ANSWER SHEET DAC 2024

PRELIMINARY ROUND

Data Analysis Competition 2024

DATA ANALYSIS COMPETITION 2024

TEAM NAME: VDE

TEAM ID: 0121

CHAPTER I: Introduction

1.1 Background of the Problem

As an abundant renewable energy source, solar energy has attracted increasing attention in recent years. Increasing awareness of the importance of clean and sustainable energy is encouraging the development of increasingly efficient solar panel technology. The Indonesian government itself is targeting the construction of rooftop Solar Power Plants (PLTS) to reach 2,145 Megawatts throughout 2021 – 2030.

In developing solar energy, there are several challenges that the government must face. One of the challenges in utilizing solar energy is fluctuations in output which are influenced by various factors such as weather, time of day and environmental conditions. Fluctuations in output make it difficult to predict energy production accurately. This makes energy supply planning more complex.

Uncontrolled output fluctuations cause various problems such as excess or shortage of electricity supply, and inaccuracy in allocating funds for developing solar energy sources. Therefore, the ability to accurately predict solar panel output is critical. Thus, this research aims to develop a prediction model that can estimate future solar panel output accurately.

1.2 Problem Formulation

Based on the background of the problem above, the problem formulation in this research is:

1. What are the characteristics of the given solar panel output data?
2. What prediction model is most appropriate for predicting solar panel output based on the available data?
3. How accurate is the resulting prediction model in predicting future solar panel output?

1.3 Research Objectives

The aims of this research are:

1. Analyze the characteristics of solar panel output data to understand existing patterns and trends.
2. Develop a prediction model that can provide accurate estimates of future solar panel output.
3. Evaluate the performance of the resulting prediction model.

1.4 Benefits of Research

It is hoped that the results of this research can contribute to the development of more efficient renewable energy systems, better decision making in energy management, or support further research in the field of solar energy.

CHAPTER II: Theoretical Framework

2.1 Introduction

The advent of machine learning and statistical modeling has revolutionized various domains, providing powerful tools for predictive analytics. In this chapter, we explore a suite of models that are pivotal in contemporary research, including neural networks, regression techniques, and ensemble methods. The selection of these models is driven by their widespread application and robust performance in handling complex datasets.

2.2 Multi-Layer Perceptron (MLP)

The Multi-Layer Perceptron (MLP) is a fundamental architecture in artificial neural networks (ANNs) [1]. It consists of an input layer, multiple hidden layers, and an output layer. Each neuron in a layer is connected to every neuron in the subsequent layer, allowing the network to learn complex representations through backpropagation. The output □ of an MLP for input □ can be expressed as:

$$y = f(W^{(2)} \cdot f(W^{(1)} \cdot x + b^{(1)}) + b^{(2)}) \tag{2.1}$$

where □$^{(1)}$ and □$^{(2)}$ are weight matrices, □$^{(1)}$ and □$^{(2)}$ are bias vectors, and □$(\cdot)$ represents the activation function [2].

2.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber (1997), address the limitations of traditional recurrent neural networks (RNNs) by effectively capturing long-term dependencies in sequential data [3]. The following equations govern the LSTM cell:

$$\begin{aligned} f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \widetilde{C_t} &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \widetilde{C_t} \\ \sigma_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= \sigma_t \cdot \tanh(C_t) \end{aligned} \tag{2.2}$$

where f_t,i_t, and σ_t represent the forget input, and output gates, respectively, C_t is the cell state, h_t is the hidden state, and σ is the sigmoid activation function [4].

2.4 Ridge Regression

Ridge Regression is a technique designed to address multicollinearity in linear regression models by introducing a regularization parameter (λ) to the loss function [5]. The Ridge Regression cost function is expressed as:

$$\min_{\beta} \left(\sum_{i=1}^{n} (y_i - \boldsymbol{X}_i \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right) \tag{2.3}$$

where y_i are the observed values, $\boldsymbol{X}_i$ are the predictors, β_j are the coefficients, and λ is the regularization parameter [6].

2.5 LASSO Regression

The LASSO (Least Absolute Shrinkage and Selection Operator) regression employs an L1 norm penalty, which leads to sparse solutions [7]. The LASSO cost function is given by:

$$\min_{\beta}\left(\sum_{i=1}^{n}(y_i - \boldsymbol{X}_i\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right) \tag{2.4}$$

where $|\beta_j|$ denotes the absolute value of the coefficient β_j, and λ controls the strength of the regularization [8].

