P-ISSN 2586-9000
E-ISSN 2586-9027
Homepage : https://tci-thaijo.org/index.php/SciTechAsia
Science & Technology Asia
Vol.29 No.4 October - December 2024
Page: [197-212]
Original research article

Machine Learning-Driven Efficiency Estimation and Variable Analysis in Combined Cycle Power Plants

Vishakha Singh, Phisan Kaewprapha*

Department of Electrical and Computer Engineering, Faculty of Engineering, Thammasat School of Engineering, Thammasat University, Pathum Thani 12120, Thailand

Received 19 February 2024; Received in revised form 15 November 2024
Accepted 25 November 2024; Available online 27 December 2024

ABSTRACT

The combined cycle power plant (CCPP) has seen significant growth as a key player in the energy sector due to its efficient electricity generation and low greenhouse gas emissions. The growing global demand for electricity, fueled by rapid technological advancements, underscores the need for a reliable power supply. However, accurately predicting the efficiency of CCPPs is essential for optimizing performance and cost-effectiveness. The efficiency of power plants is influenced by a variety of environmental and internal factors, but traditional models often fail to capture these complexities. This study addresses these gaps by employing machine learning models to estimate the efficiency of a CCPP in Thailand, using a comprehensive dataset of fourteen input variables. Nine machine learning models, including regression and ensemble methods, were used for evaluation, with Random Forest Regression and Gradient Boosting achieving superior accuracy levels of 99.91% and 99.83%, respectively. Furthermore, the research delves into 14 distinct variables utilized for prediction and aims to determine which variables are of paramount significance in the assessment process.

Keywords: Combined cycle power plant; Efficiency; Gradient boosting; Machine learning; Random forest regression

1. Introduction

The supply of power evolves in tandem with technological advancements. Given the rapid changes in global technology consumption, there is an increasing need for a robust power supply to meet the surging demand. Consequently, the spectre of an impending energy shortage

***Corresponding author:** kphisan@engr.tu.ac.th
doi: 10.14456/scitechasia.2024.81

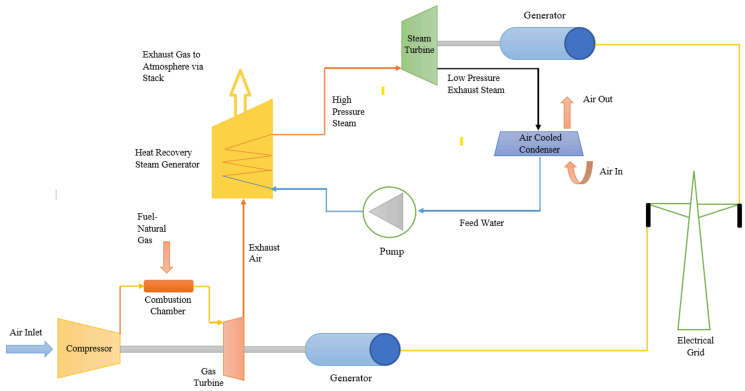

Fig. 1. Layout of a CCPP.

looms over the world. Power plants, which have been in operation for decades, remain the primary contributors to electricity generation. Owing to their ability to efficiently produce electricity with minimal greenhouse gas emissions, combined cycle power plants (CCPPs) are progressively displacing traditional power plants on a global scale [1].

The efficiency of power plants must be optimized to maintain and further reduce production costs [2]. Therefore, accurate energy efficiency prediction is essential to ensure cost-effectiveness. This study focuses on estimating the efficiency of CCPPs using various machine learning methods, taking into account diverse internal and environmental factors.

CCPPs employ a combination of a gas and steam turbine to convert the thermal energy from fuel into electrical power [3]. This procedure involves two cycles: the Brayton cycle, which mirrors a traditional gas turbine, and the Rankine Cycle, facilitated by a steam turbine. Significantly, the residual heat produced following the generation of electricity through the gas turbine is repurposed to generate steam, subsequently producing extra electrical power. In contrast, traditional power plants used to generate only about 33% of electricity, with a staggering 67% of energy going to waste. Since the introduction of CCPPs, the efficiency of electricity generation has increased by approximately 68% [4].

CCPPs are also more adaptable than conventional thermal power plants. For a visual representation, refer to Fig. 1, which provides a diagrammatic representation of a CCPP.

The production of electricity and the energy efficiency of a plant are intrinsically linked to a multitude of factors, encompassing both environmental and internal aspects. Any alteration in these factors can significantly influence a plant's efficiency. Thus, the establishment of a system to estimate this efficiency becomes a matter of paramount importance.

Typically, power plants operate at two distinct loads: Full Load and Partial Load. Plants that adhere to these load profiles adjust their power output in response to the fluctuating demand for electricity throughout the day [5]. For instance, during periods of lower electricity demand, the plant operates at partial capacity rather than running at full capacity. Conversely, when demand peaks, the plant shifts to full capacity. Combined Cycle Power Plants (CCPPs)

excel in maintaining high efficiency across a wide range of loads while simultaneously producing fewer harmful emissions and consuming less cooling water. Consequently, this paper is dedicated to assessing the efficiency of a power plant located in Thailand, taking into account various influencing factors.

While energy efficiency can be calculated through straightforward mathematical equations, relying solely on such equations may not yield accurate predictions. Often, numerous other seemingly insignificant factors come into play, exerting a substantial impact on a plant's efficiency. Mathematical formulas typically omit critical environment and inlying attributes, such as temperature, humidity, generators, and turbines. This paper aims to shed light on the significance of these internal and environmental factors in determining plant efficiency. We employ various intelligent models to estimate the competency of a CCPP, utilizing a dataset consisting of fourteen input variables.

