ISSN: 2633-4828 Vol. 6 No.1, January, 2024

International Journal of Applied Engineering & Technology

PREDICTING POWER OUTPUT OF A COMBINED CYCLE POWER PLANT USING ML MODELS

Kshitiz Singh Chauhan, Harsh and Gaurav Yadav
Delhi Technological University
[1]kshitizchauhan38@gmail.com, [2]harsh.me09.03@gmail.com and [3]gauravyadav2308@gmail.com

ABSTRACT

This research paper investigates the challenge of optimizing energy production and minimizing wastage in combined cycle power plants (CCPPs). The inability to store excess energy poses a significant hurdle in meeting consumer demand efficiently. Addressing this very pressing issue could lead to cost-effective electricity generation and improved consumer access to affordable energy. A potential solution, could potentially lie in accurately predicting power output based on various parameters affecting energy production. To explore this solution, we cunningly utilize machine learning (ML) plus ensemble ML algorithms in our study. The focal point of our research is the comparison between the performance of diverse ML methods in predicting CCPP power output. In particular, we evaluate the effectiveness of linear regression, ridge regression, lasso regression, decision tree, random forest and LGBM models in this context. The evaluation criteria for judging the models have been taken as mean square error, root mean square error and R^2 score. The results showed that tuned random forest showed the best performance and untuned decision tree showed the least accuracy in our analysis.

Index Terms - Combined Cycle Power Plant(CCPP), Machine learning, Linear Regression, Ridge Regression, Lasso Regression, Decision Trees, Random Forest, LGBM

INTRODUCTION

Combined-cycle power plants utilize both gas and steam turbines concurrently to generate electricity more efficiently compared to traditional simple-cycle plants. This method can yield up to 50% more electricity from the same amount of fuel. The waste heat produced by the gas turbine is harnessed to power the steam turbine, thereby increasing overall energy production[1]-[3].

The principle behind this is that the working fluid, or exhaust, in the first engine remains hot enough after its cycle is over for a second heat engine to use the exhaust heat to generate energy. The operation of a combined-cycle power plant involves several key stages:

- **Gas Turbine cycle:** A gas turbine, which is quite similar to a jet engine is an important component. By burning natural gas or maybe another fuel, it produces hot gasses that burn super fast. When these hot gasses are pointed across its blades, a turbine spins quite energetically. The generator links up with the turbine, converting the rotational energy into electrical power.
- **Waste heat recovery**: The exhaust gases contain a considerable quantity of heat even after the gas turbine has completed its task. This "waste heat" is not squandered in a combined cycle plant. A heat recovery steam generator is used to channel the hot exhaust gases (HRSG). This apparatus extracts the residual thermal energy.
- **Steam Turbine cycle**: The captured heat is used to produce steam from water. This steam is then directed over the blades of a steam turbine. Like the gas turbine, the steam turbine is connected to a generator, producing additional electricity.

Predicting power output in advance allows for the efficient fulfillment of consumer needs. This research aims to compare the efficacy of various machine learning models in predicting power output from combined cycle power plants (CCPPs). Accurately estimating net energy production is crucial for providing low-cost energy.

LITERATURE REVIEW

Salama Alketbi and Ali Bou Nassif compared four models including KNN, random forest etc. They conducted a study by employing three ML models which were K nearest neighbors, multilayer perceptron and random forest. In their research they found random forest as best performing model with RMSE of 3.3061[1] . Ali Alperen Islikaye and Aydin Cetin compared k-NN Regression, Simple Linear Regression, Linear Regression, Decision Trees, Bayesian Regression and RANSAC Regression. They concluded k-NN, Linear Regression and RANSAC regressions achieved the best performance among the other tested ML models [2] . C. Ahamed Saleel used deep learning models and conducted a comparative predictive analysis and trained four models of each neural network i.e ANN and DNN and found out that there output were in same range of accuracy [3]

DATASET OVERVIEW

The data has been collected through UCI machine learning repository and the link for that dataset is Link:- https://archive.ics.uci.edu/dataset/294/combined+cycle+power+plant

The dataset in question is a a really rich collection of 9,568 data points, meticulously gathered during an extensive period of six years, from 2006 to 2011. This period marks a phase when the Combined Cycle Power Plant (CCPP) were operating at full capacity, making sure all data reflects conditions of peak performance and efficiency[4] . The operation of CCPPs are highly influenced by ambient conditions, which makes this dataset particularly valuable for understanding and optimizing power plant performance under varying environmental scenarios.