2.6 Elastic-Net Regression

ElasticNet Regression combines the regularization properties of both Ridge and LASSO by incorporating both L1 and L2 penalties in its cost function [10]. The ElasticNet cost function is expressed as:

$$\min_{\beta}\left(\sum_{i=1}^{n}(y_i - \boldsymbol{X}_i\beta)^2 + \lambda_1\sum_{j=1}^{p}|\beta_j| + \lambda_2\sum_{j=1}^{p}\beta_j^2\right) \tag{2.5}$$

where λ_1 and λ_2 are the regularization parameters controlling the L1 and L2 penalties, respectively [11].

2.7 XGBoost

XGBoost, short for Extreme Gradient Boosting, is an efficient and scalable implementation of gradient boosting that has gained widespread popularity in machine learning competitions [13]. The objective function of XGBoost includes both a loss function and a regularization term:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K}\Omega(f_k) \tag{2.6}$$

where l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i, and Ω penalizes the complexity of the model f_k [14].

2.8 LightGBM

LightGBM, or Light Gradient Boosting Machine, is an advanced implementation of gradient boosting designed for speed and efficiency [16]. The splitting strategy used by LightGBM can be expressed by the following gain calculation:

$$Gain = \frac{1}{2}\left(\frac{G^2L}{H_L + \lambda} + \frac{G^2R}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right) - \gamma \tag{2.7}$$

where G_L and G_R are the sum of gradients for the left and right splits, H_L and H_R are the sum of Hessians, λ is the regularization parameter, and γ is the complexity cost [17].

2.9 Gradient Boosting

Gradient Boosting is a machine learning technique that builds models sequentially by correcting the errors of previous models [19]. The prediction function for gradient boosting can be expressed as:

$$F_m(x) = F_{m-1}(x) + v\sum_{i=1}^{n}\gamma_i h_i(x) \quad (2.8)$$

where $F_m(x)$ is the prediction of the m-th model, v is the learning rate, γ_i is the multiplier for the i-th residual, and $h_i(x)$ is the base learner (typically a decision tree) [20].

2.10 Random Forest Regression

Random Forest is an ensemble learning method that builds multiple decision trees and merges their predictions to improve accuracy and robustness [21]. The prediction $\hat{y}$ for a Random Forest can be expressed as:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} h_t(x) \quad (2.9)$$

where T is the total number of trees, and $h_t(x)$ is the prediction of the t-th tree [22].

2.11 Support Vector Regression

Support Vector Regression (SVR) is an extension of the Support Vector Machine (SVM) algorithm, adapted for regression tasks [24]. The SVR objective function is defined as:

$$\min_{\boldsymbol{w},b}\frac{1}{2}\left|\left|\boldsymbol{w}\right|\right|^2 + C\sum_{i=1}^{n}\max(0, |y_i - (\boldsymbol{w}\cdot x_i + b)| - \epsilon) \quad (2.10)$$

where $\boldsymbol{w}$ is the weight vector, b is the bias term, C is the regularization parameter, and ϵ defines the margin of tolerance [25].

2.12 Stacked Model

Stacked models, or stacking, is an ensemble learning technique that combines the predictions of multiple models to improve overall performance [27]. The general form of a stacked model prediction can be expressed as:

$$\hat{y} = g(h_1(x), h_2(x), \ldots, h_M(x)) \quad (2.11)$$

where $h_m(x)$ represents the prediction of the m-th base model, and $g(\cdot)$ is the meta-learner that combines these predictions [28].

CHAPTER III: Analytical Steps

For the precision and reliable prediction, we perform several step for the analysis :

1. Data preprocessing
2. Exploratory Data Analysis (EDA)
3. Design of Experiment
4. Modeling
5. Evaluation

CHAPTER IV: Analysis of Results

4.1 Overview of Analysis of Results

This analysis focuses on predicting solar power development using a dataset that includes variables such as solar radiation, temperature, humidity, and energy output from solar panels. The main goal of this analysis is to increase prediction accuracy by combining Multilayer Perceptron (MLP) techniques residual forecasting and stack modeling.

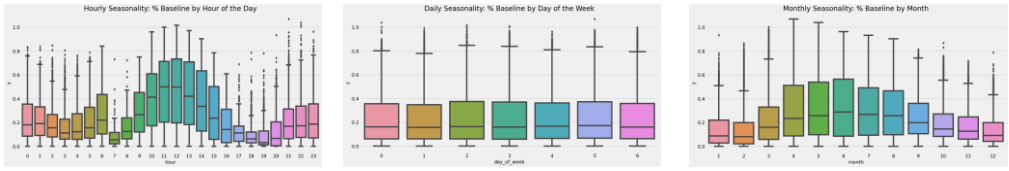
4.2 Data Preprocessing, Exploratory Data Analysis, and Design of Experiment

4.2.1 Data Preprocessing

Firstly, we perform data preprocessing to ensure that the data being used in the model are clean and reliable for the model. We extend the %Baseline so it has complete 24 hours in a day, and then for the solar irradiance dataset we join them and then filling the missing values with mean based on its cloud type. From here, we fill in the missing value because of the %Baseline extension with the MLP model of all variables that has ≥ 0.1 correlation relative to the %Baseline.

