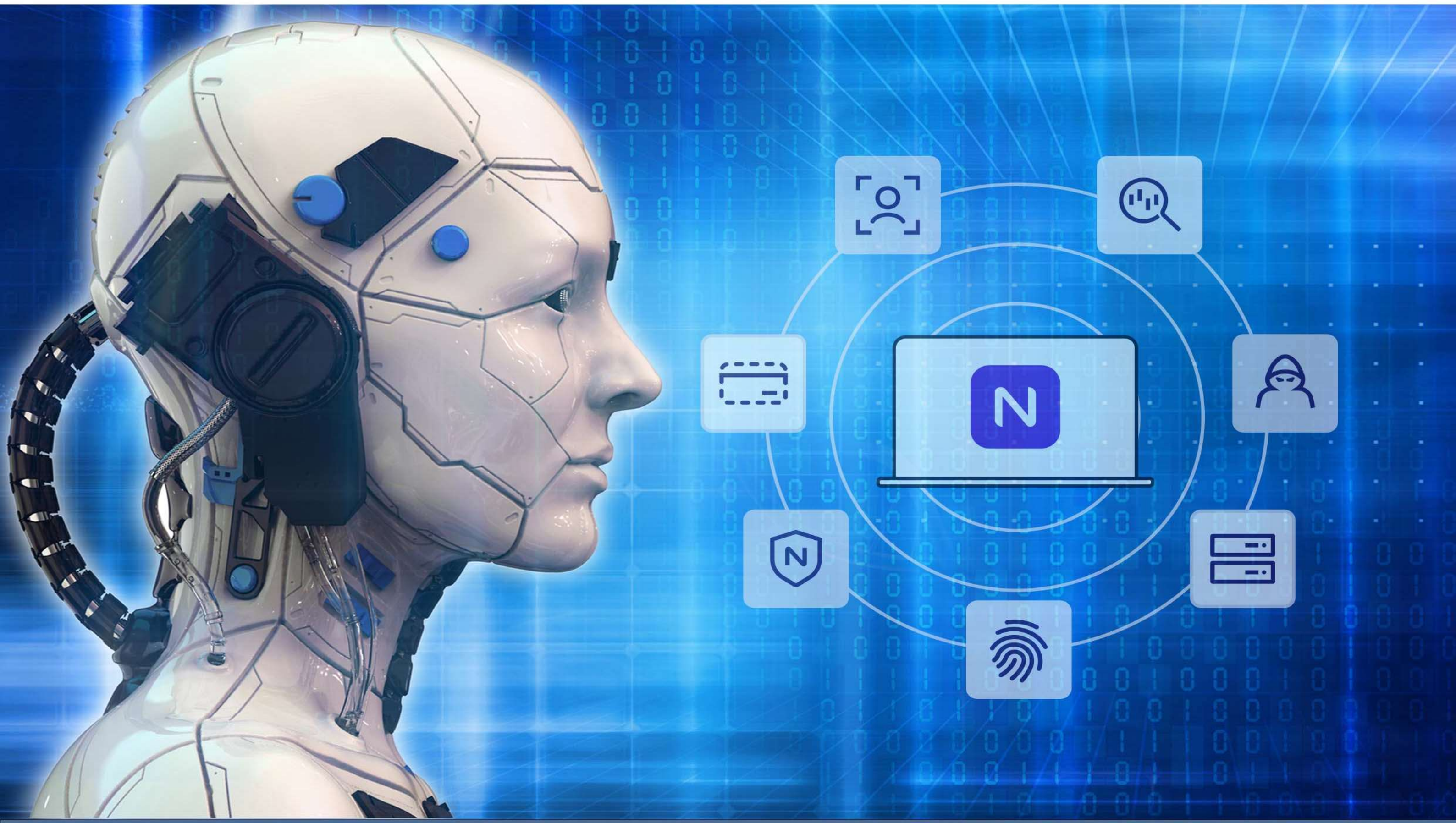
AITA

2023 Volume 3 Number 1

ARTIFICIAL INTELLIGENCE THEORY AND APPLICATIONS



RESEARCH ARTICLES

Estimating Types of Faults on Plastic Injection Molding Machines from Sensor Data for Predictive Maintenance

Comparison of Success Rates of Artificial Intelligence and Classical Methods in Estimation of Photovoltaic Energy Production: Study of İzmir Bakırçay University

An Application on Chest X Ray Images for the detection of Tuberculosis Disease by Employing Deep Convolutional Neural Networks

A Machine Learning Based Predictive Analysis Use Case for eSports Games

Effects of PCPDTBT:PCBM ratio on the electrical analysis and the prediction of I-V data using machine learning algorithms for Au/PCPDTBT:PCBM/n-Si MPS SBDs

Fraud detection on e-commerce transactions using machine learning techniques

Erratum Article

https://aitajournal.com

info@aitajournal.com

ISSN: 2757-9778

ISBN: 978-605-69730-2-4

AITA
ARTIFICIAL INTELLIGENCE
THEORY AND APPLICATIONS

Volume 3 – Issue 1
01.05.2023

e-ISSN: 2757-9778
Publisher: İzmir Bakırçay University

www.aitajournal.com
https://dergipark.org.tr/en/pub/aita

Journal Board

AITA

Artificial Intelligence Theory and Applications

Editorial Board

International Advisory Board

International Indexes

Academic Resource Index: Research Bible

https://www.researchbib.com/view/issn/2757-9778

Index Copernicus International

https://journals.indexcopernicus.com/search/details?id=125039&lang=en

About Journal

Artificial Intelligence Theory and Applications (AITA) provides coverage of the most significant work on principles of artificial intelligence, broadly interpreted. The scope of research we cover encompasses contributions of lasting value to any area of artificial intelligence. To be accepted, a paper must be judged to be truly outstanding in its field. AITA is interested in work in core artificial intelligence and at the boundaries, both the boundaries of sub-disciplines of artificial intelligence and the boundaries between artificial intelligence and other fields.

Scope

The best indicator of the scope of the journal is provided by the areas covered by its Editorial Board in theoretical (artificial intelligence and computing methodologies) and practical (artificial intelligence applications and applied computing) ways. These areas change from time to time, as the field evolves.

Year : 2023
Volume : 3
Issue : 1

ARTIFICIAL INTELLIGENCE
THEORY AND APPLICATIONS

Table of Content Page

Estimating Types of Faults on Plastic Injection Molding Machines from Sensor Data for Predictive Maintenance…………………………………………… 1-11
by Gözde Aslantaş, Tuna Alaygut, Merve Rumelli, Mustafa Özsaraç, Gözde Bakırlı, Derya Birant

Comparison of Success Rates of Artificial Intelligence and Classical Methods in Estimation of Photovoltaic Energy Production: Study of İzmir Bakırçay University…………………………………………………………………. 12-24
by Özgün Uz, Özge Tüzün Özmen

A Machine Learning Based Predictive Analysis Use Case for eSports Games………………………………………………………………………………. 25-35
by Atakan Tuzcu, Emel Gizem Ay, Ayşegül Umay Uçar, Deniz Kılınç

Effects of PCPDTBT:PCBM Ratio on the Electrical Analysis and the Prediction of I-V Data Using Machine Learning Algorithms for Au/PCPDTBT:PCBM/n-Si MPS SBDs…………………………………………….. 36-44
by Ömer Berkan Çelik, Burak Taş, Özgün Uz, Hüseyin Muzaffer Şağban, Özge Tüzün Özmen

Fraud Detection on E-Commerce Transactions Using Machine Learning Techniques………………………………………………………………………… 45-50
by Murat Gölyeri, Sedat Çelik, Fatma Bozyiğit, Deniz Kılınç,

An Application on Chest X Ray Images for the Detection of Tuberculosis Disease by Employing Deep Convolutional Neural Networks………………….. 51-64
by Hatice Koç, Kadir Hızıroğlu, Ahmet Emin Erbaycu

Erratum for Halil İbrahim Cebeci about Article “Artificial Intelligence Applications in Management Information Systems: A Comprehensive Systematic Review with Business Analytics Perspective” published on Volume 1 Issue 1 in 2021………………………………………………………… 65
By AITA Editorial Team

■ RESEARCH ARTICLE

Estimating Types of Faults on Plastic Injection Molding Machines from Sensor Data for Predictive Maintenance

Gözde Aslantaş [a†], **Tuna Alaygut** [a], **Merve Rumelli** [a], **Mustafa Özsaraç** [a], **Gözde Bakırlı** [a], **Derya Birant** [b]

[a] VESTEL Electronics, Manisa, Turkey
[b] Department of Computer Engineering, Dokuz Eylul University, Izmir, Turkey
[†] gozde.aslantas@vestel.com.tr, corresponding author

RECEIVED JULY 27, 2022
ACCEPTED AUGUST 30, 2022

CITATION Aslantaş, G., Alaygut, T., Rumelli, M., Özsaraç, M., Bakirli, G., & Birant, D. (2023). Estimating types of faults on plastic injection molding machines from sensor data for predictive maintenance. *Artificial Intelligence Theory and Applications, 3(*1), 1-11.

Abstract

Fault type detection for the plastic injection molding machines is an important problem in order to take failure-specific actions to prevent any problem in production, hence providing continuity in procurement. In this study, we treat this problem as a multi-class classification task and proposed a novel machine learning model to achieve reliable and accurate results. We applied the Random Forest (RF) and Extreme Gradient Boosting (XGBoost) algorithms with and without SMOTE (Synthetic Minority Over-sampling Technique) to a real-world dataset for predictive maintenance. According to the results, XGBoost performed better than RF. With the combination of SMOTE method, the performances of both methods increased in terms of accuracy. XGBoost with SMOTE outperformed other techniques by achieving about 98% accuracy on average.

Keywords: machine learning; predictive maintenance; classification; plastic injection molding machines; manufacturing; sensors

1. Introduction

The development of classification techniques to predict electrical machine faults has become a significant area of research and interest in the manufacturing sector since the occurrence of a fault affects production processes and leads to high financial losses and reduced efficiency for the industries. Preventing failures is essential to avoid undesired effects such as vibration, overheating, voltage unbalance, costly machinery repair, reduced safety, and a stop of the production process.

This study aims to utilize machine learning methods to classify faults on plastic injection molding (PIM) machines using features extracted from sensor data. PIM machines are widely used in various industrial plants that especially produce medical devices, white goods, and household appliances. These machines require systematic, timely, and proper maintenance since different types of faults can occur, which influence the normal operation of the equipment. PIM process quality can deteriorate due to machine or tool wear, deviations in the condition of the input material, environmental change, operator

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN : 978-605-69730-2-4 © 2023 University of Bakırçay

fatigue, and a change of material batch. A PIM machine part (i.e., piston, pump) can be damaged due to high speed, temperature, or vibration as a result of material defects, improper assembly, exceeding the initial design conditions, excessive force and overload during the production. For example, if the process-parameter settings are not properly adjusted by a process operator, unplanned PIM machine stops can occur. In this study, four fault types are detected: (i) hydraulic equipment (HE) malfunctions, (ii) heat application equipment (HAE) malfunctions, (iii) column and shear system hardware (CSSH) malfunctions, and (iv) plasticizing unit hardware (PUH) malfunctions.

The contributions of this study to the literature can be listed as follows. (i) It proposes a novel machine learning model that correctly estimates types of faults on plastic injection molding machines from sensor data. (ii) It compares different classification methods to determine which one is best suited to estimate failure types. (iii) Our study is useful for developing an effective model since it extracts useful features (i.e., entropy, kurtosis, skewness) from raw sensor data by aggregating samples over a time window.

The organization of the paper is as follows. In Section 2, we briefly review the related work in the literature. Section 3 explains our proposed approach and the details of the feature extraction process. Section 4 shows experimental results with a discussion. Finally, in Section 5, we present the conclusion and future work.

2. Related Work

Fault type classification has been studied in many different areas such as energy [1, 2], industry [3, 4], and manufacturing [5]. The literature involves some important and recent studies that use artificial intelligence methods to estimate types of faults in electrical machines and motors. More specifically, traditional machine learning techniques such as Bayesian networks (BN) [4], support vector machine (SVM) [6], k-nearest neighbors (KNN) [7], extreme learning machine (ELM) [3], logistic regression (LR) [3], decision trees (DT) [8], and neural network (NN) [1, 5] have been reported in many studies. Moreover, ensemble learning methods have also been utilized for fault type classification such as random forest (RF) [5], and extreme gradient boosting (XGBoost) [2]. Furthermore, deep learning (DL) [7] techniques have been successfully applied for predictive maintenance such as long short term memory (LSTM) [1] and convolutional neural networks (CNN) [6].

Table 1 shows a summary of the related work on fault detection and classification. Fault diagnosis has been performed for different types of machines, motors, or equipment such as power transformers [9], induction motors [10, 11], wind turbines [2], gearbox [3], steel plates [5], bearing [12], asynchronous machines [13], and rotating machinery [7]. Wang et al. [14] proposed a solution for defect diagnosis induction motors.

Chen et al. [3] demonstrated that the accuracy obtained by the optimized kernel-based extreme learning machine method was 93.97% on average for the detection and identification of faults in a gearbox. In fault detection for three types of wind turbine subsystems, Liu et al. [6] achieved 97.03% accuracy on average with the convolutional neural network. Morales et al. [8] applied different machine learning techniques for the automatic prediction of maintenance intervention types in roads and obtained a final accuracy of 93.4% with the decision tree method. Zhao et al. [9] used support vector machine to identify and classify the winding mechanical fault types and achieved an 83.3% accuracy rate on average.

Unlike the previous studies given in Table 1, our study aimed at the use of classification algorithms to estimate fault types for plastic injection molding machines. Furthermore, useful features (i.e., entropy, kurtosis, skewness) were extracted from raw sensor data by aggregating samples over a time window.

Table 1. Summary of related works

Reference	Year	Method	Description	#Faults	Equipment	Sector
Moradzadeh et al. [1]	2022	SVM, DT, KNN, CNN, LSTM	Identification of locations and types of faults	11	Transmission line	Energy
Leon-Medina et al. [2]	2022	XGBoost	Structural damage classification	4	Wind turbine	Energy
Chen et al. [3]	2021	LR, ELM	Multi-type and concurrent fault diagnosis in rotary machines	5	Gearbox	Industry
Bressan et al. [4]	2021	BN, SVM, KNN	Classification of types of faults on machines from acoustic signals	6	Induction motors	Industry
Trinh and Kwon [5]	2020	NN, KNN, SVM, RF	Classification of fault types and remaining useful life estimation	7	Steel plate	Manufacturing
Liu et al. [6]	2020	SVM, CNN	Fault detection in the process of power generation	5	Wind turbine	Energy
Liu et al. [7]	2018	KNN, NB, SVM, NN, DL	Fault diagnosis of mechanical equipment in modern industry	3	Rotating machinery	Industry
Morales et al. [8]	2018	DT, KNN, SVM, NN	Prediction of maintenance intervention types	5	Road	Transportation
Zhao et al. [9]	2017	SVM	Identification of winding mechanical fault types	3	Power transformer	Energy
Palacios et al. [10]	2015	NB, KNN, SVM, NN, DT	Fault identification in electrical machines	3	Induction motors	Industry
Aydin et al. [11]	2014	Fuzzy DT, NN, GA	Fault diagnosis in manufacturing equipment.	3	Induction motors	Industry
Ertunc et al. [12]	2013	NN	Detection and diagnosis of bearing faults	8	Bearing	Manufacturing
Barzegaran et al. [13]	2013	NN	Identification of winding failures	6	Asynchronous machines	Energy
Wang et al. [14]	2012	NB, KNN, SVM	Defect diagnosis of vital machine components	5	Induction motors	Industry

3. Material and Methods

3.1. Proposed Approach

This paper proposes a machine learning model that correctly estimates types of faults on plastic injection molding (PIM) machines from sensor data. The model is constructed by using classification algorithms. The aim of the study is to analyze sensor readings to predict PIM machine failures and their types before they occur. In this way, it is aimed to make appropriate scheduling of repairs and prevent unexpected failures of machines since machine malfunction affects production processes and leads to financial losses and reduced efficiency for the industries.

Fault detection in PIM machines is composed of a pipeline of various steps from the collection of raw data to the classification. In this process, the prediction of the remaining useful life (RUL) of the machines plays a vital role in unveiling the machines that are more likely to fail. RUL is the time interval between the current point and the point where a failure occurs or it needs maintenance action. For the time index (*t*) when the machine fails or reaches a maintenance action, the RUL is set as *t*, and after that RUL is linearly decreasing from *t* to zero at each time cycle. Therefore, RUL is calculated based on the historical maintenance and failure data of the machines. Many features of the machines affect this time interval such as clamping force, closing force, cycle time, holding pressures, the age of the machine, operational environment, the sequence of the active/idle periods, the quality of key equipment, and oil temperature. The values of these features can be obtained from sensors, process parameter records, alarm records, and tool exchange (planned or unplanned maintenance) records.

Figure 1 shows the pipeline of the proposed approach. In the first step, raw data is collected from PIM machines via sensors. In the data preprocessing step, missing values are handled either by removing or interpolating with the non-null values if the percentage of null values does not exceed a threshold. Outliers in the data are detected by z-score and discarded to obtain better results. After that, the feature extraction process is performed by using statistical methods. Later, features that contribute to the discrimination the most are selected from the data. Synthetic Minority Over-sampling Technique (SMOTE) [15] is applied to the data to re-balance it according to the fault types. Finally, in the last step, data is trained by machine learning methods. The constructed model is tested and the results are evaluated in terms of various measures such as accuracy, recall, precision, and F-score.

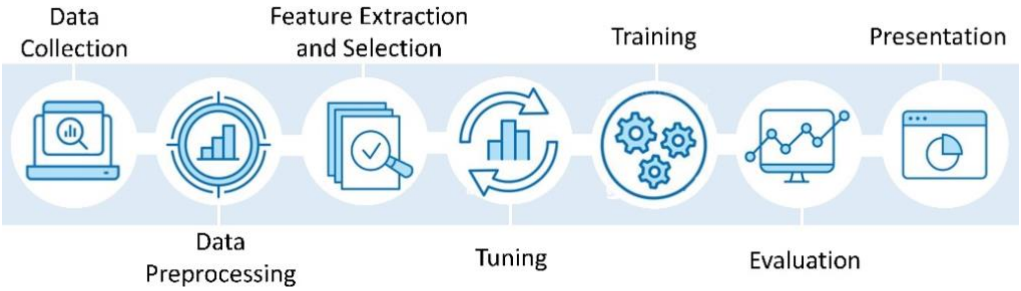


Figure 1. The pipeline of the proposed approach

3.2. Machine Learning Algorithms

In this study, two classification algorithms were used to construct machine learning models for predictive maintenance: random forest and extreme gradient boosting.

Random Forest (RF): It is an ensemble learning-based algorithm that is composed of multiple decision trees, which are called estimators. Decision trees are constructed based on a number of bootstrap samples drawn with replacements from the data. Each bootstrap is composed of a different subset of the data. Each decision tree votes on the classification of a sample, and the final estimation is obtained based on majority voting or averaging. This method is used for both regression and classification problems.

Extreme Gradient Boosting (XGBoost): It is also an ensemble learning-based method that uses gradient descent to optimize each decision tree. It finds the optimal parameter that minimizes the loss. In other words, it adds regularization terms into its loss function. The predictions of decision trees are combined with a voting mechanism. It has the ability to achieve a balance between computing speed and model performance.

3.3. SMOTE

In fault diagnosis applications, the collected data tends to be imbalanced since machines usually operate under healthy conditions prior to the occurrence of faults. When the class distribution of the data was imbalanced, the machine learning algorithms produce classifiers that may perform poorly on minority-class due to two reasons. First, since the majority class dominates a large proportion of the data, its samples are more likely to be selected for training the classifier. Second, the loss factor is calculated based on the ability of the classifier to recognize the majority class more than the minority class. In order to solve the data imbalance problem, oversampling or undersampling techniques are usually used.

SMOTE is a very popular and powerful method to expand minority sample data areas. For a sample *x* in a minority class, SMOTE searches its *k*-nearest neighbors (having the same class label) using a distance metric and randomly selects a sample *y* among them, and then creates a synthetic new sample by calculating linear interpolation between the samples *x* and *y*. The class label for the new sample is the minority class. Different synthetic samples are created based on different neighbor pairs.

In the implementation of SMOTE, the number of neighbors (*k*) is the key parameter for controlling the amount of oversampling of the minority class. For each sample belonging to the minority class in the original dataset, *k* new samples will be generated. In this study, we set *k* to 1, which doubles the number of minority samples. For example, for 100 original minority samples, SMOTE with *k*=1 produces a total of 200 minority samples (i.e. 100 original and 100 synthetics). In this study, the value for *k* was determined by GridSearchCV as an optimal value in the search space. In order to demonstrate the effect of the imbalanced class distribution, we applied RF and XGBoost algorithms with SMOTE and without SMOTE.

3.4. Feature Extraction

Raw sensor data usually do not carry sufficient information itself to describe a fault type since an observation contains one specific value at a particular time instant. For this reason, feature extraction is an important stage in the construction of a classification model and aims at the extraction of the useful information that characterizes each class. Table 2 shows the features extracted from the raw sensor data.

Feature extraction typically transforms the input data into a set of features to provide a compact but effective representation [16]. The determination of these features is a crucial issue as the choice of the classification method to be able to build a good classifier for predictive maintenance. In this study, raw sensor data was aggregated at 60-minute intervals and features were automatically extracted for each internal such as min, max, mean, skewness, kurtosis, and entropy.

Table 2. Features extracted from the raw sensor data

Feature	Description	Formula
Min	The minimum value over the segment	$\mathrm{MIN} = \min(X)$
Max	The maximum value over the segment	$\mathrm{MAX} = \max(X)$
Mean	The average value over the segment	$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
Absolute Mean	The absolute average value over the segment	$\underline{x} = \frac{1}{n}\sum_{i=1}^{n} \lvert x_i \rvert$
Standard Deviation	The standard deviation of the values over the segment	$STD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$
Median	The middle or central number in the sorted segment values	$MED = \begin{cases} \mathrm{X}\left[\frac{n}{2}\right] & if\ n\ is\ even \\ \frac{X[\frac{n-1}{2}] + X[\frac{n+1}{2}]}{2} & if\ n\ is\ odd \end{cases}$
Peak-to-Peak Value	The difference value between the maximum and minimum values over the segment	$PP = \max(x) - \min(x)$
Root Mean Squared (RMS)	The quadratic mean of the discrete values over the segment	$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$
Kurtosis	The measurement of the peakedness of values in the segment	$KV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma}\right)^4$
Skewness	The measurement of the symmetry of the distribution of values in the segment	$SV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{\sigma}\right)^3$
Crest Factor	The ratio of the maximum value to the root mean square value of samples in the segment	$CRF = \frac{\max(x)}{RMS}$
Clearance factor	The ratio of the maximum value to the squared mean of the sum of the square roots	$\mathrm{CLF} = \frac{\max(x)}{(\frac{1}{n}\sum_{i=1}^{n}\sqrt{\lvert x_i \rvert})^2}$
Shape Factor	RMS is divided by the absolute mean	$SF = \frac{RMS}{\frac{1}{n}\sum_{i=1}^{n}\lvert x_i \rvert}$
Impulse	Max values is divided by the absolute mean	$IMP = \frac{max(x)}{\frac{1}{n}\sum_{i=1}^{n}\lvert x_i \rvert}$
Entropy	The level of information of sensor data in the segment	$E = \sum_{i=1}^{n} P(x_i) log_2 P(x_i)$ $P(x_i) = \frac{x_i}{\sum_{j=1}^{n} x_j}$

4. Experimental Studies

In the experiments, we applied RF and XGBoost algorithms with SMOTE and without SMOTE versions to a real-world dataset. Hyperparameter values were selected according to the best performances of the algorithms in terms of accuracy metric by using GridSearchCV. For example, various alternative numbers of trees (50, 100, 150) were investigated for each algorithm separately. According to the experiments, the optimal values obtained for hyperparameters are given in Table 3. In addition to quality measure, four hyperparameters (*n_estimators*, *min_samples_leaf*, *max_features*, and *max_depth*) were tuned through grid search for the purpose of decreasing computational cost and saving the processing time of model training, similar to the previous studies [17, 18, 19]. Wang et al. [18] and Peng et al. [19] reported that the results were most affected by these hyperparameters.

Table 3. Hyperparameter values

Description	Parameter	Random Forest	XGBoost
Number of trees	n_estimators	150	100
Minimum number of examples to be a leaf	min_samples_leaf	5	5
Maximum depth of the tree	max_depth	12	12
Number of features to apply split	max_features	Log2	Auto
The quality measure	criterion	Gini	Mean Absolute Error (MAE)

In the evaluation phase, a 10-fold stratified cross-validation technique was used to avoid overfitting. In this technique, the original dataset is randomly split into 10 subsets with equal size. While nine subsets are used for training and the remaining subset is used for testing. The experimental results were evaluated in terms of four metrics: accuracy, precision, recall, and F-score. The performance metrics were calculated for each class, along with the macro average values of all classes.

4.1. Dataset Description

In this work, data was collected from the sensors of three plastic injection molding machines, namely HEP3204, HEP3207, and HEP3213, which are specialized in electronics of a Turkish home and professional appliances manufacturing factory. Data sizes collected from the injection machines are 251K, 205K, and 913K, and the collection time interval of each is between September 2019 - September 2021, June 2019 - September 2021, and May 2018 - September 2021, respectively. In this study, a separate predictive model was constructed for each plastic injection molding machine.

Sensor values collected from the injection machines are as follows: clamping force peak value, clamping force set value, closing force Skx value, cycle time ZU-sets value, hold pressures between 1st and 10th steps, hydraulic holding pressure peak value, material cushion smallest value, injection time, and oil temperature. A sample part of the raw data is given in Table 4. Here, the last column (fault type) is the output to be estimated while the others are the inputs.

Table 4. A sample part of the raw data

Machine	Material Cushion Smallest Value	Oil Temp.	Cycle Time	Zone1 Temp.	Zone2 Temp.	Injection Time	Closing Force	RUL	Fault Type
HEP3213	64.5	299.7	189.2	49.9	290.1	0.80	0	1	HE
HEP3213	64.6	299.8	189.2	49.9	289.9	0.80	0	1	HE
HEP3213	64.5	299.6	187.4	50.0	290.0	0.80	0	1	HE
HEP3213	20.6	280.1	46.6	20.0	284.9	1.10	0	1	HE
HEP3213	57.8	309.9	174.4	49.9	290.1	0.62	0	2	HE
HEP3213	22.4	274.9	61.4	18.9	284.9	0.60	2	2	HE
HEP3213	64.6	140.2	344.4	58.0	255.0	0.86	2	2	HE
HEP3213	20.5	280.1	47.8	20.0	284.9	1.14	0	3	HE
HEP3213	24.4	140.2	78.2	58.0	255.0	0.86	2	3	HE
HEP3213	32.5	284.9	67.8	23.0	280.0	0.64	0	1	HAE
HEP3213	20.6	284.9	48.2	18.0	270.0	0.62	0	1	HAE
HEP3213	20.6	284.9	45.8	18.0	269.9	0.62	0	1	HAE
HEP3213	21.6	280.1	49.6	21.9	244.2	0.62	0	2	HAE
HEP3213	20.6	268.1	25.2	23.0	270.0	0.62	0	2	HAE
HEP3213	22.3	285.0	46.0	20.0	269.9	0.62	0	2	HAE
HEP3213	21.6	279.6	49.0	22.0	245.7	0.62	0	3	HAE
HEP3213	21.8	265.6	25.8	21.9	275.1	0.66	0	3	HAE
HEP3213	65.3	289.7	156.6	49.9	279.9	0.92	0	3	CSSH
HEP3213	65.4	289.7	154.6	50.0	280.2	0.92	0.	3	CSSH

A large number of missing values in a data column or data row could mislead the classification problem since they increase uncertainty and unreliable conclusions. On the other hand, the removal of rows or columns having a small amount of missing data leads to a loss of important data and can cause bias in the results. Thus, determining a strategy for handling missing values has an important place in terms of making the most out of the data while having no information loss. According to previous studies like [17], the replacement of missing values with suitable ones is an important step in building an effective classifier since it avoids data loss, provides a better understanding of patterns hidden in data, and usually improves classification accuracy. Motivated by the studies in the literature [17], we successfully handled the missing values problem by using data imputation for numerical features. It is necessary to find a trade-off between the benefit of filling missing data and classification accuracy. Based on our experiences, we determined this trade-off as 40% to obtain satisfactory accuracy in the classification of the fault types. Therefore, in this study, the attributes of the dataset that have null values for less than 40% were filled with the interpolated values of the not-missing values of the same sensor. Furthermore, the rows that include sensor values with high missing rates (>40%) were eliminated.

