predicted score for that text. The boxes indicating the text regions on the invoice, along with the texts and their corresponding scores, are detailed more comprehensively in **Figure 8**.

<b>Dzellestirme No:</b>	TR1.2	1: Dzellestirme No: 0.995	14: Siparis No: 0.999
<b>Senaryo:</b>	EARSIVFATURA	2: TR1.2 0.999	15: 2019224855 1.000
<b>Fatura Tipi:</b>	SATIS	3: Senaryo: 1.000	16: Siparis Tarihi: 1.000
<b>Belge No:</b>	TYF202300000292	4: EARSIVFATURA 0.998	17: 15-05-2023 0.999
<b>Düzenleme Tarihi:</b>	31-05-2023	5: Fatura Tipi: 1.000	
<b>Düzenleme Zamani:</b>	09:22:21	6: SATIS 1.000	
<b>Siparis No:</b>	2019224855	7: Belge No: 0.993	
<b>Siparis Tarihi:</b>	15-05-2023	8: TYF202300000292 1.000	
		9: Düzenleme Tarihi: 0.983	
		10: 31-05-2023 0.998	
		11: Düzenleme 0.999	
		12: 09:22:21 1.000	
		13: Zamani: 0.989	

Figure 8. Example of extracting text and its corresponding confidence score from an invoice image.

### 3.3 Regular Expression

At the regular expression stage, we meticulously crafted regular expressions by considering various possible date formats in invoices. In some invoices, the name or abbreviation of the month is written instead of the numeric representation. To address this, we compiled lists containing both Turkish and English month names, as well as lists encompassing the abbreviations of these months in both languages. After creating the list of Turkish and English months, we proceeded to formulate regular expressions for possible date formats. The regular expression rules we crafted are presented in **Table 2**.

**Table 2.** Regular expression rules and corresponding date formats.

RegEx	Date Format
<code>\d{4}\s*-\s*\d{2}\s*-\s*\d{2}</code>	YYYY-MM-DD   YYYY-DD-MM
<code>\d{2}\s*-\s*\d{2}\s*-\s*\d{4}</code>	MM-DD-YYYY   DD-MM-YYYY
<code>\d{2}\s*/\s*\d{2}\s*/\s*\d{4}</code>	MM/DD/YYYY   DD/MM/YYYY
<code>\d{2}\s*\.\s*\d{2}\s*\.\s*\d{4}</code>	MM.DD.YYYY   DD.MM.YYYY
<code>\d{2}\s*(tr_month_regex)\s*\d{4}</code>	01 Kas 2023   01 Kas 2023
<code>\d{2}\s*(en_month_regex)\s*\d{4}</code>	01 Jan 2023   01 January 2023

After implementing these regex rules, when we tested our system, it efficiently performed the extraction of date information from invoices. Without the assistance of our developed object detection model, applying OCR directly to the entire invoice image took up to 23 seconds to locate the date information. However, in our experiments with the model, the process of finding and processing the date information was optimized and reduced to as fast as 5 seconds. This significant improvement has contributed to making our targeted system more efficient.

### 4. Future Works

In our future endeavors, we will work towards making user information extractable in addition to invoice data. By incorporating different regular expression rules into our model, which can already identify areas containing personal information, we aim to perform extraction processes for user-specific details such as name, surname, address, phone number, and email address from invoices. Additionally, we plan to enhance our model by adding more diverse invoice examples to our dataset. This ongoing improvement process will increase the accuracy of identifying information fields when presented with various invoice samples. Furthermore, we intend to introduce a new label to our dataset to capture information about the products the user has purchased, including details such as product name, quantity, tax rate, and price. We will continue refining our model with these additional labels. Throughout these endeavors, our primary focus will be on maximizing the efficiency of our model in terms of processing time.

### 5. Conclusion

In conclusion, our research introduces a robust methodology for efficient and accurate date extraction from diverse invoices. By combining custom object detection, OCR, and refined regular expressions, we overcome challenges posed by varied invoice structures. The object detection model, trained with a diverse dataset, demonstrates optimal performance with an epoch value of 50, a learning rate of 0.001, and the Adam optimizer. The integration of Paddle OCR and carefully crafted regular expressions enhances the system's precision, achieving remarkable time efficiency. Our approach outperforms traditional methods, providing a valuable contribution to the field of document processing. The interdisciplinary nature of our methodology, combining computer vision, deep learning, and text processing, positions it as a noteworthy advancement in the quest for efficiency and accuracy in date extraction from invoices.

### References

- [1] Open Source Data Labeling | Label Studio. (n.d.). Label Studio. <https://labelstud.io/>
- [2] Ultralytics. (n.d.). GitHub - ultralytics/ultralytics: NEW - YOLOv8 in PyTorch > ONNX > OpenVINO > CoreML > TFLite. GitHub. <https://github.com/ultralytics/ultralytics>
- [3] PaddlePaddle. (n.d.). GitHub - PaddlePaddle/PaddleOCR: Awesome multilingual OCR toolkits based on

PaddlePaddle (practical ultra lightweight OCR system, support 80+ languages recognition, provide data annotation and synthesis tools, support training and deployment among server, mobile, embedded and IoT devices). GitHub. <https://github.com/PaddlePaddle/PaddleOCR>

- [4] M. S. Satav, T. Varade, D. Kothavale, S. Thombare and P. Lokhande, "Data Extraction From Invoices Using Computer Vision," 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), RUPNAGAR, India, 2020, pp. 316-320, doi: 10.1109/ICIIS51140.2020.9342722.
- [5] D. A. Kosiba and R. Kasturi, "Automatic invoice interpretation: invoice structure analysis," Proceedings of 13th International Conference on Pattern Recognition, Vienna, Austria, 1996, pp. 721-725 vol.3, doi: 10.1109/ICPR.1996.547263.
- [6] H. Sidhwa, S. Kulshrestha, S. Malhotra and S. Virmani, "Text Extraction from Bills and Invoices," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018, pp. 564-568, doi: 10.1109/ICACCCN.2018.8748309.
- [7] Tesseract-Ocr. (n.d.). GitHub - tesseract-ocr/tesseract: Tesseract Open Source OCR Engine (main repository). GitHub. <https://github.com/tesseract-ocr/tesseract>
- [8] R. Smith, "An Overview of the Tesseract OCR Engine," Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, 2007, pp. 629-633, doi: 10.1109/ICDAR.2007.4376991.
- [9] Ar, I., Karşlıgil, M.E. (2007). Text Area Detection in Digital Documents Images Using Textural Features. In: Kropatsch, W.G., Kampel, M., Hanbury, A. (eds) Computer Analysis of Images and Patterns. CAIP 2007. Lecture Notes in Computer Science, vol 4673. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-540-74272-2\\_69](https://doi.org/10.1007/978-3-540-74272-2_69)
- [10] Zhang, X., Duan, L., Ma, L., Wu, J. (2017). Text Extraction for Historical Tibetan Document Images Based on Connected Component Analysis and Corner Point Detection. In: Yang, J., et al. Computer Vision. CCCV 2017. Communications in Computer and Information Science, vol 772. Springer, Singapore. [https://doi.org/10.1007/978-981-10-7302-1\\_45](https://doi.org/10.1007/978-981-10-7302-1_45)
- [11] Dachengwang,. (2011). ANALYSIS OF FORM IMAGES. International Journal of Pattern Recognition and Artificial Intelligence. 08. 10.1142/S0218001494000528.

# Generative Adversarial Networks in Anomaly Detection and Malware Detection: A Comprehensive Survey

Bishal KC <sup>1,\*</sup> , Shushant Sapkota <sup>1</sup> , Ashish Adhikari <sup>1</sup> 

<sup>1</sup> Kathmandu University, Department of Computer Science & Engineering, Nepal

## Abstract

The swiftly changing panorama of machine learning has observed first-rate leaps within the field of Generative Adversarial Networks (GANs). In the beginning, the implantation of a deep neural network seemed quite difficult and poses challenges. However, with the rapid development of huge processing power, different machine learning models such as Convolutional Neural Networks, Recurrent Neural Networks, and GANs have emerged in the past few years. Following Ian Goodfellow's proposed GANs model in 2014, there has been a huge increase in the research focused on Generative Adversarial Networks. In the present context, not only GANs are used in feature extraction, but it proves itself worthy in the domain of anomaly and malware detection having firmly established in this field. Therefore, in our research paper, we conducted a comprehensive survey of prior and current research attempts in anomaly and malware detection using GANs. This research paper aims to provides detailed insights to the reader about what types of GANs are used for anomaly and malware detection with a general overview of the different types of GANs. These results are provided by analyzing both past and present GAN surveys performed, along with detailed information regarding the datasets used in these surveyed papers. Furthermore, this paper also explores the potential future use of GANs to overcome the advancing threats and malware.

**Keywords:** *Generative Adversarial Networks (GAN); Network Security; Deep Neural Network (DNN); Research Survey; Threat detection; Malware detection; Adversarial examples.*

## 1. Introduction

Malware also known as malicious software, is undesired programs designed to harm or exploit computer systems [1]. After the outbreak of COVID-19, the change in working environment from onsite office work to a work from home has rapidly increased cybercrimes. According to Statista [2], there were 5.4 billion malware attacks detected in 2021. The number of malware attacks reached 2.8 billion by mid-2022 over a short period of time. 560,000 new instances of malware are detected every day, contributing to the over 1 billion malware programs already been discovered [3]. From the insights of this data, we can infer that the number of cybercrimes continues to rise upward in 2024, raising concerns for the government and organizations. The cybercriminals are implementing Machine Learning models to automate and increase their capabilities which leads to more sophisticated and adaptive attacks which seems impossible to detect. With the current resources, it seems daunting to detect these types of attacks. As a result, Generative Adversarial Networks (GANs) emerge as a powerful tool to be used to see the unseen cybercriminals malicious behaviors. Generative Adversarial Networks, or GANs, serve as an architecture for training generative models, such as deep convolutional neural networks used for generating images [4]. These networks have emerged as a machine learning model proficient at creating new, previously unseen data samples realistically and synthesizing large datasets based on learned classes and features from an existing dataset [5]. GANs play a key role by assisting in the development of new datasets that replicates real-world cyber threats. These usage of GANs enables researchers and cybersecurity professionals to develop defense system to fight against a diversified possible attack, finally enhancing the security of networks against upcoming cyber threats.

This survey explores different research papers that provides insights regarding the effective detection of malware through the use of the Generative Adversarial Networks (GANs), highlighting the role of GANs in enhancing network security. Along with this, we focused for every individual and provide a clear understanding of the straightforward architecture and operational principles of GANs. Further, general overview of the various types of GANs are discussed, covering both widely accepted models and recent innovations proposed by researchers. In short, this paper will detail the various applications of GANs models in the malware research and network security, highlighting specific areas where GAN research significantly contributes.

## 2. Background

Reflecting on 2023 it stands out as the most successful year for cybercriminals. One of the prevalent threats

\*Corresponding author

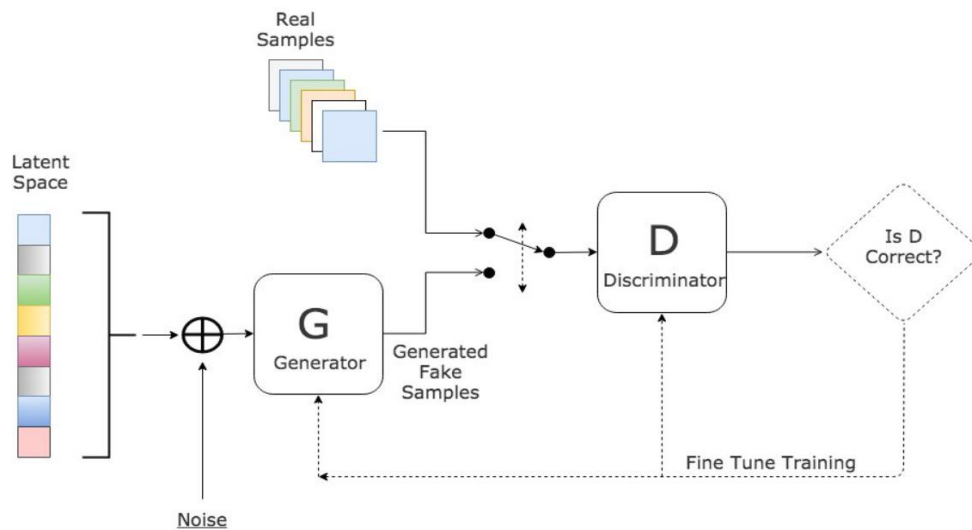
E-mail address: kcbisall@gmail.com

Received: 27/Feb/2024; Accepted: 30/Aug/2024.

to organizations globally targeted 4,368 victims, marking an increase of over 55.5% compared to the previous year. The second and third quarter alone accounted for more victims (2,903) than totaled for the entire year 2022 [6]. These attacks are becoming more and more popular today. Malware is a software program that conducts virtually any behavior malicious attacker wants to perform [1]. These goals include interrupting system operations gained access to the system and network resources and gathering personal information without user consent. In the realm of cybersecurity Generative Adversarial Networks can act as a double-edged sword. MalGAN is a notable example of GANs producing malware. MalGAN was used to generate PE malware effectively bypassing a static PE malware detection engine [7]. This approach represented a direct and forceful attack on the engine as the system was trained to observe the outputs of the model while being aware of the exploratory inputs sent to the engine. On the other hand, generative Adversarial Networks (GANs) serve as a potent and innovative tool we can harness to continually improve detection systems and prevent ransomware attacks. Section 2.1 provides an overview of the fundamental working principles of Generative Adversarial Networks (GANs). In Section 2.2, previous work in the realm of malware detection and network security using GANs is discussed. The techniques for measuring the performance of GANs are outlined in Section 2.3, while Section 2.4 details the datasets utilized in the surveyed research paper. Section 3 introduces various modes or types of GANs employed in the study. Section 4 explores the diverse areas of application for GANs, and Section 5 outlines potential future uses of GANs.

**2.1. Overview of GAN working**

The Generative Adversarial Networks (GANs) include two neural networks i.e., Generator and discriminator as shown in **Figure 1**, which are in competition with each other, forming a Zero-sum game where one agent's win is at the expense of the other's loss. This approach, a generative model originally presented for the domain of unsupervised learning, has turned out to be useful not only in that domain, but also in semi-supervised learning, fully supervised learning, and reinforcement learning [8-10]. GANs use a training set to diverge from the training set, and are an outstanding tool for classifying different learning paradigms.



**Figure 1.** Generative Adversarial Network, adopted from [11]

**2.1.1. Generator**

The generator generates synthetic data that closely mimics the actual data from the training set [12]. It takes noisy signal as a stock input, and encodes it to data that ideally is impossible to distinguish from genuine ones. This process is that of knowledge acquisition through extraction of the underlying patterns and structures from the training data. The generator's goal during the training phase is to make the discriminator think that the fake samples are real.

**2.1.2. Discriminator**

The discriminator is the binary classifier that classifies data as real and fake [13]. It has a data set, which contains both the original samples and fake ones. Discriminator tries to precisely classify the origin of the input data. With time, the discriminator learns to discern more efficiently between the real and the generated samples

**2.1.3. Feedback Loop**

The interactions between a generator and discriminator of Generative Adversarial Networks (GANs)

generate a vital feedback loop. This cyclic interaction fosters a growth mindset in both strands of training. Generator improves its skill to produce realistic data to mislead the discriminator while the discriminator is getting better at this task [14]. The Generator trains its weights in response to Discriminator signal, thereby improving the generator's performance. Through this process, the complexity of both networks gets perfected using back-propagation as the mechanism [5]. On the other hand, this kind of adjustment takes place in a black box environment where researchers only control inputs and results while the underlying operations are simulated and calculated based on assumptions. Nonetheless, Arjovsky et al., 2017's approach also studies the mathematical foundations of GANs, building a framework for a deeper understanding of how adversarial training works [15].

## 2.2. Previous Work

Within the enormous area of application, the utilization of Generative Adversarial Networks (GANs) occupies numerous areas which are capable of classification, generalization, and feature extraction. The GAN models' adaptability allows them to be used in almost any field, and therefore, listing all the applications is a challenging task. Many surveys made attempts to quantify the far-reaching activities of GAN models in their respective domains. In the study we give a brief summary of the different types of GAN models examined in the related works, offering a concise overview of their varied applications.

In Z. Cai et al., 2021 work [16], the authors focus on cybersecurity by investigating application of deep learning methods in different areas. Our paper has the same scope, on the topic of employing the Generative Adversarial Networks (GANs) in the domain of malware research, and consists of both the generation and detection. In the paper of Z. Cai et al., 2021 [16], the machine learning research in cybersecurity is acquainted with the readers first. Their functionalities are explored through the enumeration of different adversarial networks. The author dives into the GAN's privacy and security details, which are discussed through various GAN models that have been designed to protect personal information. Besides, the article does comparative assessment of GAN-Based Mechanisms for Data Privacy Protection. The authors fully delve into the issue of model privacy and compares GAN-Based schemes for model privacy protection. This article focuses on the capabilities of various GANs in cybersecurity domains, illustrating the research results on the model robustness, malware detection, fraud prevention, vehicle security, industrial protocols, and more. Similarly, our research also focuses on how GAN models can be useful in detecting and safeguarding against malware and malicious activities. In further research, this paper recommends advancing GAN-based attack methods by lowering convergence rates, and resolving the mode collapse issue. Corresponding objective functions are defined, and probably approximately correct learning may be used for optimization of data volume. The GAN-based adversarial sample detection should be employed in the practical applications. Android malware detection by statistics, dynamic analysis, and white-box attacks are shown as prospects of development. It is also important to continually improve bioinformatic identification and industrial protection with the help of GANs technologies.

The paper Navidan et al., 2021 [17] reviewed GANs application in the cybersecurity and networking sector. The survey delves into various types of GAN models, categorizing their use cases into five main domains: mobile networks, network analysis, the Internet of Things, the physical layer, and cybersecurity. Remarkably, the authors built an interesting collection of the network related papers written using GANs, and each one was categorized on its merit. Additionally, the paper provides a set of qualitative assessment criteria common for this area, and compares the results of various GANs models on datasets obtained from the reviewed papers. Our research contributes by providing readers with knowledge on variety of GANs and their applications in computer and communication network areas.

Similarly, Dong-Ok Won et al., 2022 [18], address the threat of detecting zero-day malware by developing a system that is able to learn and detect related situations that have been generated to mimic real cases. Proposed PlausMal-GAN model implements generative adversarial networks (GANs) to generate plausible malware images that are visually appealing and unique, making use of a existing malware data. The discriminator, as a detector, gets trained on both genuine and false characteristics. It has a very strong and stable prediction capacity for same group of zero-day-malware images which provides accurate prediction results on average across a different range of representative GANs models. The authors assess both the standard GAN strategy (min-max), least-squares strategy, heuristic approach and a mix of those, as well as for the DCGAN, LSGAN, WGAN and E-GAN models that are included in the proposed architecture. This demonstrates the seeming success of the framework in identifying and anticipating many new resembling zero-day malware instances especially during the testing and updating of malware detection systems.

The primary objective of our survey is to offer readers a comprehensive understanding of the utilization of Generative Adversarial Networks (GANs). In other words, this research paper served as a bridge for those individuals who are interested in GANs and its applications in cybersecurity, particularly in malware and threat-related studies. Our overview provides a detailed examination that distinguishes itself from [5] and [19]. While aligning with these works, our paper explores various approaches, providing a deep analysis of the

current use of GAN models in computer malware research and network security along with the future usage.

**Table 1.** *Different topics discussed in the surveyed research paper.*

Topics	Z. Cai et al., 2021	Navidan et al., 2021	Dong-Ok Won et al., 2022
Malware Detection	✓	✓	✓
Fraud Detection	✓	✓	
Android Malware		✓	✓
Bioinformatic-Based Recognition	✓		
Android Security		✓	✓
Industry Protocol	✓		
Zero-Day Malware Detection			✓
Black-box API attacks		✓	
Adversarial Examples	✓	✓	✓
Malware Classification			✓
Model Privacy	✓		
Password attack		✓	
Vehicle Security	✓		
Botnet Detection	✓		
Network Intrusion Detection	✓	✓	
Data Privacy	✓		

**Table 2.** *Different GANS Models discussed in surveyed research paper.*

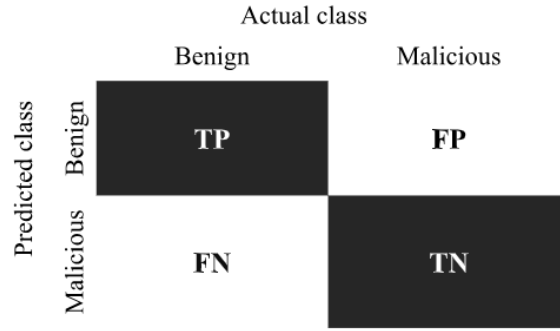
ANS Model	Z. Cai et al., 2021	Navidan et al., 2021	Dong-Ok Won et al., 2022
inilla GAN	✓	✓	✓
GAN	✓	✓	
GAN	✓		✓
GAN			✓
GAN	✓	✓	✓
GAN-GP		✓	
GAN			✓
GAN		✓	✓
GAN	✓	✓	
GAN	✓		
GAN	✓		
GAN	✓	✓	✓
GAN	✓		
GAN	✓		
GAN	✓		
GAN	✓		
GAN	✓		

We have cited three major papers that have already worked on Anomaly and Malware Detection. Table 1 presents the findings on various aspects of those surveyed papers that belong to the Anomaly and Malware Detection categories. This table provides a cross tabulation of the areas of concern. Likewise, Table 2 shows the different types of GANs described in those research papers and their uses in Anomaly and Malware Detection. This analysis in that paper demonstrates the different GAN models employed in the field and shows how each type handles the problems associated with the identification of anomalies and malware type. In this way, the presented insights can shape further research in this rapidly evolving field and offer a systematic overview of the current state of knowledge, as found in the existing literature.

**2.3 Measuring Performance**

The evaluation of the machine learning models is done through the application of a range of conventional metrics. Those statistics are universally acknowledged and are popularly used for evaluation purpose. Generally speaking, the metrics involved would be the Confusion Matrix, Classification Accuracy, Precision, Recall, F1 Score, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) as shown in **Figure 2**.





**Figure 2.** Performance Metrics for Machine learning Classifications, adopted from [20]

True Positive, True Positive refers to the number of benign samples being correctly identified as benign samples. True Negative, True Negative refers to the sum of malicious samples being correctly detected as malicious samples. False Positive, False Positive is calculated based on the number of malicious samples being incorrectly identified as benign samples. False Negative, False Negative refers to the number of benign samples that are incorrectly identified as malicious samples.

The classification accuracy of a model is simply measured by comparing the test samples that are correctly identified with total number of test samples. The accuracy of the model is given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision is calculated by comparing the number of correctly identified benign samples to the total predicted as benign. It measures the model's accuracy in classifying instances as benign and emphasizes its ability to minimize misclassifications.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

The Recall (True Positive Rate or Sensitivity), represents the ratio between the accurate detection of benign samples and the total number of actual benign samples.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1 score is the harmonic mean of precision and recall, it serves as a composite metric that captures the fundamental balance between precision and recall. The unified evaluation of the model's overall performance is provided by it which is given by,

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

The Inception Score is a metric for human evaluation of the quality of image generative models that was developed by Salimans et al. and was published in 2016 [21]. This measure shows a particular good correlation with human evaluations of the generated images in the context of CIFAR-10 dataset. The Inception Score is a measure of a pre-trained Inception v3 Network on the ImageNet dataset, based on the network's outputs applied to generated images [22].

$$IS(G) = \exp(\mathbb{E}_{x \sim p_g} D_{KL}(p(y|x) || p(y))) \tag{5}$$

where  $x \sim p_g$  refers that  $x$  is an image sampled from  $p_g$ ,  $D_{KL}(p || q)$  is the KL-divergence between the distributions  $p$  and  $q$ ,  $p(y|x)$  is the conditional class distribution, and  $p(y) = \int_x p(y|x)p_g(x)$  is the marginal class distribution.