In our previous work [6], we analysed Thailand's primary Combined Cycle Power Plant (CCPP) using a five-year dataset. For this study, we extended the dataset by three additional years and introduced a new input variable to assess its compatibility with existing variables. We applied various regression and boosting models to identify top performers and analyse the influence of each variable on efficiency, distinguishing those of greatest significance. Initially, we focused on environmental factors alone, then incorporated internal factors to observe their combined impact on efficiency.

This paper is structured as follows:

1. Introduction to CCPPs and research rationale
2. Review of relevant literature
3. Description of models used
4. Experimental results on datasets and variable selection
5. Conclusion and future research directions

2. Literature Review

In their paper titled "A Data-Driven Approach for On-line Gas Turbine Combustion Monitoring using Classification Models," Allegorico and Mantini [7] employed logistic regression and an artificial neural network to recognize anomaly patterns leading to combustion issues. They also integrated a physics-based algorithm into their study for comparison. Their findings revealed that machine learning models, particularly logistic regression, outperformed their counterparts, demonstrating the potential of these models in detecting combustion anomalies. However, their focus on combustion monitoring alone rather than on broader efficiency metrics and their analysis under specific turbine conditions limits the generalizability of their approach

Jihad and Tahiri [8], in their work "Forecasting the Heating and Cooling Load of Residential Buildings by Using a Machine Learning Algorithm 'Gradient Descent,'" predicted the energy requirements of residential buildings in the Agadir region of Morocco. They achieved remarkable accuracy rates of 98.7% and 97.6% for prediction and test data, respectively, using gradient boosting. This high performance influenced our choice of the algorithm for our research. However, their study focused solely on residential buildings, which differ significantly in efficiency and operational dynamics from industrial

power plants, thereby limiting the direct applicability of their findings to Combined Cycle Power Plants (CCPPs).

Elfaki and Ahmed [9] utilized artificial neural network (ANN)-based regression models with a four-input dataset to predict the electrical output power of a combined cycle power plant. Their research highlighted the stochastic behaviour of the regression model and concluded that increasing the dataset size led to improved predictions and enhanced the reliability of the ANN model. Yet, with only four input variables, their model potentially missed additional factors relevant to more complex energy systems like CCPPs.

Kaya, Tufecki, and Gurgen [10] conducted an experiment using a six-year dataset, incorporating temperature, humidity, pressure, and exhaust vacuum as inputs. They formed both local and global predictive models using various techniques, including conventional multivariate regression, additive regression, k-NN, feedforward ANN, and K-Means clustering. Their findings illustrated that even basic regression tools such as K-NN could forecast net yield with an average relative error of less than 1%, underscoring the potential for improved performance with the incorporation of more advanced tools and comprehensive pre-processing. This finding highlighted the potential for improved performance with advanced tools and thorough pre-processing. However, their study did not explore ensemble methods that could capture complex interactions between variables.

Siddqui et al. [11] estimated power production by a Combined Cycle Power Plant (CCPP) on an hourly basis using machine learning algorithms. They evaluated five models, including K-Nearest Neighbors, Linear Regression, Gradient-Boosted Decision Trees, Artificial Neural Network, and Deep Neural Network, with the Gradient-Boosted Decision Trees yielding the most favorable results. Their work underscores the need to evaluate multiple models to achieve optimal accuracy; however, it concentrated only on power prediction without exploring efficiency or other operational metrics

Similarly, Alketbi et al. [12] employed four machine learning methods– Multiple Linear Regression, K-Nearest Neighbors, Multilayer Perceptron, and Random Forest Regression-on four input variables. Their experiments demonstrated that Random Forest Regression produced the most promising outcomes. While effective, their approach used a limited number of input variables, possibly overlooking additional significant operational factors for power plants.

In reviewing previous studies, we noted that regression methods were commonly used but with limited input variables, often missing critical aspects of CCPP efficiency. This prompted us to expand our experiments to include not only key environmental factors but also a range of internal components. Additionally, we employed advanced machine learning techniques, such as gradient boosting, adaptive boosting, and bagging ensembles, alongside various regression methods to identify the most effective model. By incorporating a broader set of variables and sophisticated techniques, our study aims to offer a more accurate and comprehensive approach to efficiency estimation.

3. Methodology

This paper encompasses nine machine learning models, all employed in the estimation of the Combined Cycle Power Plant's (CCPP) efficiency. The selection of

the top-performing model is the initial step, and subsequently, it is utilized to identify the variables that have the most significant impact on the plant's efficiency.

In this study, we utilize four environmental variables and ten internal variables. In our previous research, we conducted separate experiments on environmental and internal variables, discovering that combining them yielded the most favourable outcomes [6]. For this study, an additional internal variable has been introduced, along with an expanded dataset. The approach entails initially shortlisting the best-performing model from the regression and boosting models. We then proceed to evaluate the estimation results of these two models by systematically removing variables individually and assessing how their absence affects the outcome. Ultimately, this process helps determine the variables of utmost importance and identifies which of the selected regression and boosting models performs best.

The input variables include temperature, humidity, pressure, heatrate, IGV1 (inlet guide vanes), generating power produced, condenser data, cooling tower data, gas turbine 1, gas turbine 2, heat recovery steam generator 1, heat recovery steam generator 2, steam turbine, and IGV2. The resulting variable is the comprehensive thermal efficiency produced.

Our machine learning models include both regression and ensemble methods, as explained by Huang et al [13]. In regression, the focus is on scrutinizing the connections between different variables, ultimately culminating in the development of a mathematical model that evaluates the value of a variable (label) based on its features. On the contrary, ensemble methods employ the strategy of generating and merging multiple models to achieve enhanced results. The models in use are as follows.

