Promoting Wind Energy by Robust Wind Speed Forecasting Using Machine Learning Algorithms Optimization

Aminuddin
National Research and Innovation Agency (BRIN)

Nurry Widya Hesty
National Research and Innovation Agency (BRIN)

Nina Konitat Supriatna
National Research and Innovation Agency (BRIN)

Akhmad, Kholid
National Research and Innovation Agency (BRIN)

他

https://doi.org/10.5109/7172293

出版情報：Evergreen. 11 (1), pp.354-370, 2024-03. 九州大学グリーンテクノロジー研究教育センター
バージョン：

Promoting Wind Energy by Robust Wind Speed Forecasting Using Machine Learning Algorithms Optimization

Aminuddin[1,2*], Nurry Widya Hesty[1*], Nina Konitat Supriatna[1,3], Kholid Akhmad[1], Arief Heru Kuncoro[1], Vetri Nurliyanti[1], Mugia Bayu Rahardja[1], Sumarsono Sudarto[2], Wiwid Mulyadi[4], Primaldi Anugrah Utama[3,4]

[1]National Research and Innovation Agency (BRIN), Indonesia
[2]Magister of Industrial Engineering, Mercu Buana University, Indonesia.
[3]Department of Mechanical Engineering, Universitas Indonesia, Indonesia
[4]Ministry of Energy and Mineral Resources, Indonesia

*Correspondence: amin010@brin.go.id (Aminuddin),
nurr010@brin.go.id (Nurry Widya Hesty)

(Received March 16, 2023: Revised December 28, 2023: Accepted January 12, 2024)

Abstract: Accurate, efficient, and stable wind prediction systems for wind turbines are critical to ensuring the operational safety and optimum design of power systems. This study deliberated hyperparameter fine-tuning of ten Machine Learning (ML) models to obtain the best short-term wind speed forecasting model by evaluating the Root-Mean-Square Error (RMSE), Mean Absolute Error (MAE), Correlation, and runtime. The Random Forest (RF) and gradient-boosted tree (GBT) had the best overall performance; however, RF has a much longer training time than GBT. This paper's findings can assist researchers and practitioners in developing the most effective data-driven methods for wind speed and power-generated forecasting.

Keywords: data mining; hyper parameter; RapidMiner; deep learning; ANN; renewable energy

1. Introduction

Significant concerns about the depletion of oil and gas supplies and global warming have made it imperative to seek energy from wind[1)] and other renewable sources[2,3)]. In addition to location[4,5)] and wind turbine selection[6,7)] wind speed forecasting has a substantial impact on energy development and optimizing production. However, the intermittency, fluctuation, and nonlinearity of wind series make it challenging to estimate wind speed accurately[8)]. ML algorithms are an option for forecasting wind speed data. ML is an area of computer science concerned with creating methods and algorithms for data-driven learning and prediction. This subset of artificial intelligence has enabled engineers to develop and implement solutions for complex challenges in various fields[9-11)].

ML algorithms have been used in numerous research to predict wind at various time scales, including the brief term (a few seconds to 30 minutes ahead), the short-range (30 minutes to 6 hours ahead), the medium-term (6 hours to 1 day ahead), and the long-term (1 day to 1 week ahead)[12)]. Moreover, many forecasting studies have been conducted in various fields using ML algorithms. One of the most popular ML techniques is ANN, which has been applied in several scientific domains, i.e., an accurate estimation and continuous forecasting of pollution levels[13)], the waste prediction problem[14)], optimizing fabrication process[15)], logistic regression and forecast risk tolerance behavior[16)], an inquiry on the process factors of the biomass gasifier[17)], prediction total and average monthly night traffic on Serbian state roads[18)], and mechanical property prediction from structure information[19)].

In terms of wind speed and power forecasting, numerous concepts have been adopted. Fuzzy logic[20)], neural networks[21)], and statistical models[22)] can be applied to ML. Deep neural networks were combined with transfer learning to smooth out the rapid transients in the wind power prediction behavior[23)]. At the same time, regression models engaged neural networks and methods such as wavelet transform, particle swarm optimization[20)], and the kNN[24)]. Moreover, wind speed predictions have also been constructed using SVM and its variant, LSSVM[25)]. The effectiveness of a SVR model based on heteroscedastic Gaussian noise in predicting wind production has been established[26)]. Several models, such as the SVR, RF, Extreme Gradient Boost, kNN, and least absolute shrinkage selector operator techniques, are compared to predict wind power values, which show decent accuracy according to several places[27].

Furthermore, many researchers utilize ensemble and hybrid techniques to enhance the accuracy of their models. A hybrid system for monthly wind speed forecasting was introduced, combining linear and nonlinear models through a nonlinear combination approach[8]. This method employs a data-driven intelligent model to determine the most effective combination, thereby improving the precision of wind speed predictions. Similarly, Nascimento proposed a hybrid model for precise wind speed prediction, incorporating time series and artificial intelligence while considering external factors like pressure, temperature, and precipitation to enhance accuracy[28]. In another study, six variations of unorganized machine models, including ELM and ESN with or without regularization coefficients, were examined[29]. The results indicated that the unorganized machines and ELM-based ensembles consistently outperformed linear models in all simulations, demonstrating superior overall performance.

It is essential to set the hyperparameters of the algorithms correctly to build effective prediction models for ML algorithms[30]. Because of their effects on model performance and the fact that the best set of values is unknown, hyperparameter setting has emerged as an important and challenging problem in using ML algorithms[31]. An essential issue in constructing effective prediction models for ML algorithms is appropriately setting the hyperparameters of the algorithm's prerequisite. Therefore, choosing and fine-tuning the ML model is crucial, similar to the physical method, where the model selection significantly impacts the forecast's accuracy[32]. However, there currently are no reliable recommendations for choosing a model, predictor, or set of hyperparameters.