In the dataset, there are 15 distinct failure messages in total. These messages are grouped into 4 fault types according to the supplied domain knowledge. Errors are listed in Table 5 and associated with the corresponding fault types.

Table 5. The types of faults in the dataset

Failure Message	**Fault Type**
Filter Error	Hydraulic Equipment (HE)
Mold Opening and Closing Error	
Hydraulic Safety Error of Mold Closing	
Mold Stroke Error	
Vise Adjustment Error	
Vise Piston Failure	
Vice Stroke Error	
Pump Failure	
High Oil Temperature Error	
Group Cylinder Temperature Error	Heat Application Equipment (HAE)
Colon Failure	Column & Shear System Hardware (CSSH)
Shear System Error	
Vise Opening/Closing Error	
Machine Failure to Inject	Plasticizing Unit Hardware (PUH)
No Error	No Error

The data was enhanced with the remaining useful life (RUL) feature. RUL was calculated by subtracting the collection date from the maintenance/failure date. After that, the data was annotated based on the condition where RUL is smaller than or equal to 3 days. The residual failure types along with their count for the three machines are listed in Table 6.

Table 6. Fault types with counts

Machine\Fault Type	**Hydraulic Equipment**	**Heat Application Equipment**	**Column & Shear System Hardware**	**No Error**
HEP3204	203	40	185	2166
HEP3207	757	338	104	1725
HEP3213	939	343	157	8425

As seen in Table 6, the class distribution in the dataset is imbalanced. The majority of samples are labeled as “No Error” for all machines. Samples that need maintenance due to hydraulic equipment type of failure are in the first rank amongst the most seen fault types. Such a class imbalance problem could mislead the classification results by showing a tendency to bias toward the class that has more data. For this reason, we used the SMOTE technique to avoid imbalanced data problems.

4.2. Experimental Results

Table 7 shows the average accuracy, precision, and recall metric values obtained by the methods: Random Forest (RF), XGBoost, RF with SMOTE, and XGBoost with SMOTE, respectively. According to the results, it is possible to say that the algorithms had no difficulty in predicting fault types successfully. The XGBoost + SMOTE algorithm achieved the best performance with an accuracy of 98% on average. In other words, the comparison in this table depicts that XGBoost + SMOTE outperformed other methods in terms of all metrics.

Table 7. Average performance results of each method

Method	Accuracy (%)	Precision	Recall
Random Forest	94	0.936	0.840
XGBoost	95	0.930	0.870
Random Forest + SMOTE	97	0.976	0.970
XGBoost + SMOTE	**98**	**0.980**	**0.980**

Figure 2 shows average macro F-score values obtained by alternative methods. While RF and RF + SMOTE achieved 0.88 and 0.97 according to the performance metric, XGBoost and XGBoost + SMOTE obtained values of 0.90 and 0.98, respectively. With the use of SMOTE technique, the performances of both methods increased. According to the results, the XGBoost + SMOTE algorithm is seen to have better performance than others on average.

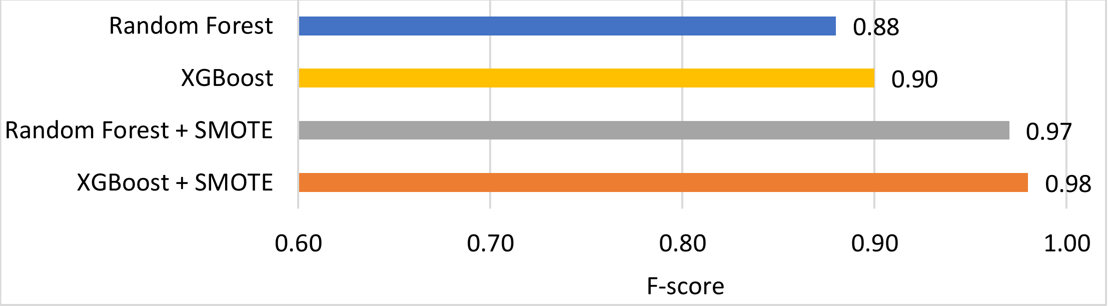


Figure 2. Average macro F-score values

5. Conclusion and Future Work

In this study, we performed an estimation of fault types of plastic injection molding machines by considering the problem as a multiclass classification for predictive maintenance. Raw sensor data was collected from three machines and then preprocessed and analyzed by using machine learning methods. In the data preprocessing step, missing values were handled since the high presence of missing values leads to uncertainty in the results, and therefore, they cause difficulty in extracting meaningful information. While filling a large amount of missing data causes unreliable conclusions, the removal of rows and columns having a small amount of missing data leads to a loss of important data. A widespread strategy is that a dataset is considered reliable if the rate of missing values is below a threshold, which was determined as 40%

in this study. Based on our experiences, the missing values were filled with the suitable interpolation of the observed values in that attribute per product type. As mentioned in Section 3.4. and listed in Table 2, useful features (i.e., entropy, kurtosis, skewness) were extracted from raw sensor data by aggregating samples over a time window. Furthermore, remaining useful life (RUL) values were calculated from data and then used to annotate data for classification. In this study, we grouped 15 distinct failure messages into 4 fault types. We applied Random Forest (RF) and Extreme Gradient Boosting (XGBoost) algorithms with and without SMOTE technique. According to the results, XGBoost performed better than RF. With the use of SMOTE technique, the performances of both methods increased since the dataset is imbalanced. XGBoost with SMOTE achieved the highest accuracy (98%) on average whereas XGBoost without SMOTE predicted with 95% accuracy.

In future work, we aim to expand this study by including other plastic injection molding machines in the manufacturing factory. An application will be implemented to show the classification results to the managers via a user interface for giving feedback about the status of the machines. In this way, the output of the model will be taken into consideration by a manager for decision-making.

References

[1] Moradzadeh, A., Teimourzadeh, H., Mohammadi-Ivatloo, B., & Pourhossein, K. (2022). Hybrid CNN-LSTM approaches for identification of type and locations of transmission line faults. *Electrical Power and Energy Systems*, *135*, 1-13.

[2] Leon-Medina, J.X., Anaya, M., Pares, N., Tibaduiza, D. A., & Pozo, F. (2021). Structural damage classification in a jacket-typewind-turbine foundation using principal component analysis and extreme gradient boosting. *Sensors*, *21*(8), 1-29.

[3] Chen, Q., Wei, H., Rashid, M., & Cai, Z. (2021). Kernel extreme learning machine-based hierarchical machine learning for multi-type and concurrent fault diagnosis. *Measurement*, *184*, 1-12.

[4] Bressan, G. A., de Azevedo, B. C. F., dos Santos, H. L., Endo, W., Agulhari, C. M., Goedtel, A., & Scalassara, P. R. (2021). Bayesian approach to infer types of faults on electrical machines from acoustic signal. *Applied Mathematics & Information Sciences*, *15*(3), 353-364.

[5] Trinh, H-C., & Kwon, Y-K. (2020). A data-independent genetic algorithm framework for fault-type classification and remaining useful life prediction. *Applied Sciences*, *10*(1), 1-20.

[6] Liu, Z., Xiao, C., Zhang, T., & Zhang, X. (2020). Research on fault detection for three types of wind turbine subsystems using machine learning. *Energies*, *13*(2), 1-21.

[7] Liu, R., Yang, B., Zio, E., & Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, *108*, 33-47.

[8] Morales, F. J., Reyes, A., Caceres, N., Romero, L., & Benitez, F. G. (2018). Automatic prediction of maintenance intervention types in roads using machine learning and historical records. *Transportation Research Record*, *2672*(44), 43-54.

[9] Zhao, Z., Tang, C., Zhou, Q., Xu, L., Gui, Y., & Yao, C. (2017). Identification of power transformer winding mechanical fault types based on online IFRA by support vector machine. *Energies*, *10*(12), 1-16.

[10] Palacios, R. H. C., da Silva, I. N., Goedtel, A., & Godoy, W. F. (2015). A comprehensive evaluation of intelligent classifiers for fault identification in three-phase induction motors. *Electric Power Systems Research*, *127*, 249-258.

[11] Aydin, I., Karakose, M., & Akin, E. (2014). An approach for automated fault diagnosis based on a fuzzy decision tree and boundary analysis of a reconstructed phase space. *ISA Transactions*, *53*, 220-229.

[12] Ertunc, H. M., Ocak, H., & Aliustaoglu, C. (2013). ANN- and ANFIS-based multi-staged decision algorithm for the detection and diagnosis of bearing faults. *Neural Computing & Applications*, *22*(1), 435-446.

[13] Barzegaran, M., Mazloomzadeh, A., & Mohammed, O. A. (2013). Fault diagnosis of the asynchronous machines through magnetic signature analysis using finite-element method and neural networks. *IEEE Transactions on Energy Conversion*, *28*(4), 1064-1071.

[14] Wang, J., Liua, S., Gaoa, R. X., Yanb, R. (2012). Current envelope analysis for defect identification and diagnosis in induction motors. *Journal of Manufacturing Systems*, *31*, 380–387.

[15] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*(1), 321–357.

[16] Zebari, R.R., Abdulazeez, A. M., Zeebaree, D. Q., Zebari, D. A., Saeed, J. N. (2020). A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. *Journal of Applied Science and Technology Trends*, *1*(2), 56-70.

[17] Charoen-Ung, P., & Mittrapiyanuruk, P. (2019). Sugarcane yield grade prediction using random forest with forward feature selection and hyper-parameter tuning. In: Unger, H., Sodsee, S., & Meesad, P. (eds) Recent Advances in Information and Communication Technology 2018. *Advances in Intelligent Systems and Computing*, *769*, 33-42.

[18] Wang, R. Gao, W., & Peng, W. (2021). Spatial downscaling method for air temperature through the correlation between land use/land cover and microclimate: A case study of the Greater Tokyo Area, Japan. *Urban Climate*, *40*, 1-16.

[19] Peng, W., Yuan, X., Gao, W., Wang, R., & Chen, W. (2021). Assessment of urban cooling effect based on downscaled land surface temperature: A case study for Fukuoka, Japan. *Urban Climate*, *36*, 1-18.

[20] Baitharu, T.R., & Pani, S.K. (2013). Effect of missing values on data classification. *Journal of Emerging Trends in Engineering and Applied Sciences*, *4*(2), 311–316.

Acknowledgement

This study has been supported by the project numbered 9190028 carried out within the scope of the TUBITAK.

■ RESEARCH ARTICLE

Comparison of Success Rates of Artificial Intelligence and Classical Methods in Estimation of Photovoltaic Energy Production: Study of İzmir Bakırçay University

Özgün Uz [a†], **Özge Tüzün Özmen**[a,b]

[a] Department of Electrical Electronics Engineering, İzmir Bakırçay University, İzmir, Turkey
[b] Department of Fundamental Sciences, İzmir Bakırçay University, İzmir, Turkey
[†] ozgunuz@hotmail.com, corresponding author

RECEIVED FEBRUARY 15, 2023
ACCEPTED MARCH 22, 2023

CITATION Uz, Ö. & Tüzün Özmen, Ö. (2023). Comparison of success rates of artificial intelligence and classical methods in estimation of photovoltaic energy production: Study of İzmir Bakırçay University. *Artificial Intelligence Theory and Applications, 3(*1), 12-24

Abstract

Global solar energy has become a popular investment choice for investors, with installed power reaching 940GW according to 2021 data. Investors are interested in profit margin estimations based on energy production, which are provided through feasibility studies conducted before building solar power plants (SPP). While classical mathematical algorithms are typically used to calculate energy production, advances in technology offer opportunities to achieve better results. In our energy production estimation studies conducted at İzmir Bakırçay University SPP, we achieved a 70.24% success rate using classical estimation algorithms based on past production and meteorological data. However, by developing an artificial neural network, we achieved a 98.23% success rate, making it a more beneficial option for investors. Our aim was to create a reliable feasibility environment.

Keywords: photovoltaic energy; energy estimation methods; artificial neural networks

1. Introduction

As technology has advanced, energy production methods have diversified, and renewable energy methods have become a necessity due to the rapid pollution of the world. They do not pollute the environment and provide sustainability. Many global agreements encourage the use of renewable energy sources to prevent global pollution [2]. Renewable energy methods provide unlimited and free resources and prevent the unconscious consumption of existing reserves, ensuring more conscious use of resources that humanity may need in the future [3].

Solar energy is one of the renewable energy sources that has various applications. Photovoltaic (PV) technology, which allows for the conversion of solar energy into electrical energy using semiconductor technology, is one of these applications. PV cells are the building blocks that convert light into electrical energy using the PV effect [4]. Although PV cells were first discovered in 1839 with low efficiency and requiring a high cost, they have gained the advantages of higher efficiency and lower cost through years

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN : 978-605-69730-2-4 © 2023 University of Bakırçay

of work [4]. In recent years, PV energy has constantly increased its share in the energy market, making it a remarkable energy source [5].

PV energy attracts investors with features such as unlimited and free renewable energy resources, low post-installation maintenance and operating costs, and the ability to be used as a passive income source after the depreciation period. However, feasibility studies are needed to use investments in PV energy more efficiently [6]. The correct estimation of energy production is the most important factor in feasibility studies because the source of income in solar power plants (SPP) depends directly on the electricity produced [7].

Computer software based on classical mathematical methods is commonly used in feasibility studies to forecast energy production [8-10]. These software packages typically estimate annual energy production by processing past meteorological data on a monthly basis using classical linear mathematical algorithms. While the estimation results from these simple algorithms can be used when there are no alternatives in the market, they require significant processing power as they take into account many parameters. Excluding parameters from the algorithms can also lead to deviation of estimation results from actual values.

Advances in computing technology have enabled the solution of non-linear problems [11]. In this context, artificial neural network technology, which mimics biological neurons in the computing environment and can provide predictive results using learning-based algorithms, can be trained for use in feasibility studies of solar power plants (SPPs).

In this study, an artificial neural network was designed, trained with measurement data from the past years of production at the İzmir Bakırçay University SPP and local meteorological stations, and used in estimation studies. Forecasting studies were also conducted using the PVSOL software, which employs classical mathematical algorithms. By comparing success rates, the aim of this study was to present more reliable new-generation methods to investors.

2. Material and Method

In this study, the energy production data from the İzmir Bakırçay University SPP between 2018-2020, as well as meteorological measurements from 5 different stations in İzmir during the same period, were used. The PVSOL software, frequently used in the market, was employed to estimate energy production using classical calculation methods. Additionally, one of the most popular and comprehensive mathematical software programs, MATLAB, was used to train and test artificial neural networks.

2.1. İzmir Bakırçay University

İzmir Bakırçay University was established in the Menemen district of İzmir, which has a coast on the Aegean Sea, in the west of Anatolia. The University aims to produce its own energy by using its own resources and to protect its national capital. The university, which turns to renewable energy sources for the purpose of producing its own energy, has a wind turbine and a SPP in its main campus.

The province of İzmir, where the university is established, is under the influence of the Mediterranean climate. Summers are hot and dry, and winters are warm and rainy [12]. As shown in Figure 1, the geographical region where Türkiye is located has solar radiation in the range of 1095-1826 kWh/m^2. İzmir province has a good solar potential with solar radiation in the range of 1534-1680 kWh/m^2 annually [13].

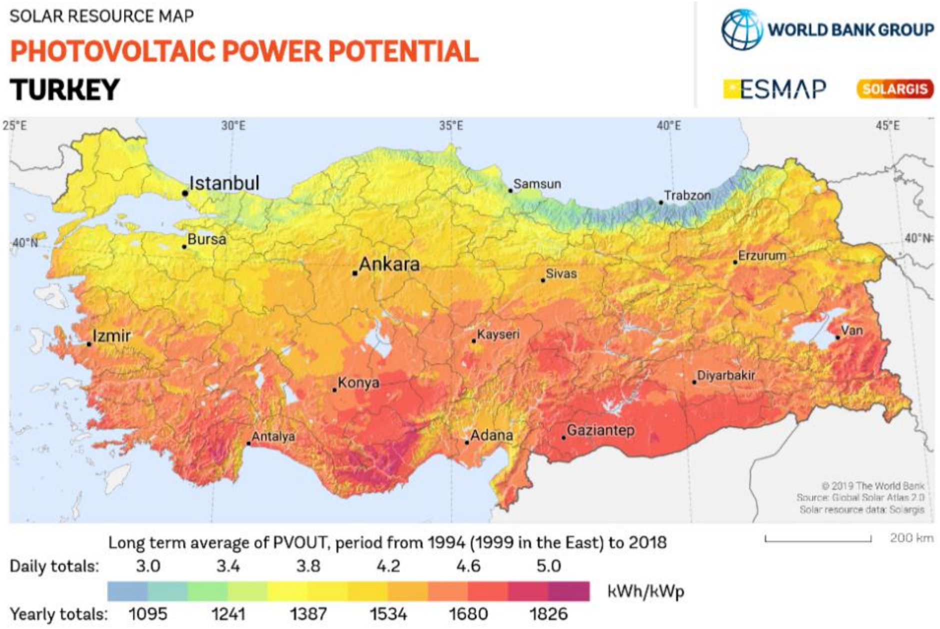

SOLAR RESOURCE MAP
PHOTOVOLTAIC POWER POTENTIAL
TURKEY
WORLD BANK GROUP
ESMAP
SOLARGIS
25°E
30°E
35°E
40°E
45°E
40°N
40°N
Istanbul
Samsun
Trabzon
Bursa
Ankara
Sivas
Erzurum
Izmir
Kayseri
Van
Konya
Diyarbakir
Antalya
Adana
Gaziantep
© 2019 The World Bank
Source: Global Solar Atlas 2.0
Solar resource data: Solargis
200 km
Long term average of PVOUT, period from 1994 (1999 in the East) to 2018
Daily totals: 3.0 3.4 3.8 4.2 4.6 5.0
kWh/kWp
Yearly totals: 1095 1241 1387 1534 1680 1826

Figure 1. PV power potential of Türkiye with Solar Radiation Map [13].

2.2. SPP of İzmir Bakırçay University

The student parking lot located in the northwest of the main campus of İzmir Bakırçay University is designed to provide shade to the vehicles and at the same time generate electricity from solar energy. The satellite image of the campus is shown in Figure 2.

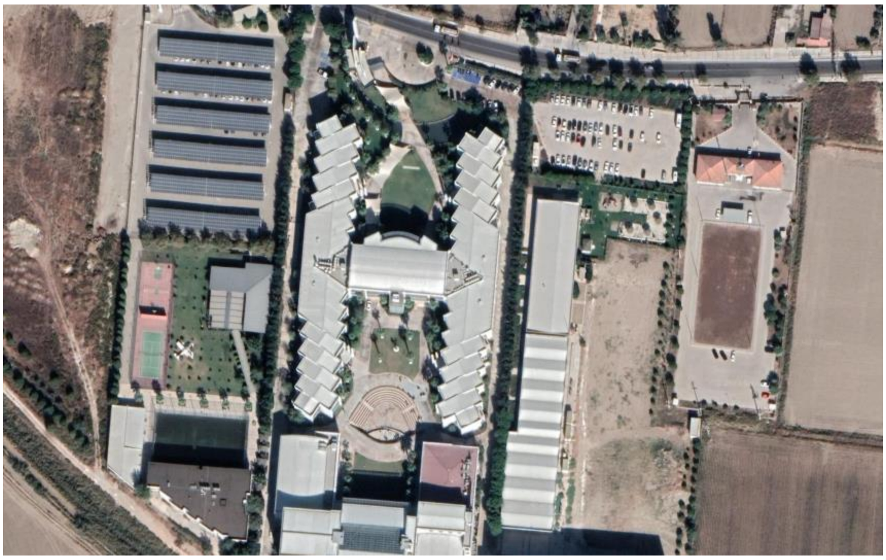


Figure 2. Satellite image of İzmir Bakırçay University main Campus.

There are a total of 1600 solar panels, 265 in 4 rows and 270 in 2 rows, in the parking lot located in the northwest of the campus, built on 6 rows of steel construction. The panels, one of which has a power of 250W, have a total power of 400kW. The technical specifications of the panels used are shown in Table 1.

Table 1. Used specifications panels.

Specification of the Panels	
Producer	PERLIGHT SOLAR
Model Number	PLM-250P-60
Cell Type	Polycrystalline Silicone
Maximum Power (W)	250
Maximum Voltage (V)	31.73
Maximum Current (A)	7.88
Open Circuit Voltage (V)	37.58
Short Circuit Current (A)	8.49
Maximum System Voltage (V)	1000
Cell Size (mm)	156×156
Module Size (mm)	1650×992×40

Considering the geographical location of Türkiye, it would be more appropriate to place the solar panels installed on the steel construction at an angle of 30 degrees facing south, in order to achieve the highest annual average efficiency [14]. However, in order to provide more shade for the parked vehicles, the panels were installed at an angle of 12 degrees to the south. Simulations carried out with the PVSOL program determined that the yield decrease due to the angle difference is 1.26%. The angle of the panels with respect to the surface is illustrated in Figure 3.

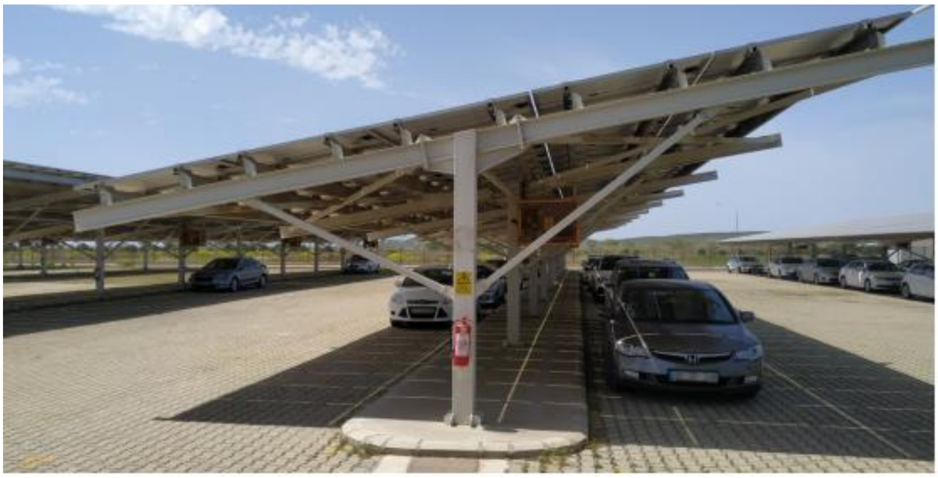


Figure 3. Angle of PV panels with ground.

A total of 24 inverters are used in the SPP in order to rectify the 400kW power produced in 1600 panels with high efficiency. The inverters are housed in steel cages, shown in Figure 3, placed under the panels to protect them from environmental factors. The technical specifications of the used inverters are given in Table 2.

Table 2. Technical specifications of the inverters used.

Specifications of the inverters	
Producer	SMA SOLAR TECHNOLOGY
Model Number	FLX PRO 17
Rated Power (kVA)	17
Number of Phases	3
Output Voltage (V) (Tolerance)	230-400 (+/- 20%)
Maximum Current (Phase-A)	3-21.7A
Rated DC Input Voltage (V)	715
Maximum DC Input Voltage (V)	1000
Maximum Yield (%)	98
Inverter Size (mm)	500×667×233

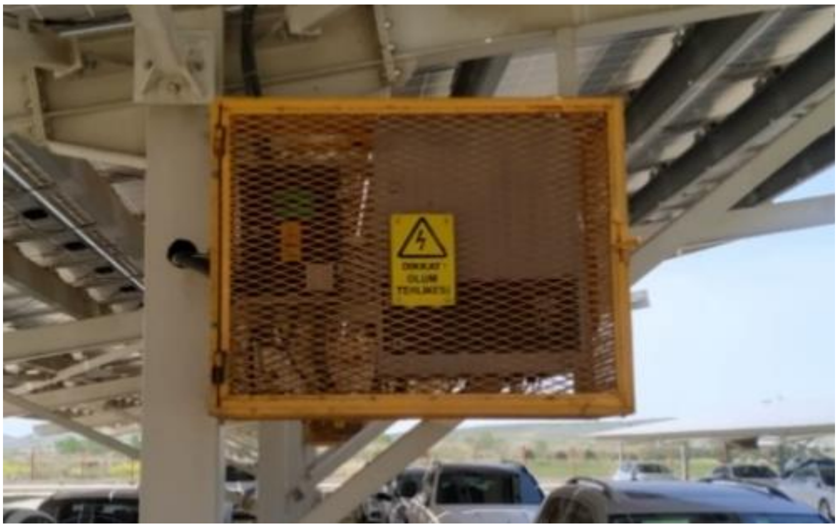


Figure 4. Inverters placed in steel cages under PV panels.

2.3. Meteorological Mesaurement Datas

The amount of energy produced in SPPs is directly dependent on the amount of light received by the panels, which can be affected by various meteorological factors. Therefore, using meteorological measurements to estimate the energy produced can be an effective method. In this study, meteorological data was collected from three different meteorological stations, as indicated by their geographical locations in Figure 4.

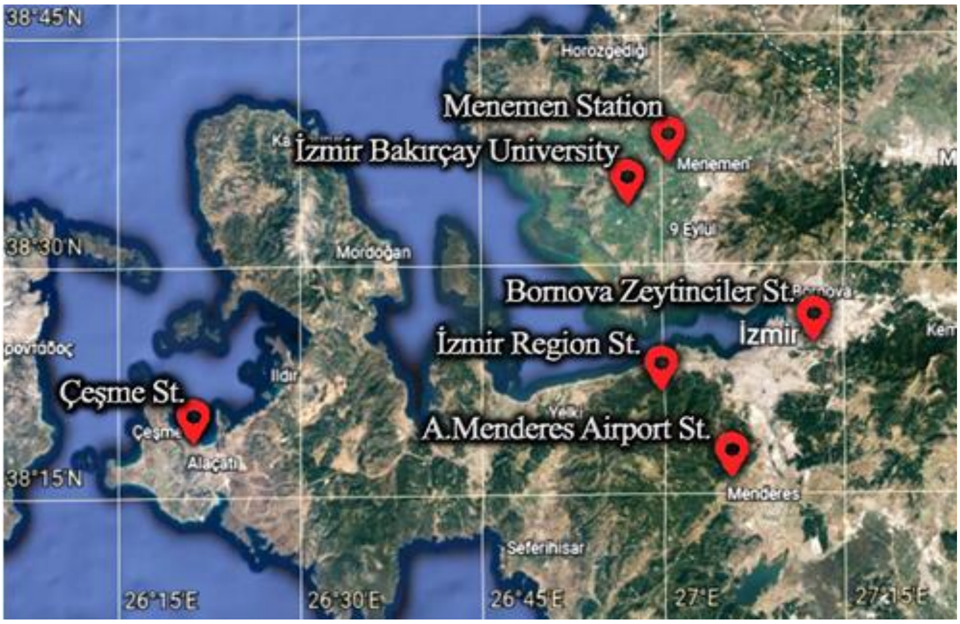


Figure 4. Geographical locations of İzmir Bakırçay University and meteorological measurement stations.

As shown in Figure 4, the closest meteorological measurement station to İzmir Bakırçay University SPP *(N38.58208, E26.96403)* is Menemen Station *(N38.62539°, E27.04255°)*. Other meteorological measurement stations located in the south and west of the region, the measurement data obtained from a total of 4 different meteorological stations, namely Çeşme Station *(N38.30408°, E26.37264°)*, İzmir Regional Station *(N38.39438°, E27.08137)*, Adnan Menderes Airport Station *(N38.29378°, E27.15173°)*, were used [16].