3.1 Linear Regression

Linear regression is a type of supervised learning model employed for conducting regression tasks. It is founded on the concept of using the independent variable to make predictions about the dependent variable, as described by Montgomery et al., in 2012 [14]. A basic regression model can be expressed using the following Eq. (3.1):

$$y = \theta_1 + \theta_2 \cdot x, \tag{3.1}$$

In this context, where x represents the input training data, specifically univariate–meaning one input variable or parameter, y is the label, $\theta_1 1$ is the intercept and θ_2 is the coefficient of x. Table 1 shows the parameters we have used for our work.

Table 1. Linear Regression Parameters.

S. no	Parameters	Value
1	fit_intercept	TRUE
2	n_jobs	-1
3	Positive	FALSE
4	Normalize	FALSE

The fit_intercept parameter decides if the model includes an intercept. Setting it to False forces the line through the origin, which can lead to a biased result if the data doesn't go through zero naturally. n_jobs controls how many CPU cores the model uses to run faster, especially with big datasets, but it doesn't affect accuracy. Positive keeps coefficients positive if we expect only positive relationships between variables. Lastly, normalize makes sure features are on a similar scale, which can help make the model more stable and improve how it performs.

3.2 K-Nearest Neighbor Regression

In the same vein, K-Nearest Neighbors (KNN) can be applied to regression tasks, not just classification. It is particularly useful when dealing with situations where non-linear boundaries define classes or values of interest. KNN leverages a feature similarity algorithm to predict the class or value for new data points. These additional data points are classified according to their closeness to the training dataset, as elucidated by Parsian [15]. The distance measurement can be accomplished through three distinct methods.

3.2.1 Euclidean Distance

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}. \quad (3.2)$$

In simple terms, Euclidean distance can be defined as a formula used to assess the distance between two points.

In Eq. (3.2), d represents the Euclidean distance. (x_1, y_1) denotes the coordinates of the first point, and (x_2, y_2) represents the coordinates of the second point.

3.2.2 Manhattan Distance

$$\sum |x_i - y_i|. \quad (3.3)$$

The distance between their real vectors.

3.2.3 Hamming Distance

$$D_H = \sum_{i=1}^{k} |x_i - y_i|. \quad (3.4)$$

Used only for categorical values, the value of D will be zero if (x) and (y) are same. Otherwise, it will be 1. Table 2 contains the parameters of KNN regression we have used. The weights parameter controls how much influence each data point has when making predictions. If set to uniform, all points count the same, but if set to distance, closer points have more influence. n_neighbors determines how many nearby points the model looks at to make a prediction. More neighbors can make predictions smoother but less sensitive to changes. The algorithm decides how the model finds the nearest neighbors, and this can affect how fast it runs, especially with big datasets. Lastly, cv stands for cross-validation, which helps test the model's performance by splitting the data into parts to make sure the model works well with new, unseen data.

Table 2. KNN Regression Parameters.

S. no	Parameters	Value
1	weights	{'unifrom', 'distance'}
2	n_neighbors	range(3, 75)
3	algorithm	auto
4	cv	5

3.3 Random Forest Regression

Random forest regression is another ML method that amalgamates various algorithms to construct a robust model capable of delivering precise predictions. The synergy among these algorithms bolsters the model's predictive capacity, ultimately enhancing its performance, as highlighted by Cutler [16]. The parameters for random forest regression are presented in Table 3. The

Table 3. Random Forest Regression Parameters.

S. no	Parameters	Value
1	max_features	{'auto', 'log2'}
2	n_estimators	700
3	Criterion	'squared error'
4	random_state	5
5	n_jobs	-1

max_features parameter controls how many

features the model looks at when building each tree. Using fewer features can make the model faster but less accurate, while using more can make it more accurate but slower. n_estimators is the number of trees in the forest; more trees generally make the model more accurate but slower to run. Criterion decides how to measure the quality of a split in the trees, with options like ”mse” for mean squared error. random_state is a seed value that ensures the model's results are reproducible, so you get the same outcome each time you run it. Finally, n_jobs controls how many CPU cores the model uses to speed up training, especially useful with large datasets.

3.4 Linear Support Vector Machine

The Linear Support Vector Machine (SVM) is exclusively utilized when dealing with datasets that demonstrate linear separability. Linear separability is a property in which data points can be effectively distinguished and separated using a single straight line, as explained by Steinwart and Christmann [17]. Table 4 displays the parameters associated with the linear SVM. The kernel parameter defines the type of function used to separate the data points (like ”linear,” ”poly,” or ”rbf”). It helps determine how the model works in higher-dimensional spaces. Loss controls how the model handles errors, with ”hinge” being the typical choice for classification tasks. C is a regularization parameter that helps balance fitting the data well and avoiding overfitting; a higher value makes the model more sensitive to the training data. cv stands for cross-validation, used to evaluate the model's performance by splitting the data into parts. Finally, random_state is a seed that ensures the model's results can be reproduced if run multiple times with the same settings.

Table 4. Linear SVM Parameters.

S. no.	Parameters	Value
1	kernel	’linear’
2	loss	’squared hinge’
3	c	[1e0, 1e1, 1e2, 1e3, 1e4]
4	cv	10
5	random_state	None

3.5 Kernel Support Vector Machine

The kernel trick in SVM takes a low dimensional input and then transfers into a higher dimensional space [17]. It can be described as follows:

$$K(\tilde{x}) = 1, \text{ if } \|\tilde{x}\| \leq 1, \qquad (3.5)$$
$$K(\tilde{x}) = 0, \text{ Otherwise.}$$

Table 5 delivers the parameters for kernelized SVM. The kernel parameter defines the type of function used to transform the data into a higher-dimensional space for better separation, with common options like ”linear,” ”poly,” and ”rbf.” Degree is used with the ”poly” kernel to control the degree of the polynomial function (higher values create more complex boundaries). C is a regularization parameter that helps balance fitting the training data well and preventing overfitting; a larger value makes the model more sensitive to the data. cv refers to cross-validation, which tests the model's performance by splitting the data into training and

Table 5. Kernelized SVM Parameters.