I. Independent Variables (Features)

The dataset features four critical ambient variables that are known to directly influence the efficiency and output of the power plant. These variables were continuously monitored and recorded as average values to capture the environmental conditions around the plant. The independent variables include:

- **Temperature (T):** Recorded in degrees Celsius (°C), this specific variable represents the ambient temperature. Temperature is really a very crucial factor affecting the thermal efficiency of the power plant, as it strongly influences the air density and thus the amount of air available for combustion processes.
- **Exhaust vacuum (V):** Measured in centimeters of mercury (cm Hg), the exhaust vacuum level is indicative of the turbine's operating condition. A really higher vacuum actually suggests more efficient turbine operation, leading to better overall plant efficiency.
- **Ambient pressure (AP):** Measured in millibars (mbar) and reflects the atmospheric pressure surrounding the power plant. Ambient pressure can surprisingly affect the air intake of the combustion system and consequently the power output.
- **Relative humidity (RH):** Expressed as a percentage (%), relative humidity measures the amount of moisture present in the air. Humidity can impact the cooling process within the plant and the density of the air, which in turn definitely affects combustion and power generation efficiency.

Sensors positioned throughout the plant assist in capturing these values by taking a secondly record of ambient variables.

II. Independent Variables (Features)

- **Net hourly energy output(PE):** It is the net hourly electrical energy output measured in Mega Watts (MW)

Table I

Parameter	AT	V	AP	RH	PE
Total count	9568	9568	9568	9568	9568
Mean value	19.651	54.305	1013.259	73.308	454.365
Standard deviation	7.452	12.707	5.938	14.600	17.066

Minimum value	1.810	25.360	992.890	25.560	420.260
Maximum value	37.110	81.560	1033.300	100.160	495.760

DATA ANALYSIS

III. Correlations

Correlations are used to determine the linear relationship between various variables in the picture. The value of correlation varies from -1 to +1. +1 indicates a perfect positive linear relationship between the variables. -1 indicates a perfect negative linear relationship between the variables whereas 0 indicates no linear relationship between them.

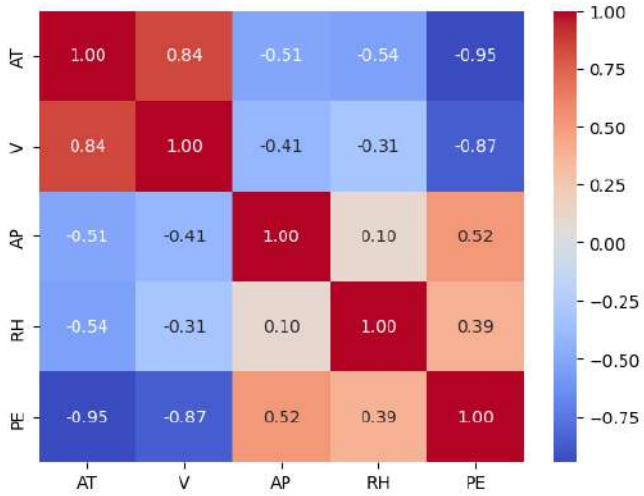

Figure I

CORRELATION MATRIX

Expanding on the observations regarding the relationship between different variables in the dataset, specifically ambient temperature (AT), exhaust vacuum (V), and power output (PE), provides deeper insights into the scenario of the power plant's operational efficiency. There is a strong negative linear relationship between ambient temperature (AT) and power output(PE), as indicated by the correlation coefficient, which is almost equal to -1. This indicates that the plant's power output tends to decrease as the surrounding temperature increases. The thermodynamic characteristics of the power generation cycle are responsible for this relationship. Increased outside temperatures reduces the effectiveness of the thermal cycle's cooling process, thereby lowering the power plant's overall efficiency. This effect is noticeable because it has a direct bearing on the plant's operating strategy, particularly when the outside temperature is high.