4.2.2 Exploratory Data Analysis (EDA)

Next up, we perform the EDA to know the big picture of the data characteristics especially the seasonal patterns and data distribution.



(a) (b) (c)

Figure 4.1. Boxplot Visualization for Seasonal Patterns (a) hourly, (b) daily, (c) monthly

The analysis of the data reveals distinct hourly and monthly seasonality, with less pronounced daily patterns. The hourly seasonality, shown in the left boxplot, highlights significant fluctuations, with activity peaking between 10:00 and 15:00 hours and diminishing during early morning and late evening. This suggests that the time of day plays a crucial role in the data's behavior, necessitating its consideration in any predictive models.

The middle boxplot shows that daily seasonality is minimal, as the data remains consistent across all days of the week. This indicates that the day of the week has little influence on the data, making it less critical for modeling purposes.

In contrast, the right boxplot indicates clear monthly seasonality, with increased values from March to July, followed by a decline towards the end of the year. This pattern underscores the importance of accounting for the time of year in analyses.

In summary, the data exhibits strong hourly and monthly seasonality, while daily patterns are relatively stable. These temporal patterns should be incorporated into any analytical or predictive modeling efforts to enhance accuracy and

relevance. Next, we will do the timeseries decomposition to reveal some hidden pattern in the data.

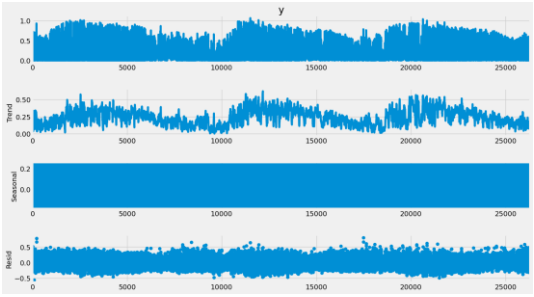


Figure 4.2. Timeseries Decomposition of Target Variable

The time series decomposition indicates that time plays a significant role in shaping the data, primarily through the observable trend component. The trend shows clear periods of increase and decrease, demonstrating that the data evolves meaningfully over time, making time a crucial factor in understanding the underlying patterns. However, the seasonal component appears minimal or constant, suggesting that while time contributes significantly, this contribution is more related to long-term shifts rather than cyclical or seasonal patterns. The residual component, with its relatively consistent unexplained variance, further emphasizes that while time captures a significant portion of the data's structure, there remains some variability not accounted for by trend or seasonality. Thus, time is a significant factor, mainly due to its influence on trends rather than seasonal effects.

4.1.3 Design of Experiment

From the EDA that we perform, we have hypothesis that time is contributing into data pattern and seasonality. Here, we will do two model scenarios as shown below.

- Scenario 1 : Feature and Time-Based Model
- Scenario 2 : Feature-Based Stacked Model with Different Meta-Learner

Although we only have two scenarios, we will have 4 models because in scenario 2, there will be three different meta-learners (XGBoost, LightGBM, and MLP).

4.2 MLP Model Performance and Residual Forecasting

In the initial stage, the MLP model is used to predict the energy output of solar panels based on input meteorological variables. The prediction results from the MLP are then used to calculate the residual, namely the difference between the prediction and the actual value. These residuals are then analyzed further with a forecasting model to identify patterns that may not have been captured by the initial MLP model.

- **MLP Performance** : The MLP model shows quite good results with a Root Mean Square Error (RMSE) of 0.0111. However, there are some complex non-linear patterns that are not fully captured by this model, as shown by the residual analysis.

Figure 4.3. MLP Model Result

- **Forecasting Residuals** : After applying the forecasting model to the residuals, RMSE decreased to 0.0106, indicating that this model was successful in capturing patterns ignored by MLP.

Figure 4.4. MLP-LSTM Model Result

4.3 Stack Modeling Results

To further improve performance, we tried to create a modeling stack by combining **XGBoost, LightGBM, Gradient Boosting, Support Vector Regression, Multi-Layer Perceptron, Random Forest, ElasticNet, Lasso, and Ridge**. Each model makes a different contribution to predicting energy output, and stack modeling allows combining the strengths of each model.