The Mode Score is the modified Inception Score [23] to overcome the sampling assessing issues by GAN. The Mode Score differs from the Inception Score in that it disregards the original probabilities. Since the generator results in few samples while the discriminator has more dominance, this score modifies the originating score [24]. Moreover, the Mode Score is an automated alternative to human annotators for quality evaluation of the DNA samples. Further details about Mode Score, including computational process, are

provided in the original article [25].

The Fréchet Inception Distance (FID) [26] which is arguably the most widely used metric for testing the feature similarity between real and fake images has gained much attention. The assessment of GAN resembles the most with the Fréchet Inception Distance (FID) that is analyzed using the Inception V3 model, which is pre-trained on the ImageNet dataset. This FID score was named due to the activation function obtained through the Inception V3 model, providing a tool for classifying how dissimilar those distributions of authentic and fake images are.

Fréchet Inception Distance (FID) technique for determining a "multivariate" normal distribution is as follows:

$$FID = || \mu_x - \mu_y ||^2 - Tr( \sum_x + \sum_y - 2\sqrt{\sum_x \sum_y} ) \tag{6}$$

where  $X$  and  $Y$  are the real and fake embeddings (activation from the Inception model) assumed to be two multivariate normal distributions.  $\mu_x$  and  $\mu_y$  are the magnitudes of the vector  $X$  and  $Y$ .  $Tr$  is the trace of the matrix and  $\sum_x$  and  $\sum_y$  are the covariance matrix of the vectors.

**2.4. Datasets**

To uncover various aspects of research, our surveyed paper used some in-depth datasets. CSI data set for wireless sensing and the Microsoft Malware Classification Challenge Dataset were the most important datasets which shaped the results. Here is a summary of the datasets employed in the survey article that we selected.

**2.4.1 WIFI RSSI**

Wi-Fi RSSI dataset [27] is designed for indoor user detection by signal strength strengthening its use by smart homes, security (finding criminals), and access point user counting. The dataset is used as a tool for building an optimized model for monitoring devices, which can be used to identify user location via Wi-Fi signal strength measurement. Signal strength data from different routers is used, and are classified as a problem of map making. The training of neural networks employs a fuzzy hybrid of Particle Swarm Optimization & Gravitational Search Algorithm (FPSOGSA) to achieve a higher accuracy. The dataset is aimed at applications that are in-house with an emphasis on the most efficient indoor user detection using Wi-Fi RSSI.

**2.4.2 CSI**

The CSI dataset [28] is an important layer for wireless sensing applications, considering the cases of activity recognition, people identification, and people counting with the support of Wi-Fi connection device. The gathered dataset, using a monitor router with Nexmon CSI, encompasses seven activities, ten clients, and more than 13.5 hours of channel readings. It establishes 242 Wi-Fi OFDM data sub-channels within this 80 MHz band to ensure a better unified foundation for the development of Wi-Fi-based wireless sensing solutions. The related work [29] highlights the possibilities of throughputs from commercial WiFi systems by utilization of Channel State Information (CSI). It concentrates on the active points CSI variations produced by body movement for activity recognition. The approach is represented by the process of feature extraction from CSI data streams and using machine learning techniques, which leads to the development of behavior recognition models.

**2.4.3 Network traffic (KDD99)**

The dataset known as the KDDCUP'99 dataset, or Network Traffic KDD99 dataset [30], features is especially important because it was created for the purpose of assessing the modern intrusion detection systems. It is a multifaceted exploit that consists of different attacks such as the connection details, traffic stats, system calls, and the user's authentication data. The data set permits the realization of multiclass classification, associating connections with safe and different attack types. Due to its numerous records, it can be used to develop models that would be helpful in security network. This is the reason why it is regarded as a critical data resource.

**2.4.4 Deepsig RadioML 2016.10A**

The "RADIOML 2016.10A" [31] historical dataset is a synthetic one which was created in GNU Radio programming. This library from Virtual Studio Technology introduced in 2016 contains 11 different types of modulation, 8 of which are digital while the remaining 3 are analog with different signal to noise ratios. These data were first reported at the 6th International Conference on GNU Radio. The 2016.05A data version is an updated and standardized version compared to the 2016.04C dataset, which is no longer available. The dataset may be designed for studies and experimentation in the domain of RF signaling processing and modulation characteristics.

### 2.4.5 Microsoft Malware Classification Challenge Dataset

The data of the Microsoft Malware Classification [32] has a large repository; its uncompressed size is almost 500 GB. The dataset is composed of nine well-known malware families that are covered by the entire dataset. This combination represents. For each malware file, the 20-bit hash is generated and then the integer is in turn converted into one of the family names. The raw data for every file is shown in a hexadecimal equivalent that represents the binary contents stripping off the headers such that only the analysis of the files is carried out without being biased. Concurrently with the raw data is given a text file, which has been enriched by different data like task names and strings which were obtained with the help of the IDA disassembler tool. The main issue of the participants was the classification of malware into the nine classes that had been already assigned.

### 2.4.6 Maling Dataset

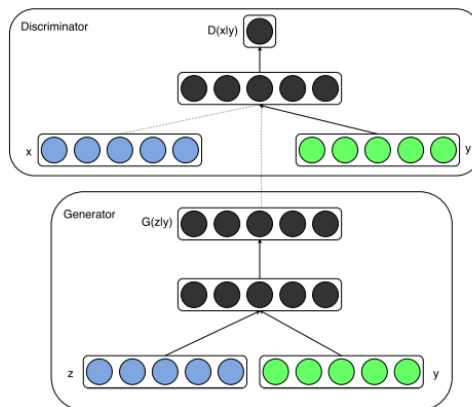
Maling Dataset [35] includes 9,339 byteplot images of malware from 25 families. It is applied as a creative approach for malware visualization and classification based on the image processing [33]. Malware binaries are converted into gray images by utilizing visual resemblance across families.

## 3. Variant Models of GAN

The reviewed papers provide a quantitative study of different GANs and come with a more detailed analysis of their utilization. This part unravels the essence of GANs, where variety is emphasized and specific features are highlighted.

### 3.1. CGAN

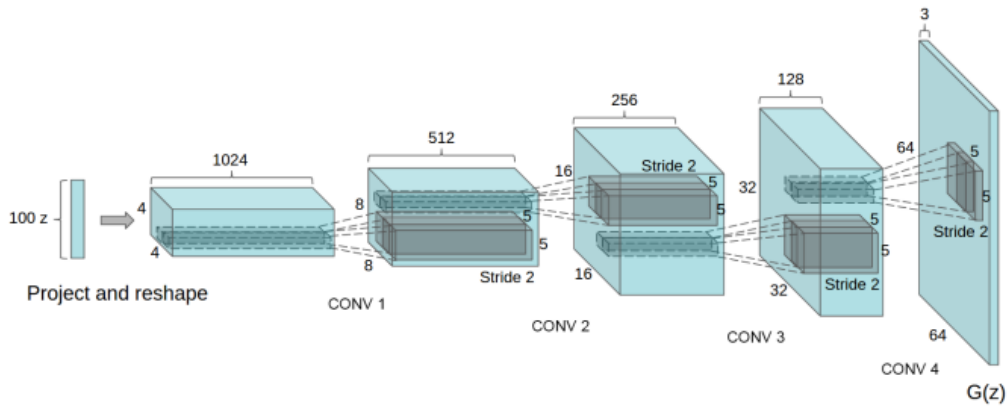
The traditional GAN [14] do not have a control over the output. The drawback of the traditional Generative Adversarial Network (GAN) is resolved in the Conditioned Generative Adversarial Network (CGAN) [34], in which the model is built to a conditional form. By using the additional information "y" as we can see in figure 3, both the generator and discriminator are connected to more information which is also labeled or comes from other modalities. This extra data is added to both the discriminator and generator as a new input layer. The whole conditioning system gives the model a capability to produce outputs with the specific features or characteristics on the basis of the provided auxiliary information [34].



**Figure 3.** The Architecture of Conditional GAN [34]

### 3.2 DCGAN

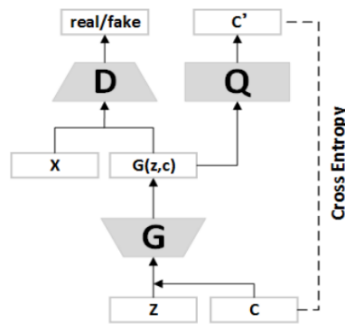
DCGANs, which was proposed by Radford [13] and currently is the most popular and best performing GAN architecture that uses convolutional layers, batch normalization, Rectified Linear Unit (ReLU) activations and strided convolutions as represented in figure 4. Attentive modifications include employing convolutional layers instead of fully connected layers and comprising particular architectural rules for both generative and discriminative parts. The intermediate layers in the generator more often use the ReLU activations, while the output layer uses tanh or sigmoid to make sure the pixel values are within the norm for image data.



**Figure 4.** The Architecture of DCGANs, adopted from [13]

**3.3 InfoGAN**

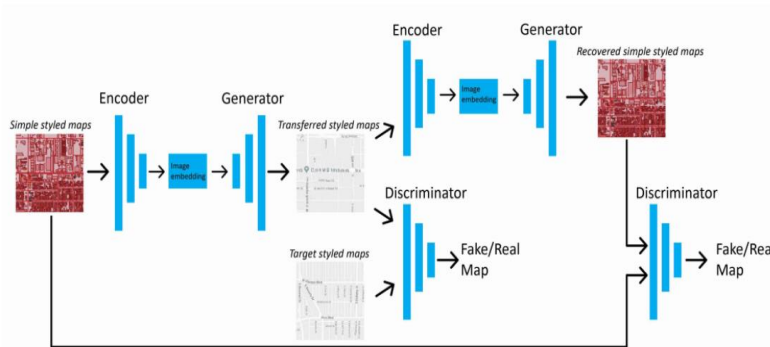
InfoGAN [35], which is an unsupervised learning model, the objective is modified to maximize the mutual information between a subset of noise variables and observed data. This is done by giving the model extra data that contains aspects of the desired features and responding random noise which results to an artificial image creation [36]. In the figure 5, we can see that  $G(z,c)$  is generator networks function where  $z$  represents a random noise and  $c$  represents additional conditional information. This change indeed leads to the revelation of interpretable representations, the performance is profitably manifested on multiple image datasets as well, including MNIST, CelebA, and SVHN. Such method implies that an information cost can be very effective at teaching the generative model to represent patterns that naturally emerge in data, thereby encouraging meaningful and disentangled representations.



**Figure 5.** The Architecture of InfoGAN, adopted from [35]

**3.4 CycleGAN**

CycleGAN [38] is an images-to-images translation model that overcomes the challenge of unpaired training data. As we can see in the architecture of CycleGAN in figure 6, it extends Pix2Pix [39] by introducing a cycle consistency loss that helps the model to learn mappings between input and output domains without the need of one-to-one correspondence. Doing so supports numerous applications, for instance, changing SAR imagery into RGB or the other way around (source to target) by means of one model. The structure involves two generators and two discriminators that are trained at the same time, providing a variety of handling image translations with unpaired datasets.



**Figure 6.** The Architecture of CycleGAN, adopted from [37]

### 3.5 ACGAN

The Auxiliary Classifier Generative Adversarial Network (AC-GAN) is one of the latest models that introduces the specific cost functions to the latent space structure of a traditional GAN in order to increase its capabilities for image resolution tasks [40]. Odena et al., 2016 [40] aimed at classifying the structure of the natural images that would include the down-sampling scheme to extract the basic features. According to Navidan et al., 2021 [17, 41], one of the main differences between ACGAN and CGAN is about setting class labels. While CGAN train generator to be classifier itself, ACGAN employ an additional decoder network to predict the labels. The model's efficacy however did not end there, as it also created malicious code for system interference, as highlighted in [41].

### 3.6 BiGAN

A crucial step forward in the development of GANs was the Bi-directional Generative Adversarial Network (BiGAN), which Donahue et al. [42] presented in 2017. The BiGAN structure incorporates three main components: a generator (G), a discriminator (D), and an extra encoder (E). They operate in a triangular relationship between each other. The generator does make the fake data, while the discriminator is responsible for the scrutinization of the real and the generated data. The last feature to be discussed is the encoder. It contributes to a bidirectional mapping between the data space and the latent space. This encoder performs not only real data to latent space mapping, but also synthetic data back to the same latent space fitting. Moreover, BiGAN [43] version has been studied further in the field of network intrusion detection. The objective here is to cut down on costs arising from extensive training while at the same time ensuring that the network data is sufficiently monitored by intrusion detectors.

### 3.7 TGAN

The model Temporal Generative Adversarial Nets (TGAN) [44] transforms the concept of video generation by introducing a dual-generator structure. Different with traditional ways, TGAN employs temporal generator which not only generates one latent variable at time but also changes it for every subsequent video frame. The purpose is to describe how the generator of the image finally synthesizes a complete video from these latent variables. The main goal of the inventive method is to make the grasping of momentary connections more accurate. The TGAN designers used the Wasserstein GAN model and introduced a new end-to-end training approach to ensure robustness during training. The time- and image-generating units show that the TGAN reaches a new level in semantic representations of unlabeled video content and generates lots of diversified and realistic video content. Moreover, Munoz et al., 2020 [45] presented a video generation model in this research which for the first time incorporated both natural spatiotemporal modeling and avoidance of computationally complex 3D architectures.

### 3.8 LSGAN

Classical GANs apply the sigmoid cross-entropy loss for training the discriminator, whereby it's possible to obtain vanishing gradients. To deal with this issue, LSGANs (the Least Squares Generative Adversarial Networks) is employed in the article [46] and it is applied least squares loss function for the discriminator. The adoption of LSGANs presents two notable advantages over regular GANs: the formation of better-quality images and higher stability in the algorithms learning process. This points out the effectiveness of LSGANs in unsupervised learning and expands their application conditions.

### 3.9 WGAN

The Wasserstein GAN (WGAN), whose conference paper was published in 2017 [47], introduces a unique approach to solve the problem encountered in the Generative Adversarial Networks (GANs). Unlike the common type of GANs based on the sigmoid cross-entropy loss function, which can be associated with vanishing gradients, WGAN instead relies on the Wasserstein distance, also known as Earth Mover (EM) distance. The main goal of the proposed WGAN architecture is to ensure stable training process, avoid mode collapses, give helpful learning curves for model optimizing (e.g. debugging and hyperparameter tuning). The distance and divergences between our two separate distributions:  $\mathbb{P}_r, \mathbb{P}_g \in \text{Prog}(X)$ , such that  $\text{Prob}(X)$  is the "space of probability measures defined on  $X$ " [47]. We are computing the distance between the two distributions by the Total Variation (TV) distance which is defined as,

$$\delta(\mathbb{P}_r, \mathbb{P}_g) = \sup_{A \in \Sigma} |\mathbb{P}_r(A) - \mathbb{P}_g(A)| \tag{7}$$

The Kullback-Leibler (KL) and the Jensen-Shannon (JS) divergence are given as,

$$\text{KL}(\mathbb{P}_r || \mathbb{P}_g) = \int \log\left(\frac{\mathbb{P}_r(x)}{\mathbb{P}_g(x)}\right) \mathbb{P}_r(x) d\mu(x) \tag{8}$$

where both  $\mathbb{P}_r$  and  $\mathbb{P}_g$  are assumed to be absolutely continuous, and therefore admit densities, with respect to the same measure  $\mu$  defined on  $X^2$ . The KL divergence is famously asymmetric and possibly infinite when there are points such that  $\mathbb{P}_g(x) = 0$  and  $\mathbb{P}_r(x) > 0$ .

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r || \mathbb{P}_m) + KL(\mathbb{P}_g || \mathbb{P}_m) \tag{9}$$

Where,  $\mathbb{P}_m$  is the mixture  $(\mathbb{P}_r + \mathbb{P}_g)/2$ . This divergence is symmetrical and always defined because we can choose  $\mu = \mathbb{P}_m$ . The core of WGAN (Wasserstein Generative Adversarial Network) contains EM-distance – a distance generated by the Earth-Mover equation. In contrast to binary evaluations, the EM-distance determines the gradual distance between the produced results and the sought goals. This brings stability and performance to the WGANs.

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y) \sim \gamma} [|x - y|] \tag{10}$$

where  $\Pi(\mathbb{P}_r, \mathbb{P}_g)$  denotes the set of all joint distributions  $\gamma(x, y)$  whose marginals are respectively  $\mathbb{P}_r$  and  $\mathbb{P}_g$ . Intuitively,  $\gamma(x, y)$  indicates how much “mass” must be transported from  $x$  to  $y$  in order to transform the distributions  $\mathbb{P}_r$  into the distribution  $\mathbb{P}_g$ . The EM distance is the “cost” of the optimal transport plan.

### 3.10 WGAN-GP

The Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) is an extension of the Wasserstein GAN (WGAN) that addresses potential issues related to weight clipping in the original WGAN. Proposed by Gulrajani et al. in 2017 [48], WGAN-GP introduces a gradient penalty term to the loss function as a regularization technique, eliminating the need for weight clipping. This modification improves the stability of the training process and mitigates potential mode collapse issues observed in standard GANs.

The WGAN-GP loss function includes the Wasserstein distance term and the gradient penalty term:

$$L = \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)] + \lambda \mathbb{E}_{\tilde{x} \sim \mathbb{P}_{\tilde{x}}} [(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2] \tag{11}$$

Where,  $\mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(\tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(x)]$  is the original critic loss,  $\lambda$  is the gradient penalty coefficient and  $\mathbb{E}_{\tilde{x} \sim \mathbb{P}_{\tilde{x}}} [(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2]$  is the gradient penalty.

### 3.11 E-GAN

Evolutionary Generative Adversarial Networks (EGAN) [49] departs from standard GANs in its incorporation of an evolutionary algorithm. In EGAN, a grouping of producers {G} continues to evolve in the discriminator-defined ecosystem. For every generator in the population, a spot in the solution space of the generative network will be assigned. The evolutionary process involves three stages at each step: Another method, called variation, creates various children from the parents after they randomly mutate; while evaluating them using a fitness function that takes the discriminator as an argument and selecting those who are better. Discriminative network (D) changes continuously, as it is responsible to tell apart the real and generated samples, and it provides an adaptive loss for driving the evolution of the generator population.

$$LD = -\mathbb{E}_{x \sim p_{data}} [\log D(x)] - \mathbb{E}_{y \sim p_g} [\log(1 - D(y))] \tag{12}$$

The fitness function gives feedback about the quality of samples that are generated in the current discriminator environment and this guides the evolution of the population towards having leading solutions.

### 3.12 BEGAN

The BEGAN model introduced by Berthelot et al. in 2017 [50], is generated by the aim of achieving equilibrium between generator and discriminator dynamics. This model applies a autoencoder-based architecture, including a notion of diversity in the loss function. The equilibrium is kept by ensuring the diversity ratio is maintain, the model can achieve stable and high-quality image. The BEGAN Loss function is given as,

$$LD = L(x) - k_t \cdot L(G(z_D)) \quad \text{for } \theta_D \tag{13}$$

$$\begin{aligned}
 LG &= L(G(z_G)) && \text{for } \theta_G \\
 k_{t+1} &= k_t + \lambda_k (\gamma L(x) - L(G(z_G))) && \text{for each training step } t
 \end{aligned}$$

### 3.13 ProGAN

The ProGAN is a breakthrough in the field of generative adversarial network architectures, achieved by Karras et al. [51] in 2017. Such technique is created by rising the resolution step by step in training corresponding to building more layers of both generator and discriminator networks. The sequential development makes the model to first of all handle major structures in the images distribution and then goes to details. Simultaneous growth of generator and discriminator networks, with carefully tuned addition of new layers, makes the process stable and robust. The ProGAN distinguishes from other models as it is able to generate high-quality images of high-resolution by using modern loss functions like WGAN-GP and LSGAN. This method agrees with the concept of getting easier questions at first while in training, what maintains the model's success.

### 3.14 MsgGAN

MsgGAN, the Multi-Scale Gradient Adversarial Generative Network, offers a solution providing these GANs with stability to quickly adapt to any new data sets. Per Karnewar & Wang, 2020 [52], MSGGAN resolves the problem of noisy gradients in 2018 by allowing for a multi-scale transfer of gradients from the discriminator to the generator. In this way, it creates an even higher resolution image synthesis that can be used as an effective substitute to standard progressive growing techniques. The MSGGAN protocol performance is stable under different datasets, resolutions, domains, loss functions, and network architecture - exceeding or matching the best GANs in the majority of cases.

## 4. Areas of Use

The emergence of Generative Adversarial Networks (GANs) has demonstrated that these tools are entirely interdisciplinary with applications in a variety of fields, including cybersecurity. These powerful models are not limited to a singular task; they find a multitude of applications through cybersecurity categories. Here GANs brought in to improve different areas of vulnerability in the system and provide new solutions and insights in this field. Here we demonstrate different areas where GANs have a potential and, by that, explain why these algorithms have such a wide use and why they are so important to the progress of the whole field.

### 4.1. Anomaly Detection

Anomaly detection based on GANs has been gaining in popularity because attacks are growing ever more complicated and new detection methods are needed. Chandola et al., 2009 [53] systematically covered the field of anomaly detection through an in-depth analysis of challenges, anomaly types, and various detection methods. Specifically, Schlegl et al., 2019 [54] proposed AnoGAN, which is based on the deep convolutional generative adversarial network (DCGAN) [13] and trains a generator and discriminator on the normal data through unsupervised learning. Efficient GAN-Based Anomaly Detection (EGBAD) was proposed by Zenati et al., 2018 [55] which makes use of the BiGAN [42] architecture allowing learning of an encoder mapping input samples to the latent representations during the adversarial training. Akcay et al., 2018 [56] demonstrated GANomaly approach, and Luo et al., 2022 [57] worked on convergence issues using exponential weighted moving average methods and Convolutional Autoencoders. Furthermore, Xia et al., 2022 [58] have summarized the issues, predicted the trends and given the research directions for the future. In the addressing of industrial limitations, Goetz & Humm, 2023 [59] presented an unsupervised, decentralized, real-time anomaly detection concept for cyber-physical production systems using multiple 1D convolutional Autoencoders. Similarly, Lim et al., 2024 [60] carried out a systematic review on GANs and their effectiveness, with suggestions for future research directions. The repeated enhancement of anomaly detection methods that overcomes the limitations while adding more capabilities is the key feature of GANs in anomaly detection.

### 4.2 Cyber Intrusion and Malware Detection

The application of the Generative Adversarial Networks (GANs) in cybersecurity also extends to different applications such as defending against malicious traffic and improving the capability of Intrusion Detection Systems (IDSs). Reflecting the vulnerability of IDS to adversarial examples [61], Usama et al., 2019 suggested a Generative Adversarial Network-based attack that used the traffic features, making the IDS unable to detect malicious traffic. To this end, a GAN-based defense was applied to enhance the resilience of IDSs against security breaches. In the last method, an innovative method called IDSGAN is proposed, based on WGAN [62], which generates adversarial attacks with the aim of bypassing the detecting system. It does that based on the generated generator, discriminator, and black-box IDSs following an unknown architecture of IDSs in real

environments. This shows the GANs capability in fitting any environment like in-vehicle networks, where the need for high intrusion detection accuracy is a priority for driver's and passengers' safety. Seo et al., 2018 [63] suggested a GIDS (GAN-based IDS) for detecting unknown attacks, and this system was shown to be an excellent real-time intrusion detector with superior performance reported.