The fact that the correlation coefficient between ambient temperature (AT) and exhaust vacuum(V) is nearly 1 indicates a strong positive linear relationship between these two. This suggests that the exhaust vacuum tends to increase along with the ambient temperature. The thermodynamic laws that control how the power plant operates provide an explanation for this relationship. Increased exhaust gas volumes may be caused by warmer ambient temperatures, which could raise the vacuum levels in the turbine's exhaust system.

RESEARCH ALGORITHMS

I. Linear regression

Linear regression is a fundamental statistical and machine learning method used to model the relationship between a dependent variable and one or more independent variables. Finding a linear equation that most accurately predicts the dependent variable from the independent variables is the aim of linear regression.

As our data contains more than one independent variables so we have used to multiple linear regression for predictive analysis. The linear equation that governs this model is:

$$Y = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots + \beta nXn \quad (1)$$

Here:

Y = Dependent variable (power output of the plant)

Xi = Value of the ith independent feature

Bi = Value of ith coefficient

$B0$ = Intercept of the regression line fitted

The first step in implementing the linear regression model is to find the possible values of β that minimize the difference between the observed values and predicted values by the model. The method of least squares, which finds the best-fitting line by minimizing the sum of the squares of the vertical distances (can also be termed as errors) of the points from the line, is usually used to accomplish this process.

II. Ridge regression

Ridge regression is an important machine learning technique that is necessary for developing robust models where multicollinearity and overfitting are likely to occur. This method modifies standard linear regression by including a penalty term proportionate to the square of the coefficients. It is particularly useful for managing highly correlated independent features. The main benefits of ridge regression over linear regression are better model generalization to perform better on testing data that hasn't been seen yet, reducing overfitting through adding complexity penalties, and managing multicollinearity by balancing the effects of correlated variables[5].

The residual sum of squares is defined as:

$$RSS = \sum_{i=1}^{n}(y - \dot{y})^2 \quad (2)$$

In ridge regression a L2 term (Ridge penalty) is added at the last which is the sum of squares of coefficients. This discourages large coefficients, leading to a model where the influence of each feature on the target variable is kept as small as possible, thus reducing model complexity and helping to prevent overfitting. The loss function in ridge regression is as follows:

$$\text{Loss}(\beta) = \sum_{i=1}^{n}(y - \dot{y})^2 + \lambda \sum_{j=1}^{p}\beta_j^2 \quad (3)$$

Here

y = Actual value

$\dot{y}$ = Predicted value

P = Total number of independent features

B_j = Coefficient associated with j^{th} feature

λ = Regularization parameter

When using ridge regression in practical situations, selecting the right regularization parameter, or lambda, is crucial. Choosing λ is an important choice. A model with an excessively large λ will be overly simplistic and may underfit the data (high bias). Should λ be excessively small, the model might overfit the data, resulting in a high variance, and the penalty effect would be negligible. Usually, cross-validation is employed to determine the optimal λ.

III. Lasso regression

Lasso regression, also known as Least Absolute Shrinkage and Selection Operator, is a kind of linear regression that, like ridge regression, has a regularization element in its objective function. The kind of penalty that is applied to the coefficients, however, is what really makes a difference. In addition to assisting in the reduction of overfitting, Lasso regression allows for variable selection by essentially eliminating certain variables from the model by shrinking their coefficients to exactly zero.

The loss function for lasso regression is:

$$\text{Loss}(\beta) = \sum_{i=1}^{n}(y - \hat{y})^2 + \lambda \sum_{j=1}^{p}|\beta_j| \qquad (4)$$

The lasso penalty promotes sparsity in the model coefficients. By penalizing the sum of the absolute values of the coefficients, lasso can drive some coefficients to zero, effectively selecting a simpler model that relies on fewer variables.

IV. Decision tree

Decision Trees are non-linear predictive models widely used in machine learning for both classification and regression tasks. In our particular research we have used the decision tree regressor for predictive analysis. They behave like humans in decision-making processes by splitting data into branches at decision nodes, leading to a set of results represented by leaf nodes. In case of continuous value target variable like in our case, decision tree gives the average of all the values present in the leaf node. Decision Trees are intuitive and easy to understand, making them popular for a wide range of applications.