In some research studies, intelligent systems leverage swarm intelligence algorithms to find the most suitable ML learning hyperparameters. A combined forecasting system utilizes four ANNs with optimal weighting coefficients determined by the MSSO[33]. The strategy boosts model performance by concurrently optimizing accuracy and stability. Similarly, adopting a hybrid approach that merges swarm intelligence optimization with artificial intelligence for predicting wind speeds has been done[34]. This method employs wavelet packet decomposition to compress and remove noise from the data, resulting in a faster and more precise model. Lastly, a versatile forecasting framework that combines empirical wavelet transform and neural network-based quantile regression to enhance the robustness and generalization of wind speed predictions has also been developed[35].

A brief review of numerous approaches for forecasting wind speed and power is provided in Table 1. Most current wind speed forecasting studies employ restricted versions of statistical approaches. The most utilized strategies are SVR and ANN algorithms. As for evaluation, there is no universal agreement on the technique, as the authors utilized different error measures. RMSE and its normalized versions stand out from the rest of the list due to their extensive use. Regarding temporal horizons, most research has been undertaken for the next day and only a few for shorter periods.

The lead time of the applicable forecast is rarely provided[36]. Hence, only a few attempts were made to assess wind speed at brief intervals. It has been demonstrated that short-term tactics increase resource coordination in real-time or almost real-time energy dispatching systems[37]. On the other hand, our approach describes the implementation of short-term (15-minute) wind speed using multiple ML models. This work intends to close a gap in the literature by evaluating 10 well-known ML models for short-term wind speed. This study's findings can guide selection models and hyperparameters for wind speed forecasting in various applications and studies.

The remainder of the article is structured as follows. The second section provides a summary of pertinent linked works. The third section presents data and model development. The materials and procedures are detailed accordingly. Also covered are exploratory data analysis, cleansing, and tweaking hyperparameter selection. The following section describes and contrasts the optimal models created through various ML approaches. Finally, the fifth section closes the report with conclusions and future research directions.

2. Materials and methods

The research methods in this paper followed the stages as shown Fig. 1. In general, the research was carried out in four stages, namely: 1) data collection, 2) data preparation, 3) model development using RapidMiner, and 4) analysis of experimental results.

2.1 Dataset input

This study used data from a meteorology mast in Baron Technopark, Daerah Istimewa Yogyakarta, Indonesia. It has a potential wind speed for power generation[38,39]. The measurement was conducted from June 2017 to March 2019[40]. The weather station measured and recorded wind speed, wind direction, solar radiation, ambient temperature, ambient humidity, and atmospheric pressure every fifteen minutes at an altitude of 30 meters.

2.2 Data preparation and science tool

Data preparation is intended to clean, transform, and filter data to prepare it for modeling. Nevertheless, a complete series must consider the impacts of seasonality and trends. Consequently, it is necessary to approximate the missing values. Therefore, the data are imported into Windographer to calculate the monthly average value. Then, the missing and outliner values were replaced with

Table 1. A summary comparison between recent wind speed and power forecasting

Algorithm	Data Source	Data Period	Forecasting Horizon	Location	Matrices	Highlight	Reference
hybrid NWP and ANN	GFS	from June 2019 to June 2020	sufficiently large prediction horizon (>6 h ahead)	Juvent Wind Farm, Switzerland	MAE, RMSE, and MedAE	The proposed models can yield accurate wind farm power forecasts at sites with high terrain and flow complexities.	41)
BNN	Wind Atlas for South Africa datasets	1 January 2016 - 30 June 2017	one hour ahead, wind speeds	the coast of South Africa.	RMSE and MAE	BNNs across ARD and non-ARD models are compared. The result shows that ARD improves predictive performance.	42)
ANN	Wind farm data	Monthly average from January 2015 to December 2019	one month ahead	"Pawan Danawi" wind farm in Sri Lanka	MSE, R, and BIAS	Three training techniques, namely Levenberg-Marquardt, SCG, and BR, were used to test the models. With 70% training and 5% validation, the LM training algorithm generates the most accurate forecasts.	43)
CSO optimizes SVM with important parameters	Wind farm data	Total of 1500 data with 10-min sampling series	Single-step forecasting and multistep forecasting	Shandong Province, China	MAE, MAPE, RMSE, WI, ENS, and Legates and McCabe Index	The model enhances forecasting accuracy and becomes helpful in aiding wind power plant management	44)
MLP, RNN, and SVM	Iowa Environmental Mesonet Data	one-minute data recordings form: Jan– Dec 2004, and Jan –Dec 2001	multistep ahead prediction	Washington region of the USA, and Strom Lake region of the USA	Prediction gain, CC, running time, and RMSE	Using a more informative training set has drastically reduced training durations.	45)
DL	measured and NWP data	January - December 2016	one hour ahead	Xiangshan, Ningbo, China	MAE and MAPE	The proposed method can effectively predict wind speed and has strong generalization performance on different data sets.	46)
3 types of DL networks: LSTM, MLP, and a hybrid of LSTM and MLP	data from a meteorology mast	15-minute data interval from 1 January 2018 to 31 March 2018	A day ahead	Baron Technopark, Daerah Istimewa Yogyakarta, Indonesia	MAE and RMSE	The resulting model achieves the lowest RMSE and MAE values when most weather data is included as input variables.	40)
TSVR, RF, and CNN	wind farm data	10-minute wind ramps for 3 – 12 months	Single-step forecasting and multistep forecasting	North Yorkshire	RMSE, MAE, and MAPE	TSVR is in good agreement for predicting 10-minute wind ramps for a shorter duration of data (3-12 months)	47)
2-stage feature selection and a supervised RF	measured and NWP data	15-minute data interval from Jan – Dec 2015	15 min	Jilin Province, China	MSE	The proposed model performs better in ensuring accuracy, efficiency, and robustness.	48)