Salem et al., 2018 [64] employed GANs as a weapon to solve the skew problem in HIDSs by converting normal data to anomalies that were so successfully dealt with, the HIDSs' performance improved significantly. Likewise, GAN for Android malware identification by Shahpasand et al., 2019 [65] is used for black-box attacks which the adversaries know nothing about the inside of the network. In this case, GANs are the driving force of such attack campaigns in that they provide the attackers with adversarial variations that enable them to manipulate the malware code and consequently avoid detection. In addition, GANs have been instrumental in the study of malware across varied operating systems including Linux and Windows among others. Kargaard et al, 2017 [66] applied GANs in malware detection by transforming malware binaries into image for GAN training. Kim et al. [67, 68] introduced an approach called the transferred deep convolutional generative adversarial network (tDCGAN) for zero-day malware detection. This example of GANs demonstrates the adaptability of GANs to deal with different types of malwares on different operating systems. These applications show a wide range of GANs applications in cybersecurity: from traffic perturbation and intrusion detection up to malware or pattern-lock system security.

### 4.3. Security Attacks

Generative Adversarial Networks (GANs) are widely implemented in the security domains for the purpose of stimulating and manipulating data. One of the key roles of GANs in cybersecurity is the generation of real but synthetic samples which help in the study of vulnerabilities, penetration testing and the development of defense mechanisms. Imitating the patterns and behaviors, GANs have an ability to strengthen the systems against the possible APTs (Advanced Persistent Threats). The studying by Chowdhary et al., 2023 [69] presents a novel autonomous web application security testing framework based on the Generative Adversarial Networks (GANs) in order to detect vulnerabilities, even the corresponding emerging threats, such as Cross-Site Scripting (XSS) [70] and SQL Injection [71]. The framework uses semantic tokenization for extracting key features for XSS attacks and conditioned sequence GAN for producing attack payloads. Moreover, vulnerable passwords such as those used by Hashcat [72] and JtR [73] can be cracked rapidly, and therefore, represent a great threat to IoT networks. Recent efforts to better the password-cracking efficiency have considered techniques like the Markov model [74] and probabilistic context-free grammar (PCFG) [75]. More significantly, PassGAN has introduced the incorporation of Generative Adversarial Networks (GANs) as a novel approach. The Nam et al., 2020 [76] propose a more advanced model which is constructed on top of the improvements done in PassGAN and PCFG [77]. The Cyber-Physical Production System (CPPS) [77] is vulnerable to cross-domain attacks from the complex interactions between the cyber and physical environments. To deal with this weakness, Chhetri et al., 2019 [78] conceived GAN-Sec that is based on CGAN (conditional generative adversarial network). Through the implementation of GAN-Sec, it is possible to determine whether the key security parameters such as confidentiality, availability and integrity, are appropriately adhered to. Rigaki et al., 2018 [79] propose using GANs that generate network traffic and consequently allow malware to mimic other network patterns by GAN-obtained parameters inclusion in the source code of malware. In conclusion, GANs play an important role in cybersecurity by imitating data patterns, conducting penetration testing, scanning for vulnerabilities, and exposing the security weaknesses in password. This adds to the effectiveness of cybersecurity.

## 5. Challenges and Future Improvements

Generative Adversarial Network (GAN) has an unparalleled impact on the aspects of contemporary human existence today. In this section, we are discussing about challenges and the improvements of GANs that we can do in the future:

### 5.1. Transfer learning and generalization capabilities

Generative models, although effective, can encounter difficulty in adapting acquired information to new and diverse conditions which restrict their practical marketability in real-life situations [80]. Scientists are on the lookout for various approaches like transfer learning [81], meta-learning [82], and domain adaptation that could be used to make the generalization capabilities of generative models better. This is very critical for diffusion to different sectors. Contrary to GAN-based anomaly detection models, the main disadvantage is that they face difficulties to adapt to diversified network structures, consequently, leading to their subpar performance across different datasets [81]. The main problem is generalizing the acquired knowledge outside the particular area the system was trained in. For this issue the answer is to let transfer learning skills developed, which should allow the model to acquire knowledge from domain one and apply it successfully in various settings without considerable retraining. In the next experiment, we will have to improve the transfer learning



methods mainly by preventing the negative transfer and exploring unsupervised methods. This would increase the model's robustness, freeing it from dependence on labelled data in such a case. However, the anomaly detection in GAN-assisted anomaly setting remains difficult due to inefficient GAN architectures, vulnerability to adversarial attacks and lower interpretation capability. Research and development work in this field may demonstrate more effective and productive techniques of network anomaly detection.

## 5.2. Hyperparameter optimization

A little acknowledged subject in discrepancy detection within GANs is the role of hyperparameter choice on the model performance, reported by Soenen et al., 2021 [83]. Hyperparameter optimization involves fine tuning the different areas of the GAN and tries to tune down learning rates and network architectures, activation functions, and loss functions to come up with an optimal balance for improvement of anomaly detection effectiveness. Future work on improving the GAN-based methods could focus on the new model's architectures and the training strategies, as proposed in the work by Xia et al., 2022 [84]. Moreover, the progression of GAN improvement could be achieved by implementing visualization techniques for attention maps obtained by residual channel attention module [85]. Efforts to strengthen GANs may include a review of aids and techniques and devising methods of correcting the missing values [86]. These fields of study are having a positive impact on the GANs performance in the area of anomaly detection and the cybersecurity sector is taking special interest in understanding the issues of model performance, comprehensibility, and reliability.

## 5.3. Legal and Regulatory Challenges: Navigating the Legal Landscape

Developing AI techniques at a remarkable speed brings some cyber security issues, especially the intellectual property and liability problems associated with anomaly detection. The generation of content that might be similar to copyrighted material poses issues of ownership and attribution which become the domain of the legal framework in relation to generated content. These consequences are especially important for the cybersecurity community which will face such problems as in the cases of misinformation or malicious use where the role of liability is still vague. Resolving these legal and regulatory issues calls for combined participation of policy makers, legal experts, AI community to suggest rules for ownership and liability. In network security and anomaly detection, the field of integrative generative AI holds huge potential for transformation. Nevertheless, legal and ethical issues should be taken into account as well as the solution of the technical difficulties so that the deployment will be performed in a safe way and to promote innovative solutions in the fight against cyber threats. Through creating a cooperation between academia, industry, and regulatory institutions, an interdisciplinary approach could be achieved by incorporating technological developments, ethical rules, and legal regulations to improve generative AI technologies in the field of cybersecurity.

## 6. Conclusion

In conclusion, this research work aims to present an extended review of Generative Adversarial Networks (GANs) and their usefulness in malware analysis and computer security. In light of these questions, the present paper provides a good starting point for understanding the potential of GANs in anomaly and malware detection by analysing the current state of GAN fundamentals, finding the deficits in the existing research, and discussing the evaluation metrics. The discussion sheds adequate light to several GAN models – Deep Convolutional GANs (DCGANs), Conditional GANs (CGANs), and Wasserstein GANs (WGANs) while enumerating their distinguishing features and areas of application. GANs hold great potential to improve the working of IDSs, malware identification, and anomaly detection along with producing realistic datasets and emulating attacks. Contrary to the traditional systems, the GANs play a dual role both as a shield against cyber-attacks and as enabler of an attack.

To guide future research, the following recommendations are made: The future work should focus on improvements in various training methods and better tuning of the model for enhancing the developments of GAN. Researchers need to concentrate on collecting better quality sets of data and focusing on issues like stability in training process and adversarial robustness. Further, the proposed idea of incorporating transfer learning and generalization of GANs in various tasks and datasets will enhance the performance of GANs even further. Hyperparameter tuning is also required to enhance the model's accuracy, and, therefore, the ability to generate more accurate outcomes. Further research works have more useful applications of GANs in advanced neural cryptography and better attack representations that are beyond the current cybersecurity measures. Thus, establishing comprehensive knowledge about GANs and their multifaceted usages, researchers can make a huge leap in furthering the developments in network security. This detailed examination also seeks to encourage and guide more research initiatives on the constantly growing cybersecurity domain, specifically in the field of anomaly and malware detection, thus assuring GANs' steady and significant contribution to combating current and future cyber threats.

**Declaration of interest:** The authors declare that there is no conflict of interest.

**Appendix A: List of Abbreviations**

Abbreviation	Description
ACGAN	Auxiliary Classifier Generative Adversarial Network
AI	Artificial Intelligence
BEGAN	Boundary Equilibrium Generative Adversarial Network
BiGAN	Bi-directional Generative Adversarial Network
CGAN	Conditioned Generative Adversarial Network
CIFAR-10	Canadian Institute for Advanced Research - 10
CNN	Convolution Neural Network
CPPS	Cyber-Physical Production System
CSI	Channel State Information
CycleGAN	Cycle-Consistent Generative Adversarial Network
DCGAN	Deep Convolutional Generative Adversarial Network
DP-SGD	Differentially-Private Stochastic Gradient Descent
E-GAN	Evolutionary Generative Adversarial Network
EM	Earth Mover
FID	Fréchet Inception Distance
FPSOGSA	Fuzzy hybrid of Particle Swarm Optimization & Gravitational Search Algorithm
GAN	Generative Adversarial Network
GNU	GNU's Not Unix
HIDS	Host-based Intrusion Detection System
IDS	Intrusion Detection System
InfoGAN	Information Maximizing Generative Adversarial Network
JtR	John the Ripper
KDD	Knowledge Discovery in Databases
KL	Kullback Leibler
LSGAN	Least Squares Generative Adversarial Network
MalGAN	Malicious Generative Adversarial Network
ML	Machine Learning
MsgGAN	Multi-Scale Gradient Generative Adversarial Network
OFDM	Orthogonal Frequency Division Multiplexing
PassGAN	Password Generative Adversarial Network
PATE-GAN	Private Aggregation of Teacher Ensembles Generative Adversarial Network
PCFG	Probabilistic Context Free Grammar
PII	Personally Identifiable Information
Pix2Pix	Image-to-Image Translation with Conditional Adversarial Networks
PlausMal-GAN	Plausible Malware Generative Adversarial Network
PPAPNet	Privacy-Preserving Adversarial Protector Network
ProGAN	Progressive Generative Adversarial Network
RadioML	Radio Machine Learning
RSSI	Received Signal Strength Indicator
tanh	hyperbolic tangent
tDCGAN	Transferred Deep-Convolutional Generative Adversarial Network
TGAN	Temporal Generative Adversarial Nets
WGAN	Wasserstein Generative Adversarial Network
WGANGP	Wasserstein Generative Adversarial Network with Gradient Penalty
XSS	Cross-Site Scripting

## References







- [1] U. Bayer, A. Moser, C. Kruegel and E. Kirda, "Dynamic Analysis of Malicious Code," vol. 2, pp. 66-67, 2006; doi: 10.1007/s11416-006-0012-2.
- [2] A. Petrosyan, "Annual number of malware attacks worldwide from 2015 to 2022," 2023. [Online]. Available: <https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide/>. [Accessed 2 2 2024].
- [3] N. J. Palatty, "30+ Malware Statistics You Need To Know In 2024," Astra, 2023. [Online]. Available: <https://www.getastra.com/blog/security-audit/malware-statistics/>. [Accessed 2 2 2024].
- [4] J. Brownlee, "How to develop an auxiliary classifier GAN (AC-GAN) from scratch with Keras," 2021. [Online]. Available: <https://machinelearningmastery.com/how-to-develop-an-auxiliary-classifier-gan-ac-gan-from-s>. [Accessed 2 2 2024].
- [5] A. Dunmore, J. Jang-Jaccard, F. Sabrina and J. Kwak, "Generative Adversarial Networks for Malware Detection: a Survey," *ArXiv*, vol. abs/2302.08558, 2023; doi: 10.48550/arXiv.2302.08558.
- [6] S. Gihon, "Ransomware Trends Q4 2023 Report," 2024. [Online]. Available: <https://cyberint.com/blog/research/ransomware-trends-and-statistics-2023-report/>. [Accessed 2 2 2024].
- [7] W. Hu and Y. Tan, "Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN," *ArXiv*, vol. abs/1702.05983, 2017.
- [8] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, "Improved Techniques for Training GANs," *ArXiv*, vol. abs/1606.03498, 2016.
- [9] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967-5976; doi: 10.1109/CVPR.2017.632.
- [10] J. Ho and S. Ermon, "Generative Adversarial Imitation Learning," in *Neural Information Processing Systems*, 2016.
- [11] A. Gharakhanian, "Generative Adversarial Networks – Hot Topic in Machine Learning," 2017. [Online]. Available: <https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html>. [Accessed 2 2 2024].
- [12] J. He, Y. Nie and Z. Mao, "Analysis of Image Generation by different Generator in GANs," *Journal of Physics: Conference Series*, vol. 1903, 2021.
- [13] A. Radford, L. Metz and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *CoRR*, vol. abs/1511.06434, 2015.
- [14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville and Y. Bengio, "Generative Adversarial Nets," in *Neural Information Processing Systems*, 2014.
- [15] M. Arjovsky and L. Bottou, "Towards Principled Methods for Training Generative Adversarial Networks," *ArXiv*, vol. abs/1701.04862, 2017.
- [16] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li and Y.-L. Pan, "Generative Adversarial Networks," *ACM Computing Surveys (CSUR)*, vol. 54, pp. 1-38, 2021.
- [17] H. Navidan, P. F. Moshiri, M. Nabati, R. Shahbazian, S. A. Ghorashi, V. Shah-Mansouri and D. Windridge, "Generative Adversarial Networks (GANs) in Networking: A Comprehensive Survey & Evaluation," *ArXiv*, vol. abs/2105.04184, 2021.
- [18] D.-O. Won, Y.-N. Jang and S.-W. Lee, "PlausMal-GAN: Plausible Malware Training Based on Generative Adversarial Networks for Analogous Zero-Day Malware Detection," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, pp. 82-94, 2023; doi: 10.1109/TETC.2022.3170544.
- [19] I. K. Dutta, B. Ghosh, A. H. Carlson, M. W. Totaro and M. A. Bayoumi, "Generative Adversarial Networks in Security: A Survey," *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0399-0405, 2020.
- [20] M. K. Prabhakaran, P. M. Sundaram and A. D. Chandrasekar, "An enhanced deep learning-based phishing detection mechanism to effectively identify malicious URLs using variational autoencoders," *IET Information Security*, vol. 17, no. 3, pp. 315-551, 2023; doi: 10.1049/ise2.12106.
- [21] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, "Improved Techniques for Training GANs," *ArXiv*, vol. abs/1606.03498, 2016.
- [22] S. T. Barratt and S. Rishi, "A Note on the Inception Score," *ArXiv*, vol. abs/1801.01973, 2018.
- [23] A. Borji, "Pros and Cons of GAN Evaluation Measures," *ArXiv*, vol. abs/1802.03446, 2018.
- [24] A. Dunmore, J. Jang-Jaccard, F. Sabrina and J. Kwak, "A Comprehensive Survey of Generative Adversarial Networks (GANs) in Cybersecurity Intrusion Detection," *IEEE Access*, vol. 11, pp. 76071-76094, 2023; doi: 10.1109/ACCESS.2023.3296707.
- [25] T. Che, Y. Li, A. P. Jacob, Y. Bengio and W. Li, "Mode Regularized Generative Adversarial Networks," *ArXiv*, vol. abs/1612.02136, 2017.
- [26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818-2826; doi: 10.1109/CVPR.2016.308.

- [27] J. G. Rohra, B. Perumal, S. J. Narayanan, P. Thakur and R. B. Bhatt, "User Localization in an Indoor Environment Using Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Algorithm with Neural Networks," in *International Conference on Soft Computing for Problem Solving*, 2016.
- [28] F. Meneghello, N. D. Fabbro, D. Garlisi, I. Tinnirello and M. Rossi, "A CSI Dataset for Wireless Human Sensing on 80 MHz Wi-Fi Channels," *IEEE Communications Magazine*, vol. 61, pp. 146-152, 2023.
- [29] S. Yousefi, N. Hirokazu, S. Dayal, S. Ermon and S. Valaee, "A Survey on Behavior Recognition Using WiFi Channel State Information," *IEEE Communications Magazine*, vol. 55, pp. 98-104, 2017.
- [30] "KDD Cup 1999 Data," 1999. [Online]. Available: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>. [Accessed 2 2 2024].
- [31] "Historical Dataset: RADIOML 2016.10A.," 2016. [Online]. Available: <https://www.deepsig.ai/datasets>. [Accessed 2 2 2024].
- [32] R. Ronen, M. Radu, C. Feuerstein, E. Yom-To and M. Ahmadi, "Microsoft Malware Classification Challenge," *ArXiv*, vol. abs/1802.10135, 2018.
- [33] L. Nataraj, S. Karthikeyan, G. Jacobe and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Visualization for Computer Security*, 2011.
- [34] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," *ArXiv*, vol. abs/1411.1784, 2014.
- [35] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever and P. Abbeel, "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," in *Neural Information Processing Systems*, 2016.
- [36] X. Li, L. Chen, L. Wang, P. Wu and W. Tong, "SCGAN: Disentangled Representation Learning by Adding Similarity Constraint on Generative Adversarial Nets," *IEEE Access*, vol. 7, pp. 147928-147938, 2019.
- [37] "How CycleGAN Works?," [Online]. Available: <https://developers.arcgis.com/python/guide/how-cyclegan-works/>. [Accessed 2 2 2024].
- [38] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242-2251, 2017.
- [39] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5967-5976, 2016; doi: 10.1109/CVPR.2017.632.
- [40] A. Odena, C. Olah and J. Shlens, "Conditional Image Synthesis with Auxiliary Classifier GANs," in *International Conference on Machine Learning*, 2016.
- [41] R. Nagaraju and M. Stamp, "Auxiliary-Classifer GAN for Malware Analysis," *ArXiv*, vol. abs/2107.01620, 2021.
- [42] J. Donahue, P. Krähenbühl and T. Darrell, "Adversarial Feature Learning," *ArXiv*, vol. abs/1605.09782, 2016.
- [43] W. Xu, J. Jang-Jaccard, T. Liu, F. Sabrina and J. Kwak, "Improved bidirectional GAN-based approach for network intrusion detection using one-class classifier," *MDPI (Basel, Switzerland)*, 2022; doi: 10.3390/computers11060085.
- [44] M. Saito, E. Matsumoto and S. Saito, "Temporal Generative Adversarial Nets with Singular Value Clipping," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2849-2858; doi: 10.1109/ICCV.2017.308.
- [45] A. Munoz, M. Zolfaghari, M. Argus and T. Brox, "Multi-Variate Temporal GAN for Large Scale Video Generation," *ArXiv*, vol. abs/2004.01823, 2020.
- [46] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang and S. P. Smolley, "Least Squares Generative Adversarial Networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2813-2821; doi: 10.1109/ICCV.2017.304.
- [47] M. Arjovsky, S. Chintala and L. Bottou, "Wasserstein GAN," *ArXiv*, vol. abs/1701.07875.
- [48] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. C. Courville, "Improved Training of Wasserstein GANs," in *Neural Information Processing Systems*, 2017.
- [49] C. Wang, C. Xu, X. Yao and D. Tao, "Evolutionary Generative Adversarial Networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 921-934, 2019; doi: 10.1109/TEVC.2019.2895748.
- [50] D. Berthelot, T. Schumm and L. Metz, "BEGAN: Boundary Equilibrium Generative Adversarial Networks," *ArXiv*, vol. abs/1703.10717, 2017.
- [51] T. Karras, T. Aila, S. Laine and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," *ArXiv*, vol. abs/1710.10196, 2017.
- [52] A. Karnewar, O. Wang and R. S. Iyengar, "MSG-GAN: Multi-Scale Gradient GAN for Stable Image Synthesis," *ArXiv*, vol. abs/1903.06048, 2019.
- [53] V. Chandola, A. Banerjee and K. Vipin, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1-15:58, 2009.
- [54] T. Schlegl, P. Seeböck, S. Waldstein, G. Lang and U. Schmidt-Erfurth, "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks," *Medical Image Analysis*, vol. 54, pp. 30-44, 2019.
- [55] C.-S. Houssam Zenati and Foo, B. Lecouat, G. Manek and V. R. Chandrasekhar, "Efficient GAN-Based Anomaly Detection," *ArXiv*, vol. abs/1802.06222, 2018.