It has a hierarchical tree structure consisting of a root node, branches, internal nodes, and leaf nodes [6].

- **Starting at the root:** At the root, the entire dataset is taken into account.To divide the data into subsets, the algorithm chooses a feature and a split point on that feature.
- **Best split determination:** Based on factors such as variance reduction in regression, entropy in classification tasks; or Gini impurity the decision to split occurs at each node, maximizing the homogeneity of the resulting nodes is the aim.
- **Recursive splitting:** This process of selecting the best feature and split points, then splitting the dataset, continues recursively for each child node.
- **Stopping criteria:** When a stopping condition is satisfied, the recursion comes to an end. Might occur when the nodes attain a certain depth to avoid overfitting, when the number of samples reaches a minimum, or when more splitting does not increase homogeneity with the resulting nodes. All these hyperparameters have been tuned in this research to achieve a robust learning model.
- **Prediction:** A decision tree uses the majority class of the samples inside it to predict the class of a leaf node in order to classify data. Regression forecasts a value by averaging the values within the leaf.

Hyperparameter tuning is also possible in decision trees where we can systematically search for the optimal set of hyperparameters which will give the best results from the model. In decision trees we can tune the parameters such as maximum depth of the tree, minimum number of samples required per leaf, minimum number of samples required to split the node.

Without constraints, decision trees can grow deep with complex structures and get vulnerable to overfitting so it's generally a good practise to tune the parameters to build a robust model that performs well on unseen data as well.

V. Random forest

Random forest is an ensemble learning method which operates by constructing multiple decision trees and merges them together to get a more accurate and stable prediction. Each tree in the forest is built from a random sample of the training data, making each tree a bit different from the others. This process is known as "bagging". In contrast, "boosting" is a term referred to the practise where weak learners are combined sequentially to create a strong learner with the highest accuracy. When building each tree, Random Forest randomly selects a subset of the features (variables) at each split in the decision tree. This adds to the diversity of the model, making it more robust and reducing the risk of overfitting.

VI. Light gradient boosting machine (LGBM)

Using tree-based learning algorithms, LGBM is an implementation of gradient boosting machines (GBM). LGBM distinguishes itself from other GBM frameworks by employing an innovative tree-building algorithm. Unlike most other boosting techniques, which grow trees horizontally (level-wise), it grows trees vertically (leaf-wise). LGBM will thus select the leaf with the largest delta loss to grow, enabling quicker convergence and increased efficiency[7][8].

Key features of LGBM are:

- **Efficiency and scalability:** Gradient boosting gains significant efficiency and scalability improvements from LGBM. Because of its ease of handling large datasets, it is appropriate for situations where computational resources are scarce.
- **Leaf wise growth strategy:** LGBM grows trees leaf-wise, in contrast to conventional LGBM frameworks that grow trees level-wise. This method can achieve much better accuracy with fewer trees because it minimizes more loss than the level-wise method.
- **Handling of categorical features:** LGBM can naturally handle categorical features by representing them in a way that is more informative for the model, without the need for extensive pre processing to convert categories into numerical values.
- **Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB):** These are the two main methods that LGBM uses to minimize the quantity of features and data instances without appreciably compromising accuracy. GOSS randomly samples the instances with small gradients while retaining all the instances with large gradients. In order to minimize the feature dimension, EFB bundles mutually exclusive features, or features that are rarely non-zero simultaneously.

VII. Performance metrics

To evaluate a model's prediction accuracy we've chosen three performance checking metrics namely, MSE(Mean square error), RMSE(Root mean square error) and coefficient of determination R2. The mathematical representation is given below:-

$$\text{MSE} = \frac{1}{n} \times \sum_{i=1}^{i=n} (y_{actual} - y_{predicted})^2 \qquad (5)$$

$$RMSE = \sqrt{\frac{1}{n} \times \sum_{i=1}^{i=n} (y_{actual} - y_{predicted})^2} \quad (6)$$

Here,

n = Total number of samples

y_{actual} = Actual value of power output

$y_{predicted}$ = Predicted value of power output

After splitting the dataset into independent and dependent variable, the complete dataset is split into training data and testing data. 70% of total entries (6697) in dataset are used for training the models and 30% of the entries (2871) will be used to test all the models.