Table 1. (continuing) A summary comparison between recent wind speed and power forecasting

Algorithm	Data Source	Data Period	Forecasting Horizon	Location	Matrices	Highlight	Reference
SVR	measured data	10 min data from a one-year wind speed dataset	Single-step forecasting and multistep forecasting	Heilongjiang Province, China	MAE, RMSE, SSE, SSR, SST, and runtime	The best regression indicator among all the models is GLM-TLSSVR.	49)
SVR and ANN	Bangladesh Meteorological Department	6-year historical weather data set of rainfall and temperature	1-day ahead, 7 days ahead, and 10 days ahead future	Chittagong, Bangladesh	RMSE and MAE	SVR can exceed the ANN in precipitation prediction, and ANN can produce better results than the SVR	50)
ANN trained using Bayesian regularisation, Decision Tree-based SGB, and GAM	Wind Atlas South Africa website	10 min data interval from 1 January 2018 to 1 March 2019	two days ahead	South Africa	MAE, MAPE, and RMSE	Interval forecasting and the combination of point forecasts improve forecast performance.	51)
ANN, SVM, random tree, RF, and RepTree.	K.A.CARE local meteorological measurement data.	The hourly data from May 2013 to July 2016	Single-Step Forecasting	Jeddah, Saudi Arabia	RMSE, MAE, and correlation.	Regarding accuracy and computational time, ANN proved to be a flexible technique.	52)
LR, SVM, and DT	Indian Meteorological Department, Pune, and from the AQUASTAT website	Monthly data from January 2015 to December 2019 (5 years, 60 data sets)	Single-Step Forecasting	Indian	MSE and correlation	DT results performed better with less computational effort than empirical models.	53)
RF, GBM, kNN, DT, and extra tree regression	www.tureb.com.tr	10 min data from 1 January - 31 December 2018	Single-Step Forecasting	West of Turkey	MAE, MAPE, RMSE, MSE	The outcomes show that the gradient boosting machine regression method has better forecasting ability.	54)
LSTM networks, DNN, NARX	Meteorological stations	daily wind speed from January 15th, 2018 to January 15th, 2019	-	Andes Mountains, Ecuador	MAPE, MSE, and correlation	The multivariable LSTM network had the highest accurate values, indicating that this model is a powerful method to predict wind speed over complex terrain and highlighting the significance of using observed variables.	55)
SVM, SVR, ANN, DT, kNN, RF, GBT, GLM, DL, LM	Meteorological stations	15 min data from 1 June 2017 - 30 March 2019	15 min	Baron Technopark, Indonesia	RMSE, MAE, correlation, runtime		This Study

them, considering the seasonal patterns for the period in which they occur. Furthermore, the data for the experiment were grouped into two classes: yearly and quarterly. In addition, a month's data was extracted to test the fitness model on the dataset source.

Qualified data series can then be processed using RapidMiner, a well-established end-to-end Java-based analytic tool for text mining, data mining, predictive analytics, and business analytics[56], to achieve an alternate rapid way for prediction[57]. This solution has been implemented in multiple contexts and is today's most popular standalone open-source solution. It is a market leader in its industry[58]. Moreover, according to a performance study, RapidMiner is better than its rival software for data mining activities with less RAM consumption[59].

2.3 Models development

The following part will discuss how RapidMiner was used in this paper, as shown in the enclosed dashed line Fig.1. The first step in exploiting RapidMiner is developing model configurations, as shown in Fig. 2, by dragging them from the selected menu. The following steps are attaching data and experimenting.

2.4 Data pre-processing

The read data step in RapidMiner imports spreadsheet data by following the import configuration wizard instruction. Targeted data was chosen and formatted into "real" type, and the wind speed role was changed into a label. The next step is windowing to transform the data series into smaller, manageable subsets for processing and analysis.

The windowing operates by moving “windows” containing a specified window plus step size, window sliding head, through the series to produce the anticipated label, i.e., the attribute value found in horizon values after the window end[60]. The term "horizon" refers to the number of steps ahead that must be forecasted to train a model to predict future value based on this learning. The step size affects the proportion of the input set fed into the ML process. For this experiment, the window size was 5, the step size was 1, and the horizon was 1.

Moreover, the dataset was separated into training, testing, and validating groups to derive the ML performance. Data separation was performed using the data splitting menu.