S. no	Parameters	Value
1	Kernel	’rbf’
2	Degree	3
3	C	[1e0, 1e1, 1e2, 1e3, 1e4]
4	Cv	10
5	random_state	None

testing sets. Finally, random_state ensures the model's results can be reproduced consistently.

3.6 Bagging ensemble

Bagging and boosting are ensemble methods employed to enhance the predictive accuracy and overall performance of models. Bagging, for instance, entails creating multiple variations of predictors and then aggregating them to construct a unified model for practical use. The parameters for Bagging related to decision trees are detailed in Table 6. The n_estimators parameter defines the number of decision trees in the model; more trees usually improve accuracy but can make the model slower. max_samples determines how many samples each tree will use when building its model. Setting it lower can help avoid overfitting by introducing more variety between trees. base_estimator is the model that will be used to build each tree, typically a decision tree, but it can be any classifier or regressor. cv refers to cross-validation, which splits the data to test how well the model generalizes. Lastly, random_state ensures the results are reproducible each time the model is run.

Table 6. Bagging DT's Parameters.

S. no	Parameters	Value
1	n_estimator	750
2	max_samples	[5, 10]
3	base_estimator	None
4	Cv	10
5	random_state	None

3.7 Adaptive Boost on Decision Tree (Ada Boost DT)

Boosting is yet another ensemble technique in which either different or a single machine learning algorithm is employed multiple times for prediction, with the aim of enhancing the overall model's performance. In the case of adaptive boosting, numerous weak classifiers are combined to create a powerful classifier, as explained by Schapire and Freund [18]. Table 7 outlines the parameters for adaptive boosting. The n_estimators parameter specifies the number of weak learners (usually decision trees) to combine into the final model; more estimators generally lead to better performance but can increase computation time. learning_rate controls how much each tree contributes to the final prediction. A smaller learning rate makes the model more robust but requires more estimators to reach the same level of accuracy. base_estimator is the model used for each weak learner, typically a decision tree, but it can be any classifier or regressor. Finally, random_state ensures the results are reproducible when running the model multiple times.

Table 7. Adaptive Boost DT Parameters.

S. no	Parameters	Value
1	n_estimator	[100, 500]
2	learning_rate	0.7
3	base_estimator	base_dt
4	random_state	5

3.8 Ada Boost on KNN Regression

In this approach, AdaBoost is applied to K-nearest Neighbor regression, leveraging the power of this machine learning algorithm multiple times to enhance the performance of the model [18]. Table 8 showcases AdaBoost KNN's parameters. The n_estimators parameter sets the number of KNN models (or weak learners) to combine into the final model. More estimators usually improve performance but increase computation time. learning_rate controls how much each KNN model contributes

Table 8. AdaBoost KNN Parameters.

S. no	Parameters	Value
1	n_estimator	[100, 500, 1000]
2	learning_rate	0.7
3	base_estimator	base_knn
4	random_state	10

to the final prediction. A lower learning rate requires more estimators to achieve the same accuracy but can make the model more stable. base_estimator specifies the base model used for each weak learner, which in this case would be KNN (but it can be other models too). Lastly, random_state ensures that the results are reproducible each time you run the model.

3.9 Gradient boosting

Gradient boosting is another algorithm that operates on the principle that the prime feasible model, in conjunction with preceding models, will reduce the prediction errors, as elucidated by [18]. The parameters utilized in our work for the gradient boosting model are presented in Table 9. The n_estimators parameter defines

Table 9. Gradient Boosting Parameters.

S. no	Parameters	Value
1	n_estimator	[100, 500, 1000]
2	learning_rate	1.1
3	n_jobs	-1
4	random_state	10
5	min_samples_split	[3,4,5]

how many decision trees the model will use; more trees generally improve accuracy but can increase computation time. learning_rate controls how much each tree's contribution is weighted. A smaller learning rate makes the model more stable, but you might need more trees to reach the same performance. n_jobs controls how many CPU cores to use for training, helping speed up the process, especially for large datasets. random_state ensures that the model's results are reproducible each time it's run. Lastly, min_samples_split determines the minimum number of samples required to split an internal node; increasing this value can prevent overfitting by making the model more conservative.

The inputs utilized in our study include the factors that influence the efficiency of the power plant. The attributes consist of Temperature, Humidity, Pressure, Heat rate, MegaWatt produced, Inlet Guide Vanes 1, Inlet Guide Vanes 2, Condenser, Cooling tower, Gas turbine 1, Gas turbine 2, Heat Recovery Steam Generator 1, Heat Recovery Steam Generator 2, and Steam Turbine. The output variable is the Net Efficiency, which varies based on the provided variables.

All model experiments were conducted using Google Colaboratory with GPU settings and a RAM of 12GB, with processing times ranging from 30 minutes to 1 hour for each of the model.

4. Experimentation

This segment focuses on the datasets employed in this research. To enhance understanding, we have visualized our data, and the performance has been evaluated using R-squared metrics. The execution of individual model is subsequently assessed collectively to determine which one superior.

4.1 Dataset

The fifteen variables are individually elucidated below. The dataset has been collected from North Bangkok Power Plant, where variables include: Temperature, Humidity, Pressure, Heat rate, Mega Watt, Inlet Guide Vanes 1, Inlet Guide Vanes 2, Condenser, Cooling tower, Gas Turbine 1, Gas turbine 2, Heat Recovery Steam Gen-

erator 1, Heat Recovery Steam Generator 2, Steam Turbine, and Net Efficiency. These variables have been collected over the past eight years at an hourly interval between data samples. Certain variables, such as Inlet Guide Vanes, Gas Turbine, and Heat Recovery Steam Generator, have two values each because the power plant is equipped with two sets of these components. Consequently, we obtained two separate readings for each of these variables.