RESULTS AND DISCUSSIONS

First three models trained were the linear regression, ridge regression and lasso regression. Linear regression resulted in a MSE of 20.3683, RMSE of 4.5131 and a R2 score of 0.93 on testing data. Ridge regressor was trained and optimal value of λ found out using GridSearchCV was 27.8. It resulted in a MSE of 20.3690, RMSE of 4.5132 and a R2 score of 0.93. Similarly lasso regressor was trained and optimal value of λ was found to be 10-6. Due to this very small value of λ lasso regressor gave similar results to linear regressor. The MSE was found out to be 20.3683, RMSE of 4.5131 and a R2 score of 0.93.

The decision tree regressor (without any hyperparameter tuning) gave an MSE of 0 on training data but MSE of 24.5744 was found on testing data indicating an overfitted decision tree. To reduce the overfitting the model was tuned with pre selected options for various hyperparameters and after the tuning the MSE on training data was found to be 10.5317. On testing data the MSE was reduced to 16.1826, RMSE was found to be 4.0227 and a R2 score of 0.944.

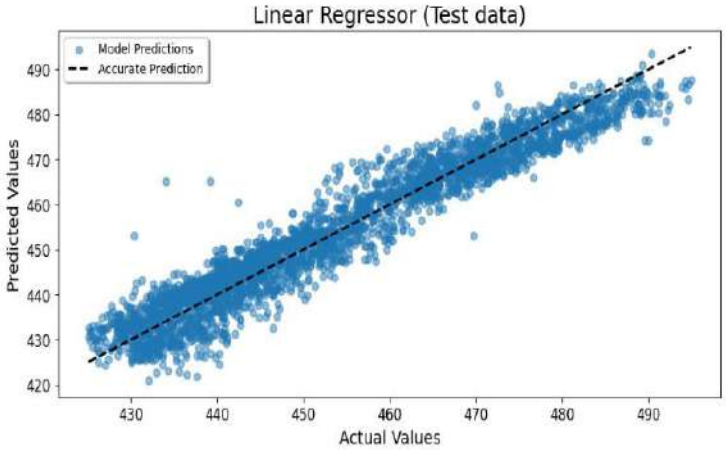

Figure II Graph between Actual and Predicted Values by Linear Regressor

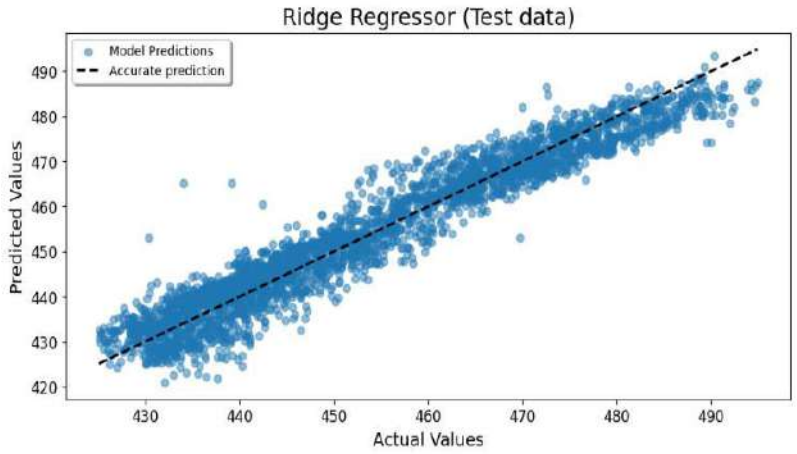

Figure III Graph between Actual and Predicted Values By Ridge Regressor

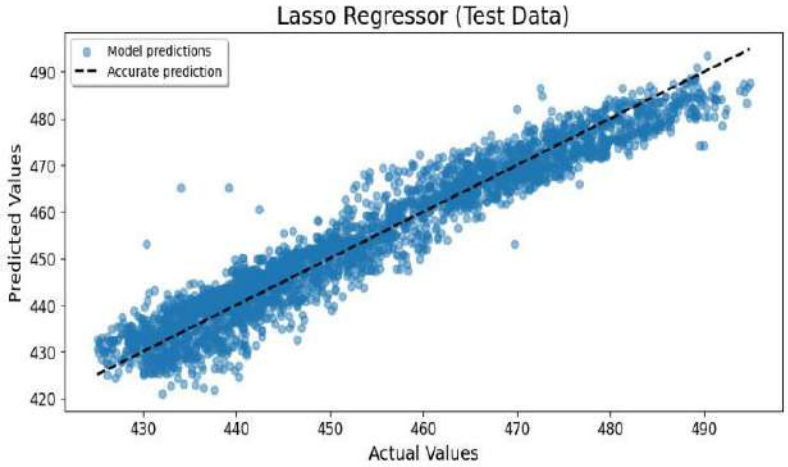

Figure IV Graph between Actual and Predicted Values by Ridge Regressor

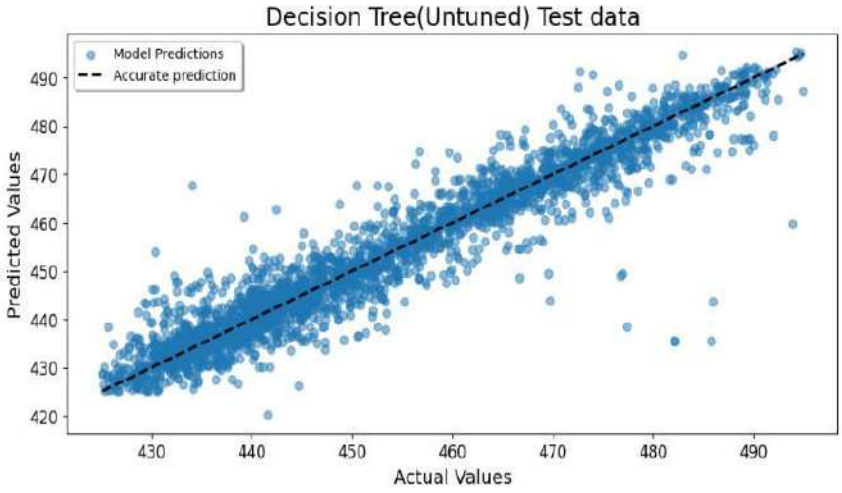

Figure V Graph Between Actual and Predicted Values by Untuned Decision Tree

We noticed many outliers in the above graph. An untuned decision tree predicted certain values quite far away from the accurate line thus increasing the errors. We could remove these outliers by fine tuning the decision tree (Table II). The graph for tuned decision tree is given below.