- [56] D. Akçay and B. D. Akçay, "Effect of media content and media use habits on aggressive behaviors in the adolescents," *The European Research Journal*, vol. 5, no. 3, 2019; doi: 10.18621/eurj.395892.
- [57] X. Luo, Y. Jiang, E. Wang and X. Men, "Anomaly detection by using a combination of generative adversarial networks and convolutional autoencoders," *EURASIP Journal on Advances in Signal Processing*, vol. 2022, pp. 1-13, 2022.
- [58] X. Xia, X. Pan, N. Li, X. He, L. Ma, X. Zhang and N. Ding, "GAN-based anomaly detection: A review," *Neurocomputing* 493, 2022; doi: 10.1016/j.neucom.2021.12.093.
- [59] C. Goetz and B. Humm, "Decentralized Real-Time Anomaly Detection in Cyber-Physical Production Systems under Industry Constraints," *Artificial Intelligence Enhanced Health Monitoring and Diagnostics*, vol. 23, no. 9, p. 4207, 2023; doi: 10.3390/s23094207.
- [60] W. Lim, S. K. C. Y. Sheng and B. T. T. C. C. L. Lau, "Future of generative adversarial networks (GAN) for anomaly detection in network security: A review," *Computers & Security*, 2024.
- [61] M. Usama, M. Asim, S. Latif, J. Qadir and Ala-Al-Fuqaha, "Generative Adversarial Networks For Launching and Thwarting Adversarial Attacks on Network Intrusion Detection Systems," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2019, pp. 78-83; doi: 10.1109/IWCMC.2019.8766353.
- [62] L. Zilong, Y.-y. Shi and X. Zhi, "IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection," *ArXiv*, vol. abs/1809.02077, 2018.
- [63] E. Seo, H. M. Song and H. K. Kim, "GIDS: GAN based Intrusion Detection System for In-Vehicle Network," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, 2018, pp. 1-6; doi: 10.1109/PST.2018.8514157.
- [64] M. A. Salem, S. Taheri and J.-S. Yuan, "Anomaly Generation Using Generative Adversarial Networks in Host-Based Intrusion Detection," *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 683-687, 2018.
- [65] M. Shahpasand, L. Hamey, D. Vatsalan and M. Xue, "Adversarial Attacks on Mobile Malware Detection," in *2019 IEEE 1st International Workshop on Artificial Intelligence for Mobile (AI4Mobile)*, 2019, pp. 17-20; doi: 10.1109/AI4Mobile.2019.8672711.
- [66] J. Kargaard, T. Drange, A.-L. Kor, H. Twafik and E. Butterfield, "Defending IT systems against intelligent malware," in *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, 2018, pp. 411-417; doi: 10.1109/DESSERT.2018.8409169.
- [67] J.-Y. Kim, S.-J. Bu and C. Sung-Bae, "Malware Detection Using Deep Transferred Generative Adversarial Networks," in *International Conference on Neural Information Processing*, 2017.
- [68] J.-Y. Kim, S.-J. Bu and S.-B. Cho, "Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders," *Inf. Sci.*, Vols. 460-461, pp. 83-102, 2018.
- [69] A. Chowdhary, K. Jha and M. Zhao, "Generative Adversarial Network (GAN)-Based Autonomous Penetration Testing for Web Applications," *Sensors (Basel, Switzerland)*, vol. 23, 2023.
- [70] M. Singh, P. Singh and P. Kumar, "An Analytical Study on Cross-Site Scripting," in *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, 2020, pp. 1-6.
- [71] R. Shobana and M. Suriakala, "A Thorough Study On Sql Injection Attack-Detection And Prevention Techniques And Research Issues," *Journal of Information and Computational Science*, vol. 10, no. 5, 2020.
- [72] C. Binnie, "Password Cracking with Hashcat," in *Linux Server Security: Hack and Defend*, 2016; doi: 10.1002/9781119283096.ch9.
- [73] K. Marchetti and P. Bodily, "John the Ripper: An Examination and Analysis of the Popular Hash Cracking Algorithm," in *2022 Intermountain Engineering, Technology and Computing (IETC)*, 2022, pp. 1-6. doi: 10.1109/IETC54973.2022.9796671; doi: 10.1109/IETC54973.2022.9796671
- [74] M. Kuperberg, "Markov Models," in *Dependability Metrics*, 2005, pp. 48-55. doi: 10.1007/978-3-540-68947-8\_8.
- [75] Z. Chi and S. Geman, "Estimation of Probabilistic Context-Free Grammars," *Computational Linguistics*, vol. 24, no. 2, pp. 298-305, 1998.
- [76] S. Nam, S. Jeon, H. Kim and J. Moon, "Recurrent GANs Password Cracker For IoT Password Security Enhancement †," *Sensors (Basel, Switzerland)*, vol. 20, 2022.
- [77] L. Monostori, "Cyber-physical production systems: roots from manufacturing science and technology," *at - Automatisierungstechnik*, vol. 63, pp. 766-776, 2015.
- [78] S. R. Chhetri, A. B. Lopez, J. Wan and M. A. Al Faruque, "GAN-Sec: Generative Adversarial Network Modeling for the Security Analysis of Cyber-Physical Production Systems," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 770-775; doi: 10.23919/DATE.2019.8715283.
- [79] M. Rigaki and S. Garcia, "Bringing a GAN to a Knife-Fight: Adapting Malware Communication to Avoid Detection," in *2018 IEEE Security and Privacy Workshops (SPW)*, 2018, pp. 70-75; doi: 10.1109/SPW.2018.00019.
- [80] D. Saxena and J. Cao, "Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1-42, 2021; doi: 10.1145/3446374

- [81] J. Li, Y. Liu and L. Qijie, "Generative Adversarial Network and Transfer Learning Based Fault Detection for Rotating Machinery with Imbalance Data Condition," *Measurement Science and Technology*, vol. 33, no. 4, 2022; doi: 10.1088/1361-6501/ac3945.
- [82] A. Yang, C. Lu, J. Li, X. Huang, T. Ji, X. Li and Y. Sheng, "Application of meta-learning in cyberspace security: a survey," *Digital Communications and Networks*, vol. 9, no. 1, pp. 67-78, 2023; doi: 10.1016/j.dcan.2022.03.007.
- [83] J. Soenen, K. Leuven, E. V. Wolputte, L. Perini, V. Vercruyssen, W. Meert, J. Davis and H. Blockeel, "The Effect of Hyperparameter Tuning on the Comparative Evaluation of Unsupervised Anomaly Detection Methods," 2021.
- [84] X. Xuan, X. Pan, N. Li, X. He, L. Ma, X. Zhang and N. Ding, "GAN-based anomaly detection: A review," *Neurocomputing*, vol. 493, pp. 497-535, 2022; doi: 10.1016/j.neucom.2021.12.093.
- [85] Z. Dehghanian, S. Saravani, M. Amirmazlaghani and M. Rahmati, "Spot The Odd One Out: Regularized Complete Cycle Consistent Anomaly Detector GAN," 2023.
- [86] J. Fu, W. Lina, J. Ke, K. Yang and R. Yu, "GANAD:A GAN-based method for network anomaly detection," 2023; doi: 10.21203/rs.3.rs-2081269/v1.

## Recent Progress on Applications of Artificial Intelligence for Sustainability of Solar Energy Technologies: An Extensive Review

Jamilu Ya'u Muhammad <sup>1\*</sup> , Abubakar Abdulkarim <sup>2</sup> , Nafi'u Muhammad Saleh <sup>3</sup> , Israel Ehile <sup>4</sup> , Nuraini Sunusi Ma'aji <sup>5</sup> , Audu Taofeek Olaniyi <sup>6</sup> 

<sup>1</sup> Department of Mechanical Engineering, Nigerian Army University Bui, Nigeria

<sup>2</sup> Department of Electrical Engineering, Bilyaminu Usman State Polytechnic, Hadejia, Jigawa State, Nigeria

<sup>3</sup> Department Corps Safety Engineering, Federal Road Safety Corps, Lagos, Nigeria

<sup>4</sup> Department of Electrical and Computer Engineering, Nazarbayev University, Kazakhstan

<sup>5</sup> Department of Electrical and Electronics Engineering, Nigerian Army University Bui, Nigeria

<sup>6</sup> Department of Mechanical Engineering, Bayero University Kano, Nigeria

### Abstract

Green energy sources are most promising energy sources in the globe, as they are non-pollutant sources. Solar energy sources are among green energy sources that are free and abundant in nature, yet solar energy sources have some shortcoming such as faults on the solar PV modules, improper maintenance and some climatic and environmental impacts. Artificial intelligences are employed to solve most of these shortcoming like prediction of the solar irradiance of the specific sites, parameters estimation on the solar PV modules, fault detection on the solar PV modules surfaces and forecasting of solar PV power output. This paper presents extensive review on application of artificial intelligences to solve problems related to solar energy systems from 2009 to 2024. It was found that from most of the literatures, artificial intelligent algorithms were more accurate and efficient than the conventional methods and it has an ability to solve complex and non-linear data. This work will help scholars to explore the relationship between solar energy technologies and artificial intelligences.

**Keywords:** Artificial neural network; Genetic algorithms; Machine learning; Particle swarm optimization; Solar irradiance.

### 1. Introduction

To have sustainable and reliable access to energy source at appropriate cost and efficient and cost-effective consumption pattern of energy are two among seventeen goals of sustainable development and utilization of green energy effectively is now a crucial research interest to many academia and industries globally.

Since burning of fuels cause pollution [1] and promoting environmental effects to the atmosphere is another critical problem [2]. Renewable energies continue improving both technical and economical for over a decade [3, 4]. Investment in these energies has been rising in the last years, even with a crisis such as the COVID-19 pandemic [3,4].

Renewable energy includes solar, wind, hydro, biomass, geothermal, etc [5, 6]. In attempt to improve the accuracy and reliability of renewable energy sources, various methods have been developed [7-9]. Nowadays, artificial intelligence techniques are applied to predict future problems and solve the problems that are difficult or impossible to be solved using conventional techniques [10].

Artificial Intelligence (AI) is a comprehensive high-technology that interact with human-based and machine intelligences. It comprises branches such as genetic algorithms (GA) [11], particle swarm optimization (PSO), simulated annealing (SA) [12], artificial neural networks (ANN), and hybrid models [13]. Synthetic intelligence methods are widely applied to renewable energy prediction because they can solve nonlinear and complex data structures [9, 14].

Artificial intelligence techniques are used in many disciplines such as renewable energy systems. In solar energy system, AI techniques can be employed to predict solar PV farms [15, 16]; and also to forecast solar irradiance of the regions [17]. A hybrid approach that combine two or more algorithms is used to improve the accuracy and reliability of power plant [18]. Most of the algorithms used are Particle Swarm Algorithms (PSA), Bee Colony Algorithm (BCA), Genetic Algorithm (GA) in order to optimize the parameters of the prediction model [19–21]. Particle Swarm Optimization (PSO) algorithm is the most widely used algorithm with fast convergence speed [22, 23].

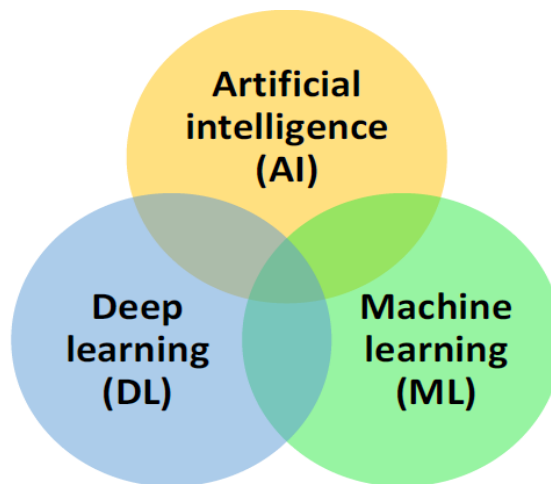
\*Corresponding author

E-mail address: muhdjy87@gmail.com

This paper tends to provide an extensive review on application of artificial intelligence to solve problems in renewable energy systems for sustainable development. The paper consists of four sections namely: section 1 gives general overview of the paper; section 2 provides the artificial intelligence techniques; section 3 consists of artificial intelligence-based solar energy technologies and section 4 drawn-out the conclusions.

## 2. Artificial Intelligence Techniques

Artificial Intelligence (AI) is new technique to process data processing in order to execute complex task by mimicking human intelligence [24]. Artificial Intelligence (AI) may be incorporated into hardware systems or dependent on software mostly in the virtual world [25]. To improve reliability, stability and performance of the renewable energy systems with complex data, AI is used for optimization, exploration, forecasting/prediction and regression easily [25–28]. Some author categorized AI into machine learning (ML) and deep learning (DL) while some said that deep learning (DL) is a subdivision of machine learning (ML), and thus both are AI techniques as shown in **Figure 1**.



**Figure 1.** Venn Diagram Showing the Relationship Between Machine Learning (ML) and Deep Learning (DL) as Categorized by [25]

### 2.1. Machine Learning

Machine learning (ML) is an artificial intelligence technique that allow systems to execute a task from the existing data stored in the system automatically without human aid [29]. In machine learning (ML), a raw data is processed by selection time-series data through extraction, training and evaluation of the feature data and then model distribution.

Machine learning (ML) is subdivided in four algorithms namely: supervised, unsupervised, semi-supervised, and reinforcement learning.

#### 2.1.1. Supervised Learning

Supervised learning is a machine learning technique that required supervisor in executing a task (**Figure 2**). The supervisor helps to guide the algorithm in predictions of the desired output and the output may lead to either classification or regression problem. Classification problem deals with a discrete variable of the output data while regression problem deals with a real value of the output data [30]. Examples of supervised learning algorithms include vector support machines, linear and logistic regression, decision trees, k-Nearest Neighbors, Neural Networks, naive Bayes and random forest.



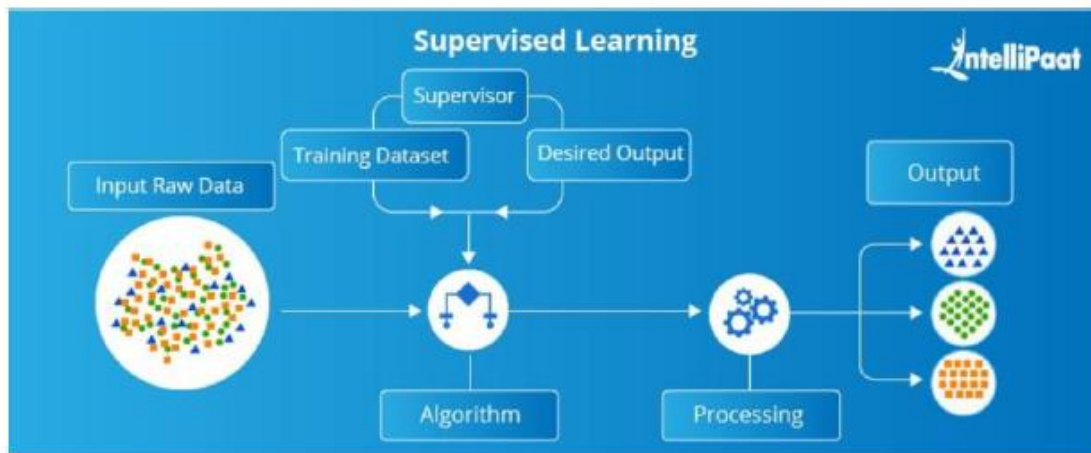


Figure 2. Schematic of supervised learning [31]

### 2.1.2. Unsupervised Learning

Unsupervised learning is a second machine learning techniques that required no supervisor to process the required output (Figure 3). Unsupervised learning algorithms trained with unknown data and expected outputs, hence they follow rules and patterns of the available database before being able to understand the actual data [26, 29]. Examples of unsupervised learning are fuzzy means, k-means and hierarchical clustering techniques.

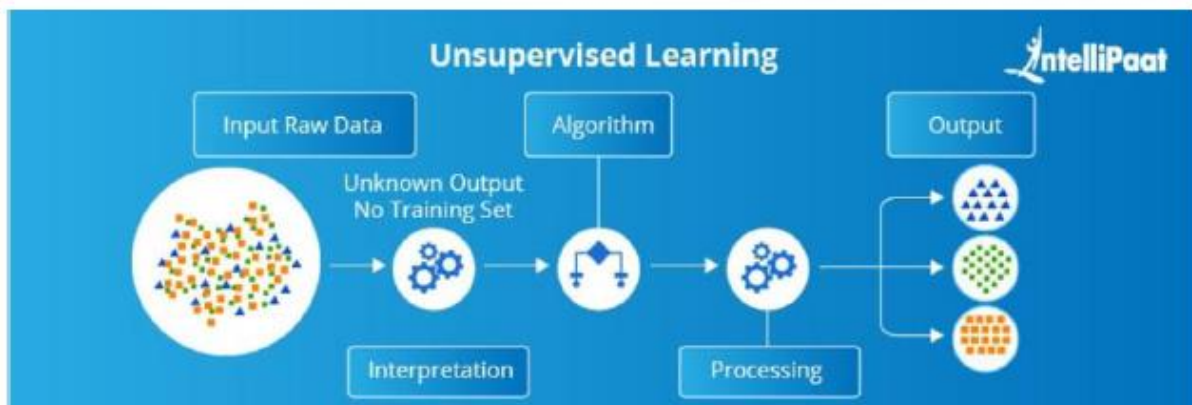


Figure 3. Schematic of unsupervised learning [31]

### 2.1.3. Semi-Supervised Learning

Semi-supervised learning combined features of both supervised and unsupervised learning [32]. This involve training an algorithm to discover a model, and the most common issues that can be solved with it are classification and clustering [25]. Generic and graph-based models are examples of semi-supervised learning.

### 2.1.4. Reinforcement Learning

Reinforcement learning is a machine learning depend on a searching goal approach by which the learner carried out different tasks to find out which task is the best task to achieve a specific target [33]. Figure 4 shows the working mechanism of the reinforcement learning, Q-Learning and Monte Carlo methods are common examples of the reinforcement learning.



Figure 4. Schematic of reinforced learning [31]

## 2.2. Deep Learning

Deep learning is an artificial intelligence that combine a set of classifiers and train them to predict a single prediction of the classifier [29, 34]. It is used for extraction of features from data sets to model the flexibility in network designs [34]. Deep learning techniques include long short-term memory (LSTM), convolutional neural networks (CNN), generative adversarial networks (GAN), and hybrid combinations [24, 26, 29, 34].

### 2.2.1. Long Short-term Memory (LSTM)

Long Short-term Memory (LSTM) is time series approach designed to process time-dependent variables of the data sets [34]. The recurrent neural network (RNN) is an algorithm of long short-term memory (LSTM) is applied when dependent input data is used. In renewable energy application, solar irradiance can be forecast using LSTM [34, 35]. Generally, LSTM is applied to improve efficiency, reliability and predictability of the systems [34].

### 2.2.2. Convolutional Neural Networks (CNNs)

Convolutional neural network (CNN) is deep learning technique whereby convolution replaced matrix multiplication in one or more layers of the conventional neural networks (NNs) [24, 35]. Forward and backward stages are involved in convolutional neural network, whereby input image with the existing parameters is served as a forward stage while executing of each parameter's gradient is a backward stage [26, 29, 34]. Convolutional neural network is usually applicable in managing and fault detection of large solar PV farms [34].

### 2.2.3. Generative Adversarial Networks (GANs)

Generative adversarial network (GAN) is another deep learning technique developed to improve performance, training stability and to generate high-quality samples [25, 26, 29, 34]. Virtual batch normalizing, feature matching and one-side label smoothing are heuristic techniques applied to enhance the training stability of generative adversarial network (GAN) [34, 36].

## 3. AI-Based Solar Energy Technologies

Artificial Intelligence is applicable in solar energy technology for prediction of solar irradiance, solar energy harnessing, fault detections on solar PV grids and so on. Many researchers conducted researches related to these and some of their findings were as follows:

Benghanem et al. [37] have developed models that forecast the daily global solar radiation using artificial neural network (ANN). The input parameters used to develop these models were air temperature, sunshine hours, relative humidity, and day of year. The authors developed six artificial neural network (ANN) and for each model, the output is the daily global solar radiation. Their findings revealed that artificial neural network (ANN) was best algorithm for forecasting global solar radiation as its correlation coefficient were 97% for each models developed.

Esen et al., [38] used artificial neural network (ANN) and wavelet neural network (WNN) algorithms for modelling of a solar air heating system. The experimental measureable parameters were used as input parameters. A comparative analysis was done between the experimental and predicted results and it was found that the developed models were more accuracy than the experimental results in terms of the efficiency of the solar air heating system.

Fadare [39] forecast the potential of solar energy utilization in Nigeria using artificial neural network (ANN). The input parameters used to develop the model were sunshine hours, mean daily temperature, relative

humidity, altitude, days of the year, longitude and latitude. These parameters were obtained from database of Nigerian Airspace Agency (NASA). The result revealed that the correlation coefficient of the predicted model was 96% accurate that the measured results.

A Multi-Layer Perceptron (MLP) network to predict one day ahead solar irradiance was presented by Mellit and Pavan [40]. The input parameters used in this work were mean daily solar irradiance and the mean daily air temperature. They compared the power produced by a 20 kWp Grid Connected Photovoltaic Plant and the predicted one using the developed MLP predictor and the result shows that MLP predicted solar irradiance performed better for the subsequent four sunny days.

A simple multi-layer feedforward perceptron was developed by Sanusi et al., [41] to forecast the sizing of solar PV array in Nigeria. The input data were the analytical parameters used for sizing solar PV array that were calculated using location's meteorological parameters obtained from Nigeria Meteorological Agency (NIMEC) and Nigerian Airspace Agency (NASA). The result obtained in the study shows that the model developed have root mean square error (RMSE) of 0.085 and 0.046 for the battery storage and solar PV array size respectively which is more accurate than the analytically derived parameters.

Bartler et al., [42] adopted VGG16 architecture using CNN consisting of 13 convolutional layers. The EL camera was used to capture the photographs of the solar modules and these photographs were serve as the datasets. To reduce the error, data augmentation and over-sampling were used to train the model. It was found that a Balance Error Rate (BER) of 7.73% was achieved if both data augmentation and over-sampling were employed.

In 2018, Deitsch and coauthors conducted a comparative study between SVM and Convolutional Neural Networks by composing of 2624 EL images of the solar modules [43]. The ELPV data sets were trained using SVM of different feature descriptors. Silicon-type solar modules (Polycrystalline and Monocrystalline solar modules) were used to test the models. This models were developed using Python with the aid of Keras for the Neural Network (NN) and the finding of this study is achieved with the accuracy of 82.4%.

A new network for optimizing the ANN for MPPT was presented by Duman et al., [44]. For real-time applications, the ANN was structured with 2-3-3-1 structure. The input of the network were solar irradiance and temperature and output of the network was maximum voltage. The authors compared P&O and conventional NN and their results revealed that the network provides more stability and efficiency.

Faragola et al., [45] developed a hybrid model that combines artificial neural network (ANN) with Support Vector Machines (SVM) [46]. The datasets which consists of solar irradiance and temperature were improved via Course Gaussian Support Vector Machine (CGSVM). The artificial neural network (ANN) was composed of two inputs, one output and a single hidden layer with 13 neurons. The results are slightly worse on power than the ANFIS, but the required time was significantly less than the ANFIS.

In the same year, Karimi et al., [47] developed a similar algorithm using SVM and CNN. The CNN was composed of two convolutional layers with leaky-relu and max-pooling. The SVM was trained with different features extracted from the images. A module with 540 cells was used to test the algorithm using its 90 images as the data sets. An accuracy of 98% was achieved in this study.

A new model with six convolutional layers using different regularization techniques was presented by Akram et al., [48]. The dataset used was the ELPV dataset that composed 2624 images. The algorithms were run on Tensorflow. The accuracy of the model was 93%.

Abdel-Naseer and Mahmoud [49] conducted a comparative analysis of five (5) different architectures of RNN: A fundamental Long Short-Term Memory (LSTM), a Long Short-Term Memory (LSTM) integrated with the window technique, Long Short-Term Memory (LSTM), incorporated with time steps, Long Short-Term Memory (LSTM) with memory between batches and stacked Long Short-Term Memory (LSTM) with memory between batches. Two datasets of different cities were used to test the three (3) models. The results show the Long Short-Term Memory (LSTM), incorporated with time steps is more reliable and have high accuracy than others since it has an RMSE of 82.15 in the first dataset and an RMSE of 136,87 in the second, which uses prior time steps in the PV series as inputs when compared with Artificial Neural Network (ANN).

Karimi et al., [50] present new models of two solar PV modules images with cracks and corrosion. The solar PV modules images were snapped with EL camera and 5400 images were used as data sets. The authors used SVM and CNN algorithms whereas Keras and Tensorflow were adopted to validated the algorithms experimentally. The CNN is composed of two convolutional layers. The SVM parameters are optimized by a grid search. An improved result of 99% accuracy was obtained using this models.

Another work was done by Torress et al., [51] using a multi-step method which decomposes the prediction challenges with Big Data and Deep Learning approach. They used to next-day-ahead forecast in 30 min intervals. Three (3) hidden layers with 12 and 32 neurons as differences was found to be the best structure. Deep learning (DL) was found to be directly proportional to the training time as the big solar data was found using deep learning approach and it more accurate than other approaches.

A modified RNN was achieved by Yao et al., [52]. The authors applied Echo State Network (ESN) as demonstrated by [53]. In this work, Echo State Network (ESN) used to replace the conventional hidden layers of RNN as a dynamical reservoir. And the number of dynamic reservoirs and input data sets were obtained using restricted Boltzmann machine (RBM) [54] and principal component analysis (PCA) [55]. These networks can obtain better results than typical RNN. A DFP Quasi-Newton algorithm [56] was employed to determine the parameters of the networks. Their findings revealed that this approach can improve the prediction of solar PV with MAPE of 0.00195% against other solar PV irradiance prediction methods.

A PSO algorithm was employed to optimize the initial weights of the neural network by Al-Majidi et al., [57]. The input datasets used by the authors are solar irradiance and temperature and power of solar PV was the output. The model proved to be more effective under various weather conditions than other ANN techniques.

Balzategui et al., [58] developed a new network with Convolutional Neural Networks using U-net approach. The input datasets used was 542 EL images with 21 convolutional layers of different sizes. The authors used Keras and Tensorflow to implement the algorithms. It was found that it was better to accept a slight decrease in the performance in order to improve the speed of the system.

A combination of static Wavelet Transform and discrete Wavelet Transform was used by Mathias et al., [59] to harness edge and textural features of the solar modules. The authors used 2300 EL images for input datasets while SVM and FFNN were employed as the classifiers. The result of the study revealed that SVM is less accurate with 92.6% and 93.6% for the FFNN.