Temperature, which is a physical quantity that numerically represents the degree of hotness or coldness, fluctuates within the range of 24 °C to 35 °C in Thailand. The probability density is depicted in Fig. 2. Humidity is a measure of the per-

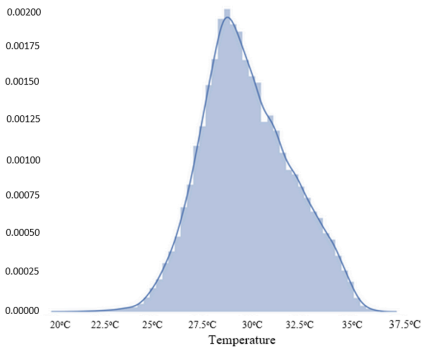

Fig. 2. Temperature (Celsius).

centage of water vapor in the atmosphere and typically falls within the range of 30% to 90%, as illustrated in Fig. 3. Air pressure, as shown in Fig. 4, experiences variations between 995 and 1025 hPa. These fluctuations are primarily influenced by the power plant's location, which is situated at sea level. Fig. 5 illustrates a steam turbine, a device that operates by utilizing a heat source to raise the temperature of water to extremely high levels, resulting in its conversion into steam, as described by Hegde [18]. The running efficiency of this steam turbine is depicted, and it is approximately

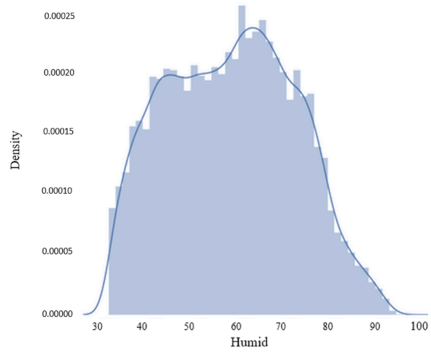

Fig. 3. Humidity.

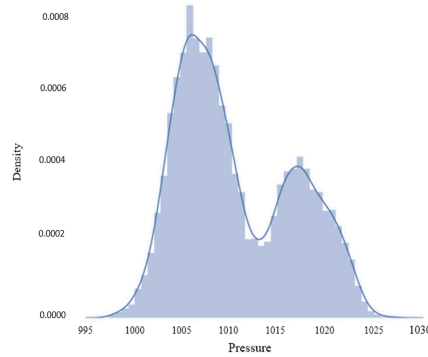

Fig. 4. Pressure (hPa).

around 40%. The main function of the In-

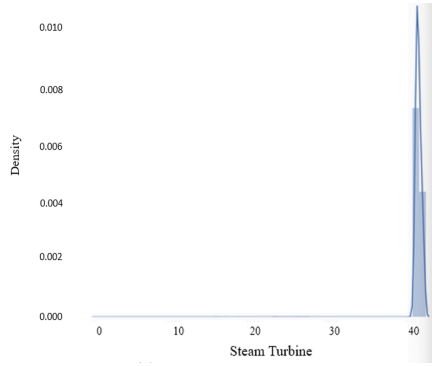

Fig. 5. Steam Turbine.

let Guide Vanes (IGV) is to regulate the airflow and pressure that enters the initial stage of compression in a centrifugal compressor. The IGV values are presented in Fig. 6 below. The Heat Recovery Steam

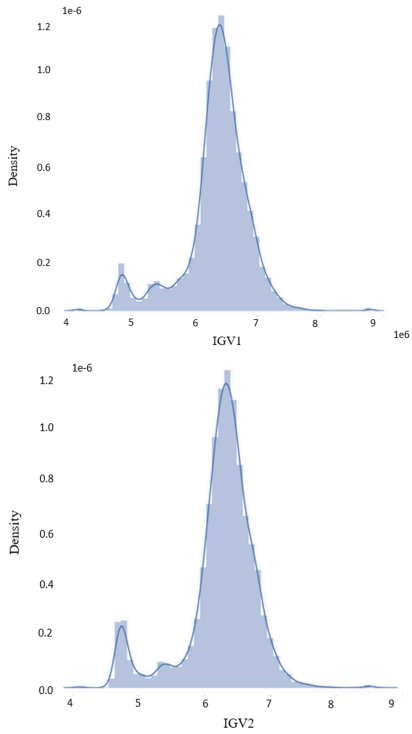

Fig. 6. Inlet Guide Vane 1 and Inlet Guide Vane 2.

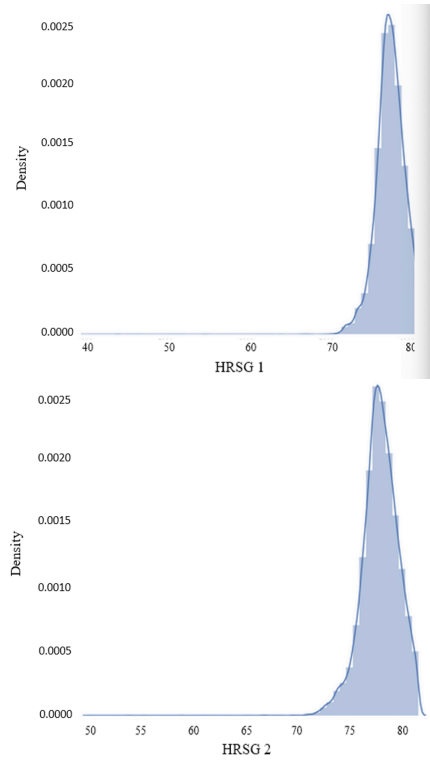

Fig. 7. Heat Recovery Steam Generator 1 and Heat Recovery Steam Generator 2.