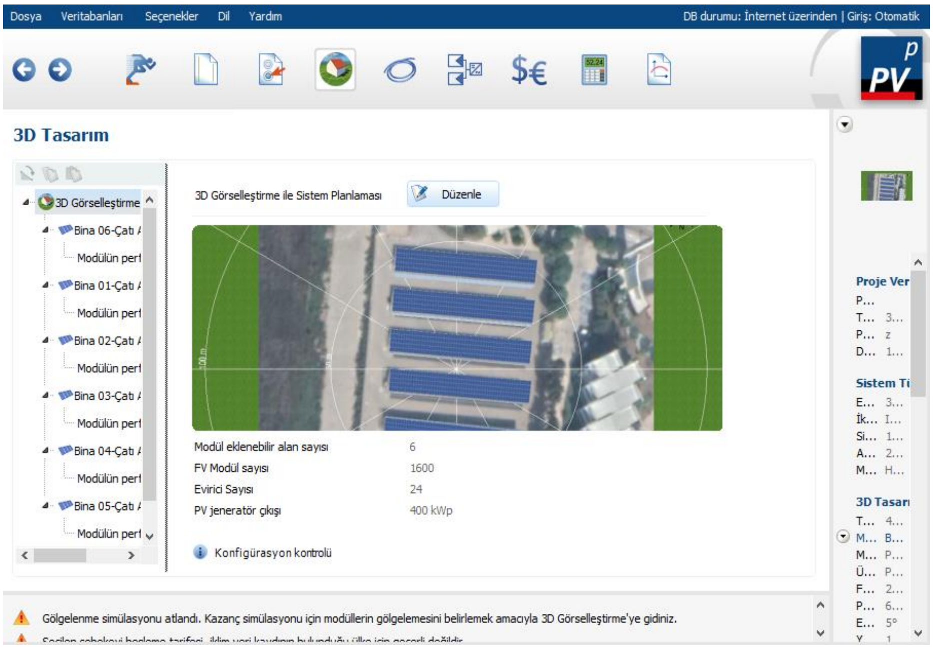
The main meteorological data used in the study consist of the number of cloudy days, solar radiation, average ambient temperature and minimum relative humidity measured between 2018 and 2020 by the relevant meteorology stations.

3. Estimation of Solar Energy Production

PVSOL software is one of the most popular programs for simulating and reporting on the feasibility of solar photovoltaic power plants (SPPs). The program allows users to design an SPP from scratch and generate cost analyses by estimating energy production. It uses meteorological data from previous years in its database to make energy production forecasts based on classical mathematical methods.

3.1. Estimation with PVSOL

PVSOL software is one of the most popular SPP simulation and feasibility reporting programs. The program generally allows you to design an SPP from scratch and then generate cost analysis by generating energy production estimates. Using the meteorological data of the previous years in the database, it makes energy production forecasts based on classical mathematical methods. The interface of the PVSOL program is shown in Figure 5.



Modül eklenebilir alan sayısı	6
FV Modül sayısı	1600
Evirici Sayısı	24
PV jeneratör çıkışı	400 kWp

Konfigürasyon kontrolü

Gölgelenme simülasyonu atlandı. Kazanç simülasyonu için modüllerin gölgelemesini belirlemek amacıyla 3D Görselleştirme'ye gidiniz.

Seçilen şebekeyi besleme tarifesi, iklim veri kaydının bulunduğu ülke için geçerli değildir.

Figure 5. Interface of PVSOL program during İzmir Bakırçay University SPP drawing.

As can be seen in Figure 5, while modeling İzmir Bakırçay University in the PVSOL program, the modeling was carried out considering the parameters given in Table 3.

Table 3: PVSOL SPP installation parameters.

Parameter	Value
Location	İzmir/Türkiye
Data resolution	1 saat
Horizontal diffuse radiation simulation	Hofmann
Simulation of radiation on inclined surface	Hay&Davies
PV module model	PLM-250P-60
Producer	Perlight Solar Co.LTD.
Slope	12°
Layout direction	186°
Inverter model	Danfoss SMA FLX Pro 17
Construction 1 Number of PV modules	270
Construction 2 Number of PV modules	270
Construction 3 Number of PV modules	265
Construction 4 Number of PV modules	265
Construction 5 Number of PV modules	265
Construction 6 Number of PV modules	265
Total number of PV modules	1600
Total number of inverters	24

The simulation results made with the SPP model modeled with the PVSOL software are shown in Table 4.

Table 4. PVSOL simulation results.

Parameter	Value
SPP output	400 kWp
Mains supply in the first year	607,904 kWh/year
Consumption in standby	283 kWh/year
Avoided CO2 emissions	285,583 kg/year
Accumulated cash flow (year 2020)	509,994.30 ₺
Amortization period	9.1 years
Electricity production cost	0.05 ₺ / kWh
Certain investment expenses	1,500.00 ₺ / kWp
Investment costs	600.000₺

The monthly energy production calculated in the simulation is shown in the graph in Figure 6.

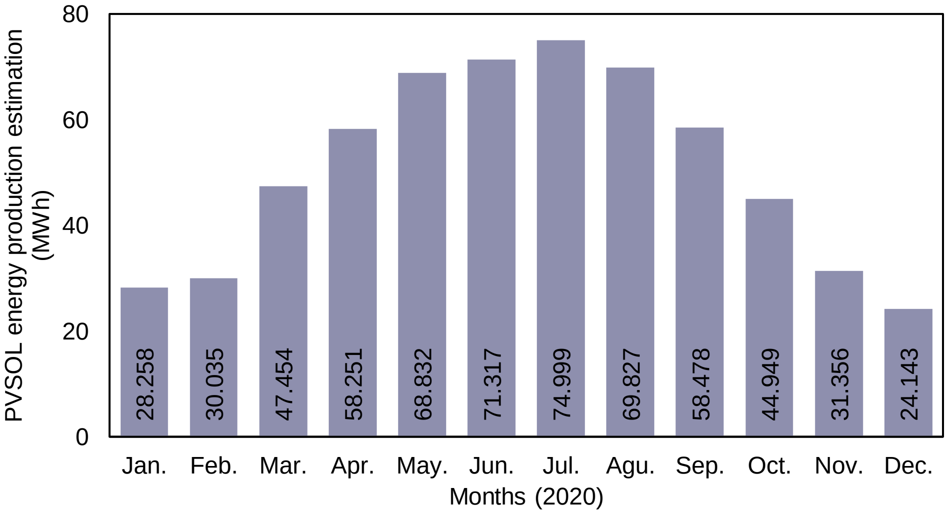


Figure 6. PVSOL annual energy production forecast.

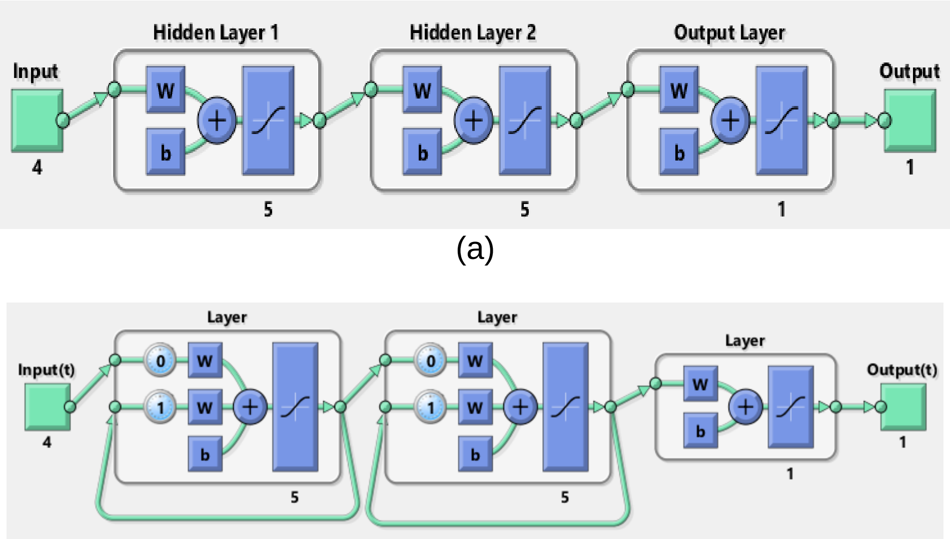
3.2. Estimation with ANN

In this study, data obtained from 5 different measurement stations and the İzmir Bakırçay University Rectorate covering the years 2018-2020 were used to train artificial neural networks (ANNs). The meteorological factors presented as inputs to the ANNs include insolation duration, insolation intensity, number of cloudy days, and average ambient temperature data. The data targeted as output is the monthly energy production data produced by the İzmir Bakırçay University SPP. Of the total 2400 meteorological measurements and power generation data, 86% were used to train ANNs, while the remaining 14% were used for testing ANNs.

The ANNs used in the study were designed with Feed-Forward Backpropagation (FFB) shown in Figure 7a and Elman Backpropagation (ELMB) architecture shown in Figure 7b. These network architectures have initial weights with randomly determined small values in the range of [0,1]. In addition, data with large values in the input data update the weights more effectively than necessary and slow down the training speed [15]. To increase the training speed and success rate, the input data was scaled to the range [0-1] using the Min-Max normalization method shown in Equation 1.

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \qquad [1]$$

In this equation, x'_j represents i th normalized data, x_j represents i th data, x_{min} represents lowest valued data in the dataset and x_{max} represents highest valued data of the dataset.



(a)

(b)

Figure 7. Diagram of architecture and parameters of (a) FFB and (b) ELMB network.

The reason why FFB network is preferred in this study is that it is one of the simplest structures of ANNs. Operations in FFB networks can be simply expressed as a transformation that maps an input sequence to an output sequence with the help of randomly assigned very small weights. [17]. This transformation process is shown in Equation 2.

$$Y_i = f\left(\sum W_{ij} X_j\right) \qquad [2]$$

In this equation, Y_j represents i th output, X_j represents j th input vector and W_{ij} represents ij th vector's weight

While 2 of the 4 different ANN models were designed with FFB architecture, the other 2 were designed with ELMB architecture. After the networks constructed with the parameters specified in Table 5 were trained with meteorological data and energy production data from the past years, they were simulated with a part of the data that was never shown. Tangent sigmoid (TANSIG) and logarithmic sigmoid (LOGSIG) functions were used as activation functions in artificial networks. TANSIG shows nonlinear and dynamic variations in the [-1, 1] range and LOGSIG in the [0, 1] range.

The mean absolute percentage error (MAPE) method, the formula of which is given in Equation 3, was used to calculate the ANN error values.

$$\text{MAPE }(\%) = \frac{\sum_{i=1}^{n} \frac{|y_i' - y_i|}{y_i'}}{n} \times 100 \qquad [3]$$

In this equation,y_i'represents target (actual) output value, y_i represents estimated output value, and n represents total number of outputs.

The success and MAPE% error rates of the networks as a result of the simulation are also shown in Table 5.

Table 5. Training and simulation parameters and results with 4 different ANN models.

Parameter	Value			
Model number	1	3	3	4
Architecture	FFB	FFB	ELMB	ELMB
Training algorithm	TRNGDX	TRNGDX	TRAINLM	TRAINLM
Training function	LRNGDM	LRNGDM	LRNGDM	LRNGDM
Performance function	MSE	MSE	MSE	MSE
Layer count	3	3	3	3
Neurons at layer 1	5	5	5	5
Transfer function at layer 1	TANSIG	LOGSIG	TANSIG	LOGSIG
Neurons at layer 2	5	5	10	10
Transfer function at layer 2	TANSIG	LOGSIG	TANSIG	LOGSIG
Minimum gradient	1.00E-07	1.00E-07	1.00E-07	1.00E-07
Training iterations	400	400	153	107
Best iteration	201	362	103	6
Training regression (%)	98.48	85.87	-	-
Validation regression (%)	99.28	92.89	-	-
Test regression (%)	98.71	90.91	-	-
General regression (%)	98.51	89.17	-	-
Error rate (MAPE%)	4.43	11.09	1.77	9.99
Success rate (%)	95.57	88.91	98.23	90.01

When the performances in Table C are examined, it is seen that the most successful network model is model number 3 designed with ELMB architecture. Also, when models 3 and 4 are compared, using TANSIG as the transfer function gave an 8.22% increase in success compared to using LOGSIG.

When models 1 and 2 designed with FFB architecture are examined, it is seen that using TANSIG as a transfer function provides higher performance and success rate.

4. Comparison of the Findings

SPP output estimation values obtained from PVSOL program and estimation values obtained from ANN model were compared with actual values. The comparison made is shown in the graph in Figure 8.

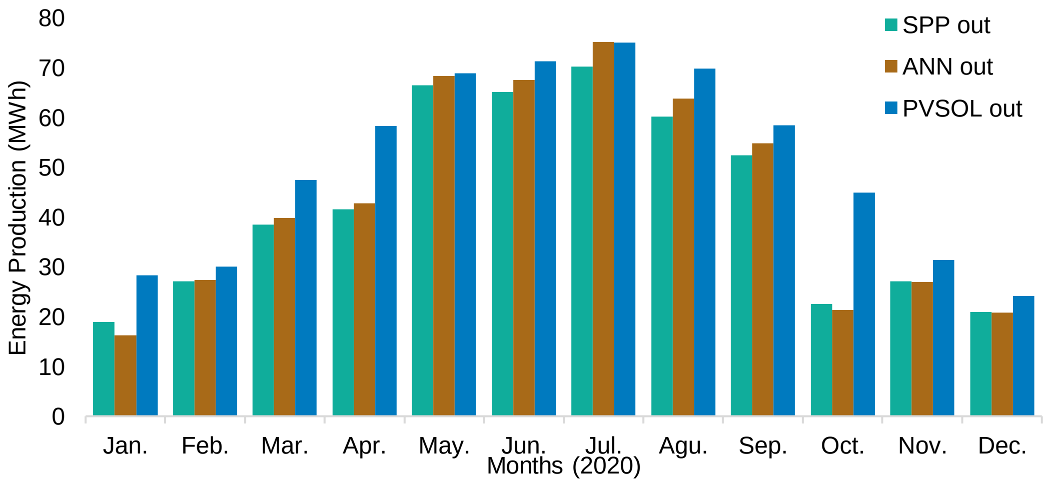


Figure 8. Comparison of the estimate power output results of ANN and PVSOL.

As shown in Figure 8, the estimation results obtained with the PVSOL software are higher than the actual values. While the software provided close estimations to the real values in some months such as February, May, July, September, November, and December, the forecast values in January, March, April, June, August, and October were less accurate. Notably, the error rates in January, April, and October were up to two times higher.

In contrast, the ANN model provided much more accurate estimations, showing more consistent success in all months. For instance, in October, the PVSOL software estimated the electricity generation as 45,553 kWh, while the number 3 ANN model estimated it as 21,314.51 kWh, which is closer to the actual value of 22,500.09 kWh. The comparison of estimation errors of the ANN model and PVSOL model is presented in Figure 9, using MAPE values.

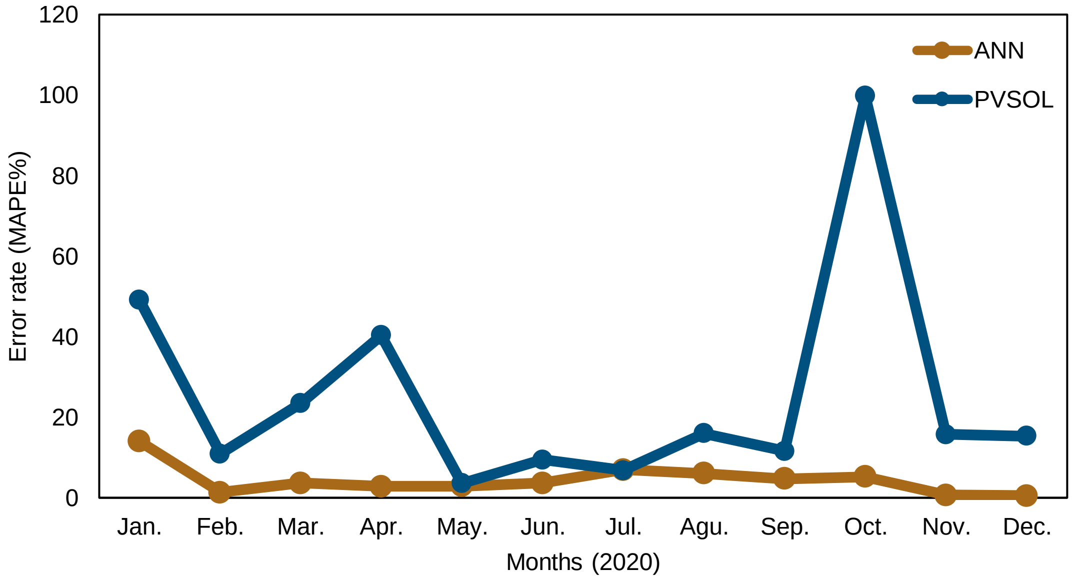


Figure 9. Comparison of the estimation error rates ANN and PVSOL.

5. Result

Before starting the construction of the SPPs, complex mathematical calculations should be used to consider many geographical and meteorological parameters. Classical mathematical calculations require a lot of time for the analyzing SPPs. With the development of technology and the availability of high processing power, artificial intelligence provided by ANNs has become much simpler and more accessible. The idea of using ANNs for forecasting and analysis of PV panels and SPPs has been the subject of many studies recently.

In this study, meteorological data of 4from four different measurement stations located in the same geographical region with the SPP energy production data located in İzmir Bakırçay University Seyrek Campus for the years 2018-2020 were used. The sunshine duration (h), sunshine intensity (kcal/cm2), number of cloudy days (days/month) , average ambient temperature (°C) and SPP production data (kWh) were normalized and presented to the ANNs. Eighty-six percent of the data was used for training and fourteen percent for testing. The data used for training were verified by examining the data on the ANNs. In the trials with test data, estimations were made with an average error rate of 1.77% and 11.09%, according to the MAPE calculation method resulting from four different models.

Regression analyzes and MAPE values were first examined for the successful results of the estimations made. Considering the regression analyzes and MAPE values, ANN modeling with ELMB architecture, using TANSIG transfer function and trainlm learning algorithm, achieved the most successful result, reaching a MAPE rate of 1.77% and a training regression value of 98.23%.

References

[1] Mohtasham, J. (2015). Review Article-Renewable Energies. *Energy Procedia,* 74, pp.1289–1297.

[2] Li, L., Lin, J., Wu, N., Xie, S., Meng, C., Zheng, Y., Zhao, Y., (2020). Review and Outlook on the International Renewable Energy Development. *Energy and Built Environment,* 3(2), pp.139-157.

[3] Panwar, N. L., Kaushik, S. C., & Kothari, S., (2011). Role of renewable energy sources in environmental protection: A review. *Renewable and Sustainable Energy Reviews,* 15(3), pp. 1513–1524.

[4] Marques Lameirinhas, Ricardo A., João Paulo N. Torres, and João P. de Melo Cunha. (2022). "A photovoltaic technology review: history, fundamentals and applications" *Energies,* 15(5), pp.1823-1867.

[5] Kumar Sahu, B. (2015). A study on global solar PV energy developments and policies with special focus on the top ten solar PV power producing countries. *Renewable and Sustainable Energy Reviews*, 43, 621-634.

[6] Kassem, Y.; Çamur, H.; Alhuoti, S.M.A., (2020). Solar Energy technology for Northern Cyprus: assessment, statistical analysis, and feasibility study. *Energies*, 13(4), pp. 940-969.

[7] Beltran, H., Perez, E., Aparicio, N., & Rodriguez, P. (2013). Daily solar energy estimation for minimizing energy storage requirements in pv power plants. *IEEE Transactions on Sustainable Energy*, 4(2), pp. 474-481.

[8] Sharma, R., & Gidwani, L. (2017). Grid connected solar PV system design and calculation by using PV*SOL premium simulation tool for campus hostels of RTU Kota. 2017 International Conference on Circuit,Power and Computing Technologies (ICCPCT) 20-21 April 2017, Kollam, India. doi:10.1109/iccpct.2017.8074315

[9] Kaczorowska, D., Leonowicz, Z., Rezmer, J., & Janik, P. (2017). Long term performance of a PV system with monocrystalline PV cells — a case study. *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*. Milan, Italy. doi:10.1109/eeeic.2017.7977838

[10] Milosavljević, Dragana D., Kevkić, Tijana S. and Jovanović, Slavica J. (2022). Review and validation of photovoltaic solar simulation tools/software based on case study. *Open Physics*, 20(1), pp. 431-451.

[11] Villarrubia, G., De Paz, J. F., Chamoso, P., & la Prieta, F. D. (2018). Artificial neural networks used in optimization problems. *Neurocomputing*, 272, 10–16.

[12] URL-1 Izmir Governorship <http://izmir.gov.tr/izmir-hakkinda> [Accessed 12 February 2023]

[13] URL-2 The World Bank Global Solar Atlas Data <https://globalsolaratlas.info/download/turkey> [Accessed 12 February 2023]

[14] Bakirci, K. (2012). General models for optimum tilt angles of solar panels: Turkey case study. Renewable and Sustainable Energy Reviews, 16(8), pp. 6149–6159.

[15] Aksu, G., Güzeller, C. O. & Eser, M. T. (2019). The effect of the normalization method used in different sample sizes on the success of artificial neural network model, *International Journal of Assessment Tools in Education,* 6 (2), pp.170-192.

[16] URL-3 2. Regional Meteorology Directorate - İzmir. <https://izmir.mgm.gov.tr/gozlem-sebekesi.aspx> [Accessed 12 February 2023]

[17] Sazlı, M.H., 2006. A brief review of feed-forward neural networks. *Communications Faculty of Science University of Ankara*, 50(1), pp.11-17.

ACKNOWLEDGEMENT

We would like to express our endless thanks to the late Assoc.Prof.Dr. Selçuk ÖZMEN, who was of great help in obtaining the meteorological data used in this study and passed away during the studies. Rest in peace.

AITA
ARTIFICIAL INTELLIGENCE
THEORY AND APPLICATIONS

■ RESEARCH ARTICLE

A Machine Learning Based Predictive Analysis Use Case For eSports Games

Atakan Tuzcu [a], **Emel Gizem Ay** [a†], **Ayşegül Umay Uçar** [a], **Deniz Kılınç** [a]

[a] Department of Computer Engineering, University of Bakırçay, İzmir, Turkey
[†] aemelgizem@gmail.com, corresponding author

RECEIVED MARCH 5, 2023
ACCEPTED APRIL 25, 2023

CITATION Tuzcu, A., Ay, E.G., Uçar, A. U., & Kılınç. D. (2023). A machine learning based predictive analysis use case for eSports games. *Artificial Intelligence Theory and Applications, 3(*1), 25-35.

Abstract

League of Legends (LoL) is a popular multiplayer online battle arena (MOBA) game that is highly recognized in the professional esports scene due to its competitive environment, strategic gameplay, and large prize pools. This study aims to predict the outcome of LoL matches and observe the impact of feature selection on model performance using machine learning classification algorithms on historical game data obtained through the official API provided by Riot Games. Detailed examinations were conducted at both team and player levels, and missing data in the dataset were addressed. A total of 1045 data were used for training team-based models, and 5232 data were used for training player-based models. Seven different machine learning models were trained and their performances were compared. Models trained on team data achieved the highest accuracy of over 98% with the AdaBoost algorithm. The top 10 features that had the most impact on the prediction outcome were identified among the 47 features in the dataset, and a new dataset was created from team data to retrain the models. After feature selection, the results showed that the accuracy of Logistic Regression increased from 89% to 98% and the accuracy of Gradient Boosting algorithm increased from 96% to 98%.

Keywords: league of legends; riot game; machine learning; random forest; gradient boosting

1. Introduction

Multiplayer Online Battle Arena (MOBA) games are a genre of games that offer a team-based combat experience, requiring strategy, coordination, and skill. The primary objective in MOBA games is to destroy the opponent team's main base. Sports analytics is a method used in analyzing player performance, team strategies, and predicting competitive outcomes by utilizing data obtained from such games. This study was conducted using data from one of the MOBA games, League of Legends (LoL). Similar to other MOBA games, LoL follows a 5v5 game style, where teams consist of 5 players in roles such as top lane, mid lane, jungle, marksman, and support. The tasks of players based on these roles vary according to different strategies. Due to the combination of limited parameters in the game, many possible game strategies can be formed, as the items obtained during the game can elicit different reactions from the characters.

League of Legends (LoL) is a team game, and the data of all five players in the team should be taken into consideration. Poor performance of some players can be

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN : 978-605-69730-2-4 © 2023 University of Bakırçay

compensated to a certain extent, and there is still a possibility of the team winning. Individual player evaluations can lead to inaccurate predictions of game outcomes. When creating the dataset, match data was retrieved using an API, specifically focusing on recent matches that are closer to the current date. The dataset includes data for each player in the match. To perform team-based analysis, the data was grouped by teams and transformed into a new dataset. The analysis was conducted on these two datasets. Another important aspect of this study is to identify the game criteria that most significantly impact the match outcome. To achieve this, feature selection was performed on the dataset to identify the most influential features. These features can assist the team in forming a strategy and put the team in a more advantageous position. The problem in this study is a classification problem. The most used machine learning algorithms for classification problems in the literature were utilized in this project. The goal is to predict the outcome of a match (win or lose) based on team data.

The remainder of the paper is structured as follows: Section two provides a comprehensive literature review on game analytics. Section three briefly describes the materials and methods used in the study, including the dataset collection process, preprocessing techniques, machine learning algorithms, and model performance evaluation criteria. In section four, we present the results of our experimental study, discuss the findings, and analyze the impact of feature selection on model performance. Finally, the conclusion summarizes the entire study.

2. Related Works

Even before the advent of computers and digitalization, data was generated from sports competitions, much like in all activities today. Analysis based on this data allowed for inferences to be made about game strategies that would give teams an advantageous position over others in these competitions.

The study conducted by Y. Yang et al. [2] stands apart from previous studies by incorporating data obtained during the game, in addition to pre-game data. This approach resulted in changes in the expected winning team, based on the in-game data. The researchers chose a logistic regression model as their prediction model and conducted their study on Dota 2. They used their trained model with real-time data and presented their results graphically. Their findings revealed that the team expected to win until the 7th minute of the game was different from the team that eventually won the game. This study illustrates how the use of in-game data can influence the accuracy of the output. However, by solely relying on logistic regression in their trained model, the researchers overlooked other models that could potentially have resulted in higher accuracy.

In their study, A. Silva and colleagues [3] aimed to compare RNN [4] models by leveraging the inherent characteristics of the data. They tested simple RNN, LSTM [5], and GRU [6] models and found that the simple RNN model had the highest accuracy rate. The researchers utilized a dataset where each row represented a minute of the game, with the goal of capturing changes in the data as the game progressed. Their results showed that the simple RNN model achieved a consistent accuracy rate of 76.29%. However, the researchers acknowledged that the model's performance may be affected by game updates and may not work as consistently.

In a separate study, Hitar-Garcia and colleagues [7] utilized pre-game data to predict the winning team in professional matches. They created new features with the aim of revealing the dynamics of player-to-player matchups and relationships. Classification algorithms were employed in alignment with their defined objectives. However, the most critical factors for team success were not addressed.

In another related study, Q. Shen [8] conducted research on data from diamond-ranked games. Popular machine learning algorithms were employed, and a voting classifier was built to predict game outcomes. The accuracy of the voting classifier was found to be 72.68%. However, feature selection was not applied, and the impact of features on game outcomes was not elucidated.

In a study conducted by F. Bahrololloomi and colleagues [9] individual players were evaluated considering their positions and roles in the game. They developed a simple win prediction model that could predict match outcomes when given the names of ten players divided into two teams. They obtained scores for players and teams overall. By considering the highness of the team score, they made predictions and recorded an accuracy rate of 86%.

In the study conducted by T.D. Do and his colleagues, [10] they predicted game outcomes based on the champions chosen by players within the game using deep learning. They achieved a prediction accuracy of 75.1% when predicting game outcomes even before the start of the game, based on the champions chosen by the players.

A project on live professional match prediction was conducted by Victoria Hodge and her colleagues [11] using data from a different MOBA game, DotA 2. They utilized Random Forest and Logistic Regression algorithms. After a 5-minute game of DotA 2, an accuracy of 85% was achieved in the prediction of match outcomes.