A complex hybrid algorithm using Neural Network (NN) was developed by Niu et al., [60]. To determine the most essential factors that affect solar PV performance, the authors used a Random Forest (RF) [61] and a Complementary Ensemble Empirical Mode Decomposition (CEEMD) was employed to decomposed the original sequence of the data sets [62], this will help to stabilized the original data sets and to ensured that the quality of the data sets was enhanced. The authors modified the PSO by optimizing the neural network (NN) using dynamic inertial factor particle swarm optimization (DIFPSO) [63, 64]. They finally improved the accuracy of the solar PV irradiance prediction with MAE of 2.84, 13.01 and 10.12 on sunny hours, rainy hours and cloudy hours respectively.

A hybrid algorithm integrated with Wavelet Transform (WT), PSO and Radial Basic Neural Feed Networks (RBFNN) was applied for prediction from 1 to 6 hours ahead by Wen et al., [65]. Data selection (Filtering) on the immediate fifteen (15) days before prediction days was done using Wavelet Transform (WT), and optimization of RBFNN was done using PSO algorithm. The solar irradiance and temperature were used as the input data sets. The authors achieved the network with better performance than the other methods, and MAE of 4.22%, 7.04% and 9.13% for a 1-hour-ahead prediction, for a 3-hour-ahead prediction and 9.13% for 6-hour- ahead prediction.

Zhang et al., [66] used Dendritic Neuron Network (DNN) as a new Neural Network (NN) for prediction of solar PV power. Synaptic layer, branch layer, membrane layer and cell-body layer are the layers of Dendritic Neuron Network (DNN) [67]. The temperature and irradiance of the city (Input data sets) were feed to the first (synaptic) layers which transformed the input data sets and selected the essential data sets and moved them to the second (branch) layer, the numerical work was done by cell-body layer. When the selected data sets exceed a required threshold, the cell-body layer will move them to other neurons through the axon. The findings of this work show that using MATLAB, this approach performed better with MAPE of 4.55 and 10.9 with weak data fluctuations and with strong data fluctuations respectively.

Deep-Feature-Based network was developed using convolutional neural networks by Demirci et al., [68]. The SVM and forward neural network (FNN) algorithms were used as classifiers of the convolutional neural networks. 2624 images of the solar cells were served as the data sets of the study. DarkNet-19, Resnet-50, VGG-16 and VGG-19 were selected to extract the features of the data sets. The result shows that SVM with 89.63% and 94.52% accuracies in four-class and in two-class classification were better than forward neural network (FNN) algorithms.

Sattar et al., [69] presents a Marine Predator Algorithm [70] to estimate parameters of the solar PV modules. The algorithm developed was run using MATLAB to extracted solar PV and it was found that the RSME on a single-diode of the solar cell and double-diode of the solar cell were 0.000773 and 0.000765 respectively.

Sibtain et al., [71] developed a multistage hybrid model for forecasting solar irradiance based on a multivariate meteorological data. They initially evaluate five standalone models, including recurrent deterministic policy gradient (RDPG), long short term memory (LSTM) neural network, extreme gradient boosting (XGB), Gaussian process regression (GPR), and support vector regression (SVR). The RDPG model outperforms the standalone counterparts by demonstrating 2.485 W/m<sup>2</sup>, 20.591 W/m<sup>2</sup>, 18.316 W/m<sup>2</sup> and 23.176 W/m<sup>2</sup> reductions in RMSE compared to the LSTM, XGB, GPR, and SVR models. Afterward, the performance of the RDPG model is further enhanced, by developing two-stage hybrid models, including improved complete ensemble empirical mode decomposition with additive noise-RDPG (ICEEMDAN-RDPG) and variational mode decomposition-RDPG (VMD-RDPG). The subsequent construction of the hybrid model

ICEEMDANSE-VMD-RDPG (ISVR) results in the further improvement of the two-stage hybrid models. The ISVR surpasses all the established models including VMD-RDPG, ICEEMDAN-RDPG, RDPG, LSTM, XGB, GPR, and SVR by displaying 18.401 W/m<sup>2</sup>, 13.908 W/m<sup>2</sup>, 33.223 W/m<sup>2</sup>, 53.111 W/m<sup>2</sup>, 67.704 W/m<sup>2</sup>, 67.502 W/m<sup>2</sup> and 69.943 W/m<sup>2</sup> respectively, decrease in RMSE.

Su et al., [72] presents Complementary Attention Network (CAN) that consist of subnetwork connected with a spatial attention subnetwork. As stated by [73] Complementary Attention Network (CAN) is grouped with any CNN, Fast R. 2129 and 2029 EL images were used as datasets. The authors used Python to implement the algorithms of the developed network. The model developed achieved an accuracy of 99.17% and a precision of 87.38%.

Solar Power Generation Prediction System was developed by Lee and Shin [74]. Neural network, SVM, and deep learning were used as prediction algorithms, and the optimal algorithm was selected by using the root mean square error (RMSE). They developed a predictive model that improve the prediction rate by changing the algorithm structure and modifying constants. Then, a defect detection system is developed by applying the predicted results to the domestic regional data.

Martinez Lopez et al., [75] quantified the existing relation between irradiance variations and efficiency loss of the logic of the Perturb-and-Observe MPPT algorithm, along with the sensitivity of the MPPT to its control parameters. It was found that when the algorithm parameters are not tuned properly, its efficiency will drop to nearly 2% and the solar irradiance variability causes a systematic energy loss of the algorithm that can only be quantified by ignoring the hardware components. The authors provide an additional efficiency loss to be considered in the calculations in order to improve the energy yield estimation.

A novel hybrid GWO–PSO-based maximum power point tracking for photovoltaic systems operating under partial shading conditions was presented by Smail et al., [76]. The feasibility and effectiveness of the hybrid GWO–PSO-based MPPT method were verified via a co-simulation technique that combines MATLAB/SIMULINK and PSIM software environments, while comparing its performance against GWO, PSO and P&O based MPPT methods. The simulation results carried out under dynamic environmental conditions have shown the satisfactory effectiveness of the hybrid MPPT method in terms of tracking accuracy, convergence speed to GMPP and efficiency, compared to other methods.

In solar water pumping system, Sumathi and Abitha [77] presented a novel system which consisted a hybridized MPPT technique based on Gravitational Search Algorithm (GSA) and Particle Swarm Optimization (PSO). The dynamic and steady-state performance of a permanent magnet brushless DC (BLDC) motor coupled to a centrifugal water pump fed by the SPV array-SEPIC was assessed, and its applicability was confirmed using simulated results in the MATLAB/Simulink environment. The Cuk converter and Single Ended Primary Inductor Converter (CUKSEPIC) achieves high settling time of 0.9 for a 5.0 kW output power, improving power efficiency.

Sangsang et al., [78] designed an optimum system for harnessing solar energy system using Fuzzy Logic (FL). They developed a maximum power point tracking (MPPT) to optimize the energy potential harvested from the solar PV system and Fuzzy logic (FL) was used to operate the MPPT technique on the converter. And the result of the study proved that FL's tracking speed algorithm for tracking MPP is twice as fast as conventional P&O.

A novel hybrid optimization approach for fault detection in Photovoltaic arrays and inverters using Artificial Intelligence (AI) and statistical learning techniques was developed by Abubakar et al., [79]. Their proposed technique integrated Elman neural network (ENN), boosted tree algorithms (BTA), multi-layer perceptron (MLP), and Gaussian processes regression (GPR) for enhanced accuracy and reliability in fault diagnosis. Feature engineering-based sensitivity analysis was utilized for feature extraction. The fault detection and diagnosis were assessed using several statistical criteria including PBAIS, MAE, NSE, RMSE, and MAPE. Two intelligent learning scenarios are carried out. The first scenario was conducted for PV array fault detection with DC power (DCP) as output. The second scenario was conducted for inverter fault detection with AC power (ACP) as the output. The proposed technique was capable of detecting faults in PV arrays and inverters, providing a reliable solution for enhancing the performance and reliability of solar energy systems. A real-world solar energy dataset was used to evaluate the proposed technique with results compared to existing detection techniques and obtained results showing that it outperforms existing fault detection techniques, achieving higher accuracy and better performance. The GPR-M4 optimization justified its reliably among all the models with MAPE = 0.0393 and MAE = 0.002 for inverter fault detection, and MAPE = 0.091 and MAE = 0.000 for PV array fault detection.

Alba et al., [80] worked on the effect of climate on Photovoltaic yield prediction using machine learning models. An extensive data gathering process was performed and open-data sources were prioritized. A website [www.tudelft.nl/open-source-pv-power-databases](http://www.tudelft.nl/open-source-pv-power-databases) was created with all found open data sources for future research. Five machine learning algorithms and a baseline one was trained for each solar PV system. Their results showed that the performance ranking of the algorithms was independent of climate. Systems in dry climates depict on average the lowest Normalized Root Mean Squared Error (NRMSE) of 47.6 %, while those

in tropical presented the highest of 60.2%. In mild and continental climates, the NRMSEs were 51.6% and 54.5%, respectively. When using a model trained in one climate to predict the power of a system located in another climate, on average systems located in cold climates show a lower generalization error, with an additional NRMSE as low as 5.6% depending on the climate of the test set. Robustness evaluations were also conducted that increase the validity of the results.

Arash et al., [81] proposed a global solar radiation forecasting approach based on federated learning (FL) and convolutional neural network (CNN). CNN input for network training in each client used were data related to eight regions of Iran with different climatic features. To test the effectiveness of the global supermodel, data related to three new regions of Iran named Abadeh, Jarqavieh, and Arak were used. It can be seen that the global forecasting supermodel was able to forecast solar radiation for Abadeh, Jarqavieh, and Arak regions with 95%, 92%, and 90% accuracy coefficients, respectively. Finally, in a comparative scenario, various conventional machine learning and deep learning models were employed to forecast solar radiation in each of the study regions. The results of the above approaches were compared and evaluated with the results of the proposed FL-based method. The results show that, since no training data were available from regions of Abadeh, Jarqavieh, and Arak, the conventional methods were not able to forecast solar radiation in these regions. This evaluation confirms the high ability of the presented FL approach to make acceptable predictions while preserving privacy and eliminating model reliance on training data.

Numerous methods based on artificial intelligence (AI) were proposed by Azad et al., [82] to address the issue of the appearance of multiple peaks in the performance of solar panels caused by partial shading. The authors presented the energy-valley optimizer-based optimization (EVO) technique, which is designed to efficiently and dependably tackle the issue of partial shading (PS) in detecting the maximum power point (MPP) for photovoltaic (PV) systems. The EVO algorithm enhanced the speed of tracking and minimized the power output fluctuations during the tracking phase. Extensive validation of the proposed technique was conducted using the Typhoon hardware-in-the-loop (HIL) 402 emulator. The effectiveness of the suggested method was compared with the established cuckoo search algorithm for achieving maximum power point tracking (MPPT) within a photovoltaic (PV) system. This comparison takes place under equivalent conditions to ensure a fair performance evaluation.

Another solar PV power prediction model was developed based on Wavelet Neural Network (WNN) by Bo et al., [83]. The authors provide data reference for the scheduling department through the prediction of distributed PV power. Initially, they studied sensing layer, prediction layer, and service layer on the distributed PV power prediction architecture, and then the specific functions of the prediction layer were designed based on a wavelet neural network. They finally concluded that the proposed model can effectively achieve the power prediction of distributed PV.

Dae et al., [84] proposed an approach to develop a solar radiation model with spatial portability based on deep neural networks (DNNs). The data for development and internal testing of the DNNs, respectively were collected from weather station networks in South Korea between 33.5–37.9° N latitude. Multiple sets of weather station data were selected for cross-validation of the DNNs by standard distance deviation (SDD) among training sites. The DNNs tended to have greater spatial portability when a threshold of spatial dispersion among training sites, e.g. 190 km of SDD, was met. The final formulation of the deep solar radiation (DSR) model was obtained from training sites associated with the threshold of SDD. The DSR model had RMSE values  $<4 \text{ MJm}^{-2}\text{d}^{-1}$  at external test sites in Japan that were within  $\pm 6^\circ$  of the latitude boundary of the training sites. The relative difference between the outputs of crop yield simulations using observed versus estimated solar radiation inputs from the DSR model was about 4% at the test sites within the given boundary. These results indicate that the identification of the spatial dispersion threshold among training sites would aid the development of DNN models with reasonable spatial portability for estimation of solar radiation.

John et al., [85] developed a maximum power point tracking of a partially shaded solar photovoltaic system using a modified firefly algorithm-based controller. They modified firefly algorithm-based controller, tied operationally with a DC–DC boost converter. A model was developed and simulated on MATLAB, for tracking the maximum power point of the system, both at constant solar irradiance and at PSC.

Khamees et al., [86] evaluated three radiation Schemes of the WRF Solar model for global surface solar radiation forecast in Egypt. Dudhia, rrtmg, and Goddard schemes were used to simulate Global Horizontal Irradiance (SR) during 2017. Sixteen stations were selected to represent the different climate conditions in Egypt. The observed SR data is only available at five stations and used to evaluate the accuracy of ERA5 and WRF output. SR data at the other eleven stations were extracted from ERA5 to evaluate the corresponding values from WRF simulation. The results showed that ERA5 and WRF Dudhia scheme have reliable results as compared to the observations at the five stations. Also, the WRF Dudhia scheme simulates the SR at the other eleven stations better than rrtmg, and Goddard schemes as compared to ERA5 dataset. Where, the three WRF radiation schemes overestimate the SR along the eleven stations and underestimate at Alexandria station by  $-0.15 \text{ KWh/m}^2$ , which may due to the weather conditions over these sites. But the Dudhia scheme over the

eleven stations has the lowest MBE ( $< 0.34 \text{ KWh/m}^2$ ), RMSE ( $< 0.58 \text{ KWh/m}^2$ ), and MAPE ( $< -2 \%$ ), with  $R^2$  more than 0.985 at most stations.

A Dual-Stage model that forecast the solar power to reflects uncertainties in weather forecasts was generated by Lee et al., [87] aiming to minimize the prediction errors during solar time. The proposed method comprises two stages. The first stage was the construction of the Solar Base Model by extracting characteristics from input variables. In the second stage, the prediction error period was detected using the Solar Change Point, which measures the difference between the predicted output from the Solar Base Model and the actual power generation. The performance evaluation was restricted to July and August. The average improvement rate in predicted power generation was 24.5%. Using the proposed model, updates to weather forecast status information were implemented, leading to enhanced accuracy in predicting solar power generation.

Lu et al., [88] compared with physical models, WRF-Solar, as an excellent numerical forecasting model, includes abundant novel cloud physical and dynamical processes, which enables the high-frequency output of radiation components which are urgently needed by the solar energy industry. This study assessed the accuracy of the improved numerical weather prediction (WRF-Solar) model in simulating global and diffuse radiation. Aerosol optical properties at 550 nm, which were provided by a moderate resolution imaging spectroradiometer, were used as input to analyze the differences in accuracies obtained by the model with/without aerosol input. The sensitivity of WRF-Solar to aerosol and cloud optical properties and solar zenith angle (SZA) was also analyzed. The results show the superiority of WRF-Solar to WRF-Dudhia in terms of their root mean square error (RMSE) and mean absolute error (MAE). The coefficients of determination between WRF-Solar and WRF-Dudhia revealed no statistically significant difference, with values greater than 0.9 for the parent and nested domains. In addition, the relative RMSE (RRMSE%) reached 46.60%. The experiment on WRF-Solar and WRF-Dudhia revealed a negative bias for global radiation, but WRF-Solar attained a slightly lower RMSE and higher correlation coefficient than WRF-Dudhia. The WRF-Solar-simulated results on diffuse radiation under clear sky conditions were slightly poorer, with RMSE, RRMSE, mean percentage error and MAE of  $181.93 \text{ Wm}^{-2}$ , 170.52%, 93.04% and  $138 \text{ Wm}^{-2}$ , respectively.

A new feature selection approach for Photovoltaic power forecasting using Sequential Forward Selection (SFS) with Kernel Conditional Density Estimator (KCDE) was presented by Macaire et al., [89] as forward selection (FS) to forecast day-ahead regional PV power production in French Guiana. This method was compared to three other FS methods used in earlier studies: The Pearson correlation method, the RReliefF (RRF) method, and SFS using a linear regression. It has been shown that SFS-KCDE outperforms other FS methods, particularly for overcast sky conditions. Moreover, Wrapper methods show better forecasting performance than filter methods and should be used.

A novel multi-scale ensemble method and multi-scale ensemble neural network was presented by Qin et al., [90]. The neural network uses long short-term memory, gate recurrent units, and temporal convolutional network as the basic model. By coupling the stochastic weight averaging ensemble method and differential evolution ensemble method, these deep learning networks were assembled from single-model scale and multi-model scale, respectively, thereby effectively improving the model prediction accuracy. For predicting the power load of Hubei Province in China, meteorological features and time features were in consideration. The proposed model was trained and compared with eleven intelligent short-term load forecasting models, including machine learning, deep learning and ensemble deep learning models. Simulations show that the proposed model has the best comprehensive prediction performance. This study highlights the power of ensemble deep learning models coupled with multiple ensemble techniques and the promising prospect of our proposed model in short-term load forecasting.

Ribeiro and Fanzeres [91] present descriptive analytics of the time-linked hourly-based daily dynamics of wind speed and solar irradiance in the main resourceful regions of Brazil. They focused on identifying similar days over the years, Representative Days, that can depict the fundamental underlying behavior of each source using Leveraging on unsupervised Machine Learning methods. Their analysis was based on a historical dataset of different sites with the highest potential and installed capacity of each source spread over the country: three in the Northeast and one in the South Regions, for wind speed; and three in the Northeast and one in the Southeast Regions, for solar irradiance. They used two Partitioning Methods (K-Means and K-Medoids), the Hierarchical Ward's Method, and a Model-Based Method (Self-Organizing Maps). And identified that wind speed and solar irradiance can be effectively represented, respectively, by only two representative days, and two or three days, depending on the region and method (segments data with respect to the intensity of each source). Analysis with higher Representative Days highlighted important hidden patterns such as different wind speed modulations and solar irradiance peak-hours along the days.

Sebastian et al., [92] introduces a novel approach for site adaptation of solar irradiance based on machine learning techniques. They analyzed seven machine-learning algorithms and compared with conventional statistical approaches to identify Sweden's most accurate algorithms for site adaptation. Solar irradiance data gathered from three weather stations of SMHI were used for training and validation. The results show that machine learning can substantially improve the STRÅNG model's accuracy. However, due to the

spatiotemporal heterogeneity in model performance, no universal machine learning model can be identified, which suggests that site adaptation was a location-dependent procedure.

Sheng et al., [93] proposed a support vector machine (SVM) model based on hybrid competitive particle swarm optimization (HCPSO) with consideration of spatial correlation (SC), for realizing short-term PV power prediction tasks. Firstly, the spatial correlation analysis was conducted on the distributed PV stations. The k-means clustering method based on morphological similarity distance improvement and mutual information function was used to select the best reference station and the best delay, which generates strongly correlated solar PV sequences. Then, a hybrid algorithm of particle swarm optimization (PSO) and sine cosine algorithm (SCA) in a competitive framework (HCPSO) was proposed, aiming to fuse the fast convergence capability of PSO algorithm with the global search capability of SCA algorithm, while enabling the algorithm to effectively handle high-dimensional optimization problems based on a competitive mechanism. Finally, the HCPSO algorithm was combined with SVM algorithm, which expands the applicable scenarios of the SVM model and effectively improves the accuracy of PV short-term prediction.

Shihan et al., [94] decomposed the normalized and pre-processed raw solar PV data into different components by the quadratic frequency domain decomposition method, and they then used the random forest algorithm and the GRU prediction neural network optimized by the genetic algorithm to process the different components, and then reconstructed the results of the different prediction methods to obtain a short-term PV forecasting model. The improved quadratic frequency domain decomposition forecasting model was applied to the forecasting problem and compared with the traditional forecasting method, and the improved quadratic frequency domain decomposition forecasting model was proved to obtain more accurate forecasting results.

A theoretical model was developed to determine the Shockley–Queisser efficiency limit of solar thermophotovoltaic (STPV) cells with single- or double-junction photovoltaic (PV) cells and a simple radiation shield considering the divergence nature of concentrated solar radiation by Wen and Bhaskar [95]. A combination of adaptive parametric sweep and graphic-based methods was developed to solve the highly nonlinear correlations of energy and carrier transports in the theoretical model to find the optimized operating conditions of STPVs with high stability. The theoretical model predicts that the Shockley–Queisser efficiency limit of STPV under 1000 times solar concentration and a simple radiation shield is ~50.1% with InGaAsSb PV cells, ~49.1% with GaSb PV cells, and ~53.2% with InGaAsSb/GaSb double-junction PV cells. The operating temperatures are ~1719.5 K, ~1794.1 K, and 1640.0 K, respectively.

Zhao et al., [96] conducted a research study on solar PV power prediction using long-term monitoring data of output power, various meteorological data, and solar irradiation intensity of photovoltaic modules. The authors developed the functional relationship between the output power of photovoltaic modules and the irradiation intensity through Pearson correlation analysis. By deducing the distribution relationship of irradiation intensity, the prediction model of irradiation intensity based on peak sunshine hours and sunshine duration was constructed and based on 340 sites across the country 64 years peak sunshine hours and sunshine duration query database. And the theoretical value of the prediction model on sunny days was averagely 0.952 which is close to the measured value. The solar radiation intensity on rainy days is weak, and the prediction accuracy is low ( $R^2 = 0.884$ ). The relative errors between the sunshine duration and the peak sunshine hours in the database are less than 4.55% and 4.79%, respectively, under sunny conditions in each quarter, indicating that the accuracy of the database meets the actual needs.

Zhu et al., [97] present SL-Transformer as a novel method rooted in the deep learning paradigm tailored for green energy power forecasting. They employed the SG filter and LOF algorithm for data cleansing and incorporating the system with a self-attention mechanism, to improve the model's ability to discern and dynamically fine-tune input data weights. For solar energy forecasting, the SL-Transformer has achieved a SMAPE of 4.2156%, signifying a commendable improvement of 15% over competing models.

Li et al., [98] proposed a novel deep learning-based model named PLSTNet for ultra-short-term prediction of photovoltaic power over a 5 min time span. This model is a novel dual-path prediction. On one hand, it effectively captures short-term fluctuations in time series data by combining CNN and RNN. On the other hand, it further captures and analyzes long-term trends in fluctuations through the use of a smoothing layer and RNN's recurrent skip layer. In one-step and multi-step forecasting experiments on annual and seasonal datasets, the authors compared the performance of the PLSTNet model with LSTNet, PHILNet, TCN\_GRU, and ResCNN to assess its performance. In one-step and multi-step forecasting using the annual dataset, the MAE of the PLSTNet model is at least 15.5% lower than that of other models. Similarly, for seasonal datasets, the MAE of the PLSTNet model is at least 13.2% lower than other models.

Liu and Li [99] proposes a regional PV power forecasting model based on an improved time-series dense encoder and graph attention network (ITDE-GAT), which takes into account the spatio-temporal correlations among the regional PV plants. Firstly, an improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN) was used to extract the clear-sky and fluctuation components from solar PV data. Secondly, the combined ITDE-GAT was applied to perform the regional solar PV power forecasting. The authors constructed an improved dense encoder network (ITDE) in order to extract the temporal and spatial



relationships of regional solar PV. Graph attention network (GAT) was then utilized to explore the spatial correlations among the regional solar PV. The results demonstrate that compared to various advanced deep learning methods, the  $R^2$  evaluation metric of the approach proposed in this paper demonstrates, respectively, maximum improvements of 3.4 %, 6.5 %, and 7.8 % for the 1 h, 3 h, and 6 h ahead predictions.