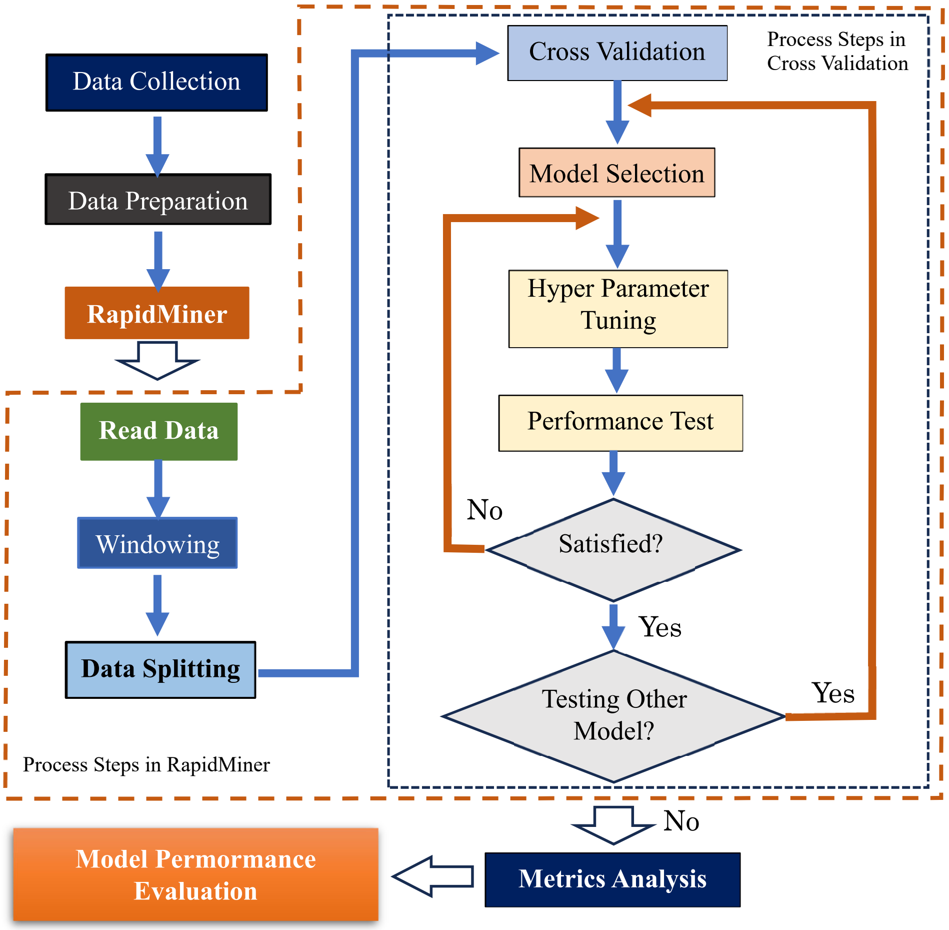

Fig. 1 Research steps flow diagram

2.5 Model training and testing

Following completion of the data engineering and pre-processing stages, the dataset was subjected to cross-validation, as shown in the enclosed dotted line in Fig. 1. This study employed a 10-fold cross-validation technique using the data splitting 80% of training and 20% of testing to assess the accuracy of the model. This technique aims to test the datasets against the model in ten iterations, thereby obtaining the average accuracy percentage. This approach allows for evaluating the model's performance across specific segments of the dataset and the complete dataset. The ML modules used for training in this experiment incorporate the cross-validation operator, which furnishes a collection of performance metrics based on the stated performance configurations for each ML technique.

2.6 Tested ML techniques

Ten model MLs were used in Rapidminer to forecast wind speed, namely, LM, GLM, DL, RF, GBT, DT, ANN, kNN, SVM, and SVR. The model development and procedure using RapidMiner studio, along with their embodied operators, is shown in Fig. 2. Conversely, Table shows the hyperparameter tuning used by each algorithm. Determining hyperparameter values followed the babysitting method[61)].

2.7 Metrics for model performance evaluation

This study evaluates the performance of the deployed prediction model utilizing a set of assessment metrics. In comparing the efficacy of the model for projecting energy use and generation, the following indicators and accompanying formulas are employed: (Eq. 1) RMSE to assess the precision of model prediction performance, (Eq. 2) MAE to characterize the mean of the absolute differences between projected and actual values, and (Eq. 3) correlation to depict how well trends in the anticipated values track trends in the actual values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(F_i - O_i)^2} \qquad \text{(Eq. 1)}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|F_i - O_i| \qquad \text{(Eq. 2)}$$

$$R = \frac{\sum_{i=1}^{n}(O_i-\bar{O})(F_i-\bar{F})}{\sum_{i=1}^{n}(O_i-\bar{O})^2 . \sum_{i=1}^{n}(F_i-\bar{F})^2} \qquad \text{(Eq. 3)}$$

Where O_i = the actual value, F_i = forecasted value, $\bar{O}$ and $\bar{F}$ are the means of actual and forecasted, and n = number of data samples The fourth indicator is runtime, which describes the model's computational requirements. It is calculated as the total of the training and prediction times. The procedure used a standard laptop with 8 GB of RAM and an Intel® Core™ i7-3537U CPU.

2.8 Model performance evaluation

We conducted several tests to optimize our model's performance by varying the sampling technique, data split ratio, and input data quantity. The sampling type in ML algorithm testing influences the model performance due to the unique data. Therefore, we tested our model with