Generator (HRSG) serves as an energy recovery heat exchanger designed to recover heat from hot gas steam. In our study, two HRSG blocks were employed, as illustrated in Fig. 7. Fig. 8 displays the condenser, also referred to as a phase transition unit, which functions as a heat exchanger converting steam from its gaseous state to its liquid state, as explained by Hegde [19]. The data in Fig. 8 ranges between 80% and 100%. Fig. 9 presents data related to the generated MegaWatts in the power plant. The power plant's output generation typically ranges between 580-680 MW when it is operating at full capacity and decreases to 400-500 MW when it operates at partial capacity. Net efficiency is a measure of how effectively a power plant converts energy input into electricity output, considering losses and waste in the process. Fig.

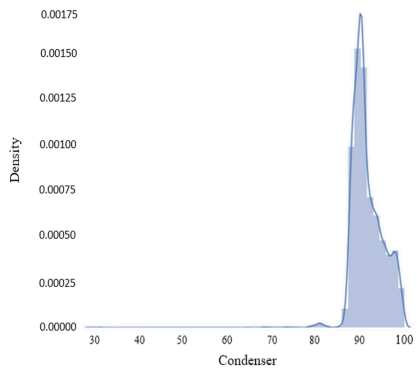

Fig. 8. Condenser.

10 illustrates the net efficiency, with values falling within the range of 46% to 55%. A Cooling Tower is utilized to dissipate heat by spraying water down through the tower, with the primary objective of maximizing the evaporation of water. The data

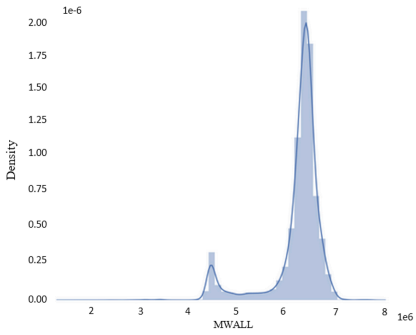

Fig. 9. Mega Watt produced (MW).

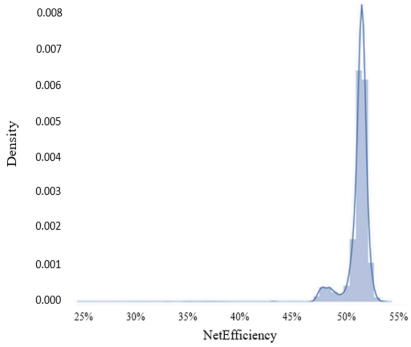

Fig. 10. Net Efficiency (%).

related to the cooling tower is depicted in Fig. 11. The gas turbine consolidates the

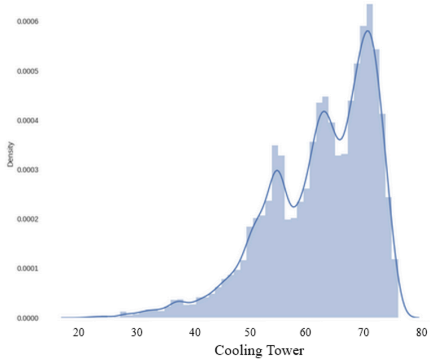

Fig. 11. Cooling Tower.

air and blends it together alongside fuel; it is then heated to immense temperatures for power generation. Data from the two turbines are presented in Fig. 12. The heat rate within a combined cycle power plant

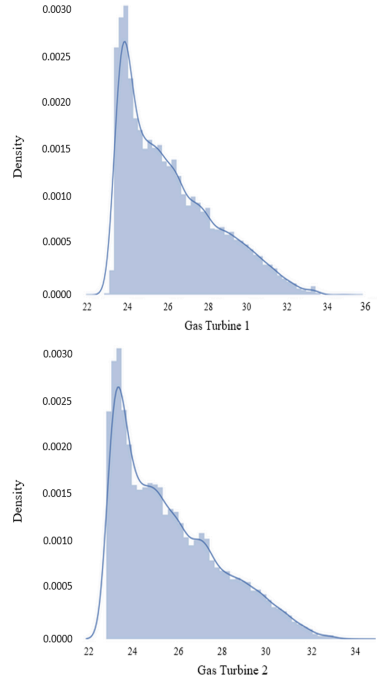

Fig. 12. Gas turbine 1 and 2.

quantifies the energy input, usually measured in British Thermal Units (BTUs), essential for generating a single kilowatt-hour (kWh) of electricity. This metric serves as a fundamental gauge of the power plant's effectiveness in converting fuel into electrical energy. A reduced heat rate signifies a more efficient power plant, signifying that it requires less energy to generate a specific quantity of electricity. The heatrate data is displayed in Fig. 13 below.

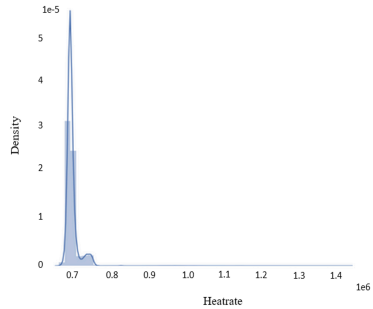

Fig. 13. Heatrate.