Murugan et al., [100] estimated the photovoltaic models using an enhanced Henry gas solubility optimization algorithm with first-order adaptive damping Berndt-Hall-Hall-Hausman method. The designed algorithm was done by incorporating the first-order Berndt-Hall-Hall-Hausman (BHHH) numerical method, along with the non-linear damping parameter of the Levenberg-Marquardt technique (LM). By implementing this approach, the authors have significantly improved the precision and reliability of estimating the initial root parameters in solar PV models, effectively filling the theoretical void in this specific research area. Then in terms of methodology, the Enhanced Henry Gas Solubility Optimization (EHGSO) algorithm was combined with the Sine-Cosine mutualism phase of Symbiotic Organisms Search (SOS) for efficiently estimating the unknown parameters of PV models. The keystone of EHGSO in terms of methodology enhances exploration at the beginning of optimization and intensifies exploitation in later iterations. The proposed EHGSO methodology based on the adaptive damping BHHH technique (EHGSOAdBHHH) was tested on Single Diode (SD), and Double Diode (DD) solar PV models using actual experimental data. EHGSOAdBHHH exhibits outstanding accordance with attained experimental data compared with other algorithms, and its superiority was validated using several statistical criteria.

The hourly solar PV production was estimated using two models based on feedforward neural networks (FFNNs) by Nicoletti and Bevilacqua [101]. The numerical weather prediction (NWP) data: ambient temperature, relative humidity, and wind speed were used input data. They used multiple inputs in the first proposed model, while the second one used only the necessary information. It was concluded that the hourly temperature trend was the most important variable for prediction. The models' accuracy was tested using experimental and NWP data, with the second model having almost the same accuracy as the first despite using fewer input data. The results obtained using experimental data as inputs show a coefficient of determination ( $R^2$ ) of 0.95 for the hourly solar PV energy produced. The RMSE was about 6.4% of the panel peak power. When NWP data were used as inputs,  $R^2$  was 0.879 and the RMSE was 10.5%. These models can have a significant impact by enabling individual energy communities to make their forecasts, resulting in energy savings and increased self-consumed energy.

A long short-term memory (LSTM)-based multi-step prediction model to forecast and fill in missing data was presented by Wang et al., [102]. The authors proposed a federated-learning-based stacking ensemble gate recurrent unit algorithm (FL-SE-GRU) for electricity theft detection and cyberattack classification, and its effectiveness is verified by comparison with existing methods. The results of experiments show that the LSTM-based multi-step prediction model exhibits a remarkable data interpolation effect. Meanwhile, FL-SE-GRU achieves 95.0% accuracy, 96.6% precision, 93.8% sensitivity, and 95.1% F1 score in detecting electricity theft, and reaches 96.8% accuracy, 96.1% precision, 97.4% sensitivity, and 96.7% F1 score in classifying 9 kinds of cyberattacks, respectively.

#### 4. Conclusion

Solar energy technologies are green technologies that generates electricity and heat using solar PV modules and solar collectors respectively without emission of the pollution or harmful to the environment. With these applications of solar energy technology, yet it has some problems that hinder the efficiency of the system such as partial shading, mismatching of the components, inadequate monitoring and maintenance practices, unforeseen fault in solar PV cells, inefficient inverters and improper design or sizing.

Artificial Intelligence is a promising technique that employed to solve real-life problems in the areas of engineering, medicine, business and so on. In renewable energy, Artificial intelligence is used to solve problems with complex and non-linear datasets. Solar irradiance and power prediction, fault detection, solar PV parameters, MPPT modeling and so on.

This paper explored the literatures that applied artificial intelligence to solve the problems in solar energy systems. Various Artificial intelligence techniques used in solar energy systems have been reviewed. Available literature summaries published in this area is also presented. To have convenient reading, the authors explained the difficulties and contributions of different authors in the area of solar energy technology. Based on results, machine learning and deep learning are used to improve the efficiency of the solar energy systems, detection of faults in the surface of solar PV modules, estimation of model parameters. Moreover, hybrid learning techniques is more accurate and efficient for optimizing solar energy systems. Therefore, it is recommended to use hybrid Artificial Intelligence learning techniques in the future to deal with solar energy generation problems.

#### Declaration of interest

The authors declare that there is no conflict of interest.

## References

- [1] Kampa M., Castanas E. “Human health effects of air pollution”. *Environmental Pollution*, (2008), 151, 362–367. <https://doi.org/10.1016/J.ENVPOL.2007.06.012>.
- [2] Hersbach H., Bell B., Berrisford P., Hirahara S., Horányi A., Muñoz-Sabater J., Nicolas J., Peubey C. Radu R., Schepers D. et al. “The ERA5 global reanalysis”. *Q. J. R. Meteorol. Soc.*, (2020), 146, 1999–2049. <https://doi.org/10.1002/QJ.3803>.
- [3] Adib R., Zervos A., Eckhart M., David M.E.A., Kirsty H., Peter H. Governments R. Bariloche F. “Renewables 2021: Global Status Report”. In *REN21 Renewables Now*, (2021). <https://www.iea.org/reports/renewables-2021>.
- [4] Eroglu H. “Effects of Covid-19 outbreak on environment and renewable energy sector”. *Environ. Dev. Sustain.*, (2021), 23, 4782–4790. <https://doi.org/10.1007/S10668-020-00837-4/FIGURES/5>.
- [5] Xin-gang, Z.; You, Z. “Technological progress and industrial performance: A case study of solar photovoltaic industry”, *Renew. Sustain. Energy Rev.*, 81 (2018) 929–936.
- [6] Wang, H.; Lei, Z.; Zhang, X.; Zhou, B.; Peng, J. “A review of deep learning for renewable energy forecasting”, *Energy Convers. Manage.*, 198 (2019) 111799.
- [7] Mellit, A.; Benghanem, M.; Kalogirou, S. A. “Modeling and simulation of a stand-alone photovoltaic system using an adaptive artificial neural network: Proposition for a new sizing procedure”, *Renew. Energy*, 32(2) (2007), 285–313.
- [8] Das, U.K.; Tey, K.S.; Seyedmahmoudian, M.; Mekhilef, S.; Idris, M.Y.I.; van Deventer, W.; Horan, B.; Stojcevski, A. “Forecasting of photovoltaic power generation and model optimization: A review” *Renew. Sustain. Energy Rev.* 18 (2018) 912–928.
- [9] Sobri, S.; Koohi-Kamali, S.; Rahim, N. A. “Solar photovoltaic generation forecasting methods: A review”, *Energy Convers. Manage.*, 156 (2018) 459–497.
- [10] Pallathadka, H.; Ramirez-Asis, E.H.; Loli-Poma, T.P.; Kaliyaperumal, K.; Ventayen, R.J.M.; Naved, M. “Applications of artificial intelligence in business management, e-commerce and finance”, *Mater. Today Proc. In press.* (2021). <https://doi.org/10.1016/J.MATPR.2021.06.419>.
- [11] Gonçalves, J.F.; Mendes, J.J.M.; Resende, M.G.C. “A genetic algorithm for the resource constrained multi-project scheduling problem”, *European J. Oper. Res.*, 189(3) (2008) 1171–1190.
- [12] Giannakoudis, G.; Papadopoulos, A.I.; Seferlis, P.; Voutetakis, S. “Optimum design and operation under uncertainty of power systems using renewable energy sources and hydrogen storage”, *Int. J. Hydrogen Energy*, 35(3) (2010) 872–891.
- [13] Alsayed, M.; Cacciato, M.; Scarcella, G.; Scelba, G. “Multi-criteria optimal sizing of photovoltaic-wind turbine grid connected systems”, *IEEE Trans. Energy Convers.*, 28 (2) (2013) 370–379.
- [14] Daut, M.A.M.; Hassan, M.Y.; Abdullah, H.; Rahman, H.A.; Abdullah, M.P.; Hussin, F. “Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review”, *Renew. Sustain. Energy Rev.*, 70 (2017) 1108–1118.
- [15] Alshahrani, A.; Omer, S.; Su, Y.; Mohamed, E.; Alotaibi, S. “The technical challenges facing the integration of small-scale and large-scale PV systems into the grid: A critical review”, *Electronics*, 8: (2019) 1443; <https://doi.org/10.3390/ELECTRONICS8121443>.
- [16] Valer, L.R.; Manito, A.R.; Ribeiro, T.B.; Zilles, R.; Pinho, J.T. “Issues in PV systems applied to rural electrification in Brazil”. *Renew. Sustain. Energy Rev.*, 78 (2017) 1033–1043. <https://doi.org/10.1016/J.RSER.2017.05.016>.
- [17] Bosman, L.B.; Leon-Salas, W.D.; Hutzal, W.; Soto, E.A. “PV System Predictive Maintenance: Challenges, Current Approaches, and Opportunities” *Energies*, 13 (2020) 1398. <https://doi.org/10.3390/EN13061398>.
- [18] Thi, H.; Thu, N.; Quoc Bao, P. “Hourly day ahead wind speed forecasting based on a hybrid model of EEMD, CNN-Bi-LSTM embedded with GA optimization”, *Energy Rep.*, 8 (2022) 53–60.
- [19] Yang, J.X. “A novel short-term multi-input–multi-output prediction model of wind speed and wind power with LSSVM based on improved ant colony algorithm optimization”, *Cluster Comput.* 22(2) (2019) 3293–3300.
- [20] Neeraj, K.; Sudha, K.; Kusum, T. “Wind power prediction analysis by ANFIS, GA-ANFIS and PSO-ANFIS”, *J. Info. Opt. Sci.* 43(3) (2022) 481–486.
- [21] Wang, L.; Tao, R.; Hu, H.L. et al. “Effective wind power prediction using novel deep learning network: Stacked independently recurrent auto-encoder”, *Renew. Energ.*, 164(C) (2021) 642–655.

- [22] Banik, A.; Behera, C.; Sarathkumar, T.V.; Goswami, A.K. “Uncertain wind power forecasting using LSTM-based prediction interval”, *IET Renew. Power. Gener.*, 14 (2020) 2657–2667.
- [23] Semero, Y.K.; Zhang, J.H.; Zheng, D.H. “EMD-PSO-ANFIS-based hybrid approach for short-term load forecasting in microgrids”, *IJET Gener. Transm. Distrib.*, 14(3) (2022) 470–475.
- [24] Márquez, F.P.G.; Gonzalo, A.P. “A comprehensive review of Artificial Intelligence and wind energy”, *Archives of Computational Methods in Engineering*, 29 (2021) 2935-2958. <https://doi.org/10.1007/s11831-021-09678-4>
- [25] Mellit, A.; Kalogirou, S. “Artificial intelligence and internet of things to improve efficacy of diagnosis and remote sensing of solar photovoltaic systems: Challenges, recommendations and future directions”, *Renewable and Sustainable Energy Reviews*, 143 (2021) 110889. <https://doi.org/10.1016/j.rser.2021.110889>
- [26] Mellit, A.; Kalogirou, S.A.; Hontoria, L.; Shaari, S. “Artificial intelligence techniques for sizing photovoltaic systems: A review”, *Renewable and Sustainable Energy Reviews*, 13 (2009) 406-419. <https://doi.org/10.1016/j.rser.2008.01.006>
- [27] Jha, S.K.; Bilalovic, J.; Jha, A.; Patel, N.; Zhang, H. “Renewable energy: Present research and future scope of Artificial Intelligence”, *Renewable and Sustainable Energy Reviews*, 77 (2017) 297-317. <https://doi.org/10.1016/j.rser.2017.04.018>
- [28] Belu, R. “Artificial intelligence techniques for solar energy and photovoltaic applications”, in: Pour, M.K.; Clarke, S.; Jennex, M.E.; Anttiroiko, A.V.; Kamel, S.; Lee, I.; Kisielnicki, J.; Gupta, A.; Slyke, C.V.; Wang, J.; Weerakkody, V. “Robotics: Concepts, methodologies, tools, and applications”, *IGI Global*, (2014) 1662-1720. <https://doi.org/10.4018/978-1-4666-4607-0.ch081>
- [29] Shehab, M.; Abualigah, L.; Shambour, Q.; Hashem, M.A.A.; Shambour, M.K.Y.; Salibi, A.I.A.; Gandomi, A.H. “Machine learning in medical applications: A review of state-of the-art methods” *Computers in Biology and Medicine*, 145 (2022) 105458. <https://doi.org/10.1016/j.compbiomed.2022.105458>
- [30] Friedman, J.; Hastie, T.; Tibshirani, R. “The elements of statistical learning”, New York: Springer, (2001).
- [31] Intellipat. “Supervised Learning vs Unsupervised Learning vs Reinforcement Learning”, (2019). Retrieved from: <https://intellipaat.com/blog/supervised-learning-vs-unsupervised-learning-vs-reinforcement-learning/>
- [32] Klass, L. “Machine Learning: Definition and Application Examples”, Spotlight Metal, (2018). Available at: [https://www.spotlightmetal.com/machine-learning--definition-and-application-examples-a-746226/?cmp=go-ta-art-trf-SLM\\_DSA-20180820&gclid=CjwKCAjwkoz7BRBPEiwAeKw3q30qrjWJ-kiSAkfp6E6Oe\\_BxzFqk66RL3o2idJPKF1GBXIC94LgOuBoCTwMQAvD\\_BwE](https://www.spotlightmetal.com/machine-learning--definition-and-application-examples-a-746226/?cmp=go-ta-art-trf-SLM_DSA-20180820&gclid=CjwKCAjwkoz7BRBPEiwAeKw3q30qrjWJ-kiSAkfp6E6Oe_BxzFqk66RL3o2idJPKF1GBXIC94LgOuBoCTwMQAvD_BwE)
- [33] Sutton, R.; Barto, A. “Reinforcement learning: An introduction”, Cambridge: MIT press, (1998).
- [34] Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. “Deep learning for visual understanding: A review”, *Neurocomputing*, 187 (2016) 27-48. <https://doi.org/10.1016/j.neucom.2015.09.116>
- [35] Shahri, O.A.A.; Ismail, F.B.; Hannan, M.A.; Lipu, M.S.H.; Shetwi, A.Q.A.; Begum, R.A.; Muhsen, N.F.O.A.; Soujeri, E. “Solar photovoltaic energy optimization methods, challenges and issues: A comprehensive review”, *Journal of Cleaner Production*, 284 (2021) 125465. <https://doi.org/10.1016/j.jclepro.2020.125465>
- [36] Kurukuru, V.S.B.; Haque, A.; Khan, M.A.; Sahoo, S.; Malik, A.; Blaabjerg, F. “A review on Artificial Intelligence applications for grid-connected solar photovoltaic systems” *Energies*, 14 (2021) 4690. <https://doi.org/10.3390/en14154690>
- [37] Benganem, M.; Mellit, A.; Alamri S.N. “ANN-based modelling and estimation of daily global solar radiation data: A case study”, *Energy Conversion and Management*, 50 (2009) 1644–1655.
- [38] Esen, H.; Ozgen, F.; Esen, M.; Sengur, A. “Artificial neural network and wavelet neural network approaches for modelling of a solar air heater”, *Expert Systems with Applications*, 36 (2009) 11240–11248.
- [39] Fadare, D. “Modelling of solar energy potential in Nigeria using an artificial neural network model”, *Applied Energy*, 86(9) (2009) 1410–1422.
- [40] Mellit, A.; Pavan A.M. “A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy”, *Solar Energy*, 84(5) (2010) 807-821.

- [41] Sanusi, Y. K.; Abisoye, S. G.; Awodugba, A. O. "Application of Neural Networks for Predicting the Optimal Sizing Parameters of Stand-Alone Photovoltaic Systems", In *SOP Transaction on Applied Physics*, 1(1) (2014) 13-15.
- [42] Bartler, A.; Mauch, L.; Yang, B.; Reuter, M.; Stoicescu, L. "Automated detection of solar cell defects with deep learning", In proceedings of the 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September (2018) 2035–2039. <https://doi.org/10.23919/EUSIPCO.2018.8553025>.
- [43] Deitsch, S.; Buerhop-Lutz, C.; Sovetkin, E.; Steland, A.; Maier, A.; Gallwitz, F.; Riess, C. "Segmentation of Photovoltaic Module Cells in Un-calibrated Electroluminescence Images", *Mach. Vis. Appl.*, 32 (2018); <https://doi.org/10.1007/s00138-021-01191-9>.
- [44] Duman, S.; Yorukeren, N.; Altas, I.H. "A novel MPPT algorithm based on optimized artificial neural network by using FPSOGSA for standalone photovoltaic energy systems", *Neural Comput. Appl.*, 29 (2018) 257–278. <https://doi.org/10.1007/S00521-016-2447-9>.
- [45] Farayola, A.M.; Hasan, A.N.; Ali, A. "Efficient photovoltaic MPPT system using coarse gaussian support vector machine and artificial neural network techniques", *Int. J. Innov. Comput. Inf. Control.*, 14 (2018) 323–339. <https://doi.org/10.24507/IJICIC.14.01.323>.
- [46] Kecman, V. "Support Vector Machines: An Introduction", Springer: Berlin/Heidelberg, Germany; 177 (2005) 605. <https://doi.org/10.1007/109846971>.
- [47] Karimi, A.M.; Fada, J.S.; Liu, J.; Braid, J.L.; Koyuturk, M.; French, R.H. "Feature Extraction, Supervised and Unsupervised Machine Learning Classification of PV Cell Electroluminescence Images", In Proceedings of the IEEE 7<sup>th</sup> World Conference on Photovoltaic Energy Conversion, WCPEC 2018-A Joint Conference of 45<sup>th</sup> IEEE PVSC, 28<sup>th</sup> PVSEC and 34<sup>th</sup> EU PVSEC, 2018, Waikoloa, HI, USA, 10–15 June (2018) 418–424. <https://doi.org/10.1109/PVSC.2018.8547739>
- [48] Akram, M.W.; Li, G.; Jin, Y.; Chen, X.; Zhu, C.; Zhao, X.; Khaliq, A.; Faheem, M.; Ahmad, A. "CNN based automatic detection of photovoltaic cell defects in electroluminescence images", *Energy*, 189 (2019) 116319. <https://doi.org/10.1016/J.ENERGY.2019.116319>.
- [49] Abdel-Nasser, M.; Mahmoud, K. "Accurate photovoltaic power forecasting models using deep LSTM-RNN", *Neural Comput. Appl.*, 31 (2019) 2727–2740. <https://doi.org/10.1007/s00521-017-3225-z>.
- [50] Karimi, A.M.; Fada, J.S.; Hossain, M.A.; Yang, S.; Peshek, T.J.; Braid, J.L.; French, R.H. "Automated Pipeline for Photovoltaic Module Electroluminescence Image Processing and Degradation Feature Classification", *IEEE J. Photovoltaics*, 9 (2019) 1324–1335. <https://doi.org/10.1109/JPHOTOV.2019.2920732>.
- [51] Torres, J.F.; Troncoso, A.; Koprinska, I.; Wang, Z.; Martínez-Álvarez, F. "Big data solar power forecasting based on deep learning and multiple data sources", *Expert Syst.*, 36 (2019); <https://doi.org/10.1111/EXSY.12394>.
- [52] Yao, X.; Wang, Z.; Zhang, H. "A novel photovoltaic power forecasting model based on echo state network", *Neurocomputing*, 325 (2019) 182–189. <https://doi.org/10.1016/J.NEUCOM.2018.10.022>.
- [53] Gallicchio, C.; Micheli, A. "Deep Echo State Network (Deep ESN): A Brief Survey", *arXiv*, arXiv:abs/1712.04323, (2017).
- [54] Hinton, G.E. "A Practical Guide to Training Restricted Boltzmann Machines", In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, (2012) 7700 Lecture N: 599–619. [https://doi.org/10.1007/978-3-642-35289-8\\_32](https://doi.org/10.1007/978-3-642-35289-8_32).
- [55] Jolliffe, I.T. "Principal Component Analysis", Springer: Berlin/Heidelberg, Germany, (2002). <https://doi.org/10.1007/B98835>.
- [56] Davidon, W. "Variable Metric Method for Minimization", Technical Report, Argonne National Laboratory (ANL): Lemont, IL, USA, (1959); <https://doi.org/10.2172/4252678>.
- [57] Al-Majidi, S.D.; Abbod, M.F.; Al-Raweshidy, H.S. "A particle swarm optimisation-trained feedforward neural network for predicting the Maximum Power Point of a photovoltaic array", *Eng. Appl. Artif. Intell.*, 92 (2020); <https://doi.org/10.1016/J.ENGAPAI.2020.103688>.
- [58] Balzategui, J.; Eciolaza, L.; Arana-Arexolaleiba, N. "Defect detection on Polycrystalline solar cells using Electroluminescence and Fully Convolutional Neural Networks", In Proceedings of the 2020 IEEE/SICE International Symposium on System Integration, SII (2020) Honolulu, HI, USA, 12–15 January 2020: 949–953. <https://doi.org/10.1109/SII46433.2020.9026211>.
- [59] Mathias, N.; Shaikh, F.; Thakur, C.; Shetty, S.; Dumane, P.; Chavan, D.S. "Detection of Micro-Cracks in Electroluminescence Images of Photovoltaic Modules", *SSRN Electron. J.*, (2020). <https://doi.org/10.2139/SSRN.3563821>.