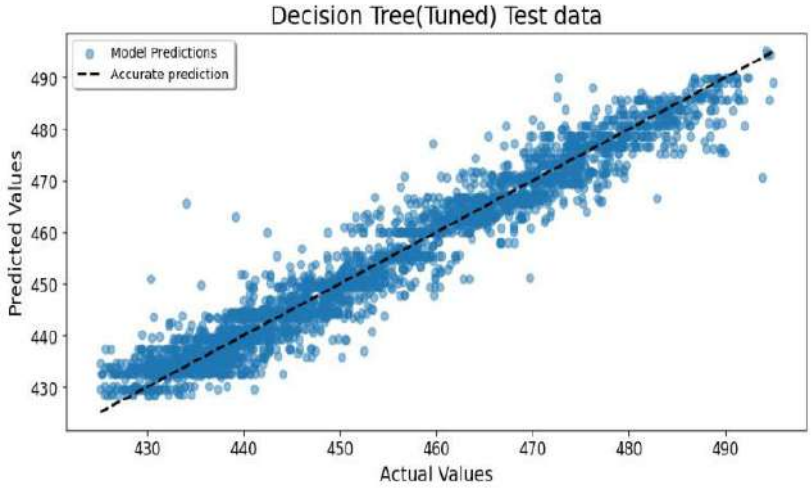

Figure VI Graph Between Actual And Predicted Values By Tuned Decision Tree

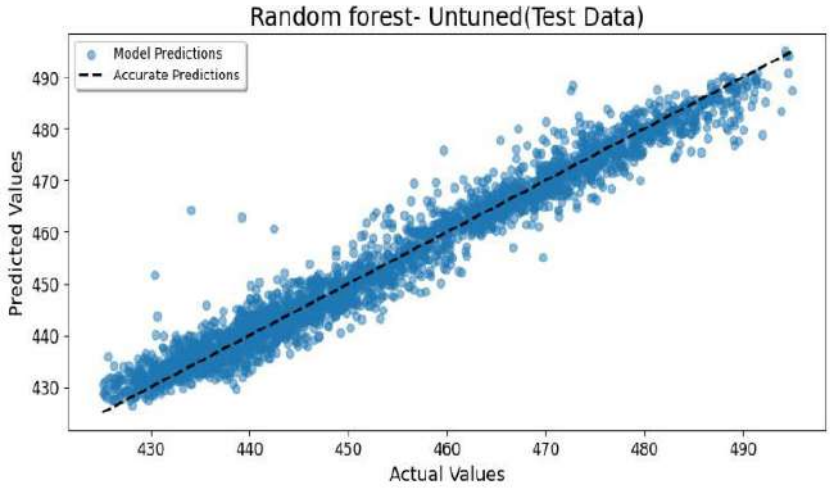

Figure VII Graph between Actual and Predicted Values By Untuned Random Forest

Random forest regressor without any hyperparameter tuning gave a MSE of 11.0897, RMSE of 3.3301 and a R2 score of 0.96 on testing data. To reduce the MSE even more we attempted hyperparameter of the model. After hyperparameter tuning the MSE was reduced to 10.7825, RMSE was 3.2836 and a R2 score of 0.96. The best selected parameters are given in Table III.

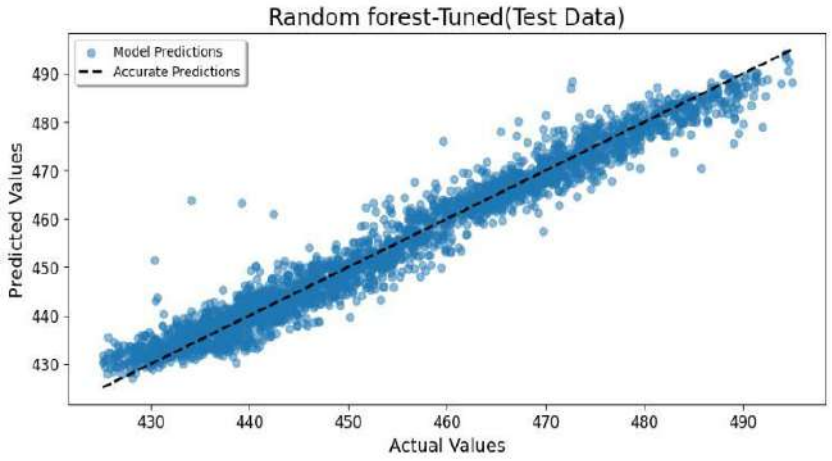

Figure VIII Graph Between Actual and Predicted Values By Tuned Random Forest

LGBM model gave a MSE of 11.0215, RMSE of 3.3198 and a R^2 score of 0.96.

Table II Tuned Parameters for Decision Tree

Tuning parameters	Value selected	Description
ccp_alpha	0	To reduce the complexity of simulation
criterion	MSE	MSE was chosen as criteria to evaluate the error
max_features	auto	The number of features to consider when looking for the best split
max_depth	9	The maximum depth of the tree
min_impurity_decrease	0	Stop a split if the amount of split is less than zero
max_leaf_nodes	None	Leaf nodes were not limited
min_samples_split	8	The minimum required samples to split an internal node
min_samples_leaf	2	The leaf node's minimum samples