In this study, data was collected from the last matches via the API platform. The dataset was analyzed for both teams and players. The classification algorithms were trained on both datasets. In addition, feature selection was performed to identify the most important factors affecting the outcome of the match. Seven different machine learning algorithms, which are commonly used in the literature for classification problems, were employed, and as a result of the trainings, a success rate of 98% was achieved.

3. Materials and Methods

3.1. Dataset

The dataset used in this study was created by obtaining game data from an online platform through the Riot API, which is provided by the game's developer. The Riot API is a tool used by developers to integrate Riot Games into their applications. Although Riot Games offers numerous APIs to researchers, only two were utilized in this project. Figure 1 illustrates the data extraction steps for the API used in this study.

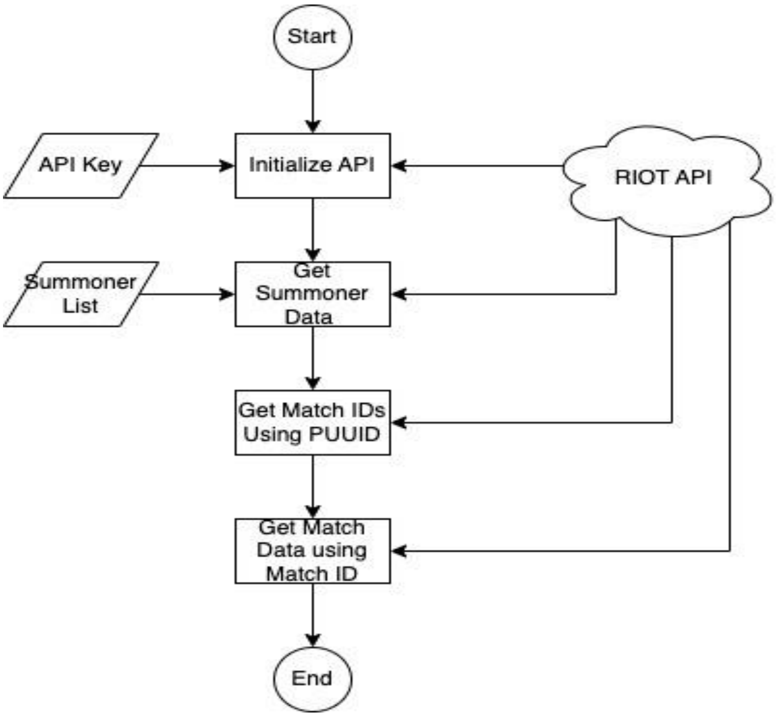


Figure 1. Data Collection with RIOT API

Summoner-v4: "Summoner v4" refers to the fourth version of the API that provides access to user account-related data in the game, League of Legends. This API version allows access to user account information, champion statistics, match history, and other account-based data. The API used in this study offers 6 different methods for obtaining summoner information. The method used in this research retrieves summoner data using the summoner name and stores the response value for retrieving the PUUID, a unique value for each summoner. This method is a GET method that requires the summoner name and region as input and returns a summoner object as response.

Match-v5: "Match v5" refers to the fifth version of the League of Legends API provided by Riot Games. This version allows access to in-game match data and enables retrieval of detailed information about matches. This API offers three methods that developers can use to retrieve information on match games. In our study, we utilized two of these methods to obtain game IDs and subsequently access each game's data. These methods are both GET methods, with one taking the PUUID as a parameter and responding with game IDs, while the other takes game IDs as a parameter and responds with the corresponding game data.

The datasets are labeled with a binary label, where 0 indicates losing team and 1 indicates winning team. The numeric features of the datasets are presented in Table 1.

Table 1. Numerical Information of The Classes in The Datasets

Dataset	Data Groups (Labels)	Data Counts	Feature Count	Total Instance Count
DS1. Team-Dataset	Loser	587		
	Winner	591	47	1,045
DS2. Player-Dataset	Loser	2,594		
	Winner	2,638	47	5,232

The dataset contains a total of 47 features. Some important attributes in the dataset and their definitions are shown in Table 2.

Table 2. Description Of Some Features

Feature Name	Description
turretsLost	The number of towers lost
turretKills	The number of destroyed towers.
inhibitorKills	The number of destroyed inhibitors.
inhibitorTakedowns	The number of inhibitors destroyed by the player.
largestKillingSpree	The highest killing spree count.
deaths	The number of deaths of the player.
damageDealtToObjectives	Damage dealt to objectives.
totalTimeSpentDead	The time spent dead in the game.
kills	The number of kills.

When examining the data distribution based on classes in the dataset, it is known that the current dataset is balanced, meaning that the data is evenly distributed among different classes. The dataset was split into training and test data with a test data ratio of 20%. The data used in the test set was not used in any way in the training set. The 80/20 ratio [12] is often used because it provides a reasonable balance between having enough data for training a machine learning model and having enough data for evaluating the model's performance.

The allocation of 80% of the data as the training dataset allows the model to learn the underlying patterns and relationships. The remaining 20% serves as an independent test dataset to evaluate the model's performance and assess its ability to generalize to unseen data. This ratio was chosen based on the size of the dataset.

3.2. Pre-processing

In this stage of the study, the game data collected with the RIOT API was pre-processed to ensure that the classification models would produce accurate results. Firstly, the attributes in the dataset were examined separately, and missing values were detected in some of the attributes; if these missing values exceeded 80%, they were deleted. The "platform id" and "game id" attributes in the dataset were combined into a single column, and the dataset was grouped based on this column to obtain a team-based dataset. As a result, the DS1 dataset based on teams and the DS2 dataset based on players were obtained for model training.

3.3. Machine Learning Algorithms

The use of machine learning algorithms in game analytics has been increasingly prevalent in recent years. In this study, after preprocessing steps were completed on the dataset, various categories of machine learning classification algorithms were applied to DS1 and DS2 datasets. The selection of classification algorithms for this study was based on a literature review of commonly used algorithms in the field. Ensemble learning algorithms were also included among the chosen algorithms. The classification algorithms used in this study were as follows: Random Forest [13], Decision Tree [14], Logistic Regression [15], LightGBM [16], Naive Bayes Classifier [17], Gradient Boosting [18], and AdaBoost [19]. These algorithms have different approaches and advantages to solve classification problems. Random Forest is an ensemble learning algorithm and one of its features is feature selection, which measures the impact of each feature on prediction. The system workflow is illustrated in Figure 2.

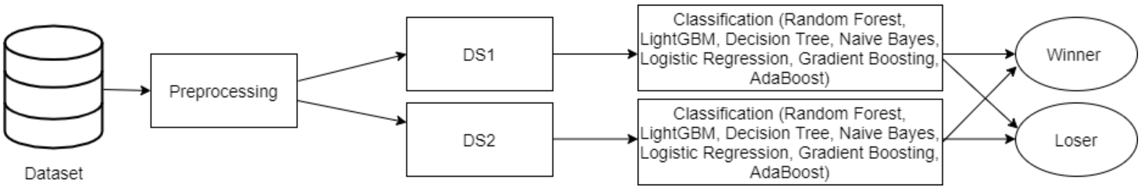

Dataset
Preprocessing
DS1
DS2
Classification (Random Forest, LightGBM, Decision Tree, Naive Bayes, Logistic Regression, Gradient Boosting, AdaBoost)
Classification (Random Forest, LightGBM, Decision Tree, Naive Bayes, Logistic Regression, Gradient Boosting, AdaBoost)
Winner
Loser

Figure 2. Operating schema of the system

3.4. Evaluation Criteria

To evaluate the accuracy of the system that performs classification using machine learning algorithms, a confusion matrix was used. The confusion matrix is a commonly used evaluation matrix in classification problems to assess the performance of a model. It aids in evaluating the performance of a model by comparing the true class labels with the predicted class labels. Table 3 shows the structure of a two-class (positive, negative) confusion matrix [20].

Table 3. Confusion Matrix

		Actual Values	
		Positive	**Negative**
Predicted Values	**Positive**	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

In this study, the performance evaluation metric of accuracy, which can be calculated from the confusion matrix, was utilized to assess the performance of the models. One of the reasons for using this metric is that the dataset is balanced. Accuracy is a commonly used metric to measure the performance of a model. The accuracy value is calculated by the ratio of the total number of correctly predicted classes in the model to the entire dataset. True Positive and True Negative refer to the areas where the model correctly predicted, while False Positive and False Negative refer to the areas where the model incorrectly predicted. The equation for the accuracy metric used to evaluate the model's performance is shown in Equation 1.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \qquad [1]$$

4. Experimental Study and Results

The selected machine learning algorithms were trained with the preprocessed datasets. When examining the results of this training, the most successful model among the team-based dataset was the AdaBoost algorithm with an accuracy rate of 0.9847. On the other hand, the most successful models among the player-based dataset were the Gradient Boosting and LightGBM algorithms with an accuracy rate of 0.95. The accuracy rates of the trained machine learning models are provided in Table 4.

Table 4. Performance Comparison of Models

Algorithm Name	DS1	DS2
Random Forest	0.9732	**0.9533**
Decision Tree	0.9503	0.9388
Naive Bayes	0.8015	0.7265
Logistic Regression	0.8969	0.7624
Gradient Boosting	0.9656	0.9541
LightGBM	**0.9770**	**0.9541**
AdaBoost	**0.9847**	0.9526

The confusion matrices of the top 2 models with the highest accuracy rates for both DS1 and DS2 datasets have been shown. Figures 3 and 4 represent the confusion matrices for the player dataset, while Figures 5 and 6 represent the confusion matrices for the team dataset.

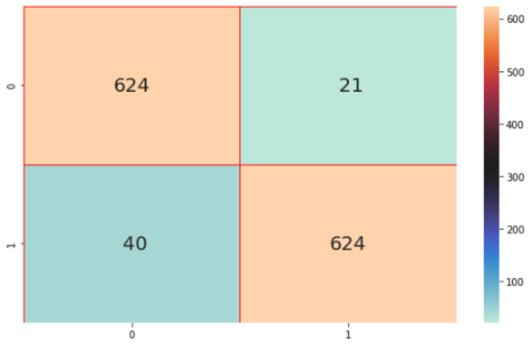


Figure 3. Random Forest

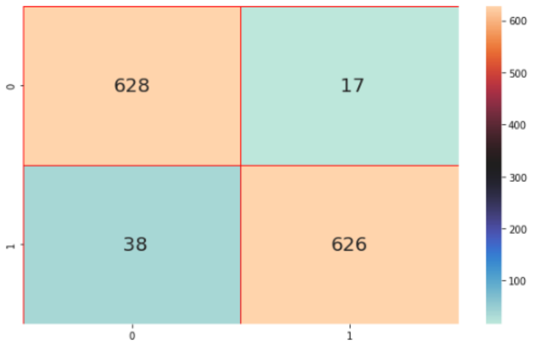


Figure 4. Gradient Boosting

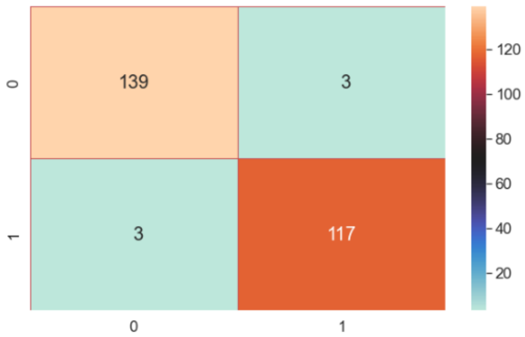


Figure 5. LightGBM

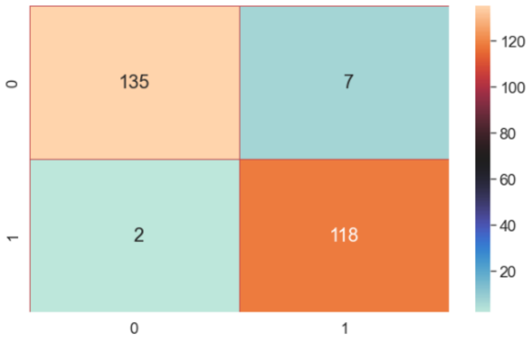


Figure 6. Gradient Boosting

Although the current dataset is not a very large dataset, it is a dataset with a high concentration of numerical data. Upon examining the structures of the algorithms used, their performance, and considering the current dataset, Random Forest and Gradient Boosting algorithms have emerged as prominent options in the study. Both of these models are ensemble models. Random Forest is an ensemble method that combines multiple decision trees to create a prediction model. It offers resistance to noise in the dataset and provides high prediction accuracy with low training time. On the other hand, Gradient Boosting is an ensemble method that progressively improves prediction models and achieves high prediction accuracy. It also provides resistance to noise and is tailored for numerical data.

4.1. Feature Selection

When examining the team dataset, it is known that the total number of data points is 1045 and the dataset contains 47 features. Feature selection is the process of reducing the number of input variables when developing a prediction-based model. It is desirable to reduce the number of input variables to decrease the computational cost of modeling and, in some cases, improve the model's performance [21]. The decision tree algorithms used in the study prune the branches of the tree based on the importance of the input variable.

In this study, a Gini score-based algorithm was used for feature selection, and the top 10 features that have the most impact on classification (Figure 7) were selected to train models and calculate their accuracy values.

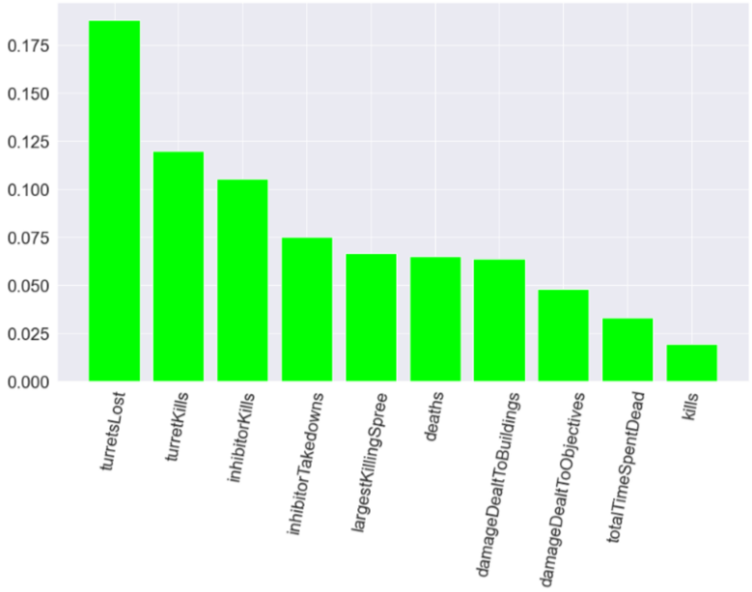


Figure 7. The Results of Gini Score-Based Feature Selection

When analyzed, 10 factors that have the most impact on the outcome of the game can be observed. These factors indicate the qualities that a team should possess against their opponents during the match. Teams can devise strategies based on these qualities.

4.2. The Effect of Feature Selection

In this study, feature selection was performed on a data set with 47 attributes to aim for model training with fewer features. Out of the 7 models trained with the team data set, performance improvement was observed in 5 models. The most significant performance increase was observed in the Naïve Bayes and Logistic Regression algorithms. According to the accuracy values obtained after the feature selection process, the most successful models were Logistic Regression and Gradient Boosting, as seen in Table 5.

Table 5. Performance Comparison of Algorithms After Feature Selection

Algorithm Name	**Accuracy Value Before Feature Selection**	**Accuracy Value After Feature Selection**
Random Forest	0.9732	0.9809
Decision Tree	0.9503	0.9618
Naive Bayes	0.8015	0.9656
Logistic Regression	0.8969	0.9847
Gradient Boosting	0.9656	0.9847
LightGBM	0.9770	0.9770
AdaBoost	0.9847	0.9809

4.2.1. Comparison of Confusion Matrices for Naïve Bayes

After the feature selection process, it was observed that the accuracy value of the Naïve Bayes model increased by 0.16%. The confusion matrices of the algorithm before and after feature selection are shown in Figure 7 and Figure 8, respectively.

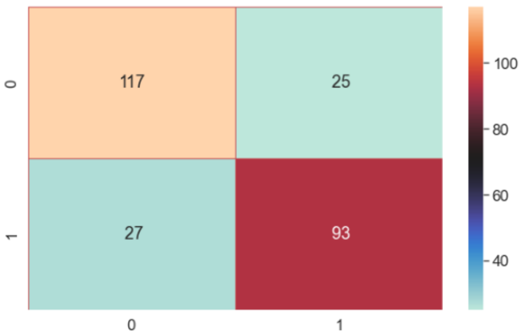


Figure 7. Before Feature Selection

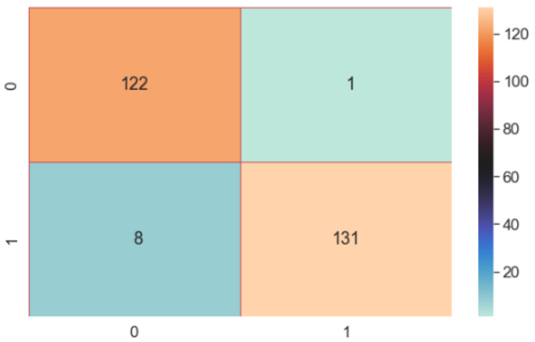


Figure 8. After Feature Selection

It can be observed that the Naïve Bayes algorithm made successful predictions on 210 out of 262 test data before feature selection. After feature selection, it made successful predictions on 253 out of 262 test data. This indicates that the performance of the model has improved after feature selection, as evident in the results.

4.2.2. Comparison of Confusion Matrices for Logistic Regression

After the feature selection process, it was observed that the accuracy value of Logistic Regression model increased by 0.16%. The confusion matrices of the algorithm before and after feature selection are shown in Figure 7 and Figure 8, respectively.

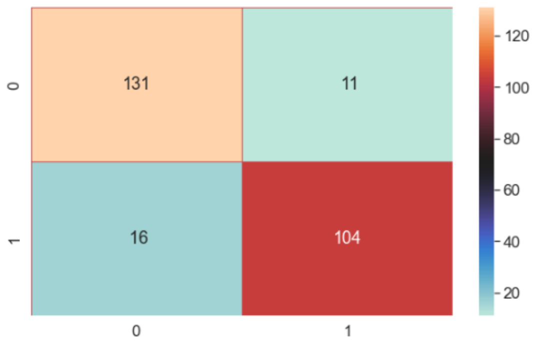


Figure 9. Before Feature Selection

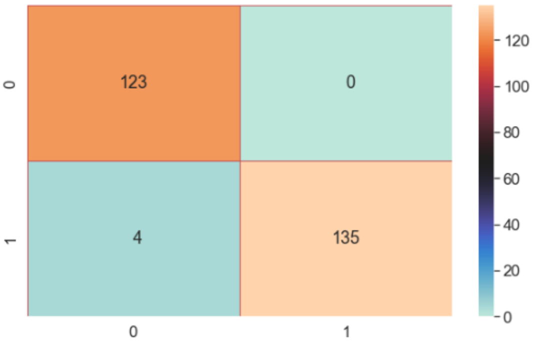


Figure 10. After Feature Selection

4.2.3. Comparison of Confusion Matrices for Gradient Boosting

It has been observed that the accuracy value of the Gradient Boosting algorithm, which is one of the most successful models, increased by 0.01% after the feature selection process. The confusion matrices before and after the feature selection are shown in Figure 11 and Figure 12, respectively.

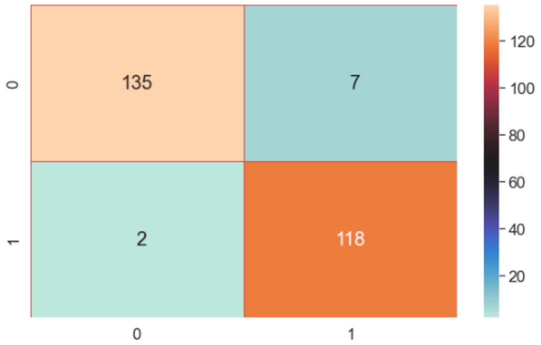


Figure 11. Before Feature Selection

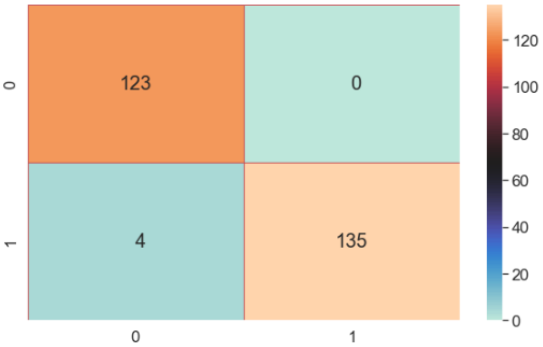


Figure 12. After Feature Selection

5. Conclusion and Future Works

The aim of this study is to predict the outcome of League of Legends games using historical game data obtained through the official API provided by Riot Games. The game data presents a classification problem, and machine learning models including Random Forest, Decision Tree, Logistic Regression, Light GBM, Naive Bayes Classifier, Gradient Boosting, and AdaBoost algorithms were used for classification. The highest accuracy rate of 98.41% was achieved with the AdaBoost algorithm on the team dataset. It was observed that selecting important features and training models with these features can result in high performance and using only 21% of the features in the dataset reduces the workload of the model. After the feature selection process, Logistic Regression and Gradient Boosting were identified as the most successful algorithms with an accuracy rate of 98.41%. It was also observed that the same accuracy rate was achieved with the AdaBoost algorithm without the feature selection process.

The result of this study clearly shows that identifying the most influential features on the game outcome through feature selection provides teams with insights for planning their strategies. Moreover, higher accuracy scores were obtained in machine learning with the support of the feature selection process.

In the future, deep learning models can be constructed and optimized to achieve higher success rates for classification. Moreover, more comprehensive and complex models can be trained with real-time data flow to improve the accuracy of game outcome predictions. Strategies can be provided to players during gameplay.

References

[1] Mora-Cantallops, M., & Sicilia, M. Á. (2018). MOBA games: A literature review. Entertainment computing, 26, 128-138.

[2] Yang, Y., Qin, T., & Lei, Y. H. (2016). Real-time e-sports match result prediction. arXiv preprint arXiv:1701.03162.

[3] Silva, A. L. C., Pappa, G. L., & Chaimowicz, L. (2018). Continuous outcome prediction of league of legends competitive matches using recurrent neural networks. In SBC-Proceedings of SBCGames (pp. 2179-2259).

[4] Medsker, L. R., & Jain, L. C. (2001). Recurrent neural networks. *Design and Applications*, *5*, 64-67.

[5] Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586.

[6] Dey, R., & Salem, F. M. (2017, August). Gate-variants of gated recurrent unit (GRU) neural networks. In 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS) (pp. 1597-1600). IEEE.

[7] Hitar-Garcia, J. A., Moran-Fernandez, L., & Bolon-Canedo, V. (2022). Machine learning methods for predicting league of legends game outcome.

[8] Shen, Q. (2022, February). A machine learning approach to predict the result of League of Legends. In 2022 International Conference on Machine Learning and Knowledge Engineering (MLKE) (pp. 38-45). IEEE.

[9] Bahrololloomi, F., Sauer, S., Klonowski, F., Horst, R., & Dörner, R. (2022). A Machine Learning based Analysis of e-Sports Player Performances in League of Legends for Winning Prediction based on Player Roles and Performances. In VISIGRAPP (2: HUCAPP) (pp. 68-76).

[10] Do, T. D., Wang, S. I., Yu, D. S., McMillian, M. G., & McMahan, R. P. (2021, August). Using machine learning to predict game outcomes based on player-champion experience in League of Legends. In Proceedings of the 16th International Conference on the Foundations of Digital Games (pp. 1-5).

[11] Hodge, V. J., Devlin, S., Sephton, N., Block, F., Cowling, P. I., & Drachen, A. (2019). Win prediction in multiplayer esports: Live professional match prediction. IEEE Transactions on Games, 13(4), 368-379.

[12] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics, 21(1), 1-13.

[13] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[14] Freund, Y., & Mason, L. (1999, June). The alternating decision tree learning algorithm. In icml (Vol. 99, pp. 124-133).

[15] LaValley, M. P. (2008). Logistic regression. Circulation, 117(18), 2395-2399.

[16] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

[17] Murphy, K. P. (2006). Naive bayes classifiers. University of British Columbia, 18(60), 1-8.

[18] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

[19] Schapire, R. E. (2013). Explaining adaboost. In Empirical inference (pp. 37-52). Springer, Berlin, Heidelberg.

[20] Liang, J. (2022). Confusion Matrix: Machine Learning. POGIL Activity Clearinghouse, 3(4).

[21] Fashoto, S. G., Mbunge, E., Ogunleye, G., & den Burg, J. V. (2021). Implementation of machine learning for predicting maize crop yields using multiple linear regression and backward elimination. Malaysian Journal of Computing (MJoC), 6(1), 679-697.

AITA
ARTIFICIAL INTELLIGENCE
THEORY AND APPLICATIONS

■ RESEARCH ARTICLE

A Machine Learning Based Predictive Analysis Use Case For eSports Games

Atakan Tuzcu [a], **Emel Gizem Ay** [a†], **Ayşegül Umay Uçar** [a], **Deniz Kılınç** [a]

[a] Department of Computer Engineering, University of Bakırçay, İzmir, Turkey
[†] aemelgizem@gmail.com, corresponding author

RECEIVED MARCH 5, 2023
ACCEPTED APRIL 25, 2023

CITATION Tuzcu, A., Ay, E.G., Uçar, A. U., & Kılınç. D. (2023). A machine learning based predictive analysis use case for eSports games. *Artificial Intelligence Theory and Applications, 3(*1), 25-35.

Abstract

League of Legends (LoL) is a popular multiplayer online battle arena (MOBA) game that is highly recognized in the professional esports scene due to its competitive environment, strategic gameplay, and large prize pools. This study aims to predict the outcome of LoL matches and observe the impact of feature selection on model performance using machine learning classification algorithms on historical game data obtained through the official API provided by Riot Games. Detailed examinations were conducted at both team and player levels, and missing data in the dataset were addressed. A total of 1045 data were used for training team-based models, and 5232 data were used for training player-based models. Seven different machine learning models were trained and their performances were compared. Models trained on team data achieved the highest accuracy of over 98% with the AdaBoost algorithm. The top 10 features that had the most impact on the prediction outcome were identified among the 47 features in the dataset, and a new dataset was created from team data to retrain the models. After feature selection, the results showed that the accuracy of Logistic Regression increased from 89% to 98% and the accuracy of Gradient Boosting algorithm increased from 96% to 98%.

Keywords: league of legends; riot game; machine learning; random forest; gradient boosting

1. Introduction

Multiplayer Online Battle Arena (MOBA) games are a genre of games that offer a team-based combat experience, requiring strategy, coordination, and skill. The primary objective in MOBA games is to destroy the opponent team's main base. Sports analytics is a method used in analyzing player performance, team strategies, and predicting competitive outcomes by utilizing data obtained from such games. This study was conducted using data from one of the MOBA games, League of Legends (LoL). Similar to other MOBA games, LoL follows a 5v5 game style, where teams consist of 5 players in roles such as top lane, mid lane, jungle, marksman, and support. The tasks of players based on these roles vary according to different strategies. Due to the combination of limited parameters in the game, many possible game strategies can be formed, as the items obtained during the game can elicit different reactions from the characters.

League of Legends (LoL) is a team game, and the data of all five players in the team should be taken into consideration. Poor performance of some players can be

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN : 978-605-69730-2-4 © 2023 University of Bakırçay

compensated to a certain extent, and there is still a possibility of the team winning. Individual player evaluations can lead to inaccurate predictions of game outcomes. When creating the dataset, match data was retrieved using an API, specifically focusing on recent matches that are closer to the current date. The dataset includes data for each player in the match. To perform team-based analysis, the data was grouped by teams and transformed into a new dataset. The analysis was conducted on these two datasets. Another important aspect of this study is to identify the game criteria that most significantly impact the match outcome. To achieve this, feature selection was performed on the dataset to identify the most influential features. These features can assist the team in forming a strategy and put the team in a more advantageous position. The problem in this study is a classification problem. The most used machine learning algorithms for classification problems in the literature were utilized in this project. The goal is to predict the outcome of a match (win or lose) based on team data.