- [60] Niu, D.; Wang, K.; Sun, L.; Wu, J.; Xu, X. “Short-term photovoltaic power generation forecasting based on random forest feature selection and CEEMD: A case study”, *Appl. Soft Comput. J.*, 93 (2020); <https://doi.org/10.1016/J.ASOC.2020.106389>.
- [61] Breiman, L. “Random Forests”, *Machine Learning*, 45 (2001) 5–32. <https://doi.org/10.1023/A:1010933404324>.
- [62] Yeh, J.R.; Shieh, J.S.; Huang, N.E. “Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method” *Adv. Adapt. Data Anal.*, 2 (2011) 135–156. <https://doi.org/10.1142/S1793536910000422>.
- [63] Li, H.; Tan, Q. “A BP neural network based on improved particle swarm optimization and its application in reliability forecasting”, *Res. J. Appl. Sci. Eng. Technol.*, 6 (2013) 1246–1251. <https://doi.org/10.19026/RJASET.6.3939>.
- [64] Jiao, B.; Lian, Z.; Gu, X. “A dynamic inertia weight particle swarm optimization algorithm”, *Chaos Solitons Fractals*, 37 (2008) 698–705. <https://doi.org/10.1016/J.CHAOS.2006.09.063>.
- [65] Wen, Y.; AlHakeem, D.; Mandal, P.; Chakraborty, S.; Wu, Y.K.; Senjyu, T.; Paudyal, S.; Tseng, T.L. “Performance Evaluation of Probabilistic Methods Based on Bootstrap and Quantile Regression to Quantify PV Power Point Forecast Uncertainty”, *IEEE Trans. Neural Network Learn. Syst.*, 31 (2020) 1134–1144. <https://doi.org/10.1109/TNNLS.2019.2918795>.
- [66] Zhang, T.; Lv, C.; Ma, F.; Zhao, K.; Wang, H.; O’Hare, G.M. “A photovoltaic power forecasting model based on dendritic neuron networks with the aid of wavelet transform”, *Neurocomputing*, 397 (2020) 438–446. <https://doi.org/10.1016/J.NEUCOM.2019.08.105>.
- [67] Jiang, T.; Wang, D.; Ji, J.; Todo, Y.; Gao, S. “Single dendritic neuron with nonlinear computation capacity: A case study on XOR problem”. In Proceedings of 2015 IEEE International Conference on Progress in Informatics and Computing, PIC (2015), Nanjing, China, 18–20 December 2016: 20–24. <https://doi.org/10.1109/PIC.2015.7489802>.
- [68] Demirci, M.Y.; Besli, N.; Gümüşçü, A. “Efficient deep feature extraction and classification for identifying defective photovoltaic module cells in Electroluminescence images”, *Expert Syst. Appl.*, 175 (2021). <https://doi.org/10.1016/j.eswa.2021.114810>.
- [69] Sattar, M.A.E.; Sumaiti, A.A.; Ali, H.; Diab, A.A. “Marine predators’ algorithm for parameters estimation of photovoltaic modules considering various weather conditions”, *Neural Comput. Appl.*, 33 (2021) 11799–11819. <https://doi.org/10.1007/S00521-021-05822-0>.
- [70] Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. “Marine Predators Algorithm: A nature-inspired metaheuristic”, *Expert Syst. Appl.*, 152 (2020) 113377; <https://doi.org/10.1016/J.ESWA.2020.113377>.
- [71] Sibtain, M.; Xianshan, L.; Snoober, S.; Qurat-Ul-Ain; Muhammad, S. A.; Touseef, T.; Halit, A. “Multistage Hybrid Model ICEEMDAN-SE-VMD-RDPG for a Multivariate Solar Irradiance Forecasting”, *IEEE Access*, 9 (2021) 37334–37363; <https://doi.org/10.1109/ACCESS.2021.3062764>
- [72] Su, B.; Chen, H.; Chen, P.; Bian, G.; Liu, K.; Liu, W. “Deep Learning-Based Solar-Cell Manufacturing Defect Detection with Complementary Attention Network”, *IEEE Trans. Ind. Inform.*, 17 (2021) 4084–4095; <https://doi.org/10.1109/TII.2020.3008021>.
- [73] Girshick, R. “Fast R-CNN”, In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December (2015) 1440–1448; <https://doi.org/10.1109/ICCV.2015.169>.
- [74] Lee, J. Y.; Shin, D. H. “Development of Solar Power Generation Prediction System using Artificial Intelligence”, In Proceedings of the First Australian International Conference on Industrial Engineering and Operations Management, Sydney, Australia, December 20-21, (2022) 1649-1654
- [75] Martinez Lopez, V.A.; Žindžiute, U.; Ziar, H.; Zeman, M.; Isabella, O. “Study on the Effect of Irradiance Variability on the Efficiency of the Perturb-and-Observe Maximum Power Point Tracking Algorithm” *Energies*, 15 (2022) 7562; <https://doi.org/10.3390/en15207562>
- [76] Smail, C.; Saad, M.; El Hammoumi, A.; Aissa, C.; Abou Soufiane, B.; El Ghzizal, A.; Aziz, D.; Mohamed, A.; Askar, S. S. “A novel hybrid GWO–PSO-based maximum power point tracking for photovoltaic systems operating under partial shading conditions”, *Scientific Reports*, 12 (2022) 10637; <https://doi.org/10.1038/s41598-022-14733-6>
- [77] Sumathi, S.; Abitha, S. “A Novel Method for Solar Water Pumping System Using Machine Learning Techniques”, In Third International Conference on Advances in Physical Sciences and Materials: ICAPSM 2022, AIP Conf. Proc. 2901, 070001-1–070001-11; <https://doi.org/10.1063/5.0179061>

- [78] Sangsang, S.; Sasmowiyono, S.; Fadlil, A.; Subrata, A. C. “Optimum solar energy harvesting system using artificial intelligence”, *ETRI Journal*, (2022) 1–11; <https://doi.org/10.4218/etrij.2022-0184>.
- [79] Abubakar, A.; Jibril, M.M.; Almeida, C.F.M.; Gemignani, M.; Yahya, M.N.; Abba, S.I. “A Novel Hybrid Optimization Approach for Fault Detection in Photovoltaic Arrays and Inverters Using AI and Statistical Learning Techniques: A Focus on Sustainable Environment”, *Processes*, 11 (2023) 2549; <https://doi.org/10.3390/pr11092549>
- [80] Alba, A.; Anders, V. L.; Miro, Z.; Hesani, Z.; Olindo, I. “Effect of Climate on Photovoltaic Yield Prediction Using Machine Learning Models”, *Global Challenges*, 7 (2023) 2200166; <https://doi.org/10.1002/gch2.202200166>
- [81] Arash, M.; Hamed, M.; Behnam, M. I.; António, P. A.; Amjad, A. M.; Zulkurnain, A. “Generalized global solar radiation forecasting model via cyber-secure deep federated learning”, *Environmental Science and Pollution Research*, (2023); <https://doi.org/10.1007/s11356-023-30224-1> 2023
- [82] Azad, M.A.; Sajid, I.; Lu, S.-D.; Sarwar, A.; Tariq, M.; Ahmad, S.; Liu, H.-D.; Lin, C.-H.; Mahmoud, H.A. “Energy Valley Optimizer (EVO) for Tracking the Global Maximum Power Point in a Solar PV System under Shading”, *Processes*, 11 (2023) 2986; <https://doi.org/10.3390/pr11102986>
- [83] Bo, L.; Zhan, S.; Zhihua, Y.; Xiuzhu, W. “Wavelet neural network-based distributed photovoltaic grid-connected power prediction method”, In Proc. SPIE 12788, Second International Conference on Energy, Power, and Electrical Technology (ICEPET 2023), 1278825 (25 September 2023), (2023); <https://doi.org/10.1117/12.3004356>
- [84] Dae, G. K.; Yean-Uk, K.; Shinwoo, H.; Kwang, S. K.; Junhwan, K.; Chung-Kuen, L.; Atsushi, M.; Robert, M. B.; David, H. F. “Identification of a spatial distribution threshold for the development of a solar radiation model using deep neural networks”, *Environ. Res. Lett.*, 18 (2023) 104020; <https://doi.org/10.1088/1748-9326/acf6d4>
- [85] John, I. S.; Abubakar, A.; Gbenga A. O. “Maximum power point tracking of a partially shaded solar photovoltaic system using a modified firefly algorithm-based controller”, *Journal of Electrical Systems and Inf. Technol.*, 10(48) (2023) 1-16; <https://doi.org/10.1186/s43067-023-00114-0>
- [86] Khamees, A.S.; Sayad, T.; Morsy, M.; Ali Rahoma, U.; Hassan, A.H. “Evaluation of three radiation Schemes of the WRF-Solar model for global surface solar radiation forecast: A case study in Egypt”, *Advances in Space Research*, (2023); <https://doi.org/10.1016/j.asr.2023.12.010>
- [87] Lee, J.; Choi, J.; Park, W.; Lee, I. “A Dual-Stage Solar Power Prediction Model That Reflects Uncertainties in Weather Forecasts”, *Energies*, 16 (2023) 7321; <https://doi.org/10.3390/en16217321>
- [88] Lu, Y. B.; Wang, L. C.; Zhou, J.J.; Niu, Z.G.; Zhang, M.; Qin, W.M. “Assessment of the high-resolution estimations of global and diffuse solar radiation using WRF-Solar”, *Advances in Climate Change Research*, 14 (2023) 720-731. <https://doi.org/10.1016/j.accre.2023.09.009>
- [89] Macaire, J.; Zermani, S.; Linguet, L. “New Feature Selection Approach for Photovoltaic Power Forecasting Using KCDE”, *Energies*, 16 (2023) 6842. <https://doi.org/10.3390/en16196842>
- [90] Qin, S.; Li, M.; Guanjuan, L.; Jianzhong, Z.; Yongchuan, Z.; Pinan, R. “Short-Term Load Forecasting Based on Multi-Scale Ensemble Deep Learning Neural Network”, *IEEE Access*, 11 (2023) 111963-111975, <https://doi.org/10.1109/ACCESS.2023.3322167>
- [91] Ribeiro, R.; Fanzeres, B. “Identifying representative days of solar irradiance and wind speed in Brazil using machine learning techniques”, *Energy and AI*, (2023). <https://doi.org/10.1016/j.egyai.2023.100320>.
- [92] Sebastian, Z.; Dazhi, Y.; Tomas, L.; Pietro, E. C. “Site adaptation with machine learning for a Northern Europe gridded global solar irradiance product”, *Energy and AI*, (2023). <https://doi.org/10.1016/j.egyai.2023.100331>
- [93] Sheng, W.; Li, R.; Shi, L.; Lu, T. “Distributed photovoltaic short-term power forecasting using hybrid competitive particle swarm optimization support vector machines based on spatial correlation analysis”, *IET Renew. Power Gener.*, (2023) 1–14. <https://doi.org/10.1049/rpg2.12860>
- [94] Shihan, L.; Chun, Y.; Yuan, G. “Short-term photovoltaic power prediction model based on quadratic frequency domain decomposition algorithm for neural networks”, In Proc. SPIE 12788, Second International Conference on Energy, Power, and Electrical Technology (ICEPET 2023), 1278811 (25 September 2023); <https://doi.org/10.1117/12.3004418>
- [95] Wen, S.-B.; Bhaskar, A. “The Shockley–Queisser Efficiency Limit of Solar Thermophotovoltaic (STPV) Cells using Different Photovoltaic Cells and a Radiation Shield Considering the Étendue of Solar Radiation”, *Energies*, 16 (2023) 7085. <https://doi.org/10.3390/en16207085>

- [96] Zhao, H.; Zhu, D.; Yang, Y.; Li, Q.; Zhang, E. "Study on photovoltaic power forecasting model based on peak sunshine hours and sunshine duration", *Energy Sci. Eng.*, (2023) 1-11. <https://doi.org/10.1002/ese3.1598>
- [97] Zhu, J.; Zhao, Z.; Zheng, X.; An, Z.; Guo, Q.; Li, Z.; Sun, J.; Guo, Y. "Time-Series Power Forecasting for Wind and Solar Energy Based on the SL-Transformer"; *Energies*, 16 (2023) 7610. <https://doi.org/10.3390/en16227610>
- [98] Li, G.; Ding, C.; Zhao, N.; Wei, J.; Guo, Y.; Meng, C.; Huang, K.; Zhu, R. "Research on a novel photovoltaic power forecasting model based on parallel long and short-term time series network", *Energy*, 293 (2024) 130621. <https://doi.org/10.1016/j.energy.2024.130621>
- [99] Liu, J.; Li, T. "Multi-step power forecasting for regional photovoltaic plants based on ITDE-GAT model", *Energy*, 293 (2024) 130468. <https://doi.org/10.1016/j.energy.2024.130468>
- [100] Murugan, R.; Arunachalam, S.; Hussein, M. R.; Seyedali, M. "Estimation of photovoltaic models using an enhanced Henry gas solubility optimization algorithm with first-order adaptive damping Berndt-Hall-Hausman method", *Energy Conversion and Management*, 299 (2024) 117831, <https://doi.org/10.1016/j.enconman.2023.117831>
- [101] Nicoletti, F.; Bevilacqua, P. "Hourly Photovoltaic Production Prediction using Numerical Weather Data and Neural Networks for Solar Energy Decision Support", *Energies*, 17 (2024) 466. <https://doi.org/10.3390/en17020466>
- [102] Wang, J.; Si, Y.; Zhu, Y.; Zhang, K.; Yin, S.; Liu, B. "Cyberattack detection for electricity theft in smart grids via stacking ensemble GRU optimization algorithm using federated learning framework", *Electrical Power and Energy Systems*, 157 (2024) 109848; <https://doi.org/10.1016/j.ijepes.2024.109848>

# Artificial Neural Network Parameter Optimization: Improving Meteorological Data Predictions through Machine Learning

Ceyhun Kapucu <sup>1,\*</sup> , Oğuz Akpolat <sup>2</sup> 

<sup>1</sup> Mugla Sitki Kocman University, Department of Informatics, Mugla, Turkey

<sup>2</sup> Mugla Sitki Kocman University, Science Faculty, Department of Chemistry, Mugla, Turkey

## Abstract

This study aims to create an artificial neural network (ANN) based model to predict solar irradiance using open-sourced meteorological data. A neural network that is feed-forward with backpropagation was employed to build the model. A large combination of model parameters including learning algorithms, transfer functions, number of hidden layers, and neurons was used to customize the neural network. The data used in this study is a part of the publicly available dataset containing real outdoor measurements provided by The National Renewable Energy Laboratory (NREL). The proposed model has been validated by measuring prediction errors using normalized mean squared error (NMSE) and prediction accuracies using regression value (R). The lowest value of the NMSE error was obtained with a neural network model based on three hidden layers employing 40, 8, and 5 neurons respectively. The R-value of this model was the highest among all models. The results have shown that the ascending/descending distribution of neurons in hidden layers is an important factor among other parameters.

**Keywords:** Artificial neural network; solar irradiance; meteorological modeling; curve fitting

## 1. Introduction

Today, the reserves of fossil fuels, which are used as the main energy source, are gradually decreasing. However, considering the continuous increase in energy demand and the harm of fossil fuels to the environment all over the world, the tendency towards alternative energy sources becomes increasingly important. Renewable energy is the best alternative energy source for fossil fuels thanks to its limitless resources, environmental friendliness and increasing efficiency with developing the technology. Solar energy is one of the most widely used renewable energy sources. Determining the solar energy potential is important for the proper design of solar applications. Therefore, many countries have measuring stations to measure meteorological data including solar irradiance. It is essential to develop methods for estimating solar irradiance from meteorological data available in areas where measurement stations are not available, or at intervals not measured.

Making a mathematical model of a system is one of the best methods to describe how it functions. It might be required to incorporate experimental or simulation data from the system to be modeled while building a mathematical model. Usually, these data are in the form of numerical quantities and can be visually represented as data points. These data points are essentially independent variables within the system and dependent variables affected by these variables. According to independent variables, the function capable of supplying the dependent variable or variables correctly is the target function. In other words, the target function is a parametric function that will define the closest possible points to all data points. This function defines the relationship between the data and the physical process or actual system represented in the background. The model thus obtained can be used to understand the responses of the system and/or to predict data that has not yet been measured. In this way, choosing a suitable function or selecting the most suitable parameters for a basic function requires a good understanding of the system targeted for the model.

If the target function is indicated as  $f(x) = y$ , here,  $x$  is the independent variable and  $y$  is the dependent variable and  $f$  is the target function. However, it may not always be possible to reach the perfect target function that defines all these variables continuously. This is a common problem in data analysis. This problem is known as the "Curve Fitting" problem, which can be explained as determining the function closest to the target function or searching for new functions to simplify the calculations by switching the functions that are difficult to use. The function obtained while trying to reach the target function is called a hypothesis or hypothesis function. When the hypothesis function is shown as  $h(x) = y'$ ,  $x$  is the independent variable,  $y'$  is the prediction, and  $h$  is the hypothesis.

As computer technology advances, the use of artificial intelligence techniques in curve fitting is gradually increasing compared to traditional iterative approaches. These techniques include artificial neural networks (ANN), genetic algorithm (GA), and fuzzy logic (FL). ANN in curve fitting has been a hot research topic in the literature with its nonlinear nature, flexible structure, self-adaptability, ability to work with missing data

\*Corresponding author

E-mail address: ceyhun@mu.edu.tr



and ease of use.

Sözen et al. proposed a study that is mapping Turkey's solar potential by using ANNs [1]. The studied ANN models differ from each other in terms of the number of hidden layers, the number of neurons in these layers, and used training algorithms. They employed the logistic sigmoid transfer function in ANN structures together with the Levenberg-Marquardt (LM), scaled conjugate gradient (SCG), and Polak-Ribière conjugate gradient (CGP) algorithms. They chose to use latitude, longitude, altitude, month, average sunshine duration, and average temperature features for the input layer of the ANN models. They showed that for the estimate of solar radiation, the ANN-based estimation approach is preferable to the traditional regression models put forward in the literature. According to their results, a trained ANN model seems with the potential for estimating solar radiation even in regions not having monitoring stations established.

Şenkal and Kuleli proposed a similar study for the estimation of solar radiation with the help of ANN [2]. In their study, they used different learning algorithms including resilient propagation (RP), the SCG learning algorithms to train ANN models. They used latitude, longitude, altitude, month, average diffuse radiation, and average beam radiation features in the input layer of the ANN models. The dataset in the study comes from twelve stations in twelve different cities in Turkey. Data from stations in nine cities were used to train ANN, while the rest were used to test. They suggested that building solar databases and estimating solar radiation may be done affordably and efficiently by employing ANN.

In order to estimate the solar radiation for the Mediterranean area of Anatolia, Turkey, Koca et al. employed ANN in their work [3]. Data on solar radiation from two cities were used to train the proposed ANN model, while data from five cities were utilized to evaluate the model. ANN models are initialized with feature sets including different numbers of input parameters. Feature sets are formed by selecting from latitude, longitude, altitude, month, average of cloudiness and sunshine duration parameters. The number of parameters in a feature set changes from four to six. They claimed that the most important component in the calculation of solar radiation was the quantity of input parameters.

Wang et al. proposed a short-term solar irradiance prediction model using ANN [4]. They created several neural networks with different structures and then determined the most suitable network by comparing all the models with cross-validation. The models differ from each other in terms of having single or double hidden layers, and by the number of neurons in the hidden layer(s). The solar radiation data from The National Renewable Energy Laboratory (NREL) is used for training and testing the prediction model [5]. The proposed prediction model employs a multi-layered feed-forward neural network with backpropagation. They reported that the model with double hidden layers containing 18-13 neurons respectively was the most successful one.

Ozgoren et al. suggested a research to create an ANN model for calculating the sun radiation of any location in Turkey using a multi-nonlinear regression technique [6]. Using the various combinations of neural network inputs, they produced 10 distinct feed-forward back-propagation neural network models. The input parameters were selected using the stepwise multiple regression analysis. The meteorological data used in the study was collected in 2000-2006 years, from 31 different stations spread throughout Turkey. While the data from 27 stations were used for training, the rest were used for the test. After selecting the most successful model, trial and error procedure was used to determine the best network architecture depending on the number of neurons in the hidden layer. They then decided to use a network consisting of one input layer with 10 input parameters and one hidden layer with 10 neurons, as the final model to estimate the monthly global solar radiation at any location in Turkey.

Renno et al. used ANN in their study to develop a tool in order to estimate the solar energy potential of the University of Salerno [7]. They have investigated two different ANN models to predict the daily global radiation and the hourly direct normal irradiance. The first ANN model for predicting the daily global radiation consists of one hidden layer with ten neurons. The second model for predicting the hourly direct normal irradiance has one hidden layer with five neurons. Both models have a sigmoid transfer function for hidden layers and a linear function for the output layer. The models have been used with feature sets including different numbers of input parameters. They stated that the created ANN models might serve as an effective instrument for evaluating a cleaner energy system, guaranteeing an accurate assessment of the solar potential for various locations.

Bou-Rabee et al. proposed an ANN-based model to forecast for the daily average solar radiation in Kuwait by using average radiation data collected for five years in a row from five different locations in Kuwait [8]. They used a multi-layered feed-forward neural network with the back-propagation learning method. After normalizing the collected data, years 2007-2010 data were used for training the proposed model. To validate the proposed model, the year 2011 data was used. The neural network design utilized in the model was determined after comprehensive testing of several potential configurations. It comprises an input layer with four parameters and a hidden layer with ten neurons. According to the researchers, Kuwait's solar radiation may be accurately predicted using the forecasting model they devised.

Xue, in his study, used an ANN technique to predict daily diffuse solar radiation [9]. The back-propagation neural network model's efficiency and capacity for generalization were enhanced by the suggested study with

the use of two optimization techniques: particle swarm optimization and genetic algorithm. Seven input parameters including the month of the year, duration of sunshine, average temperature, rainfall, relative humidity, wind speed, and daily global solar radiation were used to predict the daily diffuse solar radiation as output. He reported that the performance of the neural network model optimized by particle swarm optimization is better than the plain neural network and the neural network model optimized by genetic algorithm.

Rodríguez et al. proposed an ANN model to forecast the amount of solar energy generated by photovoltaic units [10]. After data analysis, a vector of 146 input values were given as training input to the network. These values consisted of the season of the day, the time of day, and the irradiation values in 10 minute intervals for the last 24 hours that were the remaining 144 input values. The single output of the ANN was the predicted irradiation value. According to the researchers, the suggested method's accuracy was high enough to be implemented in systems with built-in solar generators.

This research presents the development of an ANN that can model and forecast the intensity of solar radiation utilizing independent variables such as air temperature, relative humidity, atmospheric pressure, cumulative daily total precipitation, and exact time of measurement. MATLAB [11] software was used to generate the produced ANN, and the parameters pertaining to the solution approach were tuned.

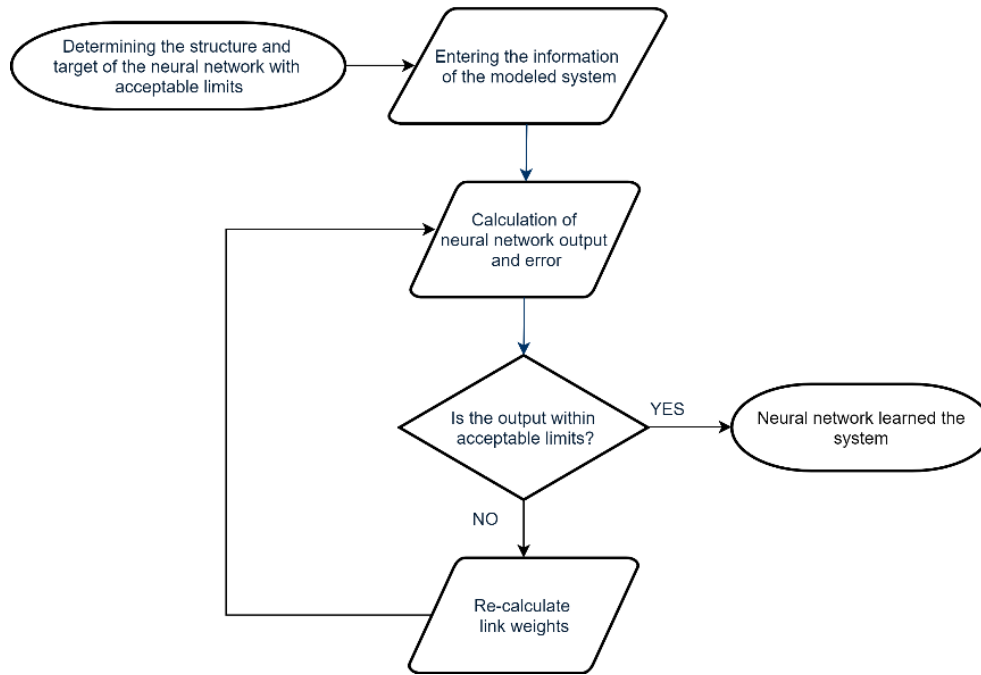
## 2. Artificial Neural Network

ANN is a distributed, parallel information processing architecture, which consists of interconnected processing units, each with its memory and inspired by the human brain [12]. This structure consists of a combination of artificial nerve cells –neurons– obtained by mimicking the ways in which biological nerve cells work. Artificial neurons are completely nature-inspired units and information systems that can mimic the learning ability of the human brain. The network structure formed by the connection of neurons to each other is called a neuron network or neural network [13]. The binding of neurons to each other can be likened to the binding of biological neurons to each other via synapses. Data from the external environment or from the outputs of other neurons are applied to the inputs of the next linked neuron. These inputs are then processed by the neuron and then the output is exported as the neuron output or as an input for another neuron.

ANN has the ability to learn and generalize. The ability to generalize can be summarized as the ability of a trained ANN to make accurate predictions against data it has never encountered before. An ANN consists of input, output, and intermediate layers. There are independent variables in an input layer. The output layer includes dependent variables or variables that are expected to be predicted. Increasing the number of intermediate layers helps to solve more complex problems but makes it difficult to train the network.

The neural network is successful in modeling, classifying, estimating and finding the most appropriate value. Conventional computers are faster and more successful in precise arithmetic operations and calculations based on a particular algorithm. However, ANN is successful in data that contains noise or missing parts, unlike traditional computers. While traditional computers perform only the tasks they are programmed to, the ANN does not need to be programmed. ANN can learn the instructions itself. While conventional computers store information in a specific location in memory, ANN stores information in a distributed manner throughout the network, in fact the information is distributed over the connections and weights of the network.