4.2 Training dataset

The dataset is divided into a ratio of 90:10, with 90% of the data designated for training and the residual 10% for testing. Due to the extensive nature of the data, min-max scaling is implemented to standardize the values within the 0-1 range. For this research, we have opted to use the R-squared score as the accuracy metric. This metric is utilized to evaluate the model's performance on the provided data. It is defined as the ratio of the Sum of Squares Regression (SSR) to the Sum of Squares Total (SST). A higher R-squared value indicates better results, as explained by Kane [20]. The R-squared is expressed as follows:

$$\text{R-Squared} = \frac{\text{Sum of Squares Regression}}{\text{Sum of squares Total}}.$$

4.3 Model evaluation result

The R-squared evaluation results of the models employed are presented in Table 9, and in Fig. 14. From the table it is evident that, in the context of a regression model, Random Forest Regression exhibited the highest performance, achieving an accuracy of 99.10% on the training dataset and 99.42% on the testing dataset. Meanwhile, for the ensemble method, Gradient Boosting emerged as the top-performing model, attaining 99.83% accuracy on the training dataset and 97.85% on the testing dataset. Following the outcomes described above,

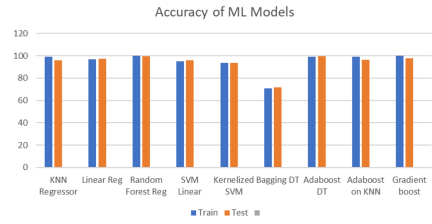

Fig. 14. Accuracy of different ML models.

we opted to proceed with Random Forest Regression and Gradient Boosting for additional experimentation. Our approach involved systematically eliminating variables one by one and assessing the impact on accuracy.

Table 10. Training and testing accuracy on different models.

Model	Train	Test
KNN Regressor	99	96.0597
Linear Reg	96.6942	97.1136
Random Forest Reg	99.9137	99.4238
SVM Linear	94.9827	95.7191
Kernelized SVM	93.5148	93.783
Bagging DT	70.7964	71.866
Adaboost DT	99	99.409
Adaboost on KNN	99.1012	96.49
Gradient boost	99.8306	97.8523

Subsequently, we identified the variables that had both positive and negative effects on accuracy. This process led us to determine the most favourable combination of variables for predicting efficiency. In Table 10, it is evident that the absence of temperature, humidity, pressure, HRSG 2, Gas Turbine 1, and Gas Turbine 2 led to an increase in accuracy. Conversely, when the other parameters were removed, accuracy declined. Subsequently, we trained the model using the remaining 8 variables, namely Heatrate, IGV 1 & 2, Condenser, Cooling Tower, HRSG 1, MegaWatt produced, and Steam Turbine. The results of this training are presented in Table 11 below.

The results are encouraging with the selected 8 parameters, but the highest accuracy was attained when only the pressure data was excluded. As demonstrated in Table 10, upon removing pressure from the model, Random Forest Regression achieved an accuracy of 99.97% on the training data and 99.93% on the testing data. Similarly, Gradient Boosting delivered an

Table 11. Accuracies after parameter removal.

Variables	**Random Forest Regression**		Gradient Boosting	
	Train	Test	Train	Test
All Variables included	99.91%	99.42%	99.83%	97.85%
Temperature absent	99.99%	99.42%	99.86%	98.05%
Pressure absent	**99.97%**	**99.93%**	**99.96%**	**99.93%**
Humid absent	99.92%	99.45%	99.86%	98.13%
Heatrate absent	99.91%	99.42%	99.76%	98.54%
IGV1 absent	99.79%	98.42%	99.25%	94.48%
IGV2 absent	99.49%	96.01%	99.31%	90.50%
HRSG 1 absent	99.91%	99.41%	99.86%	98.55%
HRSG 2 absent	99.91%	99.45%	99.85%	98.36%
Condenser absent	99.85%	98.29%	99.85%	98.29%
Cooling Tower absent	99.91%	99.43%	99.84%	98.62%
Steam Turbine absent	99.92%	99.20%	99.84%	98.21%
Gas Turbine 1 absent	99.92%	99.44%	99.86%	98.54%
Gas Turbine 2 absent	99.92%	99.43%	99.86%	97.91%
Mega Watt Produced absent	99.80%	98.97%	99.68%	97.72%

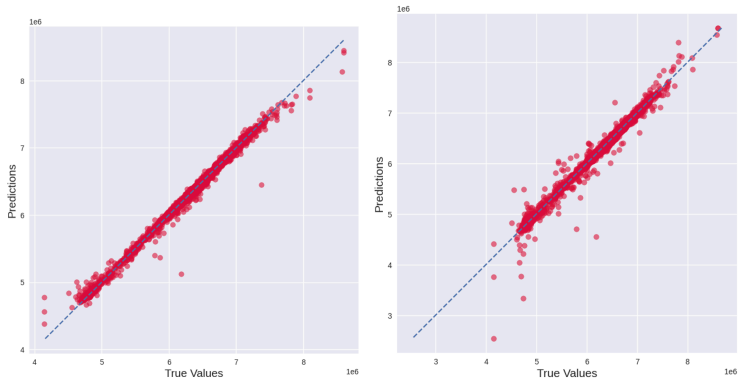

Fig. 15. Actual and Predicted data for Random Forest Regression and Gradient Boosting.

Table 12. Accuracy with 8 parameters.

Model	Train	Test
Random Forest Regression	99.92%	99.48%
Gradient Boosting	99.86%	99.14%

accuracy of 99.96% for the training data and 99.93% for the testing data. The accuracy increased in both models, suggesting that the combination of both environmental and internal variables is essential for a positive impact, and this combination performs optimally when the pressure data is omitted. Hence, for our final model, we have chosen to exclude the pressure variable and proceed with the remaining 13 variables.

Fig. 15 illustrates that our results have provided valuable insights into how our datasets have interacted with the models and how well they have executed predictions. The experiments have showcased the proximity between the actual and predicted data.

Overall, the results have demonstrated the effectiveness of incorporating both internal and environmental variables in

our models. This highlights the importance of considering various factors when making predictions and decisions, whether it is in the field of science, business, or any other field that relies on data analysis.