The remainder of the paper is structured as follows: Section two provides a comprehensive literature review on game analytics. Section three briefly describes the materials and methods used in the study, including the dataset collection process, preprocessing techniques, machine learning algorithms, and model performance evaluation criteria. In section four, we present the results of our experimental study, discuss the findings, and analyze the impact of feature selection on model performance. Finally, the conclusion summarizes the entire study.

2. Related Works

Even before the advent of computers and digitalization, data was generated from sports competitions, much like in all activities today. Analysis based on this data allowed for inferences to be made about game strategies that would give teams an advantageous position over others in these competitions.

The study conducted by Y. Yang et al. [2] stands apart from previous studies by incorporating data obtained during the game, in addition to pre-game data. This approach resulted in changes in the expected winning team, based on the in-game data. The researchers chose a logistic regression model as their prediction model and conducted their study on Dota 2. They used their trained model with real-time data and presented their results graphically. Their findings revealed that the team expected to win until the 7th minute of the game was different from the team that eventually won the game. This study illustrates how the use of in-game data can influence the accuracy of the output. However, by solely relying on logistic regression in their trained model, the researchers overlooked other models that could potentially have resulted in higher accuracy.

In their study, A. Silva and colleagues [3] aimed to compare RNN [4] models by leveraging the inherent characteristics of the data. They tested simple RNN, LSTM [5], and GRU [6] models and found that the simple RNN model had the highest accuracy rate. The researchers utilized a dataset where each row represented a minute of the game, with the goal of capturing changes in the data as the game progressed. Their results showed that the simple RNN model achieved a consistent accuracy rate of 76.29%. However, the researchers acknowledged that the model's performance may be affected by game updates and may not work as consistently.

In a separate study, Hitar-Garcia and colleagues [7] utilized pre-game data to predict the winning team in professional matches. They created new features with the aim of revealing the dynamics of player-to-player matchups and relationships. Classification algorithms were employed in alignment with their defined objectives. However, the most critical factors for team success were not addressed.

In another related study, Q. Shen [8] conducted research on data from diamond-ranked games. Popular machine learning algorithms were employed, and a voting classifier was built to predict game outcomes. The accuracy of the voting classifier was found to be 72.68%. However, feature selection was not applied, and the impact of features on game outcomes was not elucidated.

In a study conducted by F. Bahrololloomi and colleagues [9] individual players were evaluated considering their positions and roles in the game. They developed a simple win prediction model that could predict match outcomes when given the names of ten players divided into two teams. They obtained scores for players and teams overall. By considering the highness of the team score, they made predictions and recorded an accuracy rate of 86%.

In the study conducted by T.D. Do and his colleagues, [10] they predicted game outcomes based on the champions chosen by players within the game using deep learning. They achieved a prediction accuracy of 75.1% when predicting game outcomes even before the start of the game, based on the champions chosen by the players.

A project on live professional match prediction was conducted by Victoria Hodge and her colleagues [11] using data from a different MOBA game, DotA 2. They utilized Random Forest and Logistic Regression algorithms. After a 5-minute game of DotA 2, an accuracy of 85% was achieved in the prediction of match outcomes.

In this study, data was collected from the last matches via the API platform. The dataset was analyzed for both teams and players. The classification algorithms were trained on both datasets. In addition, feature selection was performed to identify the most important factors affecting the outcome of the match. Seven different machine learning algorithms, which are commonly used in the literature for classification problems, were employed, and as a result of the trainings, a success rate of 98% was achieved.

3. Materials and Methods

3.1. Dataset

The dataset used in this study was created by obtaining game data from an online platform through the Riot API, which is provided by the game's developer. The Riot API is a tool used by developers to integrate Riot Games into their applications. Although Riot Games offers numerous APIs to researchers, only two were utilized in this project. Figure 1 illustrates the data extraction steps for the API used in this study.

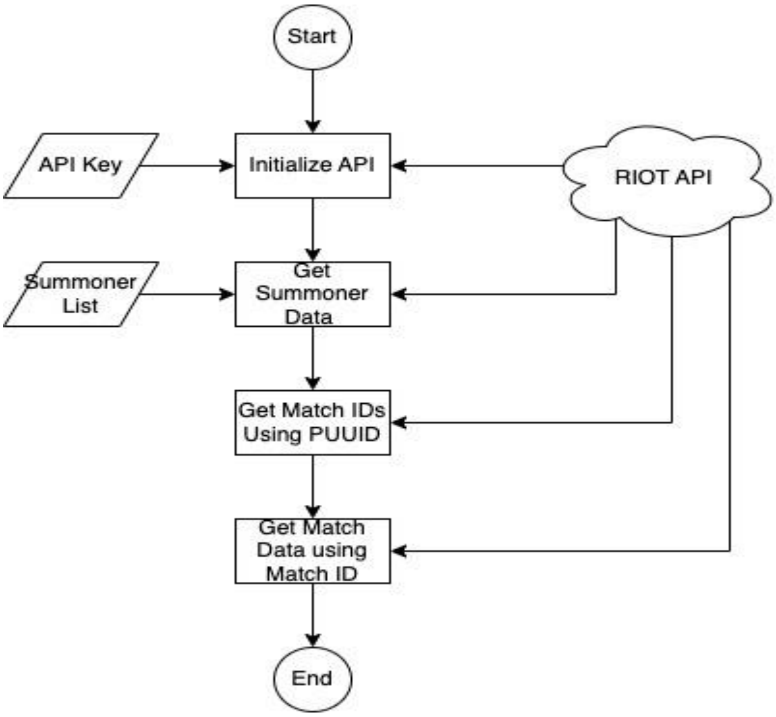


Figure 1. Data Collection with RIOT API

Summoner-v4: "Summoner v4" refers to the fourth version of the API that provides access to user account-related data in the game, League of Legends. This API version allows access to user account information, champion statistics, match history, and other account-based data. The API used in this study offers 6 different methods for obtaining summoner information. The method used in this research retrieves summoner data using the summoner name and stores the response value for retrieving the PUUID, a unique value for each summoner. This method is a GET method that requires the summoner name and region as input and returns a summoner object as response.

Match-v5: "Match v5" refers to the fifth version of the League of Legends API provided by Riot Games. This version allows access to in-game match data and enables retrieval of detailed information about matches. This API offers three methods that developers can use to retrieve information on match games. In our study, we utilized two of these methods to obtain game IDs and subsequently access each game's data. These methods are both GET methods, with one taking the PUUID as a parameter and responding with game IDs, while the other takes game IDs as a parameter and responds with the corresponding game data.

The datasets are labeled with a binary label, where 0 indicates losing team and 1 indicates winning team. The numeric features of the datasets are presented in Table 1.

Table 1. Numerical Information of The Classes in The Datasets

Dataset	Data Groups (Labels)	Data Counts	Feature Count	Total Instance Count
DS1. Team-Dataset	Loser	587		
	Winner	591	47	1,045
DS2. Player-Dataset	Loser	2,594		
	Winner	2,638	47	5,232

The dataset contains a total of 47 features. Some important attributes in the dataset and their definitions are shown in Table 2.

Table 2. Description Of Some Features

Feature Name	Description
turretsLost	The number of towers lost
turretKills	The number of destroyed towers.
inhibitorKills	The number of destroyed inhibitors.
inhibitorTakedowns	The number of inhibitors destroyed by the player.
largestKillingSpree	The highest killing spree count.
deaths	The number of deaths of the player.
damageDealtToObjectives	Damage dealt to objectives.
totalTimeSpentDead	The time spent dead in the game.
kills	The number of kills.

When examining the data distribution based on classes in the dataset, it is known that the current dataset is balanced, meaning that the data is evenly distributed among different classes. The dataset was split into training and test data with a test data ratio of 20%. The data used in the test set was not used in any way in the training set. The 80/20 ratio [12] is often used because it provides a reasonable balance between having enough data for training a machine learning model and having enough data for evaluating the model's performance.

The allocation of 80% of the data as the training dataset allows the model to learn the underlying patterns and relationships. The remaining 20% serves as an independent test dataset to evaluate the model's performance and assess its ability to generalize to unseen data. This ratio was chosen based on the size of the dataset.

3.2. Pre-processing

In this stage of the study, the game data collected with the RIOT API was pre-processed to ensure that the classification models would produce accurate results. Firstly, the attributes in the dataset were examined separately, and missing values were detected in some of the attributes; if these missing values exceeded 80%, they were deleted. The "platform id" and "game id" attributes in the dataset were combined into a single column, and the dataset was grouped based on this column to obtain a team-based dataset. As a result, the DS1 dataset based on teams and the DS2 dataset based on players were obtained for model training.

3.3. Machine Learning Algorithms

The use of machine learning algorithms in game analytics has been increasingly prevalent in recent years. In this study, after preprocessing steps were completed on the dataset, various categories of machine learning classification algorithms were applied to DS1 and DS2 datasets. The selection of classification algorithms for this study was based on a literature review of commonly used algorithms in the field. Ensemble learning algorithms were also included among the chosen algorithms. The classification algorithms used in this study were as follows: Random Forest [13], Decision Tree [14], Logistic Regression [15], LightGBM [16], Naive Bayes Classifier [17], Gradient Boosting [18], and AdaBoost [19]. These algorithms have different approaches and advantages to solve classification problems. Random Forest is an ensemble learning algorithm and one of its features is feature selection, which measures the impact of each feature on prediction. The system workflow is illustrated in Figure 2.

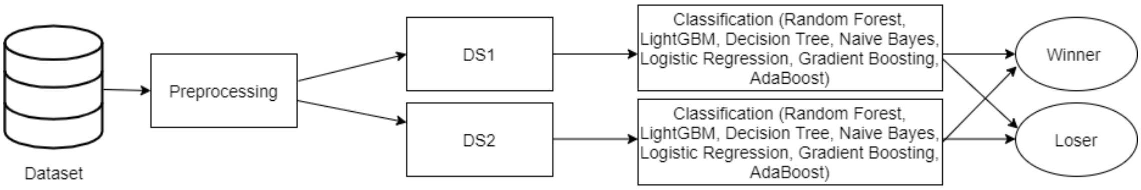

Dataset
Preprocessing
DS1
DS2
Classification (Random Forest, LightGBM, Decision Tree, Naive Bayes, Logistic Regression, Gradient Boosting, AdaBoost)
Classification (Random Forest, LightGBM, Decision Tree, Naive Bayes, Logistic Regression, Gradient Boosting, AdaBoost)
Winner
Loser

Figure 2. Operating schema of the system

3.4. Evaluation Criteria

To evaluate the accuracy of the system that performs classification using machine learning algorithms, a confusion matrix was used. The confusion matrix is a commonly used evaluation matrix in classification problems to assess the performance of a model. It aids in evaluating the performance of a model by comparing the true class labels with the predicted class labels. Table 3 shows the structure of a two-class (positive, negative) confusion matrix [20].

Table 3. Confusion Matrix

		Actual Values	
		Positive	**Negative**
Predicted Values	**Positive**	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

In this study, the performance evaluation metric of accuracy, which can be calculated from the confusion matrix, was utilized to assess the performance of the models. One of the reasons for using this metric is that the dataset is balanced. Accuracy is a commonly used metric to measure the performance of a model. The accuracy value is calculated by the ratio of the total number of correctly predicted classes in the model to the entire dataset. True Positive and True Negative refer to the areas where the model correctly predicted, while False Positive and False Negative refer to the areas where the model incorrectly predicted. The equation for the accuracy metric used to evaluate the model's performance is shown in Equation 1.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \qquad [1]$$

4. Experimental Study and Results

The selected machine learning algorithms were trained with the preprocessed datasets. When examining the results of this training, the most successful model among the team-based dataset was the AdaBoost algorithm with an accuracy rate of 0.9847. On the other hand, the most successful models among the player-based dataset were the Gradient Boosting and LightGBM algorithms with an accuracy rate of 0.95. The accuracy rates of the trained machine learning models are provided in Table 4.

Table 4. Performance Comparison of Models

Algorithm Name	DS1	DS2
Random Forest	0.9732	**0.9533**
Decision Tree	0.9503	0.9388
Naive Bayes	0.8015	0.7265
Logistic Regression	0.8969	0.7624
Gradient Boosting	0.9656	0.9541
LightGBM	**0.9770**	**0.9541**
AdaBoost	**0.9847**	0.9526

The confusion matrices of the top 2 models with the highest accuracy rates for both DS1 and DS2 datasets have been shown. Figures 3 and 4 represent the confusion matrices for the player dataset, while Figures 5 and 6 represent the confusion matrices for the team dataset.

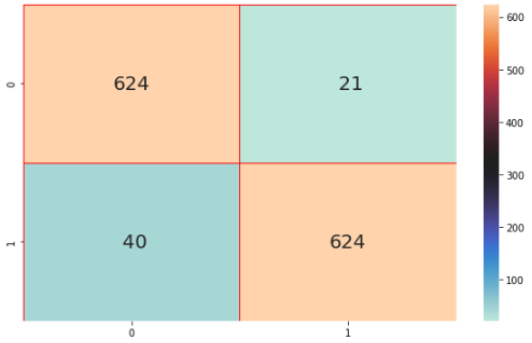


Figure 3. Random Forest

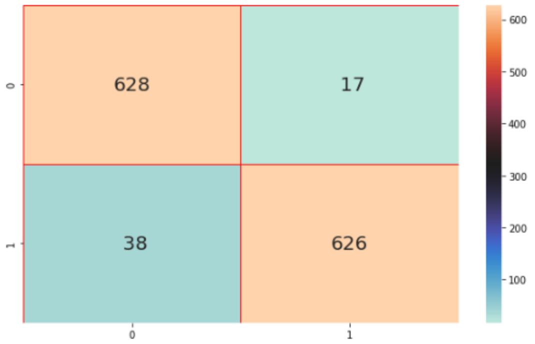


Figure 4. Gradient Boosting

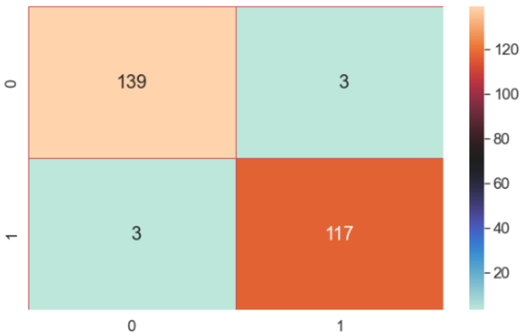


Figure 5. LightGBM

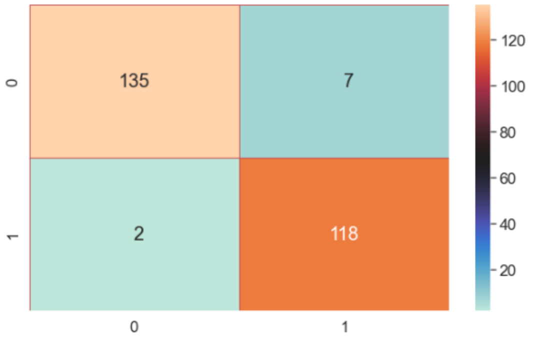


Figure 6. Gradient Boosting

Although the current dataset is not a very large dataset, it is a dataset with a high concentration of numerical data. Upon examining the structures of the algorithms used, their performance, and considering the current dataset, Random Forest and Gradient Boosting algorithms have emerged as prominent options in the study. Both of these models are ensemble models. Random Forest is an ensemble method that combines multiple decision trees to create a prediction model. It offers resistance to noise in the dataset and provides high prediction accuracy with low training time. On the other hand, Gradient Boosting is an ensemble method that progressively improves prediction models and achieves high prediction accuracy. It also provides resistance to noise and is tailored for numerical data.

4.1. Feature Selection

When examining the team dataset, it is known that the total number of data points is 1045 and the dataset contains 47 features. Feature selection is the process of reducing the number of input variables when developing a prediction-based model. It is desirable to reduce the number of input variables to decrease the computational cost of modeling and, in some cases, improve the model's performance [21]. The decision tree algorithms used in the study prune the branches of the tree based on the importance of the input variable.

In this study, a Gini score-based algorithm was used for feature selection, and the top 10 features that have the most impact on classification (Figure 7) were selected to train models and calculate their accuracy values.

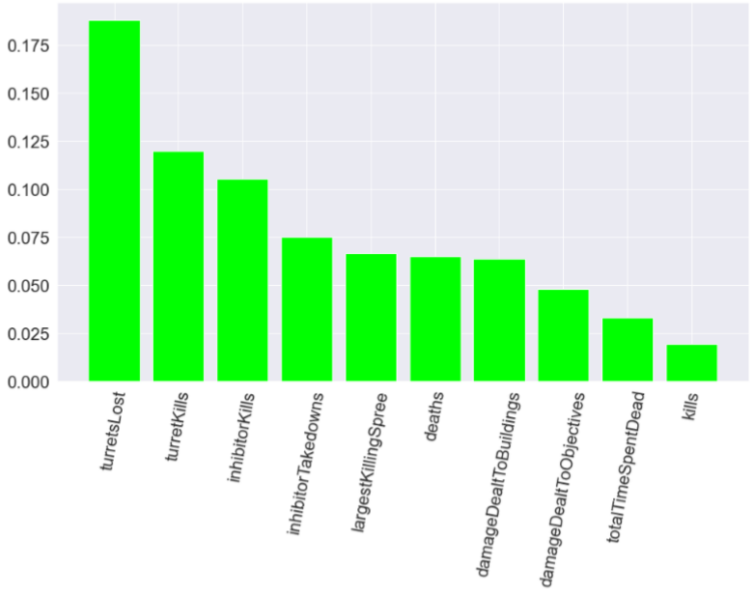


Figure 7. The Results of Gini Score-Based Feature Selection

When analyzed, 10 factors that have the most impact on the outcome of the game can be observed. These factors indicate the qualities that a team should possess against their opponents during the match. Teams can devise strategies based on these qualities.

4.2. The Effect of Feature Selection

In this study, feature selection was performed on a data set with 47 attributes to aim for model training with fewer features. Out of the 7 models trained with the team data set, performance improvement was observed in 5 models. The most significant performance increase was observed in the Naïve Bayes and Logistic Regression algorithms. According to the accuracy values obtained after the feature selection process, the most successful models were Logistic Regression and Gradient Boosting, as seen in Table 5.

Table 5. Performance Comparison of Algorithms After Feature Selection

Algorithm Name	**Accuracy Value Before Feature Selection**	**Accuracy Value After Feature Selection**
Random Forest	0.9732	0.9809
Decision Tree	0.9503	0.9618
Naive Bayes	0.8015	0.9656
Logistic Regression	0.8969	0.9847
Gradient Boosting	0.9656	0.9847
LightGBM	0.9770	0.9770
AdaBoost	0.9847	0.9809

4.2.1. Comparison of Confusion Matrices for Naïve Bayes

After the feature selection process, it was observed that the accuracy value of the Naïve Bayes model increased by 0.16%. The confusion matrices of the algorithm before and after feature selection are shown in Figure 7 and Figure 8, respectively.

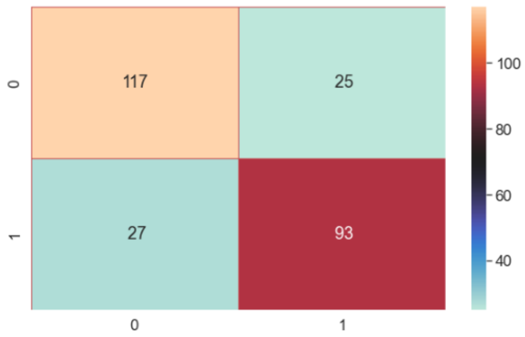


Figure 7. Before Feature Selection

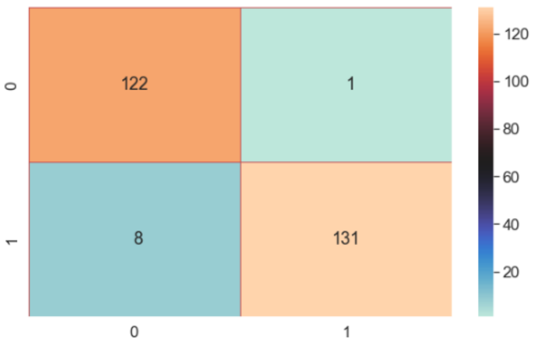


Figure 8. After Feature Selection

It can be observed that the Naïve Bayes algorithm made successful predictions on 210 out of 262 test data before feature selection. After feature selection, it made successful predictions on 253 out of 262 test data. This indicates that the performance of the model has improved after feature selection, as evident in the results.

4.2.2. Comparison of Confusion Matrices for Logistic Regression

After the feature selection process, it was observed that the accuracy value of Logistic Regression model increased by 0.16%. The confusion matrices of the algorithm before and after feature selection are shown in Figure 7 and Figure 8, respectively.

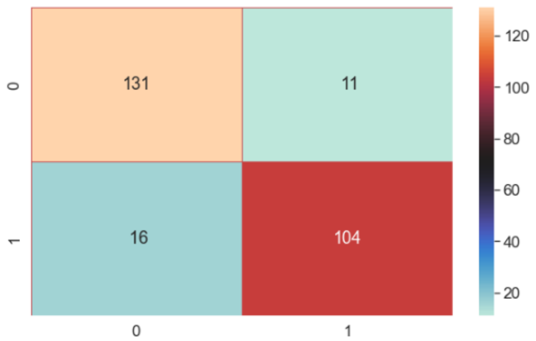


Figure 9. Before Feature Selection

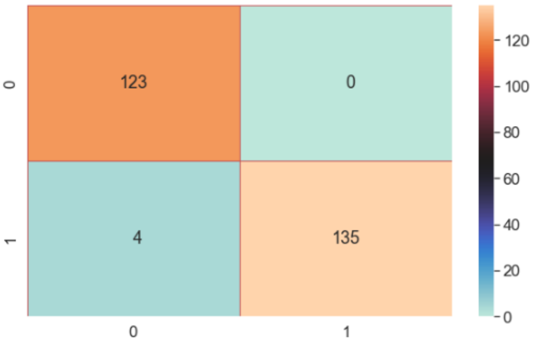


Figure 10. After Feature Selection

4.2.3. Comparison of Confusion Matrices for Gradient Boosting

It has been observed that the accuracy value of the Gradient Boosting algorithm, which is one of the most successful models, increased by 0.01% after the feature selection process. The confusion matrices before and after the feature selection are shown in Figure 11 and Figure 12, respectively.

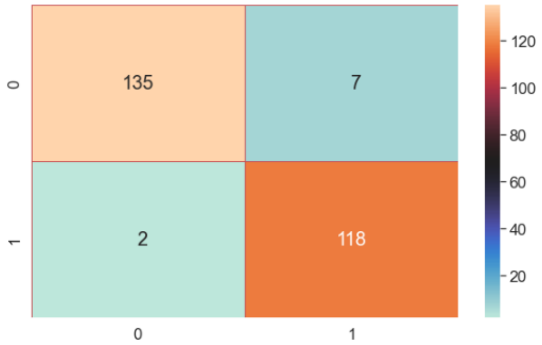


Figure 11. Before Feature Selection

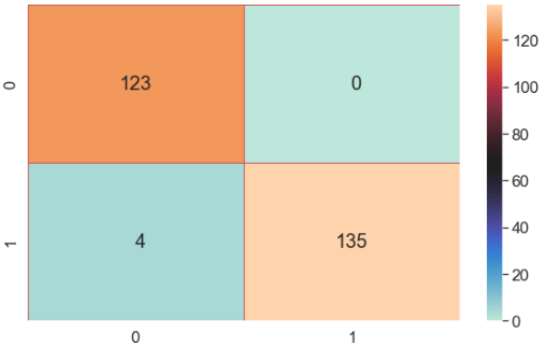


Figure 12. After Feature Selection

5. Conclusion and Future Works

The aim of this study is to predict the outcome of League of Legends games using historical game data obtained through the official API provided by Riot Games. The game data presents a classification problem, and machine learning models including Random Forest, Decision Tree, Logistic Regression, Light GBM, Naive Bayes Classifier, Gradient Boosting, and AdaBoost algorithms were used for classification. The highest accuracy rate of 98.41% was achieved with the AdaBoost algorithm on the team dataset. It was observed that selecting important features and training models with these features can result in high performance and using only 21% of the features in the dataset reduces the workload of the model. After the feature selection process, Logistic Regression and Gradient Boosting were identified as the most successful algorithms with an accuracy rate of 98.41%. It was also observed that the same accuracy rate was achieved with the AdaBoost algorithm without the feature selection process.

The result of this study clearly shows that identifying the most influential features on the game outcome through feature selection provides teams with insights for planning their strategies. Moreover, higher accuracy scores were obtained in machine learning with the support of the feature selection process.

In the future, deep learning models can be constructed and optimized to achieve higher success rates for classification. Moreover, more comprehensive and complex models can be trained with real-time data flow to improve the accuracy of game outcome predictions. Strategies can be provided to players during gameplay.

References

[1] Mora-Cantallops, M., & Sicilia, M. Á. (2018). MOBA games: A literature review. Entertainment computing, 26, 128-138.

[2] Yang, Y., Qin, T., & Lei, Y. H. (2016). Real-time e-sports match result prediction. arXiv preprint arXiv:1701.03162.

[3] Silva, A. L. C., Pappa, G. L., & Chaimowicz, L. (2018). Continuous outcome prediction of league of legends competitive matches using recurrent neural networks. In SBC-Proceedings of SBCGames (pp. 2179-2259).

[4] Medsker, L. R., & Jain, L. C. (2001). Recurrent neural networks. *Design and Applications*, *5*, 64-67.

[5] Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586.

[6] Dey, R., & Salem, F. M. (2017, August). Gate-variants of gated recurrent unit (GRU) neural networks. In 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS) (pp. 1597-1600). IEEE.

[7] Hitar-Garcia, J. A., Moran-Fernandez, L., & Bolon-Canedo, V. (2022). Machine learning methods for predicting league of legends game outcome.

[8] Shen, Q. (2022, February). A machine learning approach to predict the result of League of Legends. In 2022 International Conference on Machine Learning and Knowledge Engineering (MLKE) (pp. 38-45). IEEE.

[9] Bahrololloomi, F., Sauer, S., Klonowski, F., Horst, R., & Dörner, R. (2022). A Machine Learning based Analysis of e-Sports Player Performances in League of Legends for Winning Prediction based on Player Roles and Performances. In VISIGRAPP (2: HUCAPP) (pp. 68-76).

[10] Do, T. D., Wang, S. I., Yu, D. S., McMillian, M. G., & McMahan, R. P. (2021, August). Using machine learning to predict game outcomes based on player-champion experience in League of Legends. In Proceedings of the 16th International Conference on the Foundations of Digital Games (pp. 1-5).

[11] Hodge, V. J., Devlin, S., Sephton, N., Block, F., Cowling, P. I., & Drachen, A. (2019). Win prediction in multiplayer esports: Live professional match prediction. IEEE Transactions on Games, 13(4), 368-379.

[12] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics, 21(1), 1-13.

[13] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[14] Freund, Y., & Mason, L. (1999, June). The alternating decision tree learning algorithm. In icml (Vol. 99, pp. 124-133).

[15] LaValley, M. P. (2008). Logistic regression. Circulation, 117(18), 2395-2399.

[16] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

[17] Murphy, K. P. (2006). Naive bayes classifiers. University of British Columbia, 18(60), 1-8.

[18] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

[19] Schapire, R. E. (2013). Explaining adaboost. In Empirical inference (pp. 37-52). Springer, Berlin, Heidelberg.

[20] Liang, J. (2022). Confusion Matrix: Machine Learning. POGIL Activity Clearinghouse, 3(4).

[21] Fashoto, S. G., Mbunge, E., Ogunleye, G., & den Burg, J. V. (2021). Implementation of machine learning for predicting maize crop yields using multiple linear regression and backward elimination. Malaysian Journal of Computing (MJoC), 6(1), 679-697.