The objective of an ANN is to learn the system that is being modeled. ANN generally performs the learning process as in the flow chart shown in Figure 1. As shown in the figure, it is necessary to present the data of this system to the neural network and compare the response of the neural network with the expected response. The difference is used by a cost function to update the weights of all connections in the network. The learning algorithm recalculates and updates the weights in each cycle. These cycles continue until the neural network's response and the predicted response diverge less than acceptable bounds.



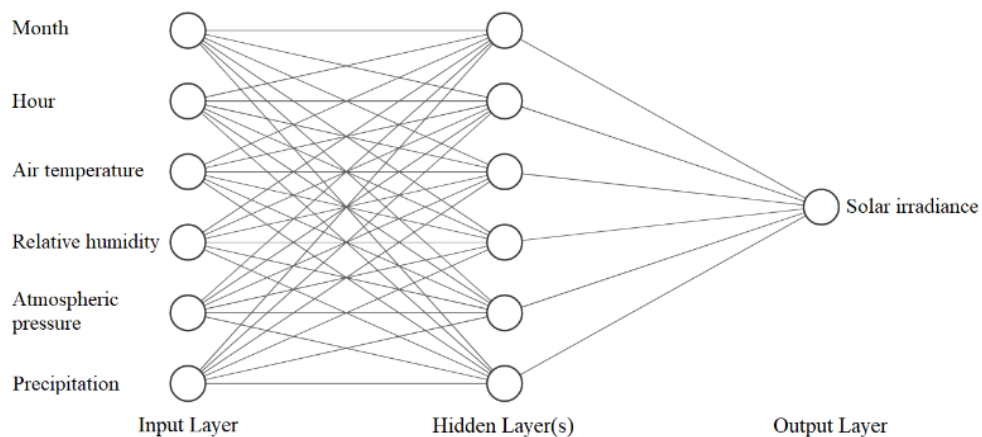
**Figure 1.** Learning process of a neural network.

### 3. Materials and Methods

#### 3.1. Dataset

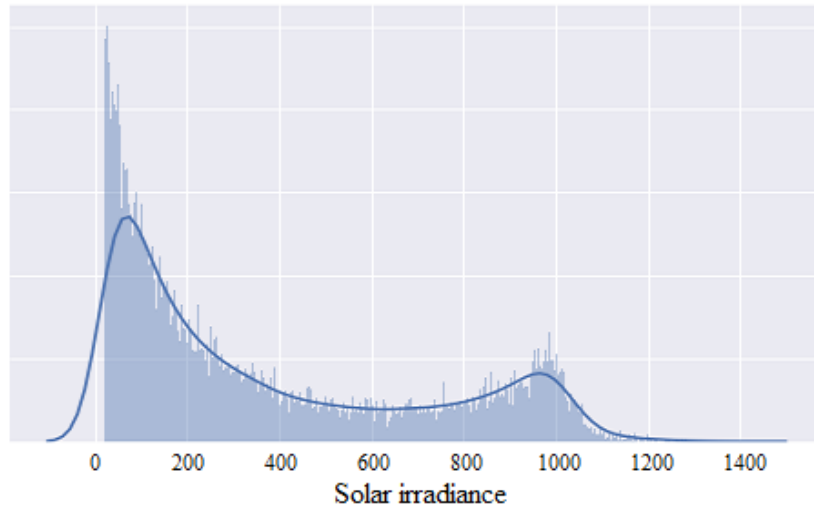
The raw data set used in this study contains meteorological data as well as electrical measurement data provided by NREL [14], [15]. Since the main purpose of the study was to model solar irradiance using some meteorological data, electrical data in the raw data set were not used. The meteorological data used are solar irradiance, air temperature, relative humidity, atmospheric pressure and total daily precipitation. In addition to these data, the month and hour of the measurement were used.

The ANN model to be developed within the scope of the study was used to estimate the intensity of solar irradiance using other data. In terms of the curve-fitting problem, the solar irradiance intensity is the output of the ANN while the other data are used as the inputs of the ANN as shown in **Figure 2**.



**Figure 2.** Inputs and output for the target ANN structure.

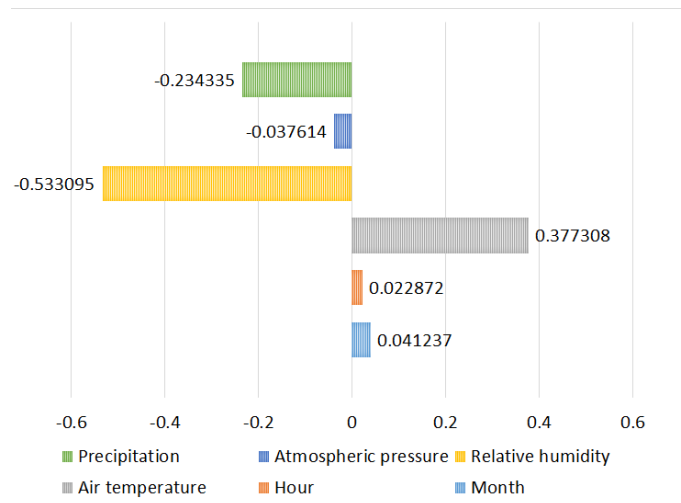
The timestamp of the measurement is a text-type statement and it is stored in the raw format of year-month-day, hour : minute : second. To use in the input variables, this raw string needs to be converted to numerical values. For this reason, the month and hour values in the timestamp were extracted and normalized. Month value varies between one and twelve, while hour value is between zero and one. For example, the hour of a measurement taken at 12:00 p.m. is assumed to be 0.5 after normalization.



**Figure 3.** Distribution of the output variable.

**Figure 3.** shows the solar irradiance values' distribution in the dataset. The minimum and the maximum values for the solar irradiance data are 20W/m<sup>2</sup> and 1377.9W/m<sup>2</sup> respectively. As shown in the figure, there is a non-normal distribution in the solar irradiance data. Because of the non-normal distribution, the Spearman test has applied to see the correlations between the input variables and solar irradiance.

**Figure 4.** shows the correlation coefficients obtained from the Spearman correlation test, between all of the input variables and the solar irradiance output variable. As shown in the figure, there are significant positive and negative correlations between some of the input variables and solar irradiance.



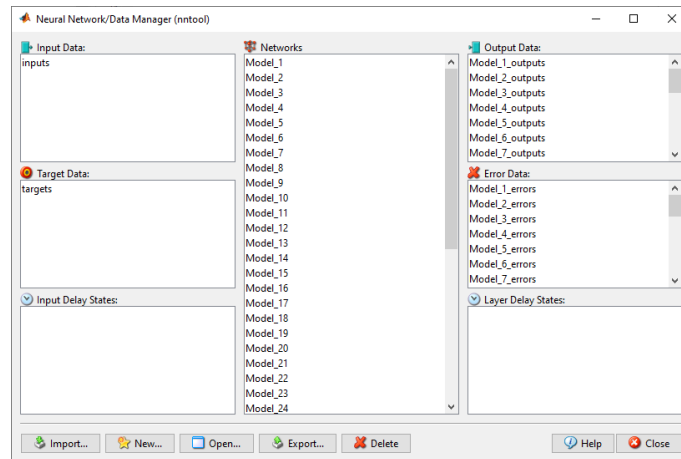
**Figure 4.** Correlation coefficients between input variables and output.

### 3.2. Creating and Training ANN Models

ANN models to be used in the study are distinguished from each other by parameters such as training algorithms, hidden layer numbers, number of neurons in the hidden layers, and transfer functions used in hidden layer neurons. However, some parameters were set the same in all ANN models. These parameters, which do not change according to the ANN model, are the type of neural network, the transfer function used in neurons in the output layer, and the distinction between training, validation, and test data set size. All ANN models were created in the type of feedforward and backpropagation neural networks and the linear transfer function was used in the output layer. This function is called "purelin" in MATLAB. Transfer functions are functions that calculate the output using the plain input of a neuron [16]. In addition, 70% of the data set was utilized for training, 15% for validation and the rest 15% for testing in all models.

Different ANN models have been created by using hyperbolic tangent sigmoid transfer function (*tansig*), and logarithmic sigmoid transfer function (*logsig*) in the hidden layers. This is the first parameter (*Parameter*

1) used for the optimization of the models. However, the number of hidden layers (*Parameter 2*) of the ANN models, the number of the neurons (*Parameter 3*) in these layers and the training algorithms (*Parameter 4*) are the other optimization parameters used to create different ANN models. With these parameters, 51 different ANN models were created. The neural network/data management tool (*Matlab nntool*), which provides more detailed optimization options than the curve fitting tool (*Matlab nftool*), was used to create these models. Prediction outputs and errors that were obtained after training of all ANN models are listed in the rightmost pane on the neural network/data management tool screen shown in **Figure 5**.



**Figure 5.** Creating and Training ANN Models.

In the study, regression R-value and the normalized mean squared error (NMSE) used as a measure of the neural network's prediction performance are commonly used statistical metrics [17]. An R-value approaching 1.0 means that the solar irradiance values predicted by the neural network have a close relationship with the actual values. If the R-value approaches 0.0, that means a random relationship. The MSE and NMSE values can be calculated with the following equations. Lower error values are better.

$$MSE(t, y) = \frac{1}{n} \sum_{i=1}^n (t_i - y_i)^2 \quad (1)$$

$$NMSE(t, y) = \frac{MSE(t, y)}{MSE(t, 0)} \quad (2)$$

In both equations, the variable  $t$  is the actual solar irradiance values; the  $y$  variable is the solar irradiance values predicted by the network.

The results including solar irradiance predictions, errors for predictions, NMSE values, and regression curves presented in MATLAB environment, R-values, the elapsed times, and the numbers of cycles for the training of the ANN models were recorded. Evaluations regarding these recorded results will be discussed in the following section.

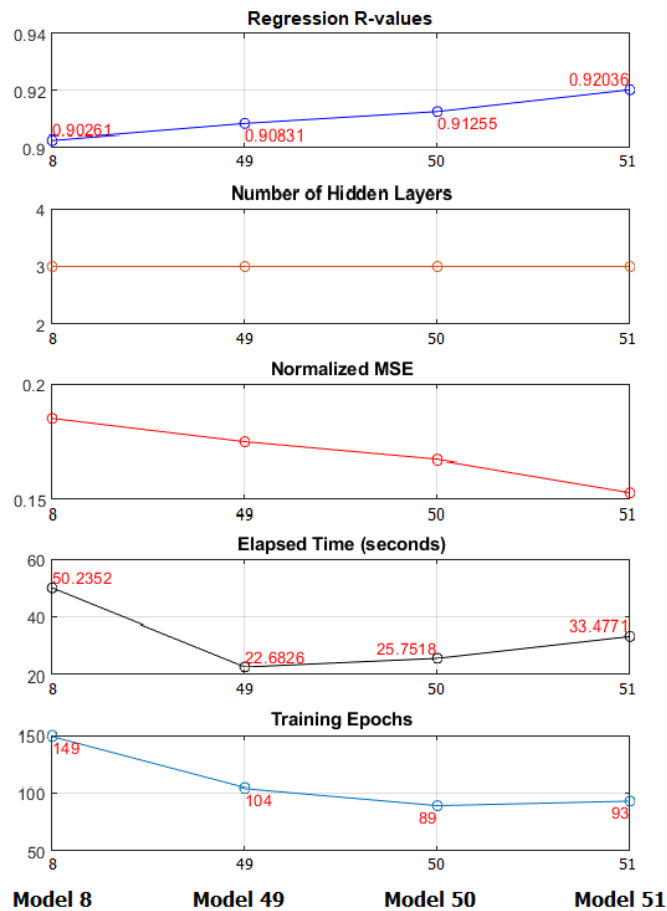
#### 4. Results and Discussion

When the results are examined, the following findings are obtained. Increasing the number of hidden layers in ANN models increases the accuracy of the predictions but the required time for training also increases. The learning algorithm of seven of the top 10 most successful models is the Levenberg-Marquardt algorithm. Model 8 with 3 hidden layers and 10-20-10 neuron distribution achieved an R-value of 0.90261 in 50.24 seconds, Model 49 with 3 hidden layers and 20-10-10 neuron distribution achieved an R-value of 0.90831 in 22.68 seconds. Here the only difference between Model 8 and Model 49 was the neuron distribution. Only changing the distribution of the neurons from 10-20-10 to 20-10-10 reduced the training time by nearly 50% besides increased the accuracy slightly.

Similarly, Model 50 with 30-10-3 neuron distribution achieved an R-value of 0.91255 in 25.75 seconds, Model 51 with 40-8-5 neuron distribution achieved an R-value of 0.92036 in 33.48 seconds.

Briefly, a decreasing neuron distribution from the first hidden layer to the last, significantly reduced training time and increased the prediction accuracy. The results of the four most successful models among the 51 ANN models obtained with different parameters are shown below in the **Figure 6**. In this figure, the regression R-values, the number of hidden layers, NMSE values, elapsed time, and required epochs for the training are

shown in a visual manner to show the performance of these four most successful models.



**Figure 6.** The results of the four most successful models.

Model 8 with 3 hidden layers and 10-20-10 neuron distribution achieved an R-value of 0.90261 in 50.24 seconds, Model 49 with 3 hidden layers and 20-10-10 neuron distribution achieved an R-value of 0.90831 in 22.68. Here the only difference between Model 8 and Model 49 was the neuron distribution. Only changing neuron distribution from 10-20-10 to 20-10-10 reduced the training time by nearly 50% and slightly increased the accuracy. Similarly, Model 50 with 30-10-3 neuron distribution achieved an R-value of 0.91255 in 25.75 seconds, Model 51 with 40-8-5 neuron distribution achieved an R-value of 0.92036 in 33.48 seconds. Briefly, a decreasing neuron distribution from the first hidden layer to the last, significantly reduced training time and increased the prediction accuracy.

The four regression plots in Figure 7 show the relationship between the outputs from the four most successful models and the actual values. The plots show prediction accuracies for the entire data set. The closer predictions to the actual values cause a more homogeneous and collective distribution of the data points on the 45-degree green dashed line in the graphs. The detailed regression plot in Figure 7 has been formed as an alternative to the black-and-white regression plot obtained from MATLAB, which shows the prediction accuracy in the whole data set. Although the data points in the regression plot originating from MATLAB are actually the predicted outputs and the actual target values, they are not distinguishable because they are all drawn in black colour. In the alternative regression plot in Figure 7, red data points are the actual target values, while blue data points show the predicted outputs. The green and dashed line in the figure is the first order polynomial curve between the network's predicted values and actual target data. The black coloured and continuous line in the figure is the regression curve formed from the values obtained as a result of the regression analysis that measures the relationship between the predicted values and the actual data.

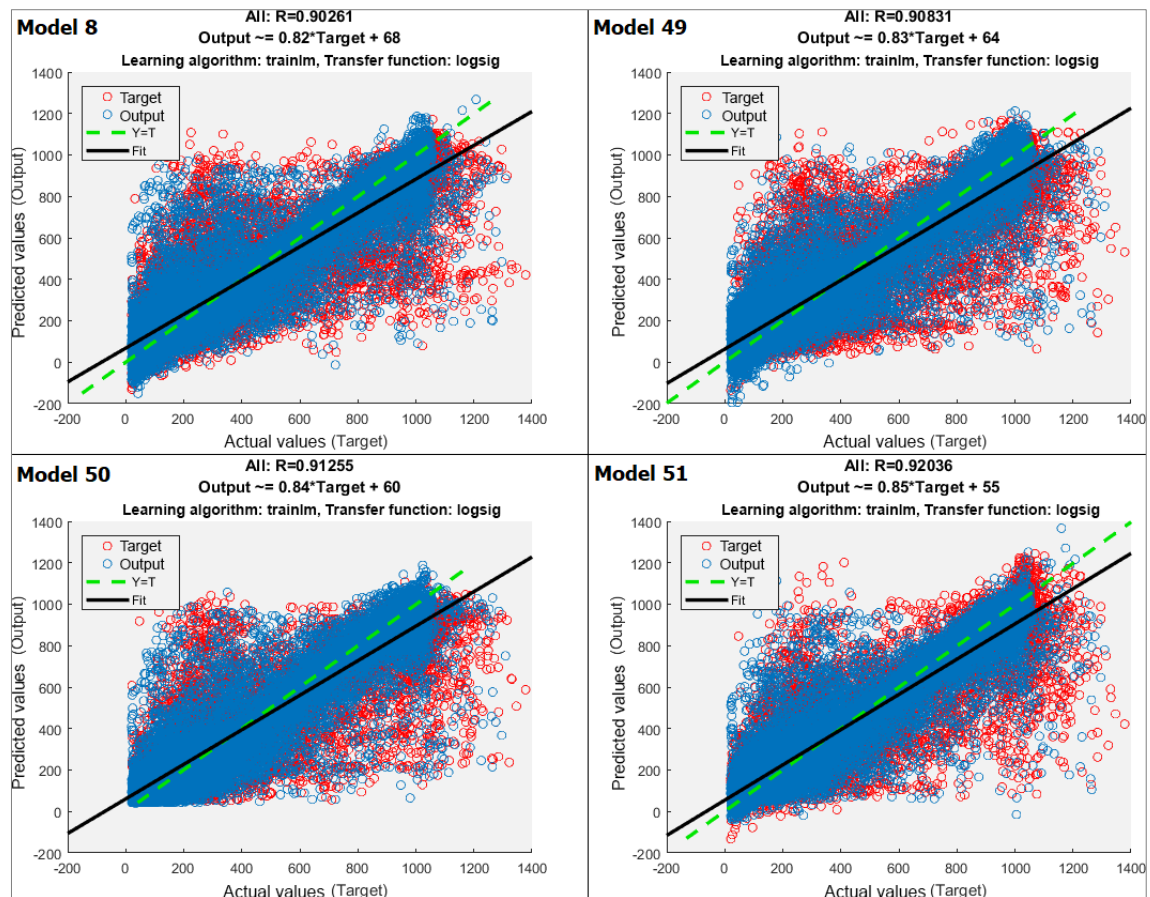


Figure 7. The regression plots of the four most successful models.

## 5. Conclusion

In this study, it was aimed to reveal and optimize the parameters in the development of the most suitable ANN structure for modeling solar irradiance using some climatic data. In the study, 51 different ANN models, which are separated from each other with parameters related to the solution method, were designed and examined.

The dataset was selected from the data collected at the measuring station installed in the Oregon University campus by NREL researchers. The measurement data had been collected between 20 December 2012 and 20 January 2014. Within the scope of the study, only measurement data of 2013 were used. The meteorological data used in the study are the solar irradiance as the target value and the independent variables such as the air temperature, relative humidity, atmospheric pressure, and cumulative daily total precipitation. Along with these data, the month and the hour values extracted from the measurement timestamp was used. After removing the samples with missing values, 70% of the data was used to train the ANN models, 15% for validation, and the remaining 15% were used to test the generalization ability against the new data faced by the trained models. The performances of the models were compared in terms of the regression R-values, NMSE error, the time spent training the models, the number of epochs required in the training.

The results obtained from all ANN models showed that the selected training algorithms and the number of hidden layers forming the architectural structure of the network and the number of neurons within these layers have important effects on the network's predictive performance. Since the transfer functions used in neurons were selected according to regression problems, they have been less determinant of the performance of the models. When using a transfer function that is not suitable for regression problems, the network has lost almost all of its prediction capabilities. Therefore, transfer functions suitable for problems such as classification and clustering were not addressed. Certain training algorithms such as trainlm and trainrp showed a significant increase in their predictive quality by increasing the number of hidden layers used in the ANN model or changing the neuron distribution in the hidden layers. Both algorithms achieved the highest predictive performance with the ANN model with 3 hidden layers. When neurons in hidden layers were distributed from the first hidden layer to the last in descending order, both accuracy increased and processing time shortened significantly. Using the advantage of reduced processing time, adding more neurons to hidden layers has led

to more successful models. This advantage can be used until the processing time-accuracy tradeoff is broken. As a further study, models using deep learning can be created and compared with the models obtained in this study.

### Declaration of interest

There is no conflict of interest, according to the authors.

### Nomenclature

#### Abbreviations

ANN	artificial neural network
CGP	polak-ribière conjugate gradient algorithm
FL	fuzzy logic
GA	genetic algorithm
LM	levenberg-marquardt algorithm
MSE	mean squared error
NMSE	normalized mean squared error
NREL	national renewable energy laboratory
R	regression value
SCG	scaled conjugate gradient algorithm

### References

- [1] A. Sözen, E. Arcaklioglu, M. Özalp, and E. G. Kanit, "Use of artificial neural networks for mapping of solar potential in Turkey," *Applied Energy*, vol. 77, no. 3, pp. 273–286, Mar. 2004.
- [2] O. Şenkal and T. Kuleli, "Estimation of solar radiation over Turkey using artificial neural network and satellite data," *Applied Energy*, vol. 86, no. 7–8, pp. 1222–1228, 2009.
- [3] A. Koca, H. F. Oztop, Y. Varol, and G. O. Koca, "Estimation of solar radiation using artificial neural networks with different input parameters for Mediterranean region of Anatolia in Turkey," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8756–8762, 2011.
- [4] Z. Wang, F. Wang, and S. Su, "Solar irradiance short-term prediction model based on BP neural network," *Energy Procedia*, vol. 12, pp. 488–494, 2011.
- [5] B. Marion *et al.*, "Data for Validating Models for PV Module Performance," 2014.
- [6] M. Ozgoren, M. Bilgili, and B. Sahin, "Estimation of global solar radiation using ANN over Turkey," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5043–5051, 2012.
- [7] C. Renno, F. Petito, and A. Gatto, "ANN model for predicting the direct normal irradiance and the global radiation for a solar application to a residential building," *Journal of Cleaner Production*, vol. 135, pp. 1298–1316, 2016.
- [8] M. Bou-Rabee, S. A. Sulaiman, M. S. Saleh, and S. Marafi, "Using artificial neural networks to estimate solar radiation in Kuwait," *Renewable and Sustainable Energy Reviews*, vol. 72, no. November 2016, pp. 434–438, 2017.
- [9] X. Xue, "Prediction of daily diffuse solar radiation using artificial neural networks," *International Journal of Hydrogen Energy*, vol. 42, no. 47, pp. 28214–28221, 2017.
- [10] F. Rodríguez, A. Fleetwood, A. Galarza, and L. Fontán, "Predicting solar energy generation through artificial neural networks using weather forecasts for microgrid control," *Renewable Energy*, vol. 126, pp. 855–864, 2018.
- [11] "Matlab 2017b," 2017. [Online]. Available: <https://ww2.mathworks.cn/en/>. [Accessed: 04-May-2019].
- [12] Ç. Elmas, *Yapay Zeka Uygulamaları Yapay Sinir Ağları – Bulanık Mantık– Genetik Algoritma*, 4th ed. 2011.
- [13] S. Haykin, *Neural Networks and Learning Machines, 3d Edition*, 3rd ed. ew Jersey: Pearson Education, 2008.
- [14] "Photovoltaic Research, NREL," 2019. [Online]. Available: <https://www.nrel.gov/pv/index.html>. [Accessed: 07-May-2019].
- [15] B. Marion *et al.*, "New data set for validating PV module performance models," in *2014 IEEE 40th Photovoltaic Specialist Conference (PVSC)*, 2014, pp. 1362–1366.
- [16] C. Kubat, *MATLAB Yapay Zeka ve Mühendislik Uygulamaları*. Abaküs, 2019.
- [17] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, vol. 103, no. 4. New York, NY: Springer New York, 2013.