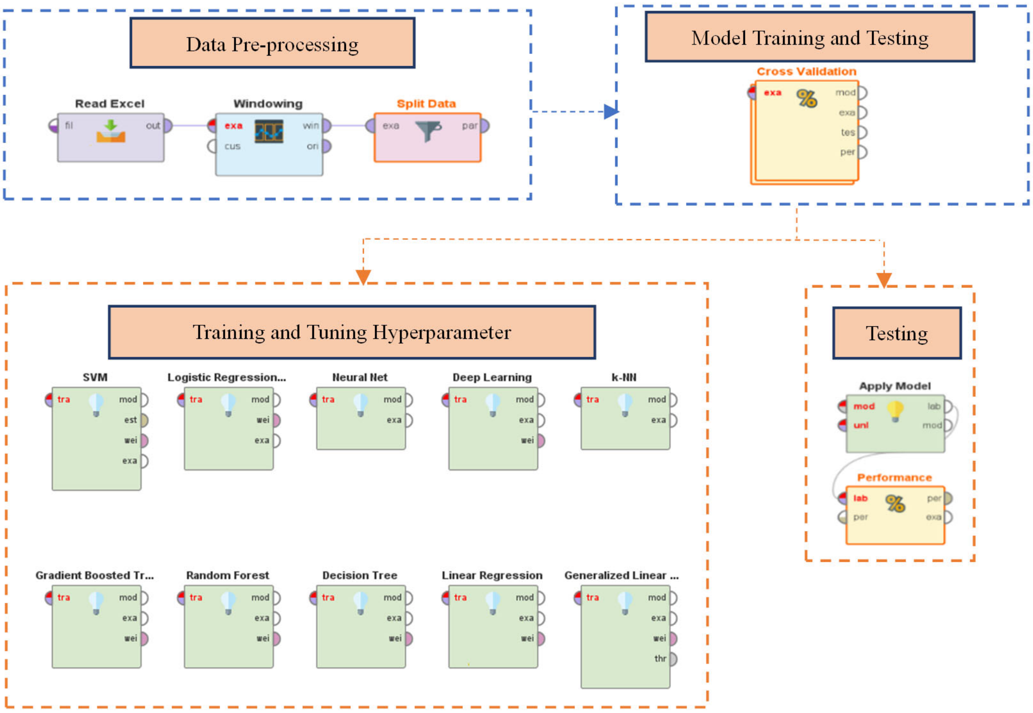

Fig. 2 Model process development using RapidMiner

Table 2. The hyperparameter tuning

No	Algorithm	Hyperparameter (Tuning Value)
1	SVM	Kernel type (radial; polynomial; anova), Gamma (Kernel: Radial) value (0.001; 0.01; 0.1; 0.5; 1.0), C (Kernel: Radial, Gamma: 0.01) value (0.001; 0.01; 0.1; 0.5; 1.0; 5.0; 10.0)
2	SVR	Kernel type (RBF; linier; poly; sigmoid), Gamma (Kernel:RBF) value (0.001; 0.01; 0.1; 0.5), C (Kernel: RBF, Gamma: 0.01) value (0.01; 0.1; 1.0; 5.0)
3	ANN	Training Cycles (200; 400; 600; 1000; 1500), Hidden Layers (HL:2; HL1:2, HL2:2; HL1:2, HL2:2, HL3:2), Learning Rate (0.001; 0.01; 0.1)
4	DT	Maximal Depth (2; 10; 50; 100; 500), Minimum Gain (0.00001; 0.001; 0.01; 0.1), Minimal Leaf (1; 2; 3; 4)
5	kNN	k values (1; 2; 5; 10; 50; 100), measured types (mixed meaures; numerical measures; bregman divergences)
6	RF	Maximal Depth (2; 5; 7; 10), Number of Trees (MD: 10, 50; 100; 200; 500)
7	GBT	Maximal Depth (2; 10; 50; 100), Number of Trees (MD: 10, 50; 100; 200; 500)
8	GLM	Family types (auto; gaussian; poisson; gamma; twedie; negative binomial), lambda (family: auto) value (0.01; 0.1; 0.5; 1.0), alpha (family: auto) value (0; 0.1; 0.5; 1.0)
9	DL	Activation types (rectifier; tanh; maxout; exp rectifier), Epoch values (20; 50; 100; 200), Hidden layer sizes (HL1:10, HL2:10; HL1:50, HL2:50; HL1:100, HL2:100)
10	LR	Feature Selection (M5 prime; greedy; t test; iterative t test), Minimum Tolerance (0.05; 0.1; 0.5; 1.0), Ridge values (0.001; 0.1; 1.0; 10)

various sampling types in the Split Data operator, namely automatic, linear, shuffled, and stratified sampling. That approach will ensure that our model's performances are robust, reliable, and capable of generalizing to different data scenarios.

In ML, it is essential to evaluate a model's performance on data it has not seen before. That is done to ensure that the model is more than memorizing the training data and will generalize well to new data. Splitting data into two sets of training and testing is commonly practiced. This study revealed the effect of splitting the dataset by considering two splitting scenarios. First, in S1, the data were split into 80% training, 20% testing, and 0% validation. Second, in S2, the data were split into 70% training, 10% testing, and 20% validation.

In addition, by using different data quantities to fit the model, one can better understand how the forecasting model performs under different conditions. We tested the model with one month of data, seasonal data spanning three months, and annual data, ensuring its suitability for both short-term and long-term forecasting scenarios.

3. Results and discussion

The hyperparameters for each algorithm have a significant impact on the model performance. Therefore, it is crucial to carefully tune the hyperparameters for each algorithm to achieve the best results. Here, the experimental results are presented and explained.

3.1 Dataset description and features

The collected data characteristics are exposed Table 3 and are depicted as a correlation in Fig. 3 and Fig. 4. Upon examination of the monthly boxplots depicted in Fig. 3, it is evident that there exists a discernible annual seasonal pattern. June presented a higher mean wind speed than other months, especially February and March. The highest mean value was 10.09 m/s in June, and the lowest was 3.81 m/s in February. In addition, Fig. 4 shows the scatter plot of the data, a correlation between wind speed and other weather parameters. A negative correlation between wind speed and ambient humidity is exposed, while the correlation between wind speed and other parameters is positive.