■ RESEARCH ARTICLE

Effects of PCPDTBT:PCBM Ratio on the Electrical Analysis and the Prediction Of I-V Data Using Machine Learning Algorithms for Au/PCPDTBT:PCBM/n-Si MPS SBDs

Ömer Berkan Çelik [a], **Burak Taş** [a†], **Özgün UZ** [a], **Hüseyin Muzaffer Şağban** [b],

Özge Tüzün Özmen [a,c]

[a] Department of Electrical Electronics Engineering, İzmir Bakırçay University, İzmir, Turkey
[b] Beykent University, İstanbul, Turkey
[c] İzmir Bakırçay University, İzmir, Turkey
† burak.tas@bakircay.edu.tr, corresponding author

RECEIVED MARCH 20, 2023
ACCEPTED APRIL 25, 2023

CITATION Çelik Ö.B, Taş B., Uz, Ö., Şağban H.M. & Tüzün Özmen, Ö., (2023). Effects of PCPDTBT:PCBM ratio on the electrical analysis and the prediction of I-V data using machine learning algorithms for Au/PCPDTBT:PCBM/n-Si MPS SBDs. *Artificial Intelligence Theory and Applications, 3(*1), 36-44.

Abstract

In this study, Au/Poly[2,6-(4,4-bis-(2-ethylhexyl)-4H-cyclopenta[2,1-b;3,4-b′]dithiophene)-alt-4,7(2,1,3-benzothiadiazole)] (PCPDTBT) : [6,6]-phenyl C61 butyric acid methyl ester (PCBM) /n-Si heterojunction Schottky barrier diodes (SBDs) with 1:1 and 2:1 PCPDTBT:PCBM doping ratios were produced, and the electrical analysis of metal-polimer-semiconductor (MPS) SBDs with different concentrations was investigated. Ideality factor (n), saturation current values (I_0) and barrier heights (F_0) of the materials were obtained based on the current-voltage (I-V) measurements performed. According to the results obtained, the PCBM concentration has significant effects on the electrical properties of the Au/PCPDTBT:PCBM/n-Si MPS SBD. To predict the electrical characterization of a system in detail, based on its doping concentration, the I-V data set consisting of 2 samples is typically split into a 70% training set and a 30% test set, which is used to train machine learning algorithms. Various methods, including Fine Tree, Cubic SVM, Fine KNN, Boosted Trees, Bagged Trees, Subspace KNN, RUSBoosted Trees, Wide Neural Network, Trilayered Neural Network, and Logistic Regression Kernel, have been analyzed. The obtained results indicate that certain algorithms can predict the I-V data of Au/PCPDTBT:PCBM/n-Si MPS SBD with full accuracy, i.e., 100%.

Keywords: Schottky barrier diode; PCPDTBT:PCBM ratio; electrical analysis; I-V data; machine learning

1. Introduction

The vast majority of electronic devices used today are made of semiconductor materials. Semiconductor devices are commonly used in power-consuming devices, such as computers, televisions, mobile phones, and other electronic devices used in daily life [1]. Organic semiconductors have many advantages, such as easy production technologies,

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN : 978-605-69730-2-4 © 2023 University of Bakırçay

low production costs, and wide surface application areas [2]. Thanks to these advantages, electronic components such as organic field-effect transistors (OFET), organic light-emitting diodes (OLED), organic photodiodes (OPD), organic photovoltaic cells (OPV), and organic Schottky diodes are highly preferred using organic semiconductors [3]. Organic semiconductors are divided into two groups: carbon-based small molecules and polymers [4]. Since polymers are more often in solution, spin coating is the most commonly used organic growth method [5]. In contrast, small molecules are usually grown by methods such as vacuum evaporation or sublimation [6]. Metal-semiconductor (MY) Schottky contacts are widely used in optoelectronics and electronics due to their advantages such as easy conduction even at a voltage value of 0.25 V, low noise levels, and high efficiency [7]. In metal-semiconductor contacts, the performance, efficiency and electrical properties of the material can be directly changed by placing an interface material between the metal and the semiconductor. A Schottky barrier diode called metal-insulator-semiconductor (MIS) can be created by placing an insulating interface between the MY structure, and a Schottky barrier diode called metal-polymer-semiconductor (MPS) can be created by placing a polymer interface between the MY structure [9,10]. To improve the performance of MPS Schottky barrier diodes, it is important to understand and analyze their electrical properties in detail.

In this study, PCPDTBT{Poly[2,6-(4,4-bis-(2-ethylhexyl)-4H-cyclopenta[2,1-b;3,4-b′]dithiophene)-alt-4,7(2,1,3-benzothiadiazole)]}:PCBM{[6,6]-phenyl C61 butyric acid methyl ester} concentrations prepared with 1:1 and 2:1 doping ratios were used as an interface in MPS Schottky barrier diodes, and the electrical parameters obtained from the I-V measurements of these Au/PCPDTBT:PCBM/n-Si (MPS) Schottky barrier diodes in the dark, under vacuum, and at room temperature were investigated.

In the second part of the study, the main aim was to predict the effect of PCPDTBT:PCBM concentration on the electrical characterization of Au/PCPDTBT:PCBM/n-Si MPS SBD. To achieve this goal, 10 different machine learning methods were trained using feature engineering operations on the same data set and their performances were compared.

1. Material and Method

1.1. Fabrication And Characterization of Schottky Barrier Diodes

In this study, PCPDTBT:PCBM organic compounds were purchased from Sigma-Aldrich Company Ltd. The PCPDTBT and PCBM powders were dissolved separately in chloroform at a concentration of 25mg/ml at 30°C and stirred for 3 hours with magnetic stirrers to form a solution to create a polymer interface layer at different doping concentrations. Then, mixtures were prepared at 1:1 and 2:1 concentrations and left to stir overnight at 30°C. N-type (phosphorus-doped), single-crystal silicon (Si) wafer with <100> orientation and a thickness of 200±25μm and a resistivity of 4.8Ω.cm was used as a substrate to produce Au/PCPDTBT:PCBM/n-Si (MPS) Schottky barrier diodes with different PCPDTBT:PCBM doping ratios. The polished side of the Si wafer was chemically cleaned by the Radio Corporation of America (RCA) cleaning method is the basic procedure developed by Werner Kern in 1965 while working at the RCA [11]. The back surface of the cleaned Si wafer was coated with ~1500Å thick silver (Ag) metal without a mask, using a thermal evaporation system. Then, the Ag metal was annealed in a tube furnace under N_2 flow at 450°C for 30 minutes to create a good ohmic contact on the back surface of the n-Si wafer. After the formation of the ohmic contact, the front surface of the n-Si wafer was cleaned with 50% hydrofluoric (HF) acid to remove any thin oxide layer that might have formed. Following this oxide cleaning process, organic compounds with 1:1 and 2:1 (PCPDTBT:PCBM) doping ratios were spin-coated onto the front surface of the samples. The samples were heated at 45^{o}C on a hot plate for 15 minutes to evaporate the solvent in the organic film. To fabricate Au/PCPDTBT:PCBM/n-Si (MPS) Schottky barrier diodes, circular-shaped gold (Au) rectifying contacts with a thickness of ~1500Å were formed on the PCPDTBT:PCBM organic film using a mask containing 1mm diameter circles (Figure 1). The Au coating process also used thermal evaporation system, and the evaporation process was carried out at a pressure of ~1x10^{-6} Torr. At the same time, the thickness of the circular-shaped Au contacts was monitored using a digital thickness measurement monitor in the thermal evaporation system. The schematic representations of the produced MPS SBDs are shown in Figure 2.

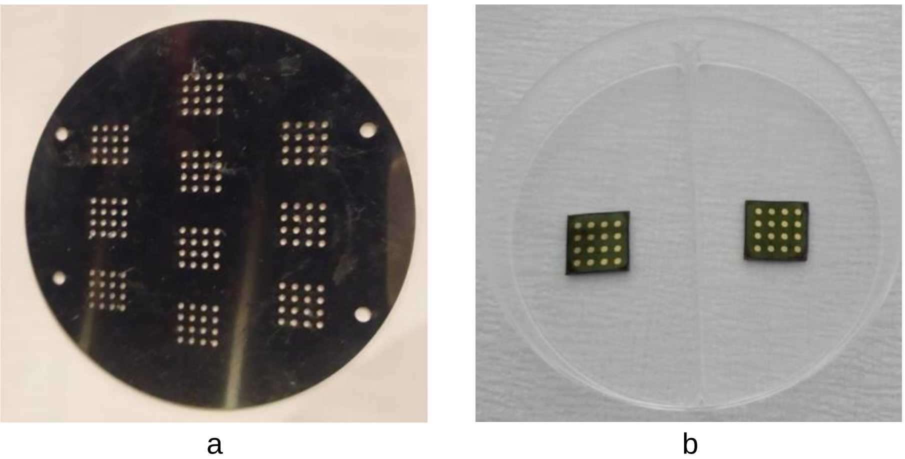


Figure 1. (a) The mask used for the rectifier contacts (b) Top view after coating the rectifier contacts.

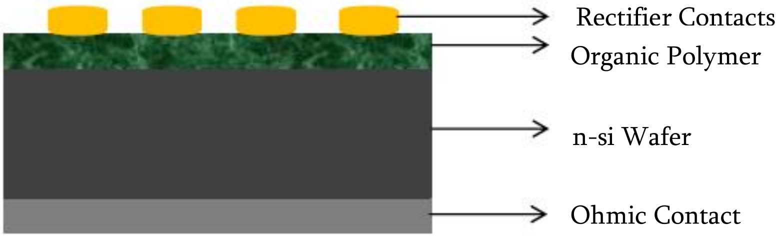


Figure 2. Schematic representation of SBD.

The electrical characterization of Au/PCPDTBT:PCBM/n-Si (MPS) Schottky barrier diodes produced using 1:1 and 2:1 PCPDTBT:PCBM doping ratios were analyzed by I-V measurements performed under vacuum and in the dark at room temperature.

1.2. Machine Learning Algorithms

In essence, machine learning provides computers with the ability to "learn from experience", a capability naturally found in humans. Unlike relying on a pre-determined equation as a model, machine learning algorithms utilize computational methods to extract information directly from data. As the number of available samples for learning increases, these algorithms can adaptively improve their performance [12]. With the ability to make effective and error-free estimations, machine learning algorithms can be used for various purposes such as classification, estimation, and forecasting [13]. Essentially, machine learning aims to predict future outcomes based on past experiences [14]. This is achieved through the use of software design that can learn rules from data, adapt to changes, and improve its performance with experience. The field of machine learning is focused on developing computer programs that can automatically improve their performance using sample data or past experience [15,16].

In this study, different classifier models with varying structures were developed using MATLAB's machine learning toolbox. The objective was to estimate the impact of PCPDTBT:PCBM concentration on the electrical characteristics of Au/PCPDTBT:PCBM/n-Si MPS SBD. The models employed in this study included multi-layered neural networks, NB classifiers, KNN algorithms, DT algorithms, and SVM. All numerical results were obtained using MATLAB R2020 on an Intel processor running Windows 10.

1.2.1. Artificial Neural Networks (ANN)

The artificial neural network (ANN) structure, which is used for classification processes, was designed to include 1, 2, and 3 hidden layers and modelled as a single output and feedback structure.

1.2.2. Support Vector Machine (SVM)

The SVM classification approach is a two-step process. In the first step, the classifier's high-dimensional input is non-linearly mapped to another attribute space. In the second step, a new linear hyperplane is created from this attribute space to maximize the separation between the samples' parts [16].

1.2.3. Decision Tree Classifier Algorithm

One of the most widely used machine learning algorithms is tree-based learning, which falls under the category of data mining classification algorithms. In this approach, a set of multiple decision trees is created to train a model. The decision tree structure resembles a flowchart that tests attributes to determine the sample corresponding to each internal node. Each branch represents a test result, and each node represents a class. Each decision tree is built using randomly selected input data values.

1.2.4. k-Nearest Neighbor Algorithm (KNN)

The fundamental principle of the k-nearest neighbor algorithm in classification problems is that a chosen value of K can identify the nearest neighbor of a given data point. The data point is then assigned to the class with the highest frequency among its K nearest neighbors. The K value refers to the number of neighboring data points considered in the classification process.

1.2.5. Naive Bayes

The Naive Bayes classifier is a statistical classification model that is based on Bayes' theorem. It assumes that the effect of a particular attribute on a class is independent of other attributes, even if they are correlated. This simplifying assumption makes the calculations easier and is referred to as "naive".

2. Results and Discussions

In this study, measurements of Au/PCPDTBT:PCBM/n-Si heterojunction SBDs with 1:1 and 2:1 PCPDTBT:PCBM doping ratios were performed in a closed circuit cryostat at approximately 1x10^{-4} mbar pressure, at 300K room temperature in the dark, within the voltage range of -3V to +3V. Figures 3 and 4 show the I-V curves of the diodes fabricated using 1:1 and 2:1 PCPDTBT:PCBM doping ratios, respectively.

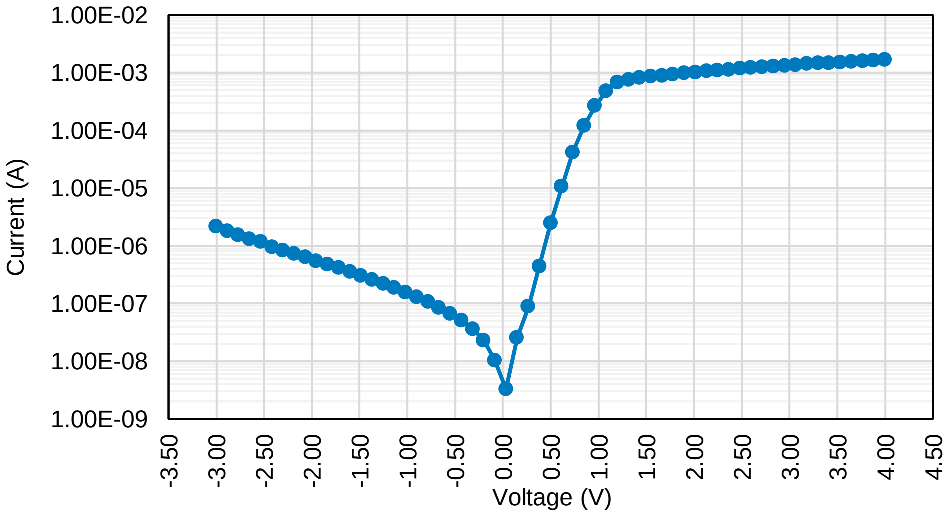


Figure 3. I-V characterization of 1:1 PCPDTBT:PCBM used Au/PCPDTBT:PCBM/n-Si MPS SBDs.

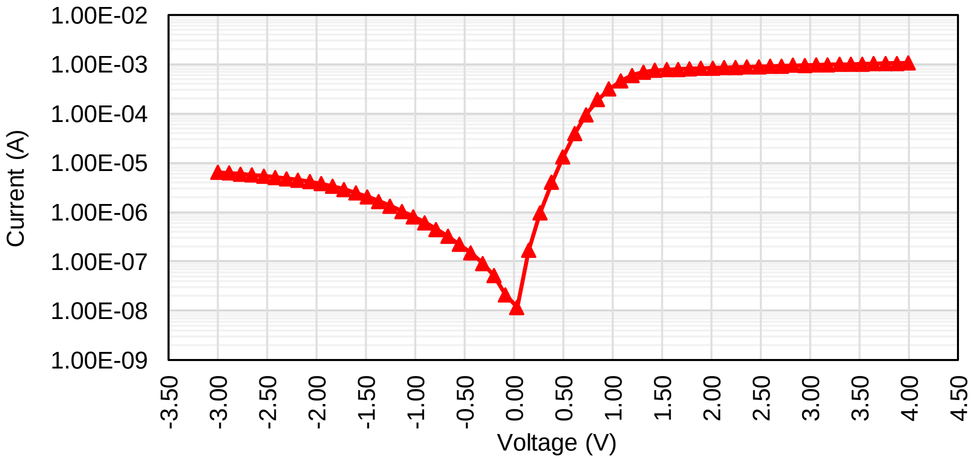


Figure 4. I-V characterization of 2:1 PCPDTBT:PCBM used Au/PCPDTBT:PCBM/n-Si MPS SBDs.

Ideality factors (n), barrier heights (ϕ_B) values, and saturation current values (I_o) values were calculated from the current-voltage (I-V) characteristics of the produced diodes. The ideality factors were obtained from the slope of the linear region of the obtained I-V graphs using the following equation [17].

$$n=q/kTtan\theta \quad [1]$$

According to equality [1], q:electron charge, k:boltzman constant and T:temprature(in ^{o}K). The ideality factors(n) obtained using the above expression are 2.97 for the diode with 1:1 PCPDTBT:PCBM ratio and 3.09 for the diode with 2:2 PCPDTBT:PCBM ratio.

The $\emptyset_B$ values are obtained using the following e$q$uation:

$$\emptyset_B = \frac{kT}{q}\ln\left(\frac{AA^{*}T^{2}}{I_0}\right) \quad [2]$$

In this equation, I_o is the saturation current value, which is obtained from the current values obtained from the I-V curve at the point where the voltage is zero. The saturation current value of the diode with a 1:1 PCPDTBT:PCBM ratio was obtained as $6.92\text{x}10^{-9}$ A in the dark. Using the saturation current, the $\emptyset_B$ value obtained using the Equation 2 was calculated as 0.88 eV.

When the same procedures were applied to the diode produced with 2:1 PCPDTBT:PCBM ratio, the saturation current value was calculated as $1.22\text{x}10^{-8}$ A and the $\emptyset_B$ value was calculated as 0.87 eV. All the obtained values are given in Table 1.

Table 1. Comparison of characteristic features of SBDs.

PCPDTBT:PCBM Ratio	***n***	***I_o(A)***	$\emptyset_B$(eV)
1:1	2.97	6.92×10^{-9}	0.88
2:1	3.09	1.22×10^{-8}	0.87

The comparison of the I-V curves obtained in the dark at 300K for Au/PCPDTBT:PCBM/n-Si heterojunction SBDs with 1:1 and 2:1 PCPDTBT:PCBM doping ratios is given in Figure 5.

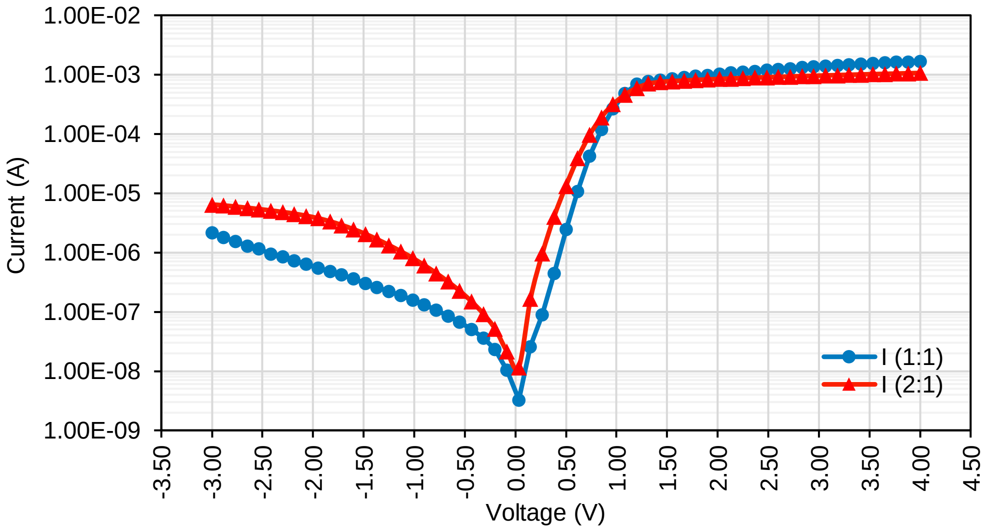


Figure 5. Comparison I-V characterization of 1:1 and 2:1 PCPDTBT:PCBM based SBD.

As shown in Figure 5, the diode produced using a 1:1 PCPDTBT:PCBM doping ratio exhibits more ideal behaviour in terms of their ideal factor, maximum current values, and leakage currents when compared to the diode produced using a 2:1 PCPDTBT:PCBM doping ratio. As previously calculated, the ideal factor of the diode with a 1:1 doping ratio is 2.97, while the ideal factor of the diode produced with a 2:1 doping ratio is 3.09. This is thought to be due to a reduction in defects and cracks within the structure as the PCBM doping increases.

RR (Rectification ratio) values of both diodes were calculated using the ratio of the forward current value at +3V and -3V voltage values to the current value at reverse bias. The RR of the diode with a 1:1 PCPDTBT:PCBM blend ratio was calculated as 7.77x10^2. RR of the diode with a 2:1 blend ratio was obtained as 1.65x10^2. The high RR is an important parameter for electronic applications [18]. It is crucial for diodes that will be used in rectifier circuit designs to have a high RR as it will provide better current control and higher efficiency during rectification. Additionally, the obtained RRs show that there is a higher injection of charge into the polymer layer in the forward bias state and much less in the reverse bias state as the PCBM blend ratio increases [19].

In conclusion, a significant increase in RR was observed in the Au/PCPDTBT:PCBM/n-Si heterojunction SBD with an increase in PCBM blend ratio. Therefore, the heterojunction SBD with a 1:1 PCPDTBT:PCBM blend ratio exhibits better diode properties.

In this part of the study, the performance of the proposed machine learning algorithms for I-V data estimation was investigated. The evaluation of PCBM concentration prediction was carried out using the dataset of Au/PCPDTBT:PCBM/n-Si MPS SBD. The dataset was used contains I-V values as input data and mixing ratio as output data. Traditional validation and k-fold cross-validation approaches were utilized to evaluate the proposed algorithms. The I-V data were tested with various machine learning techniques such as Logistic Regression, NB, Linear SVM, Cubic SVM, Quadratic SVM, Fine Gauss SVM, Medium Gaussian SVM, Coarse Gaussian SVM, Fine KNN, Medium KNN, Coarse KNN, Cosine KNN, Cubic KNN, Weighted KNN, Subspace KNN, Boosted Trees, Bagged Trees, Fine Tree, Medium Tree, Coarse Tree, RUSBoosted Trees, Subspace

Discriminant, and Multi-layer Neural Network (with 1–2–3 hidden layers) methods. The classification performance results obtained with different classifiers using all features (with PCA or not) are presented in Table 2.

Table 2. Comparison of ANN methods.

Model No	Model name	Model Type	Success Rate (%)	Cost	Estimation speed	Estimation time (s)
1	Tree	Fine Tree	30.00	42	19629	7.909
2	SVM	Cubic SVM	100.00	0	50.466	24.974
3	KNN	Fine KNN	100.00	0	9176.4	0.58721
4	Ensemble	Boosted Trees	100.00	0	1876.3	0.84399
5	Ensemble	Bagged Trees	100.00	0	1660.2	27.864
6	Ensemble	Subspace KNN	100.00	0	650.78	27.338
7	Ensemble	RUSBoosted Trees	35.00	39	1206.8	20.562
8	Neural Network	Wide Neural Network	100.00	0	21242	32.664
9	Neural Network	Trilayered Neural Network	100.00	0	15951	15.843
10	Kernel	Logistic Regression Kernel	100.00	0	55.45	45.832

In this study, the proposed machine learning algorithms showed more successful accuracy performance for the same dataset compared to other machine learning methods in the literature. Despite the imbalance in the dataset, Cubic SVM, Fine KNN, Boosted Trees, Bagged Trees, Subspace KNN, Wide Neural Network, Trilayered Neural Network and Logistic Regression Kernel algorithms that showed the best performance in I-V data achieved a successful prediction score (100%).

When examining the performance analysis of different feature engineering methods on the dataset, it was found that the structure of the used dataset is decisive. As shown in Table 2, data segmentation reduces performance in all methods similarly. In addition, different parameter variations were tried in the PCA method, and the best results are given in the table. When these results were examined, it was seen that they did not affect the performance much.

3. Conclusion

Measurements of Au/PCPDTBT:PCBM/n-Si heterojunction SBDs with 1:1 and 2:1 PCPDTBT:PCBM doping ratios were performed in a closed environment creostat at a pressure of approximately 1x10^{-4} mbar with a voltage range of -3V to +3V in the dark at 300K room temperature. When both diodes were compared, it was found that the diode with a 1:1 PCPDTBT:PCBM contribution ratio, i.e. the diode with more PCBM, both reached higher current values and had a lower ideality factor. In other words, the heterojunction diode with a high amount of PCBM showed more ideal behaviour. The reason for these values obtained as a function of the PCBM content was that the passivation of the surface defects in the structure increased as the PCBM content increased, and thus it had a better transmission mechanism.

In this study, all the features in the I-V data set for Au/PCPDTBT:PCBM/n-Si MPS SBD were classified by machine learning methods and predicted with 100% accuracy. In this respect, our study has shown that machine learning can be used effectively in the dual classification of I-V data of SBDs. For comparison, different types of machine learning methods with different variants were tried and classification accuracies between 30 and 100% were achieved.

References

[1] Sze, S.M. and Ng, K.K., (2007). *Physics of Semiconductor Devices*, (3rd ed.), John Wiley & Sons, Hoboken, New Jersey.

[2] Chiguvare, Z., Parisi, J., & Dyakonov, V. (2003). Current limiting mechanisms in indium-tin-oxide/poly3-hexylthiophene/aluminum thin film devices. *Journal of Applied Physics, 94*(4), 2440-2448.

[3] Hoppe, H. and Sariciftci, N.S., (2004). Organic solar cells: an overview. *Journal of Materials Research, 19*(7), 1924-1945.

[4] Turmuş, M., (2014). N tipi silisyum tabanlı altlık üzerine pyrene (C16H10) maddesinin kaplanarak elde edilen yapıların akım iletim mekanizmaları, [Yüksek Lisans Tezi, Bingöl Üniversitesi]. Bingöl-Türkiye.

[5] Nalçalıgil, S.Z., (2011). Perylene türevi oranik yarıiletken ince filmlerin optik özelliklerinin incelenmesi, [Yüksek Lisans Tezi, Selçuk Üniversitesi]. Konya-Türkiye.

[6] Boy, F., (2013). Organik arayüzeyli GaAs schottky diyodların elektriksel karakterizasyonu, [Yüksek Lisans Tezi, Selçuk Üniversitesi]. Konya-Türkiye.

[7] Sharma, B.L., (1984). (Ed.), Metal-semiconductor schottky barrier junctions and their applications, Plenum Press, New York.

[8] Şimşir, N., (2012). Metal/organik/inorganik schottky diyodların sıcaklığa bağlı elektriksel karakterizasyonu, [Yüksek Lisans Tezi, Selçuk Üniversitesi]. Konya-Türkiye.

[9] Özdemir, A.F., Aldemir, D.A., Kökce, A. & Altindal, S., (2009). Electrical properties of Al/conducting polymer (P2ClAn)/p-Si/Al contacts. *Synthetic Metals*, *159*(14), 1427-1432.

[10] Demirezen, S. and Altındal Ş., (2010). Possible current-transport mechanisms in the (Ni/Au)/Al0. 22Ga0. 78N/AlN/GaN schottky barrier diodes at the wide temperature range. *Current Applied Physics 10*(4), 1188-1195.

[11] Tüzün Özmen, Ö., (2014). Effects of PCBM concentration on the electrical properties of the Au/P3HT:PCBM/n-Si (MPS) schottky barrier diodes. *Microelectronics Reliability, 54*(12), 2766–2774.

[12] Yang, M., (2018). *A machine learning approach to evaluate Beijing air quality*. [Senior Thesis, University of California].

[13] Mohri, M., Rostamizadeh, A. & Talwalkar, A., (2012). *Foundations of machine learning*. The MIT Press, Cambridge

[14] Hal Daume´ III, (2017). *A course in machine learning*. <http://ciml.info/dl/v0_99/ciml-v0_99-all.pdf> [Accessed: 19 Mar 2023].

[15] Alpaydın, E., (2014). *Introduction to machine learning*. MIT Press, Cambridge.

[16] Özbay Karakuş, M. and Er, O., A 2022. comparative study on prediction of survival event of heart failure patients using machine learning algorithms. *Neural Comput & Applic, 34*, 13895–13908.

[17] Tüzün Özmen, Ö., (2014). "Effects of PCBM concentration on the electrical properties of the Au/P3HT:PCBM/n-Si (MPS) Schottky barrier diodes", *Microelectronics Reliability*, *54*, 2766-2774.