5. Conclusion

In our research, we employed a comprehensive approach by using nine different machine learning models to estimate the efficiency of a combined cycle power plant. These models included Linear Regression, KNN Regression, Random Forest Regression, Linear SVM, Kernelized SVM, Bagging Ensemble, AdaBoost on DT, AdaBoost on KNN Regression, and Gradient Boosting. Our findings showed that several models achieved over 99% accuracy, advancing current methods of efficiency estimation for combined cycle power plants. We also provided the specific hyperparameters for each model used.

To enhance our results, we expanded our dataset by three years, bringing the total to eight years, and added the ”heatrate” parameter. After thorough testing, we determined that Random Forest Regression and Gradient Boosting were the most stable and effective models. Each model was trained on 90% of the data and tested on the remaining portion to ensure accuracy and prevent overfitting.

In the second part of our research, we analysed 14 variables to understand their impact on power plant efficiency. These variables included temperature, humidity, pressure, heatrate, megawatt, inlet guide vanes, condenser, cooling tower, turbines, and heat recovery steam generators. By including a broad range of variables, we ensured a thorough examination of all factors that could affect plant efficiency. To identify the most influential variables, we conducted a systematic elimination process, removing each variable one by one and observing the impact on model accuracy. We found that removing pressure data improved the accuracy of both models, leading to the optimal parameter combination.

This research offers several key contributions: a comprehensive analysis of power plant efficiency by considering a wide range of variables, the identification of key variables through systematic elimination for targeted optimization, enhanced model accuracy by focusing on the most impactful factors, and practical implications that help industry professionals improve monitoring, control strategies, efficiency, and reduce costs and environmental impact.

In conclusion, our findings indicate that Random Forest Regression and Gradient Boosting, with the exclusion of pressure data, yield the most favorable outcome with the Random Forest Regression model coming up overall at the top of the list. For our future work, we intend to employ this model as a substitute for the direct computation of efficiency, streamlining the process that typically necessitates knowledge of multiple variables and equations.

Acknowledgements

We would like to thank the Department of Electrical and Computer Engineering, Faculty of Engineering (Thammasat University), Electricity Generating Authority of Thailand (EGAT) and North Bangkok Power Plant for their valuable support and resources.

References

[1] Araner (n.d.). What makes combined cycle power plants so effective? Retrieved Febuary 28, 2022. https://www.araner.com/blog/combined-cycle-power-plants.

[2] Breeze, P. (2020). Power Generation Technologies. Newnes, an imprint of Elsevier.

[3] Donev JMKC, et al. (2020). Combined cycle gas plant. Energy Education. Retrieved November 1, 2022. https://energyeducation.ca/encyclopedia/-Combined_cycle_gas_plant.

[4] Ramireddy, V. (2012). An overview of combined cycle power plant. Electrical Engineering Portal.

[5] Rapier, R. (2017). The Load Following Power Plant: The Peaker. Transform. Retrieved November 1, 2022. https://www.ge.com/power/-transform/article.transform.articles.-2017.jun.load-following-power-plant

[6] Kaewprapha, P, Prempaneerach, P, Singh, V., Tinikul, T, Intarangsi, N. Machine Learning Approaches for Estimating the Efficiency of Combined Cycle Power Plant. 2022 International Electrical Engineering Congress (iEECON), Khon Kaen, Thailand, 2022, pp. 1-4.

[7] Allegorico, C., Mantini, V. (2014). A data-Driven Approach for on-line Gas Turbine Combustion Monitoring using Classification Models. European Conference of The Prognostics and Health Management Society.

[8] Jihad, A. S., Tahiri, M. (2018). Forecasting the heating cooling load of residential buildings by using a learning algorithm ”gradient descent”, Morocco. Case Studies in Thermal Engineering, 12, 85-93.

[9] Elfaki, E. A., Ahmed, H. A. (2018). Prediction of Electrical Output Power of Combined Cycle Power Plant Using Regression ANN Model. Journal of Power and Energy Engineering, 6, 17-38.

[10] Kaya, H., Tufecki, P., F. S. (2012). Local and Global Learning Methods for Predicting Power of a Combined Gas & Steam Turbine. International Conference on Emerging Trends in Computer and Electronics Engineering, 13-8.

[11] Siddiqui, R., Anwar, H., Ullah, F., Ullah, R., Rehman, M. A., Jan, F., Zaman, F. (2021). Power Prediction of Combined Cycle Power Plant (CCPP) Using Machine Learning Algorithm-Based Paradigm. Wireless Communications and Mobile Computing.

[12] Alketbi, S., Nassif, A. B., Eddin, M. A., Shahin, I., Elnagar, A. (2020). Predicting the power of a combined cycle power plant using machine learning methods. International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI).

[13] Huang, F., Xie, G., Xiao, R. (2009). Research on Ensemble Learning. 2009 International Conference on Artificial Intelligence and Computational Intelligence, 249-52.

[14] Montgomery, D. C., Peck, E. A., Vining, G. G. 2012. Linear Regression Analysis (5th ed.). Wiley.

[15] Parsian, M. (2015). Data Algorithms. O’Reilly Media, Inc.

[16] Cutler, A. (2010). Random Forest Regression and Classification. Ovronnaz/Utah State University.

[17] Steinwart, I., Christmann, A. (2008). Support Vector Machines. Springer New York.

[18] Schapire, R. E., Freund, Y. (2014). Boosting: Foundations and Algorithms (Adaptive Computation and Machine Learning). The MIT Press.

[19] Hegde, R. K. (2015). Pearson Education India.

[20] Kane, F. (2017). Hands-On Data Science and Python Machine Learning. Packt Publishing.