Table 3. Dataset description

Period	**1 June 2017 – 31 March 2019**		
Interval	**15 min**		
Variable	**Mean**	**Min**	**Max**
Wind Speed (m/s)	5.33	0.02	28.90
Ambient Temperature (^{0}C)	26.6	12.2	33.2
Ambient humidity (%)	86.4	29.9	97.0
Atmospheric pressure (kPa)	756	283	761

3.2 Hyperparameter tuning

This study uses 10 ML algorithms, including regression, classification, and ensembles. The hyperparameter value listed in Table 2 was tuned by following the experimental step shown in Fig. 1 using a yearly dataset on splitting of 80% training and 20% testing. Each value was run for every algorithm and is tabulated in Table 4, showing the performance of the different ML algorithms on the best corresponding hyperparameter value. The table compares the algorithms based on four metrics: RMSE, MAE, correlation, and runtime. RMSE and MAE measure the magnitude and absolute error between the predicted and actual values, correlation measures the strength of the relationship between the predicted and actual values, and runtime measures the time each algorithm takes to run.

The RMSE, MAE, and runtime vary with algorithms on their best hyperparameter. However, the correlation values are all very high, indicating that all algorithms can learn the relationship between the input and output variables.

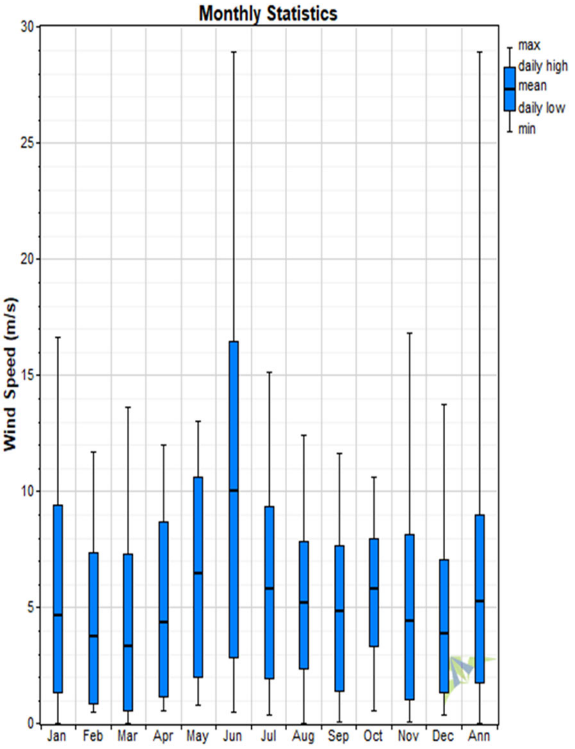

Fig. 3 Monthly box plot of the wind speed value

The best-performing algorithm in terms of RMSE is GBT and RF, with a value of 0.90, while GLM exhibits the worst performance with an RMSE of 1.02. It means that GBT and RF have the lowest average error between their predictions and the actual values, and GLM has the highest error. Compared to ANN, GBT, and RF, learning methods based on DT outperformed them in improving wind speed forecasts. It is common because RF systematically avoids correlation and enhances the model's performance by selecting a random subset of features. Therefore, it will result in different subsets for the various trees, which in turn causes less correlation between the DT and a reduction in variance in the final prediction[62]. Previous wind power forecasting studies have shown that the GBT model performed best[63,64].

Moreover, the best-performing algorithm for MAE is SVR, with a value of 0.54, while SVM exhibits the worst performance with an MAE of 0.67. It means that SVR has the lowest absolute error between its predictions and the actual values, and SVM has the highest error. However, SVR exhibits the most extended runtime of 1040 s, leading to not considering it in practice.

The three quickest algorithms for the provided dataset are DT, LR, and GLM, with corresponding runtimes of 2 s, 4 s, and 5 s, respectively. Although responsive, they exhibit a lower correlation than RF, GBT, and DL. Hence, the selection of an algorithm is contingent on the user's