[18] D. Braun ve A.J Heeger, (1991). Visible-light emission from semiconducting polymer diodes. *Applied Physics Letters*, *58,* 1982-1984.

[19] Yağlıoğlu E., and Tüzün Özmen, Ö., (2014). F4-TCNQ concentration dependence of the current voltage characteristics in the Au/P3HT:PCBM:F4-TCNQ/n-Si (MPS) Schottky barrier diode, *Chinese Physics B, 23*(11), 117306.

Acknowledgement

This study was supported by İzmir Bakırçay University Unit of Scientific Research Projects Coordination with project number BBAP.2022.012.

AITA
ARTIFICIAL INTELLIGENCE
THEORY AND APPLICATIONS

RESEARCH ARTICLE

Fraud Detection on E-commerce Transactions Using Machine Learning Techniques

Murat Gölyeri[a], Sedat Çelik[a], Fatma Bozyiğit[bc†], Deniz Kılınç[d]

[a] Boyner Group, Istanbul, Turkey
[b] Department of Computer Science, University of Antwerp, Antwerp, Belgium
[c] AnSyMo/CoSys corelab, Flanders Make, Leuven, Belgium
[d] Department of Computer Engineering, İzmir Bakırçay University, İzmir, Turkey
[†]fatma.bozyigit@uantwerpen.be

RECEIVED MARCH 30, 2023
ACCEPTED APRIL 29, 2023

CITATION Gölyeri, M., Çelik, S., Bozyiğit, F., & Kılınç. D. (2023). Fraud detection on e-commerce transactions using machine learning techniques. *Artificial Intelligence Theory and Applications, 3(*1), 45-50.

Abstract

Fraud detection is an important aspect of e-commerce transactions as it helps to prevent fraudulent activities such as unauthorized transactions, identity theft, and account takeovers. Recently, machine learning algorithms have been widely used in the literature to detect fraud in e-commerce transactions. These algorithms work by learning patterns in the data that indicate fraudulent activity. Pattern detection involves discovering the discriminative features in the data, such as unusual transaction amounts, locations, or behaviors that are out of the normal range for a particular user, to feed the machine learning method. In this study, four basic machine learning algorithms (decision tree, logistic regression, random forest, and extreme gradient boosting) are used to detect fraud in e-commerce transactions using a newly created dataset including various features about online shopping activities on Boyner Group's e-commerce website and mobile application. The study contributes to the literature by trying different machine learning classifiers and utilizing different features that differ from current approaches in the literature.

Keywords: fraud detection; e-commerce; machine learning; feature engineering

1. Introduction

E-commerce fraud refers to any fraudulent or dishonest activity conducted by individuals or groups performing unauthorised transactions, steal personal or financial information, or manipulate e-commerce systems for financial gain. Some common examples of e-commerce fraud include identity theft, phishing, chargeback fraud, affiliate fraud and false advertising.

Fraud detection is a crucial aspect of e-commerce transactions as it helps protect the customers and prevent financial losses [1]. There are some techniques that can be used to detect fraud in e-commerce transactions, such as transaction monitoring, IP address geolocation and device fingerprinting. With evolving technology, machine learning algorithms can be used to analyze transaction data and identify patterns indicative of fraudulent activity. These algorithms can be trained on historical transaction data to detect fraudulent patterns and flag suspicious transactions [2].

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN : 978-605-69730-2-4 © 2023 University of Bakırçay

In this study, four basic machine learning algorithms (decision tree [3], logistic regression [4], random forest [5] and extreme gradient boosting [6]) are used to detect fraud in e-commerce transactions using a newly created dataset. The dataset includes shopping activities during ninety days on the e-commerce website and mobile application of Boyner Group[1], a Turkish retail company operating in the fashion and apparel industry. The study contributes to the literature by trying different machine learning classifiers using various features such as cart quantity, number of items in the cart, number of successful orders in the last 24 hours, number of failed orders in the last 24 hours, number of returns in the last 24 hours, number of returns in the last week, order ID, payment method and customer status (guest or registered). The performance of the classifiers are then compared using the metrics of Precision, Recall and F1 Score. The remaining parts of the study are structured as follows. Section 2 discusses related work. Section 3 provides information about the experimental dataset, data preprocessing steps, feature engineering and machine learning methods used in the study. Section 4 presents the details of the evaluation with metrics and results. Our research shows that experts' knowledge does not differ semantically, but rather the information is interconnected. The experiments conducted have shown that the size of words and the retrieval speed of words from memory varies between individuals with different background knowledge. The results of our study could also help to provide better, personalized instructions to users in different areas and to build a more interactive dialogue between the user and an intelligent tutoring system.

2. Related Work

E-commerce has grown rapidly in recent years, making it an attractive target for fraudulent activity. Fraudsters use sophisticated techniques to bypass security measures and steal money from e-commerce businesses. Detecting and preventing fraud in e-commerce transactions is a challenging task, and researchers have explored various machine learning and data mining techniques to address this problem.

Several studies have investigated e-commerce fraud detection using machine learning techniques. Anomaly detection is a commonly used technique for fraud detection. In their study, Li et al [7] proposed a deep learning-based anomaly detection model to detect fraud in e-commerce transactions. They showed that their model can detect fraudulent transactions with high accuracy. Machine learning is another popular technique for fraud detection in e-commerce. In their study, Zhang et al [8] proposed a supervised learning approach to detect fraudulent behavior in e-commerce transactions. They used logistic regression and random forest algorithms to train their model and achieved high accuracy in detecting fraudulent transactions. Porwal et al [9] proposed a clustering-based approach for detecting fraud in e-commerce transactions. They used clustering to group similar transactions together and identified anomalous clusters that contained fraudulent transactions. In another study, Xie et al [10] proposed a decision tree-based approach for e-commerce fraud detection. They showed that their approach can detect fraudulent transactions effectively and with high accuracy.

[1] https://www.boyner.com.tr/

3. Materials and Methods

In this study, the first step is to collect data consisting of users' transactions and shopping activities. Then the data is prepared using pre-processing methods by normalizing and removing missing values, outliers and other inconsistencies. After structuring the data, the ChiSquare [11] feature selection method is applied to determine which feature is most effective in classification. Finally, the performance of the model is evaluated using metrics such as precision, recall and F1 score in the test set.

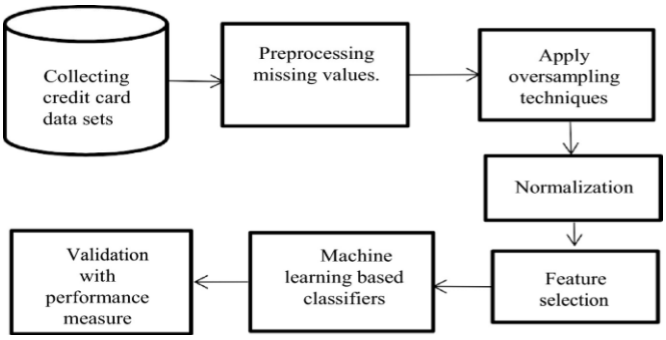

Collecting credit card data sets
Preprocessing missing values.
Apply oversampling techniques
Normalization
Feature selection
Machine learning based classifiers
Validation with performance measure

Figure 1. Workflow of the proposed approach

3.1. Dataset

To develop a machine learning method for fraud detection, a dataset containing both fraudulent and legitimate transactions is needed. In this study, the dataset includes shopping activities during ninety days on the e-commerce website and mobile application of Boyner Group, a Turkish retail company operating in the fashion and apparel industry.

In this study, we included eight different features, explained in Table 1, in a first version of the dataset. In this version, shopping activities of 1850 registered users are included. In the second version of the dataset IsGuestOrder feature is included. This feature shows the situation of the customer (guest or registered). Consideringly, 1752 guest users' transactions are added to dataset.

Table 1. Features in the dataset

Feature	Feature Name	Feature Description
Feature 1	TotalAmount	basket amount
Feature 2	OrderItemCount	number of items in the basket
Feature 3	SuccessOrder	number of successful orders in the last 24 hours
Feature 4	FailedOrder	number of failed orders in the last 24 hours
Feature 5	Last24HoursReturnOrder	number of returns in the last 24 hours
Feature 6	LastWeekReturnOrder	number of returns in the last week
Feature 7	OrderID	order ID
Feature 8	PaymentMethodCode	payment method

3.2. Data Preperation

Pre-processing is a necessary step to prepare the data for analysis. This includes dealing with missing values and scaling the data so that all features are at a similar scale.

Consideringly, SimpleImputer and StandardScaler classes from the scikit-learn library are used for this purpose.

3.3. Feature Selection

Feature selection is a technique used to select the most relevant features from a data set. Chi-square feature selection is one of the most effective techniques for selecting the most important features based on their statistical significance. By identifying the features that are most strongly associated with the target variable, it can help improve the performance of machine learning models and reduce the risk of overfitting.

3.4. Machine Learning Algorithms

Decision Tree

Decision tree is one of the widely used machine learning algorithms for classification and regression tasks. It is a supervised learning algorithm that builds a tree-like model of decisions and their possible consequences. The tree structure consists of nodes and edges, where the nodes represent the decision or outcome, and the edges represent the possible consequences of the decision.

Logistic Regression

It is a type of regression analysis used when the dependent variable is binary or dichotomous. The aim of logistic regression is to model the probability of a particular outcome based on one or more predictor variables.

Extreme Gradient Boosting

Extreme Gradient Boosting is a tree-based ensemble method that combines the predictions of multiple decision trees to produce a final prediction. The algorithm is very effective in dealing with high-dimensional data and has the ability to model non-linear relationships between variables.

Random Forest

Random Forest is a type of ensemble learning method in which multiple decision trees are created and combined to make a final prediction. In Random forest, each decision tree is created independently and the final prediction is made by averaging the predictions of all the trees. To prevent overfitting, each tree is trained on a random subset of the original dataset and a random subset of the input features is used for each split in the tree.

4. Experimental Study

In the experimental study, the default parameters are set for each classifier implemented and feature selection method since these parameters give promising experimental results. The evaluation results of each machine learning method are obtained by dividing the data set into 10 pieces by cross-validation (Table 2). To perform 10-fold cross-validation on this data, the data is divided into ten equal-sized folds, each with 362 samples. The model is trained ten times, using a different fold as the validation set and the other nine folds as the training set, to better assess the performance of the model for the entire data set.

Table 2 shows the performance comparison of classifiers in terms of precision, recall, and F1 score on the first version of the dataset. As it can be seen from Table 2, the performance of the all classifiers are over 80%.

Table 2. Performance of classifiers on the dataset containing TotalAmount, OrderItemCount, SuccessOrder, FailedOrder, Last24HoursReturnOrder, LastWeekReturnOrder, PaymentMethodCode features

Classifier	Accuracy	Precision	Recall	F-measure
Decision Tree	0.83	0.82	0.78	0.80
Logistic Regression	0.89	0.91	0.86	0.88
Extreme Gradient Boosting	0.90	0.85	0.92	0.88
Random Forest	0.81	0.80	0.87	0.83

In Table 3, the second version of the dataset is inputted for classifiers. It is seen that when the IsGuestOrder feature is included the performance of classifiers increases. For instance, the performance of logistic regression is increased 3% in terms of F1 score on the dataset including IsGuestOrder feature.

Table 3. Performance of classifiers on the dataset containing IsGuestOrder in addition to dataset version 1.

Classifier	Accuracy	Precision	Recall	F-measure
Decision Tree	0.83	0.82	0.79	0.80
Logistic Regression	0.93	0.95	0.89	0.92
Extreme Gradient Boosting	0.90	0.87	0.91	0.89
Random Forest	0.86	0.81	0.90	0.85

5. Conclusion

Detecting fraud in e-commerce is a challenging task that requires the use of sophisticated techniques to detect fraudulent transactions. The current state of the art shows that machine learning techniques are promising for detecting fraudulent activities in e-commerce transactions. In this study, four different machine learning methods (decision tree, logistic regression, extreme gradient boosting, and random forest) are performed considering different features in the collected data. The performance of the classifiers is compared against two versions of the dataset to find the most relevant attribute in detecting fraudulent activity. In the first version of the dataset, the features TotalAmount, OrderItemCount, SuccessOrder, FailedOrder, Last24HoursReturnOrder, LastWeekReturnOrder and PaymentMethodCode are used as input to the classification algorithms. In the second version of the dataset, the feature IsGuestOrder is added as an additional feature. It can be seen that the performance of the classifiers increases when the feature IsGuestOrder is included. Since the performance of the logistic regression was calculated to be over 92%, we can say that the results of the study are motivating for future work.

References

[1] Patidar, R., & Sharma, L. (2011). Credit card fraud detection using neural network. *International Journal of Soft Computing and Engineering (IJSCE)*, *1*(32-38).

[2] Yufeng Kou, Chang-Tien Lu, S. Sirwongwattana and Yo-Ping Huang, "Survey of fraud detection techniques," *IEEE International Conference on Networking, Sensing and Control, 2004*, Taipei, Taiwan, 2004, pp. 749-754 Vol.2, doi: 10.1109/ICNSC.2004.1297040.

[3] Kingsford, C., & Salzberg, S. L. (2008). What are decision trees?. *Nature biotechnology*, *26*(9), 1011-1013.

[4] Cabrera, A. F. (1994). Logistic regression analysis in higher education: An applied perspective. *Higher education: Handbook of theory and research*, *10*, 225-256.

[5] Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, *114*, 24-31.

[6] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., ... & Zhou, T. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, *1*(4), 1-4.

[7] Li, Z., Xiong, H., & Liu, Y. (2012). Mining blackhole and volcano patterns in directed graphs: a general approach. *Data Mining and Knowledge Discovery*, *25*, 577-602.

[8] Zhang, R., Zheng, F., & Min, W. (2018). Sequential behavioral data processing using deep learning and the Markov transition field in online fraud detection. *arXiv preprint arXiv:1808.05329*.

[9] Porwal, U., & Mukund, S. (2019, August). Credit card fraud detection in e-commerce. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* (pp. 280-287). IEEE.

[10] Cao, R., Liu, G., Xie, Y., & Jiang, C. (2021). Two-level attention model of representation learning for fraud detection. *IEEE Transactions on Computational Social Systems*, *8*(6), 1291-1301.

[11] Zhai, Y., Song, W., Liu, X., Liu, L., &Zhao, X. (2018, November). A chi-square statistics based feature selection method in text classification. In 2018 IEEE 9th İnternational Conference on Software Engineering and Service Science (ICSESS) pp.160-163. IEEE

■ RESEARCH ARTICLE

An Application on Chest X-Ray Images for the Detection of Tuberculosis Disease by Employing Deep Convolutional Neural Networks

Hatice Koç [a, b†], **Abdulkadir Hızıroğlu** [b], **Ahmet Emin Erbaycu** [c]

[a] Management Information Systems, Gebze Technical University, Kocaeli, Turkey
[b] Management Information Systems, İzmir Bakırçay University, İzmir, Turkey
[c] Medicine, İzmir Bakırçay University, İzmir, Turkey
[†] haticekoc@gtu.edu.tr, corresponding author

RECEIVED MARCH 1, 2023
ACCEPTED APRIL 30, 2023

CITATION Koç, H., Hızıroğlu, A., & Erbaycu. A. E. (2023). An application on chest X-ray images for the detection of tuberculosis disease by employing deep convolutional neural networks. *Artificial Intelligence Theory and Applications, 3(*1), 51-64.

Abstract

Tuberculosis is the second infectious disease causing death after COVID-19. Diagnosing it is an easy and cheap via chest radiographs. However, some countries lack medical personnel and equipment for tuberculosis detection on chest radiographs. Computer-aided diagnosis and computer-aided detection systems utilizing deep learning can be employed to identify tuberculosis on medical images. Although there are some studies, they are insufficient for unbiased systems because these systems require the datasets having different features. The aim of this study is to evaluate the performance of pretrained networks for a classification application on chest X-ray images by utilizing the dataset from the Hospital in Turkey and Montgomery Count Dataset. The predictive models were implemented with the pre-trained DCNNs such as ResNet-50, Xception, and GoogLeNet. An Xception model provides the best performance.

Keywords: tuberculosis, deep convolutional neural networks, transfer learning, classification

1. Introduction

Tuberculosis (TB) is a contagious disease and is one of the top 10 disease causing death before COVID-19 pandemic. The Global Tuberculosis Report, by WHO in 2022, expresses that TB has influenced approximately a quarter of the world's population. About 50% of TB patients have passed away since they were not treated. Besides, TB can be described as a disease of poverty due to its unfavourable impacts on 30 high TB burden countries, which these countries were influenced 87% by TB [1] [2]. 8.925 TB cases were recorded in Turkey in 2020, which the number of male patients equal 57.2% while the number of female patients has 42.8% of all TB patients [3].

Early diagnosing can prevent and treat TB. Chest X-ray (CXR) is a cheaper and effortless technology used as a part of TB diagnosis. However, it requires systematically check different points of lungs on a CXR with a physician’s interpretation [4] [5] [6]. Low-income countries have the issues arising from the lack of medical personnel and equipment for TB diagnosis. Thus, several studies evaluate if computer-aided detection (CADe) and computer-aided diagnosis (CADx) systems assist detect TB on chest radiographs (CRs)

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN : 978-605-69730-2-4 © 2023 University of Bakırçay

in order to solve these issues. These systems can collect and manipulate medical data and report medical examination by exploring potential abnormalities on CRs. They assist physicians and radiologists by identifying the localization of lesions, diseases, and/or causes of symptoms over examinations [7] [8]. That is, they are employed as second opinion during disease diagnosis [8] and medical decision-making.

Many studies have utilized artificial intelligence (AI) to create automated activities for diagnosis process and prognosis, medical decisions, and health practise by developing CADe and CADx systems. Some studies also employ AI to process medical images, biosensors, molecular data and electronic medical records (EMR). DL is an AI approach employed to examine CRs. The typical patterns of TB are easily recognized with satisfactory accuracy during medical examinations [1] [8] by using DL. Hence, many studies have used deep convolutional neural networks (DCNNs) to identify TB since medical image classification is one of major steps to improve CADe and CADx systems [9] which enable to recognize of localization of lesions, diseases, and/or symptoms [8].

On the other hand, many challenges have been faced when implementing DL based-CADe and CADx systems. There are limited data, the dataset containing outside available populations, and/or the dataset having insufficient variables and instances [10]. An unbiased DL model requires the dataset formed of balanced distribution and several demographic variables. Therefore, some research has been performed to explore how robust DCNNs algorithm to develop CADe and CADx systems determining TB on CXR images while some have investigated whether the DCNNs-based systems help tackle or not the lack of medical personnel and equipment for TB diagnosis. Although there are several research, the studies have been performed with similar datasets. Any study has never been carried out by a dataset that is not obtained from the hospitals in Turkey. This gap in academic literature has motivated to carry out this study. Expert view also encouraged for the research problem. Briefly, the purpose of the study is to fill the gap in the study utilizing CXR images from hospital in Turkey and analyse how robust the DCNNs to identify TB.

There are two experiments in this study. Firstly, exploratory data analysis (EDA) was applied to grasp the distribution of demographic variables because it allows decision support systems (DSS) to discover tendency and unique characteristics in dataset. Then, predictive analytics was applied with transfer learning approach in order to detect TB disease since it enables to CADx and CADe systems make predictions on new observation. Before implementing the models, data augmentation was applied. Performance evaluation was performed by comparing the confusion matrix, accuracy, recall, precision, misclassification rate and F1 score of each model.

The rest of the study covers four sections. In the second section, DSS, analytics, and AI were defined, and the literature was reviewed in terms of medical problems. The third section clarifies the methodology which covers five parts; model development, data collection and description, image data pre-processing, data analytics, and performance evaluation. The next section presents the results of EDA, and predictive models while the last section explains the contributions, limitations, and future of the study.

2. Literature Review

Healthcare sector requires intelligent systems that ease diagnosis, and support treatment by processing data about patients' lifestyles and having disease manifestations. CADe and CADx systems can be defined as a type of DSS that provide information of clinicians and deal with challenges from knowledge and information acquisition and management in clinical practice. Their usage grows tendency for personalized healthcare services, the request for accessibility for EMR and the distribution of convenient data [11].These systems employ analytics and internal and external data acquisition [12].

Analytics integrates actions, processes, and tools in the dimensions of descriptive, predictive, and prescriptive analytics [13]. AI provides intelligence and expertise for analytics [13]. ML, a part of AI, help develop automated and/or semi-automated processes by manipulating the data [14]. It is used to develop knowledge-based DSS employed since these systems increase accuracy rates and reduce errors in medical decisions and practices if it is intended to develop an unbiased system [15]. ANNs is one of the most commonly used ML techniques to predictive model [13] trained with structured, semi-structured, and/or unstructured data [16]. The rise in computational power and data have boosted the performance of ANNs and emerged DL. Additionally, CNNs is a type of DL technique that perform automatically feature extraction and segmentation of organs or an object [17]. Hence, several studies for CADe and CADx systems have used CNNs [18].

Some studies have analysed how robust DCNNs are when identifying TB on CXR images. They have employed transfer learning approach, implemented DCNNs models, or compared the performances of CNNs and other AI techniques. They have used open-source datasets and/or specific hospitals. Besides, the models have been implemented for either classification, feature extraction or segmentation. Their performances have been evaluated with accuracy, recall, precision, F1 score, AUC and ROCs.

The studies of [19], [20], and [21] classify lung diseases while the studies of [5], [22], [23], [24], and [25] focus on TB disease. Additionally, the studies of [22] and [25] utilized transfer learning, unlike the study of [5]. Abiyev and et. al. [19] examined the applicability of CNNs, backpropagation neural networks (BPNNs), and competitive neural networks (CpNNs) for the classification of some chest diseases. The performances of the models were compared with accuracy, MSE, and training time. The study was performed with the insufficient dataset for detecting various lung diseases, and the dataset does not include demographic variables.

Mamalakis and et. al. [20] constructed a new deep transfer learning pipeline and evaluated the performance of this pipeline on the detection of COVID-19, pneumonia, TB, and healthy patients from CXR images. The study has the dataset that contains a smaller size of cohort for COVID-19 and does not have various demographic features and multi-label lung disease.

In the study of Ölmez and et. al. [21], the biomedical classification applications were represented by using CNNs and PNNs algorithms for detecting several lung diseases in both Turkey and the World. The performances of the models were compared with accuracies. The study indicates a CNNs-based DSS can be improved for the detection of lung diseases.

On the other hand, Hwang and et. al. [5] applied the classification to determine TB disease on CXR images without transfer learning. AUROC, AUAFROC, sensitivity, and specificity were evaluated to measure the performances of the models. Besides, the performances of the models were compared with the performance of physicians. According to the results, the DLAD algorithm outperformed most physicians and enhanced the performance of non-radiology physicians as the second reader.

In the study of Lakhani and et. al. [22], a transfer learning approach was employed to examine the efficacy of DCNNs for TB detection on CRs. ROC and AUCs were compared for the performance of the models. Data augmentation was also applied when comparing the performances. The results show that greater values for AUCs were presented with the pre-trained models, the best performance was obtained with the augmented data, and the highest performance was given by the ensemble models.

Cao and et. al. [25] conducted a study to deploy a mobile device-based computing system for TB diagnostics in Peru by using a transfer learning approach and mobile health technology in order to lessen TB diagnosis time. The performances were evaluated with accuracy. The results indicated that the proposed approach is feasible.

However, in the study of [23] and [24], transfer learning was used for segmentation models, and then support vector machine (SVM) was used for the classification model. In the study of Karaca and et. al. [23], an automated DSS was proposed for TB detection. The performances of models were evaluated with accuracy and AUC. The results obtained from the study were also compared with the findings from some previous studies. Although the study was performed with insufficient data, the study indicated that data augmentation improves the accuracy of the detection of TB.

Oltu and et. al. [24] suggested an automated DSS that classifies normal and TB on CXR images. The impacts of data augmentation were also examined. Similar results were presented by MobileNet and VGG16 while the highest accuracy and AUC were presented by the MobileNet model utilized data augmentation with rotation. Poor performances were obtained when applied all data augmentation methods together and unnecessary both shifting and rotation for feature extraction.

The studies of [26], [27], and [28] employed a lung segmentation model for TB detection. Stirenko and et. al. [26] carried out the study to prove the efficiency of the lung segmentation technique without the pretrained DCNNs. EDA was also applied to understand the distribution of disease, age, and gender in the dataset and the image heights and widths. The values of accuracy were compared to evaluate the performances. The study indicated that better performance for training on the pre-processed dataset has been acquired after lung segmentation. The segmented and augmented data increased accuracy. However, the models were trained with a small and not-well-balanced dataset. The dataset is also retrospective dataset and has non-evident outliers.

Rahman and et. al. [27] utilized a transfer learning approach, and the original and segmented lungs in X-ray images in order to detect. The study has evaluated the performance of all classification models for the detection of TB. In the study, better performance was acquired with the segmented CXR images while testing the networks. The results show that classification accuracy can be increased with image segmentation.

Heo and et. al. [28] compared two DCNNs models for the detection of TB from CRs by using CXR images and demographic variables structures and image segmentation. The performances of the models were evaluated with ROC and AUC. The study represents that the demographic variables improve the performance of the CNNs model. CADe and CADx systems employing DCNNs can enable to detect TB on CXR images easily and enhance disease management. However, the study has the limitations that are the low number of demographic features and insufficient computational power.

As summarized in Table 1, the accuracy rates were 92.4% and 95.51% in the studies of Abiyev and et. al. [19] and Ölmez and et. al. [21] respectively, while the recall rate was discovered as 98.12% by Mamalakis and et. al. [20]. However, in the studies for detecting TB with classification, Hwang and et. al. [5], and Lakhani et. al. [22] ascertained the recall rates between 94.3% and 100%, and between 92% and 97.3% while the specificity rates between 84.1% and 100%, and between 94.7% and 98.7%, respectively. Cao and et. al. [25] discovered that the accuracy rate is 89.6% for the binary classification while the accuracy rate is 62.07% for the multi-classification. The results of accuracy, recall, and specificity indicate that CNNs can be used to detect TB and other lung diseases.