Table III Tuned Parameters For Random Forest

Tuning parameters	Value selected	Description
ccp_alpha	0	To reduce the complexity of simulation
criterion	MSE	MSE was chosen as criteria to evaluate the error
max_features	0.6	The number of features to consider when looking for the best split
max_depth	13	The maximum depth of the tree in RF
min_impurity_decrease	0	Stop a split if the amount of split is less than zero
max_leaf_nodes	None	Leaf nodes were not limited
min_samples_split	2	The minimum required samples to

		split an internal node
min_samples_leaf	1	The leaf node's minimum samples
max_samples	1	No. of samples(%) to train each decision tree
n_estimators	110	Number of trees in the forest

CONCLUSION

The best model with most accurate predictions was random forest. After hyperparameter tuning it gave a MSE of 10.7825, RMSE of 3.2836 and a R2 score of 0.96. The other models in order of their prediction accuracy with the most accurate model coming first are as follows:- LGBM, Random forest (Untuned), Decision tree (Tuned), Linear regression = Lasso regression, Ridge regression, Decision tree (Untuned).

ACKNOWLEDGMENT

We would like to express our profound gratitude to Professor Rajesh Kumar of the Department of Mechanical Engineering at Delhi technological university for his invaluable guidance, insightful feedback, and unwavering support throughout the course of this research project. His expertise and dedication not only shaped this work but also inspired us to explore our potential. We are truly grateful for his mentorship and for providing us with the opportunity to work under his guidance. This project would not have reached its fruition without his encouragement and constructive criticism.

REFERENCES

[1] Salama Alketbi, A. B. (2020). Predicting the power of a combined cycle power plant using Machine learning methods. *2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)* (p. 5). Sharjah, United Arab Emirates: IEEE, https://doi.org/10.1109/CCCI49893.2020.9256742

[2] Ali Alperen Islikaye, A. C. (2018). Performance of ML methods in estimating net energy produced in a combined cycle power plant. *2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)* (p. 4). Istanbul, Turkey: IEEE, https://doi.org/10.1109/SGCF.2018.8408976

[3] Saleel, C. A. (2021). Forecasting the energy output from a combined cycle thermal using deep learning models. *Case Studies in Thermal Engineering* 28 (2021) 101693, https://doi.org/10.1016/j.csite.2021.101693.

[4] Tfekci, P. a. (2014). *Combined Cycle Power Plant (Dataset)*. Retrieved from UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/294/combined+cycle+power+plant

[5] Jacob Murel Ph.D., E. K. (2023, November 21). *Ridge Regression*. Retrieved from IBM: https://www.ibm.com/topics/ridge-regression#:~:text=Ridge%20regression%E2%80%94also%20known%20as,for%20multicollinearity%20in%20regression%20analysis.

[6] Saini, A. (2024, January 5). *Decision Tree A step by step guide*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

[7] *Light Gradient Boosting Machine*. (2023, May 23). Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/lightgbm-light-gradient-boosting-machine/

[8] Microsoft. (2024). *LightGBM* . Retrieved from Lightgbm docs: https://lightgbm.readthedocs.io/en/latest/Parameters.html

[9] R. Pugliese, S. Regondi, R. Marini, Machine learning-based approach: global trends, research directions, and regulatory standpoints, Data Science and Management 4 (2021) 19–29, https://doi.org/10.1016/j.dsm.2021.12.002

[10] A.A. Mas'ud, Comparison of three machine learning models for the prediction of hourly PV output power in Saudi Arabia, Ain Shams Engineering Journal 13 (2022) 101648, https://doi.org/10.1016/j.asej.2021.11.017

[11] A. A. Islikaye and A. Cetin, "Performance of ML methods in estimating net energy produced in a combined cycle power plant," *Proc. - 2018 6th Int. Istanbul Smart Grids Cities Congr. Fair, ICSG 2018*, pp. 217–220, 2018.

AUTHOR INFORMATION

Kshitiz Singh Chauhan, Undergraduate student, Department of Mechanical engineering, Delhi Technological University, New Delhi,110042, India

Harsh, Undergraduate student, Department of Mechanical engineering, Delhi Technological University, New Delhi, 110042, India

Gaurav Yadav, Undergraduate student, Department of Mechanical engineering, Delhi Technological University, New Delhi, 110042, India