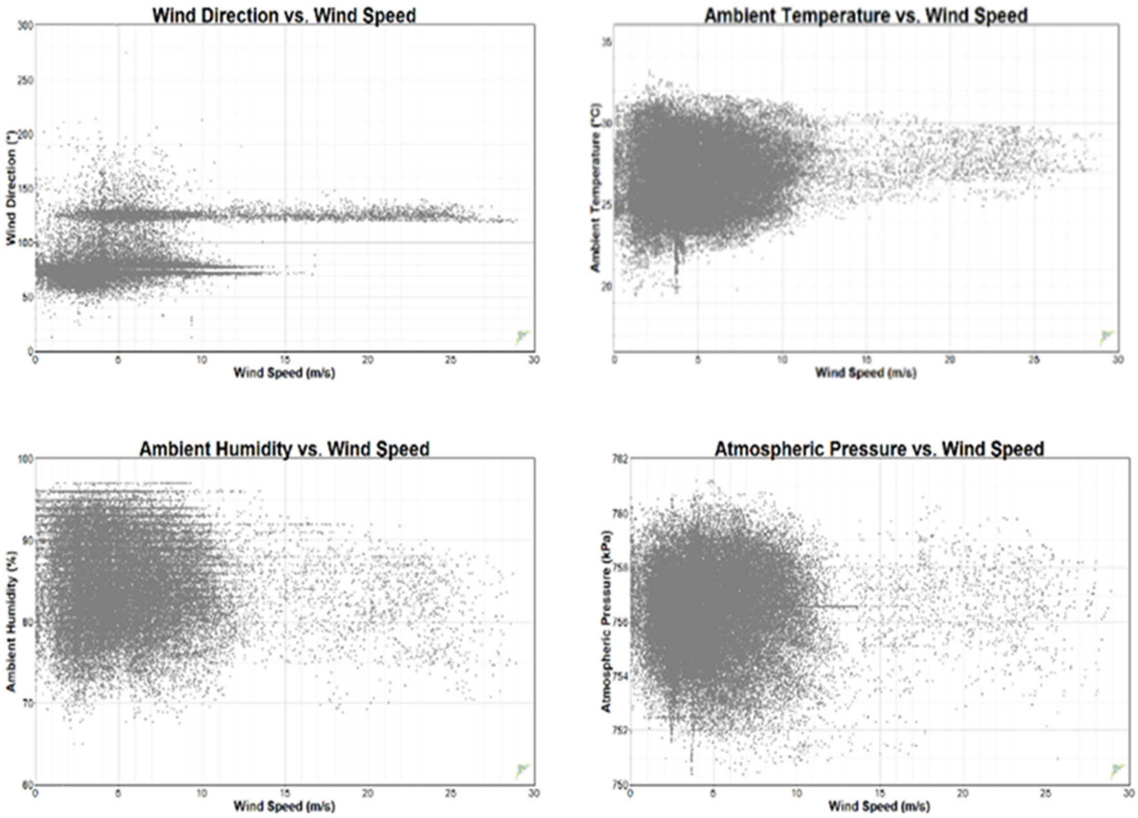

Fig. 4 The scatter plot of wind speed, temperature, humidity, and atmospheric pressure

priorities.

The table shows that the RF and GBT algorithms are the best choices for regression tasks, as they offer a good balance between accuracy and efficiency. GBT and RF have an RMSE of 0.90 and a MAE of 0.55. They also have a high correlation of 0.97, indicating that they can accurately predict the target variable. However, GBT is superior to RF at 35 s vs. 118 s runtime. GBT is a robust ensemble learning algorithm well-suited for various tasks, including classification, regression, and ranking[65].

The best ML algorithm selection is about a trade-off between accuracy and training time; it can sometimes be universally optimal since it always depends on the advantages of lower error rates and the expense of the computational resource. However, it is typical in automated ML to apply a time budget for model tuning and training during benchmarking[66]. It will not be worthwhile if we forecast 10 minutes but require 12 minutes of runtime. In practical forecasting applications, minimal errors are often seen as far more crucial than training time. Nevertheless, if numerous models have nearly comparable performance, it is advisable to choose the faster model. Moreover, it is essential to note that the best-performing algorithm for a given task will depend on the specific characteristics of the dataset. For example, if the dataset is small or noisy, a simpler algorithm such as LR may perform better than a more complex algorithm such as GBT[50].

The best ML algorithm selection is about a trade-off between accuracy and training time, and it can sometimes be universally optimal since it always depends on the advantages of lower error rates and the expense of the computational resource. However, it is typical in automated ML to apply a time budget for model tuning and training during benchmarking[67]. It will not be worthwhile if we forecast 10 minutes but require 12 minutes of runtime. In practical forecasting applications, minimal errors are often seen as being far more crucial than training time; nevertheless, if numerous models have nearly comparable performance, it is advisable to choose the faster model. It is important to note that the best-performing algorithm for a given task will depend on the specific characteristics of the dataset. For example, if the dataset is small or noisy, a simpler algorithm such as LR may perform better than a more complex algorithm such as GBT[68].

3.3 Effect of sampling selection

The splitting operator facilitates the sampling methods to be customized for the experiment. While using the auto setting in the previous result, as shown in Table 4, the question arises as to whether the sampling method influences model performance. Therefore, they were subjected to four sampling selections: automatic, linear, shuffled, and stratified.

Table 4. The best performance of the ML algorithm

Algorithm	Hyperparameter	RMSE	MAE	Correlation	Runtime (s)
SVM	Gamma 0.01	1.01	0.67	0.96	412
SVR	C 5	0.91	0.54	0.96	1040
ANN	Training Cycle 400	0.94	0.60	0.96	421
DT	Max. Depth 10	0.94	0.57	0.96	2
kNN	K:10	0.94	0.58	0.96	124
RF	Max. Depth 10	0.90	0.55	0.97	118
GBT	Max. Depth 10	0.90	0.55	0.97	35
GLM	Lambda 0.01	1.02	0.63	0.96	5
DL	Epoch 200	0.91	0.57	0.97	807
LR	Default	1.01	0.63	0.96	4

Table 5. Comparison of sampling type selection test

Algorithms	RMSE				MAE				Correlation			
Sampling Type	A	B	C	D	A	B	C	D	A	B	C	D
SVM	1.01	1.76	1.01	1.05	0.67	1.15	0.67	0.71	0.96	0.74	0.96	0.96
SVR	0.91	1.33	0.91	0.91	0.54	0.67	0.54	0.54	0.96	0.83	0.96	0.96
ANN	0.94	1.75	0.94	0.94	0.60	0.90	0.60	0.60	0.96	0.72	0.96	0.96
DT	0.94	1.64	0.94	0.94	0.57	0.78	0.57	0.57	0.96	0.74	0.96	0.96
kNN	0.94	1.37	0.94	0.94	0.58	0.74	0.58	0.58	0.96	0.82	0.96	0.96
RF	0.90	1.55	0.90	0.90	0.55	0.75	0.55	0.55	0.97	0.77	0.97	0.97
GBT	0.90	1.54	0.90	0.90	0.55	0.74	0.55	0.55	0.97	0.77	0.97	0.97
GLM	1.02	1.64	1.02	1.02	0.63	0.77	0.63	0.63	0.96	0.75	0.96	0.96
DL	0.91	1.17	0.91	0.91	0.55	0.64	0.55	0.55	0.96	0.87	0.96	0.96
LR	1.01	1.69	1.01	1.01	0.63	0.78	0.63	0.63	0.96	0.73	0.96	0.96

Note: A = Automatic, B = Linear, C = Shuffled, D = Stratified

Table 5 shows the sampling-type selection test results for splitting data. Linear sampling produces the worst performance for all models compared to other sampling types. However, the data splitting method with the lowest RMSE and MAE values for most algorithms is shuffled and automatic. Compared to stratified and linear splitting, shuffled splitting keeps the proportions of the different classes or groups in the data. Furthermore, the utilization of automatic selection sampling is deemed sufficient.