Table 1. Related empirical work

Study	Purpose	Dataset	Technique & Architecture	Performance Evaluation	Findings
[19]	To show the applicability of conventional and DL techniques for the classification of chest diseases.	The dataset that contains of 112,120 CXR images belonging to 30,805 different patients from the National Institutes of Health—Clinical Centre.	BPNNs CpNNs CNNs	Accuracy MSE Training time	**BPNNs** Accuracy:80.04% MSE: 0.0025 Training time:630 sec. **CpNNs** Accuracy: 89.57% MSE:0.0036 Training time: 2500 sec. **CNNs** Accuracy: 92.4% MSE:0.0013 Training time:2500 sec.
[20]	To develop a new deep transfer learning pipeline, called as DenResCov-19, and evaluated its performance by detecting COVID-19, pneumonia, TB, and healthy patients from CXR images.	The Pediatric CXRs Dataset, The IEEE COVID-19 CXRs Dataset, Shenzhen Dataset.	DenResCov-19 constituted by concatenating four blocks from the ResNet50 network and the DenseNet121 network with their width, height, and frames.	AUC-ROC Confusion matrix Precision Recall F1 score	**For each dataset:** AUC-ROC: 0.9960, 0.9651, 0.9370, and 0.9640. F1 score: 98.21%, 87.29%, 76.09%, and 83.17%. Recall: 98.12%, 89.38%, 59.28%, and 69.7%. Precision: 98.31%, 85.28%, 79.56%, and 82.90%.
[21]	To exemplify for biomedical classification applications using DL methods for the detection of lung diseases that are widespread in Turkey and the World.	The dataset that includes 38 different features indicating laboratory examination from the patients hospitalized because of lung diseases, and 357 subjects.	PNNs CNNs	Accuracy	**PNNs** Accuracy: 91.25%. **CNNs** Accuracy: 95.51%.
[5]	To improve a DL-based automatic detection algorithm (DLAD) for active TB on CRs as a second opinion and measure its performance with different datasets and by comparing it with	Seoul National University Hospital Dataset, Boramae Medical Center Dataset, Kyunghee University Hospital at Gangdong Dataset, Daejeon Eulji Medical Center Dataset, Montgomery Count Dataset, Shenzhen Dataset.	DCNNs	ROC Sensitivity Specificity	**For internal validation** AUROC: 0.988. AUAFROC: 0.977. **For external validation** AUROC: 0.977 to 1.000. AUAFROC: 0.973 to 1.000 Sensitivity: 94.3%-100%. Specificities: 91.1%-100%.

	physicians' performance.				
[22]	To analyse the efficacy of DCNNs for TB detection on CRs.	Belarus TB Public Health Program Dataset, Thomas Jefferson University Hospital Dataset, Montgomery Count Dataset, Shenzhen dataset.	AlexNet GoogLeNet.	ROC AUCs	**AlexNet** AUC: 0.90, 0.95, 0.98, and 0.98 Sensitivity: 92 % Specificity: 94.7% **GoogLeNet** AUC: 0.88, 0.94, 0.97, and 0.98. Sensitivity: 92 % Specificity: 98.7% **The ensemble of both** AUC: 0.99 Sensitivity: 97.3 % Specificity: 94.7%
[25]	To deploy a mobile device-based computing system that screens CXR images for TB diagnostics in Peru.	4701 CXR images were employed, which provided by Dr. Peinado in Peru.	GoogLeNet	Accuracy	**For binary classification** Accuracy: 89.6% **For multiclass categorization** Accuracy: 62.07%.
[23]	Proposed an automated DSS for TB detection.	Montgomery Count Dataset	VGG16 VGG19 DenseNet121 MobileNet InceptionV3 SVM	Accuracy AUC	**With data augmentation** Accuracy: 98.7%, 98%, 98.9%, 98.8% and 96.5%. AUC: 1.000, 0.990, 1.000, 1.000 and 0.999. **Without data augmentation** Accuracy: 87%, 86.2%, 80.4%, 74.6% and 79%. AUC: 0.900, 0.910, 0.870, 0.810 and 0.880.
[24]	Presented an automated DSS identifying TB and analysed how data augmentation influences the analysis was examined.	Montgomery Count Dataset, Shenzhen Dataset.	VGG16 MobileNet. SVM	AUC Accuracy	**MobileNet** Accuracy: 91.30%, 96.40%, 96.50%, 0.-96.60%, 96.50%, 91.40%, 91.40% and 87.80%. AUC: 0.970, 0.990, 0.990, 0.990, 0.990, 0.990, 0.970 and 0.950. **VGG16** Accuracy: 91.40%, 96.70%, 95.70%, 95.60%, 95.60%, 91.60%, 91.30% and 87.60%. AUC: 0.960, 0.990, 0.990, 0.990, 0.990, 0.970, 0.960 and 0.930.
[26]	To prove the efficiency of the lung segmentation technique with lossless and lossy data augmentation for predicting TB disease.	Shenzhen Hospital Dataset, Montgomery Count Dataset, JSRT Dataset	DCNNs Lung segmentation masks EDA	Loss Accuracy Intersection-Over-Union (IoU) or Jaccard Index F1-score	The highest training rate with lung segmentation Lower training rate for the segmented dataset with both lossless and lossy data augmentations.

[27]	To detect TB with transfer learning and the original and segmented lungs in CXR and evaluate the performance of all classification models for the detection of TB.	Kaggle CXR images dataset and corresponding lung mask dataset for the lung segmentation. For classification: National Library of Medicine datasets (Montgomery Count and Shenzhen datasets), Belarus dataset, NIAID TB dataset, and RSNA CXR dataset.	ResNet18 RestNet50 ResNet101 ChexNet InceptionV3 VGG19 DenseNet201 SqueezeNet MobileNet U-Net Modified U-Net		**ChexNet** Accuracy: 96.47% Precision: 96.62% Sensitivity: 96.47% Specificity: 96.51% F1-score: 96.47%. **DenseNet201** Accuracy: 98.6% Precision: 98.57% Sensitivity: 98.56% Specificity:98.54% F1-score: 98.56%.
[28]	To compare two DCNNs models for the detection of TB, which these models were I-CNN trained with only CXRs and D-CNN trained with demographic variables and employed the health examination data of annual workers.	Dataset contains the medical surveillance data acquired from workers at Yonsei University.	VGG19 InceptionV3 ResNet50 DenseNet121 InceptionResNet V2 U-Net	ROC AUC	**VGG19** D-CNN: ROC: 0.9213 AUC: 0.97147. I-CNN ROC:0.9075 AUC: 0.9570. **InceptionV3** D-CNN: ROC: 0.9045 AUC:0.9616. I-CNN ROC:0.8821 AUC:0.9523. **ResNet50** D-CNN: ROC:0.8955 AUC: 0.9219. I-CNN ROC: 8780 AUC:09219. **DenseNet121** D-CNN ROC:0.8864 AUC: 0.9472. I-CNN: ROC:0.8605 AUC: 0.9315. **InceptionResNetV2** D-CNN ROC: 0.8864 AUC: 0.9455. I-CNN ROC:0.8881 AUC: 0.9482. Sensitivities: D-CNN Sensitivity: 81.50%. I-CNN Sensitivity: 77.50%.

These studies reveal that DCNNs can be an alternative to determine from CXR images so as to implement automated CADe and CADx systems as a second opinion. However, an unbiased AI-based system requires several datasets covering diverse populations, instances, and variables. Hence, the DCNNs models were implemented without the lung segmentation and the feature extraction by contrast with [26], [27], [28], [23] and [24]. The binary classification was applied within in this study, unlike the studies of [19], [20], [21], and [25].

3. Methodology

3.1. Model development

The five steps are generally followed to implement a ML model, which are collecting data, data pre-processing, splitting data for developing the model, training the model, and testing and validating the model. Although feature engineering and domain expertise are parts of implementing a ML model, DL models may require some different processes [29]. The steps of a typical ML project are applied with additional processes related to using a DL technique embedded within that. The steps in Figure 1 were followed in this study in order to implement the models.

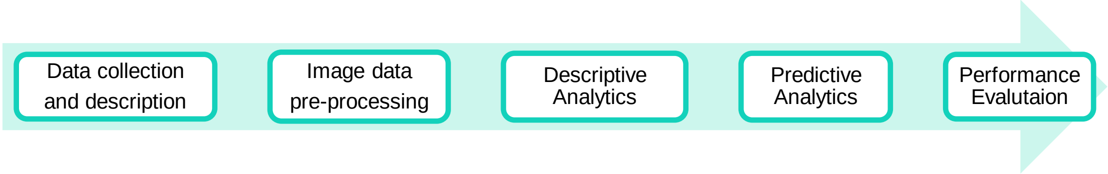
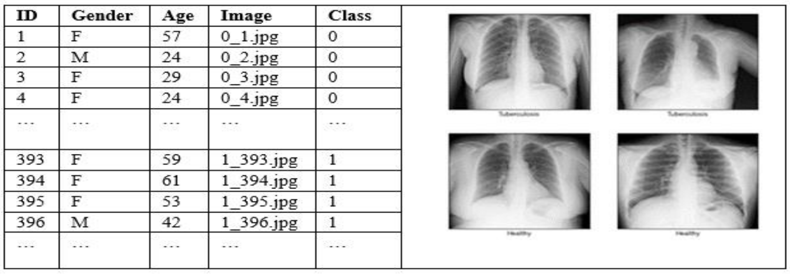


Figure 1. The steps for implementing DCNNs models

3.2. Data collection and description

Willemink and et. al. [30] proposed eight phases that are ethical approval, data access, querying data, data de-identification, downloading and storing data, data quality control, structuring data, and labeling data so as to prepare medical image for ML [30]. CXR images were prepared to implement DCNNs models by the similar phases in this study.

CXR images and demographic variables concerning these images were obtained from the Hospital in Turkey after ethical approval, which are about TB patients and healthy people. 444 of CXR images were included, which each image has the size of 3032 width and 2520 height. The total 222 data is about TB patients while the rest of data is related to healthy people. Demographic variables consist of age and gender, which were recorded “csv” file and the corresponding attributes “ID”, “Gender”, “Age”, “Images”, and “Class”. There are different two classes that are Healthy and TB. The class of TB is expressed with 1 while the class of healthy is expressed with 0. Furthermore, Montgomery Count Dataset were also ultilized. 110 CXR images were included, which each image has the size of 4020 width and 4892 height or 4892 width and 4892 heights. An example for CXR images’ view and summarizing demographic variables are given in Figure 2.



ID	Gender	Age	Image	Class
1	F	57	0_1.jpg	0
2	M	24	0_2.jpg	0
3	F	29	0_3.jpg	0
4	F	24	0_4.jpg	0
...	...	...	...	...
393	F	59	1_393.jpg	1
394	F	61	1_394.jpg	1
395	F	53	1_395.jpg	1
396	M	42	1_396.jpg	1
...	...	...	...	...

Figure 2. An example for demographic variables and CXR images’ views

3.3. Image data pre-processing

Image data pre-processing was performed in two stages which are data transformation and data augmentation in by coding in MATLAB. In the stage of data transformation, CXR images were resized as the dimensions of 224x224x3 before training the models implemented with the ResNet-50 and GoogLeNet architectures while the images were resized as the dimensions of 299x299x3 in order to train the models implemented with the Xception architecture. Besides, CXR images were transformed into colour images since these architectures require colour input images.
In the stage of data augmentation, rotation, reflection and shear intensity techniques were applied as the following: random rotation between -5 and 5, random reflection as 1, random shear intensity between -0.05 and 0.05.

3.4. Data analytics

3.4.1. Exploratory data analysis

Exploratory data analysis (EDA) is a technique used in descriptive analytics and utilizes retrospective data, statistical techniques and visualization tools before implementing a ML/DL model in order to explore useful information, inform results and support decision-making. In this study, EDA was performed to analyse demographic variables so as to understand the distributions of gender and age in terms of each class. Descriptive statistics and data visualization were employed by coding in Python on Jupyter Notebook.

3.4.1. Predictive analytics

The total of 22 models were implemented to discover the best performance for diagnosing TB disease. The models for binary classification were constituted by coding in MATLAB R2022b to predict TB disease on CXR images. Moreover, transfer learning approach was employed because implementing new DCNNs requires a large dataset as well as substantial time and cost [23]. Some characteristics of the architectures such as ResNet-50, Xception, and GoogLeNet are provided in Table 2.

Table 2. A brief for ResNet-50, Xception, and GoogLeNet

Characteristics	ResNet-50	Xception	GoogLeNet
Input dimensions	224x224x3	299x299x3	224x224x3
Class	1000	1000	1000
Layers	50	71	22
Last three layers	‘fc1000’, ‘fc1000_softmax’, 'ClassificationLayer_fc1000'	'predictions', 'predictions_softmax', 'ClassificationLayer_predictions'	‘loss3-classifier’, ‘prob’ ‘output’

The purpose of improving 22 models is to explore the best performance for diagnosing TB disease by comparing their performances. The first 11 models were trained with the data from the hospital by applying data augmentation. Nevertheless, the rest 11 model were trained with the combination of Montgomery Count Dataset and the hospital dataset after applying data augmentation. Additionally, the two datasets were split into training dataset and test dataset as 80% and 20%, respectively. Each model was built with 0.001 learning rate. The values of epoch, mini-batch size, and the types of optimizers were changed for each model, which are detailed in Table 3. The last three layers of each pre-trained network were also frozen, and then, a fully connected layer that contains two classes, a Softmax layer and classification output were added.

Table 3. The value of epoch, mini-batch size, and the types of optimizers

Dataset	Model	Optimizer	Epoch	Mini-Batch
The dataset from the hospital	ResNet-50 Model 1	Adam	10	25
	ResNet-50 Model 2	Sgdm	10	25
	Xception Model 1	Adam	10	25
	GoogLeNet Model 1	Adam	10	25
	GoogLeNet Model 2	Adam	10	32
	GoogLeNet Model 3	Adam	25	32
	GoogLeNet Model 4	Adam	32	64
	GoogLeNet Model 5	Sgdm	10	25
	GoogLeNet Model 6	Sgdm	10	32
	GoogLeNet Model 7	Sgdm	25	32
	GoogLeNet Model 8	Sgdm	32	64
The combined dataset	ResNet-50 Model 3	Adam	10	25
	ResNet-50 Model 4	Sgdm	10	25
	Xception Model 2	Adam	10	25
	GoogLeNet Model 9	Adam	10	25
	GoogLeNet Model 10	Adam	10	32
	GoogLeNet Model 11	Adam	25	32
	GoogLeNet Model 12	Adam	32	64
	GoogLeNet Model 13	Sgdm	10	25
	GoogLeNet Model 14	Sgdm	10	32
	GoogLeNet Model 15	Sgdm	25	32
	GoogLeNet Model 16	Sgdm	32	64

3.5. Performance Evaluation

Performance evaluation was performed by comparing the values of confusion matrix and performance measures of each model. For this, False Positive (FP), False Negative (FN), accuracy rate, recall rate, precision rate, F1 score, and misclassification rate were employed. Moreover, the positive class is the TB class while the negative class is the healthy class in this study.

4. Results

4.1. Exploratory Data Analysis

EDA was performed with three datasets which are the dataset from the hospital, Montgomery Count Dataset, and the dataset combined. The demographic variables were analysed with EDA. The findings from descriptive statistics are given in Table 3.

Table 4. Descriptive statistics for each dataset

Distribution of variables in the dataset	The dataset from hospital	Montgomery count dataset	The combined dataset
The total number of people	444	110	554
The total number of TB patients	222	55	277
The total number of females	242	58	300
The total number of males	202	52	254
The total number of female patients	86	19	105
The total number of male patients	136	36	172
The range for the distribution of age of TB patients	17 to 89	15 to 89	15 to 89
The range for the distribution of age of female patients	16 to 83	25 to 89	16 to 89
The range for the distribution of age of male patients	18 to 89	15 to 73	15 to 89
The average age of all people	43.19	41.07	42.77
The average age of females	39.76	42.12	40.21
The average age of males	47.30	39.90	45.78
The average age of patients	50.76	49.10	50.43
The average age of female patients	47.38	59.26	49.53
The average age of male patients	52.90	43.72	50.98

According to these findings, the age of the oldest patient for each gender in all data is the same although the age of the youngest patient for each gender is different. The average age of female patients and male patients are 50 and 51, respectively, while the average age of all TB patients is 50. The distribution of gender of TB patients indicates that the total number of male patients is 62% of all TB cases. The number of TB cases among male is 68% while the number of TB cases among female is 38%. Besides, the findings from the combined dataset show that TB cases were mostly common in adults and the number of TB cases is higher in males. This is parallel with the "Global Tuberculosis 2022" Report by WHO as the report expresses that the number of men affected by TB disease is more than the number women affected by TB disease [2].

4.2. Predictive models

The models were assessed by comparing their performances. The results of performance measures for each predictive model are presented in Table 5 while Figure 3 gives the accuracy and loss graphs for the first three best performances in terms of accuracy rate obtained from training dataset.

Table 5. A summary of comparing values of performance measures for each model

Model	Accuracy	Recall	Precision	F1 score	Misclassification	FN	FP
ResNet Model 1	78.41%	85.71%	68.18%	75.95%	21.59%	5	14
ResNet Model 2	80.68%	88.57%	70.45%	78.48%	19.32%	4	13
Xception Model 1	-	-	-	-	-	-	-
GoogLeNet Model 1	50%	-	-	-	-	-	-
GoogLeNet Model 2	50%	-	-	-	-	-	-
GoogLeNet Model 3	50%	-	-	-	-	-	-
GoogLeNet Model 4	50%	-	-	-	-	-	-
GoogLeNet Model 5	78.41%	100%	56.82%	72.47%	21.59%	0	19
GoogLeNet Model 6	71.59%	100%	43.18%	60.32%	28.41%	0	25
GoogLeNet Model 7	86.36%	90%	81.82%	85.72%	9.10%	4	8
GoogLeNet Model 8	81.82%	78%	88.64%	82.98%	5.68%	11	5
ResNet Model 3	80.90%	80.36%	81.82%	81.08%	10%	10	11
ResNet Model 4	88.18%	95.65%	80%	87.13%	10%	2	11
Xception Model 2	99.09%	100%	98.18%	99.10%	0.91%	0	1
GoogLeNet Model 9	50%	-	-	-	-	-	-
GoogLeNet Model 10	50%	-	-	-	-	-	-
GoogLeNet Model 11	50%	-	-	-	-	-	-
GoogLeNet Model 12	50%	-	-	-	-	-	-
GoogLeNet Model 13	50%	-	-	-	-	-	-
GoogLeNet Model 14	50%	-	-	-	-	-	-
GoogLeNet Model 15	50%	-	-	-	-	-	-
GoogLeNet Model 16	50%	-	-	-	-	-	-

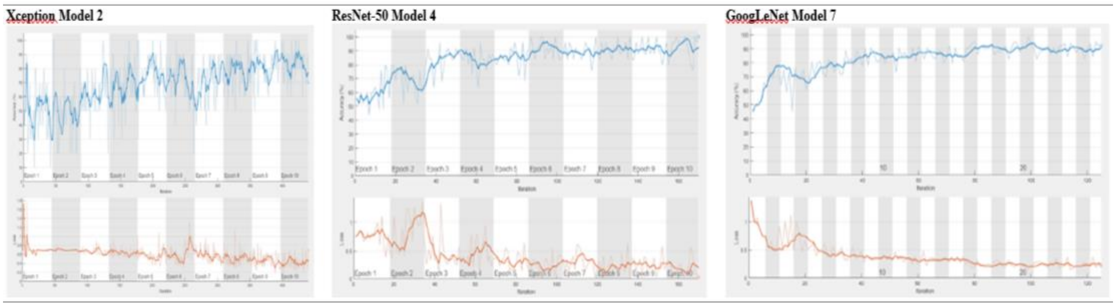


Figure 3. The accuracy and loss graphs for the first three best performances

These results show that the ResNet-50 Model 4 presents the highest accuracy rate, recall rate, and F1 score among the ResNet-50 models while it gives the lowest value of FN. When investigating GoogLeNet models, the highest accuracy rate, and F1 score are provided from the GoogLeNet Model 7. The highest precision rate is obtained from the GoogLeNet Model 8 while the GoogLeNet Model 8 gives the lowest misclassification rate. Besides, the lowest value of FP is provided from the GoogLeNet Model 8. However, the Xception Model 2 provides the best performance when evaluating in terms of accuracy, recall, misclassification, F1 score, FN, and FP among all the models.

5. Conclusion and Discussion

CADe and CADx systems are clinical DSS which can also utilize DL to manipulate CRs, deal with medical problems, support medical decision making. They enhance the productivity in diagnosis and treatment because DL rises accuracy rates and diminishes errors as well as reducing costs and time for detection. Furthermore, unbiased AI-based CADe and CADx systems require the dataset that covers balanced and different varied population and several features. Therefore, the study was performed to fill the gap in the study employed CXR images from hospital in Turkey when examining how robust the DCNNs to identify TB disease by comparing the performance of DCNNs architectures.

Experiments were carried out with EDA and predictive analytics. The distribution of demographic variables in the dataset were investigated with EDA while predictive analytics was performed for implementing the binary classification models with ResNet-50, Xception, and GoogLeNet. Before implementing the models, the data were augmented but lung segmentation were not applied. The performances of these DCNNs models were evaluated on the trained dataset from the Hospital in Turkey and Montgomery Count CXR datasets. For this, the accuracy, recall, precision, misclassification rate, F1 score, FP and FN values were compared.

The findings showed that the Xception Model 2 has the best performance, and the pretrained networks can be useful to improve CADx and CADe systems in determining TB disease on CXR images. However, this generalization should not be made because all types of DCNNs architectures have not been tested, and more than data is needed to avoid overfitting and develop a reliable system. An unbiased system can ignore any TB case or cause to apply TB treatment for any healthy individual. On the other hand, the study can encourage new research and practices to solve the issues arising from the lack of personnel and equipment for TB diagnosis. The research is also significant to reduce diagnosis time and propose an individualized treatment.

There are several limitations for the study: the data could be insufficient and may not be varied in terms of region, and age, and insufficient GPU power and time were available for a reliable DCNNs model. Hence, further experimental studies can be conducted with multi-classification or object detection on these datasets or tested on new data.

References

[1] World Health Organization, «Global tuberculosis report,» World Health Organization, Geneva, 2020.

[2] World Health Organization, «Global tuberculosis report,» World Health Organization, Geneva, 2022.

[3] T.C. Sağlık Bakanlığı, «Ulusal tüberküloz kontrol programı,» T.C. Sağlık Bakanlığı Yayın No: 1129, Ankara, 2022.

[4] T.C. Sağlık Bakanlığı Halk Sağlığı Genel Müdürlüğü, «Tüberküloz tanı ve tedavi rehberi,» T.C. Sağlık Bakanlığı Halk Sağlığı Genel Müdürlüğü, Ankara, 2019.

[5] E. J. Hwang, S. Park, K.-N. Jin, J. I. Kim, S. Y. Choi, J. H. Lee, J. M. Go, J. Aum, J.-J. Yim, J.-J. Yim ve C. M. Park, «Development and validation of a deep learning–based automatic detection algorithm for active pulmonary tuberculosis on chest radiographs,» Clinical Infectious Diseases, cilt 65, no. 5, pp. 739-747, 2019.

[6] World Health Organization, «Chest Radiography in Tuberculosis Detection: Summary of current WHO recommendations and guidance on programmatic approaches,» World Health Organization, Geneva, 2016.

[7] R. A. Castellino, «Computer aided detection (CAD): an overview,» Cancer Imaging, pp. 17-19, 23 August 2005.

[8] K. Suzuki, «Computer-aided detection of lung cancer,» Image-based computer-assisted radiation theraphy, Singapore, Springer, 2017, pp. 9-40.

[9] J. Zhang, X. Yutong, Q. Wu ve Y. Xia, «Medical image classification using synergic deep learning,» Medical image analysis, cilt 54, pp. 10-19, 2019.

[10] D. W. v. S. T. I. M. E. Matheny, «Artificial intelligence in health care a report from the national academy of medicine,» JAMA,, cilt 323, no. 6, pp. 509-510, 2020.

[11] M. A. Musen, B. Middleton ve R. A. Greenes, «Clinical decision support system,» Biomedical informatics, Cham,, Springer, 2021, pp. 795-840.

[12] K. C. Laudon ve J. P. Laudon, Management information systems: managing the digital firm, London: Pearson, 2020.

[13] R. Sharda, D. Dursun ve E. Turban, Analytics, data science, & artificial intelligence: systems for decision support, Hoboken: Pearson, 2020.

[14] E. Alpaydın, «Introduction,» Introduction to machine learning, Cambrige, MIT Press, 2010, pp. 1-19.

[15] A. M. Antoniadi, Y. Du, Y. Guendouz, L. Wei, C. Mazo, B. A. Becker ve C. Mooney, «Current challenges and future opportunities for XAI in machine learning-based clinical decision support systems: a systemeatic review,» Applied Sciences, cilt 11, no. 11, 2021.

[16] A. K. Jain, J. Mao ve K. M. Mohiuddin, «Artificial neural networks: a tutorial,» Computer, cilt 29, no. 3, pp. 31-44, 1996.

[17] R. Yamashita, M. Nishio, R. K. G. Do ve K. Togashi, «Convolutional neural networks: an overview and application in radiology,» Insights into imaging, cilt 9, no. 4, pp. 611-629, 2018.

[18] S. Kulkarni ve S. Jha, «Artificial intelligence, radiology, and tuberculosis: a review,» Academic radiology, cilt 27, no. 1, pp. 71-75, 2020.

[19] R. H. Abiyev ve M. K. S. Ma'aitah, «Deep convolutional neural networks for chest diseases detection,» Journal of healthcare engineering, 2018.

[20] M. Mamalakis, A. J. Swift, B. Vorselaars, S. Ray, S. Weeks, W. Ding, R. H. Clayton, L. Mackenzie ve A. Banerjee, «DenResCov-19: A deep transfer learning network for robust automatic classification of COVID-19, pneumonia, and tuberculosis from X-rays,» Computerized Medical Imaging and Graphics, no. 102008, p. 94, 2021.

[21] E. Ölmez, O. Er ve A. Hızıroğlu, «Deep learning in biomedical applications: detection of lung dise convolutional neural networks,» Deep learning in biomedical and health informatics, Boca Raton, CR 2021, pp. 97-115.

[22] P. Lakhani ve B. Sundaram, «Deep learning at chest radiography: automated classification of p tuberculosis by using convolutional neural networks,» Radiology, cilt 284, no. 2, pp. 574-582, 2017.

[23] B. K. Karaca, S. Güney, B. Dengiz ve M. Ağıldere, «Comparative Study for Tuberculosis Detection Deep Learning,» 2021 44th International Conference on Telecommunications and Signal Processir Brno, 2021.

[24] B. Oltu, S. Güney, B. Dengiz ve M. Ağıldere, «Automated tuberculosis detection using pre-trained an 2021 44th International Conference on Telecommunications and Signal Processing (TSP), Brno, 202

[25] Y. Cao, C. Liu, M. J. Brunette, N. Zhang, T. Sun, P. Zhang, J. Peinado, E. S. Garavito, L. L. Garcia Curioso, «Improving tuberculosis diagnostics using deep learning and mobile health technologie resource-poor and marginalized communities,» 2016 IEEE first international conference on connecte applications, systems and engineering technologies (CHASE), Washington, 2016.

[26] S. Stirenko, Y. Kochura, O. Alienin, O. Rokovyi, Y. Gordienko, P. Gang ve W. Zeng, «Chest X-ray a tuberculosis by deep learning with segmentation and augmentation,» 2018 IEEE 38th International Co on Electronics and Nanotechnology (ELNANO), Kyiv, 2019.

[27] T. Rahman, A. Khandakar, M. Abdulkadir, K. R. Islam ve K. F. Islam, «Reliable tuberculosis detecti chest X-ray with deep learning, segmentation and visualization,» IEEE Access, cilt 8, pp. 191586-1916

[28] S.-J. Heo, Y. Kim, S. Yun, S.-S. Lim, J. Kim, C.-M. Nam, E.-C. Park, I. Jung ve J.-H. Yoon, «Deep algorithms with demographic information help to detect tuberculosis in chest radiographs in annual health examination data,» International journal of environmental research and public health, cilt 16, 250, 2019.

[29] T. M. Navamani, «Efficient deep learning approaches for health informatics,» Deep learning and computing environment for bioengineering systems, St. Louis, Academic Press, 2019, pp. 123-137.

[30] M. J. Willemink, W. A. Koszek, C. Hardell, J. Wu, D. Fleischmann, H. Harvey, L. R. Folio, R. M. Sum L. Rubin ve M. P. Lungren, «Preparing medical imaging data for machine learning,» Radiology, cilt 2 pp. 4-15, 2020.

AITA
ARTIFICIAL INTELLIGENCE
THEORY AND APPLICATIONS

ARTIFICIAL INTELLIGENCE
THEORY and APPLICATIONS
ISSN: 2757-9778 **||** **ISBN :** 978-605-69730-2-4
More information available at **aitajournal.com**

May 1st, 2023

To whom it may concern

We are writing to bring to your attention an error that was made in one of the articles that was published in AITA. Specifically, we would like to address an error in the name of author, which was misspelled during listing. We understand how important it is to ensure that author names are spelled correctly for the recognition of their contributions to the field, and we should have taken greater care in this matter.

The article titled "*Artificial Intelligence Applications in Management Information Systems: A Comprehensive Systematic Review with Business Analytics Perspective*" and was published in the April 30th 2021, Volume 1, Issue 1. The author's name, listed in Dergipark platform as **Halil İbrahim ÇELEBİ** was misspelled and the correct form of the author's name is **Halil İbrahim CEBECİ**. This error is particularly significant as it may cause confusion among readers searching for other works by the author, as well as for academic citation purposes.

Wrong Form	**Correct From**	**URL**
Halil İbrahim ÇELEBİ	Halil İbrahim CEBECİ	https://dergipark.org.tr/en/pub/aita/issue/70741/1137794

We would like to express our sincerest apologies to the author for this error. We assure that we are committed to upholding the highest standards of academic publishing and that we will take steps to ensure that this type of mistake does not happen again in the future.

Sincerely,

AITA Editorial Team