- **Stack Model Evaluation** : After merging, the stack model succeeded in reducing RMSE to 0.0071, which is the best result compared to individual models. This model shows strong generalization ability and is more resistant to data variations.

(a) (b) (c)

Figure 4.5. Stack Model Results (a) XGB Meta (b) ML Meta (c) LightGBM Meta

4.4 Model Evaluation and Validation

The stack model is then validated using a separate test dataset. The stack model shows consistent performance with lower RMSE than the individual models. These results indicate that stack modeling is effective in capturing various aspects of the data and providing more accurate predictions, although there is no time related variable in the stacked model.

4.5 Implications of Analysis Results

The results of this analysis can be used to optimize solar panel operations, including maintenance scheduling and more accurate predictions of daily energy output. The stack modeling technique used can also be adapted for similar applications in other renewable energy contexts.

CHAPTER V: Conclusion and Recommendation

5.1 Conclusion

In this analysis, **the Multilayer Perceptron (MLP) approach** with **residual forecasting** and **stack modeling techniques** has been successfully applied to predict the energy output of solar panels. The analysis results show that:

- **The MLP** provides a strong basis for predictions, but there are complex patterns that are not fully captured, as seen from the residual analysis.
- By applying the forecasting model to the residuals, the model performance improved significantly, demonstrating the importance of capturing patterns that were not detected by the initial model.
- **Stack modeling** successfully combines the power of several predictive models, including **XGBoost, LightGBM, Gradient Boosting, Support Vector Regression, Multi-Layer Perceptron, Random Forest, ElasticNet, Lasso, and Ridge**. which overall reduces the prediction error to 0.0071. This confirms the effectiveness of the ensemble approach in dealing with complex and diverse data.

Overall, the model built is able to provide more accurate and reliable predictions, which can be used for better decision making in solar energy management.

5.2 Recommendations

Based on the main findings of this analysis, several recommendations that can be implemented are:

1. **Solar Panel Operational Optimization** :
   The developed prediction model can be used to plan and optimize solar panel operations, including maintenance scheduling and production adjustments based on predicted energy output.
2. **Further Model Development** :
   It is recommended to continue developing and refining the model by taking into account more weather and environmental variables that may influence energy output, as well as considering more sophisticated data processing techniques.
3. **Investment in Real-time Monitoring Systems** :
   Implementing a real-time monitoring system that is integrated with a prediction model will help monitor solar panel performance more efficiently and provide a quick response to changing conditions.
4. **Advanced Research** :
   It is recommended to carry out further research that focuses on the influence of external factors such as long-term climate change on solar power system performance, as well as exploring new methods in stack modeling.

5.3 Implications for Stakeholders

The results and recommendations from this analysis provide clear guidance for stakeholders in the renewable energy sector to improve efficiency and accuracy in planning and operating solar power systems. Implementing these recommendations can help achieve more stable and sustainable energy production targets.

REFERENCES

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533-536, Oct. 1986.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[4] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, BC, Canada, 2013, pp. 6645-6649.

[5] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55-67, Feb. 1970.

[6] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2009.

[7] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B (Methodol.)*, vol. 58, no. 1, pp. 267-288, Jan. 1996.

[8] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc. B (Stat. Methodol.)*, vol. 67, no. 2, pp. 301-320, Apr. 2005.

[9] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Stat. Softw.*, vol. 33, no. 1, pp. 1-22, Feb. 2010.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, San Francisco, CA, USA, 2016, pp. 785-794.

[11] Y. Zhang and X. Lu, "XGBoost: A powerful and scalable tool for data science," in *Mach. Learn. Data Mining Pattern Recognit.*, Cham, Switzerland: Springer, 2019, pp. 345-362.

[12] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, vol. 30, pp. 3146-3154.

[13] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5-32, Oct. 2001.

[14] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189-1232, Oct. 2001.

[15] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[16] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241-259, 1992.

[17] L. Breiman, "Stacked regressions," *Mach. Learn.*, vol. 24, no. 1, pp. 49-64, Jul. 1996.

[18] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Adv. Neural Inf. Process. Syst.*, Denver, CO, USA, 1997, pp. 155-161.

[19] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995.

[20] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123-140, Aug. 1996.

[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2001.

[22] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, vol. 30, pp. 3146-3154.

[23] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Stat. Softw.*, vol. 33, no. 1, pp. 1-22, Feb. 2010.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[25] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, San Francisco, CA, USA, 2016, pp. 785-794.

[26] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55-67, Feb. 1970.