3.4 Effect of data splitting and data quantity

The effect of splitting the dataset was revealed by considering two splitting scenarios, S1 and S2, and the data quantity effect was studied using yearly, seasonal, and August data. The experiment results are presented in Table 6.

For yearly data, RF, DL, kNN, and DT exhibit more goodness fitting than GBT, but DT is less fit than RF, DL, and kNN. GBTs are an ensemble learning algorithm combining multiple weak learners to produce a strong learner. GBTs are known for their excellent performance on various tasks, but they can be more prone to overfitting than other ML algorithms, such as RF, DL, kNN, and DT. When the data is split into three sets, the training set is smaller than when the data is split into two sets. It can lead to overfitting for GBTs.

In addition, the effect of quantity data can be studied by comparing performance results from yearly, seasonal, and August data on the S2 scenario. Generally, the RMSE worsens for all algorithms when the amount of data gets smaller. Furthermore, the MAE changes pattern follows the same trend as RMSE, except for GBT, which shows the opposite trend.

Moreover, Table 6 shows that all of the ML algorithms have a high correlation with each other, indicating that they are all making similar predictions on the dataset. However, there are some exciting trends in the table. First, the fewer data sets available, the worse the correlation value. The training sets are smaller for the more granular splitting scenarios. As a result, the algorithms are more likely to overfit the training data, which can lead to decreased correlation with the other algorithms. Second, the RF algorithm has the highest correlation with all other algorithms, suggesting that RF is the most stable and consistent algorithm. Third, the DT algorithms have the lowest correlation with all other algorithms. It suggests that DT is more sensitive to changes in the dataset and may be more prone to overfitting.

3.5 Wind speed prediction

RapidMiner evaluates performance using 20% of the uncertain testing dataset. Fig. 6 depicts the discrepancy between the relevant model's forecast and actual wind speed using S1. Overall, the ML system accurately predicted the wind speed, and there is a significant correlation between the anticipated and actual values.

Table 6. The performance of the ML algorithms for different splitting scenario

Scenario, Data Periode	RF	DL	kNN	GBT	DT
			RMSE		
S1,Yearly	0,90	0,91	0,94	0,90	0,94
S2,Yearly	0,85	0,88	0,89	2,21	0,96
S2,DJF	0,74	0,80	0,77	1,51	0,86
S2,MAM	0,89	0,92	0,91	1,77	1,02
S2,JJA	0,96	0,97	1,02	2,82	1,09
S2,SON	0,90	0,98	0,96	1,67	0,96
S2,Average	0,87	0,92	0,91	1,94	0,98
S2,Min	0,74	0,80	0,77	1,51	0,86
S2,Max	0,96	0,98	1,02	2,82	1,09
S2, Aug	1,19	1,41	1,26	1,72	1,65
			MAE		
	RF	DL	kNN	GBT	DT
S1,Yearly	0,55	0,57	0,58	0,55	0,57
S2,Yearly	0,54	0,57	0,57	1,60	0,57
S2,DJF	0,45	0,48	0,48	1,31	0,49
S2,MAM	0,57	0,63	0,61	1,43	0,61
S2,JJA	0,59	0,60	0,62	1,96	0,65
S2,SON	0,55	0,59	0,60	1,36	0,58
S2,Average	0,54	0,58	0,58	1,51	0,58
S2,Min	0,45	0,48	0,48	1,31	0,49
S2,Max	0,59	0,63	0,62	1,96	0,65
S2, Aug	0,87	1,06	0,92	1,40	1,12
			Correlation		
	RF	DL	kNN	GBT	DT
S1,Yearly	0,97	0,97	0,96	0,97	0,96
S2,Yearly	0,97	0,97	0,97	0,96	0,96
S2,DJF	0,95	0,94	0,94	0,94	0,93
S2,MAM	0,94	0,94	0,94	0,94	0,93
S2,JJA	0,98	0,98	0,97	0,98	0,97
S2,SON	0,93	0,92	0,92	0,93	0,92
S2,Average	0,95	0,95	0,94	0,95	0,94
S2,Min	0,93	0,92	0,92	0,93	0,92
S2,Max	0,98	0,98	0,97	0,98	0,97
S2, Aug	0,87	0,82	0,85	0,85	0,77

However, the graph shows that DT performance is low compared to others. That is because the deviation in tree structures variance emerges as the negative side of the method[69]. DT generally tends to have low bias and high variance[70]. Nevertheless, DT has a much shorter training time than the RF. As the number of trees in an RF increases, so does the time required to train each. It is frequently essential when working on an ML project with a tight deadline.

4. Conclusion

Ten well-known ML models (LM, GLM, DL, RF, GBT, DT, ANN, kNN, SVM, and SVR) were generated for short-term wind speed forecasting utilizing a data set from June 2017 to March 2019 from a meteorology mast in Baron Technopark, Daerah Istimewa Yogyakarta, Indonesia. The RF and GBT models performed best on the test dataset, with the GBT model doing considerably better.

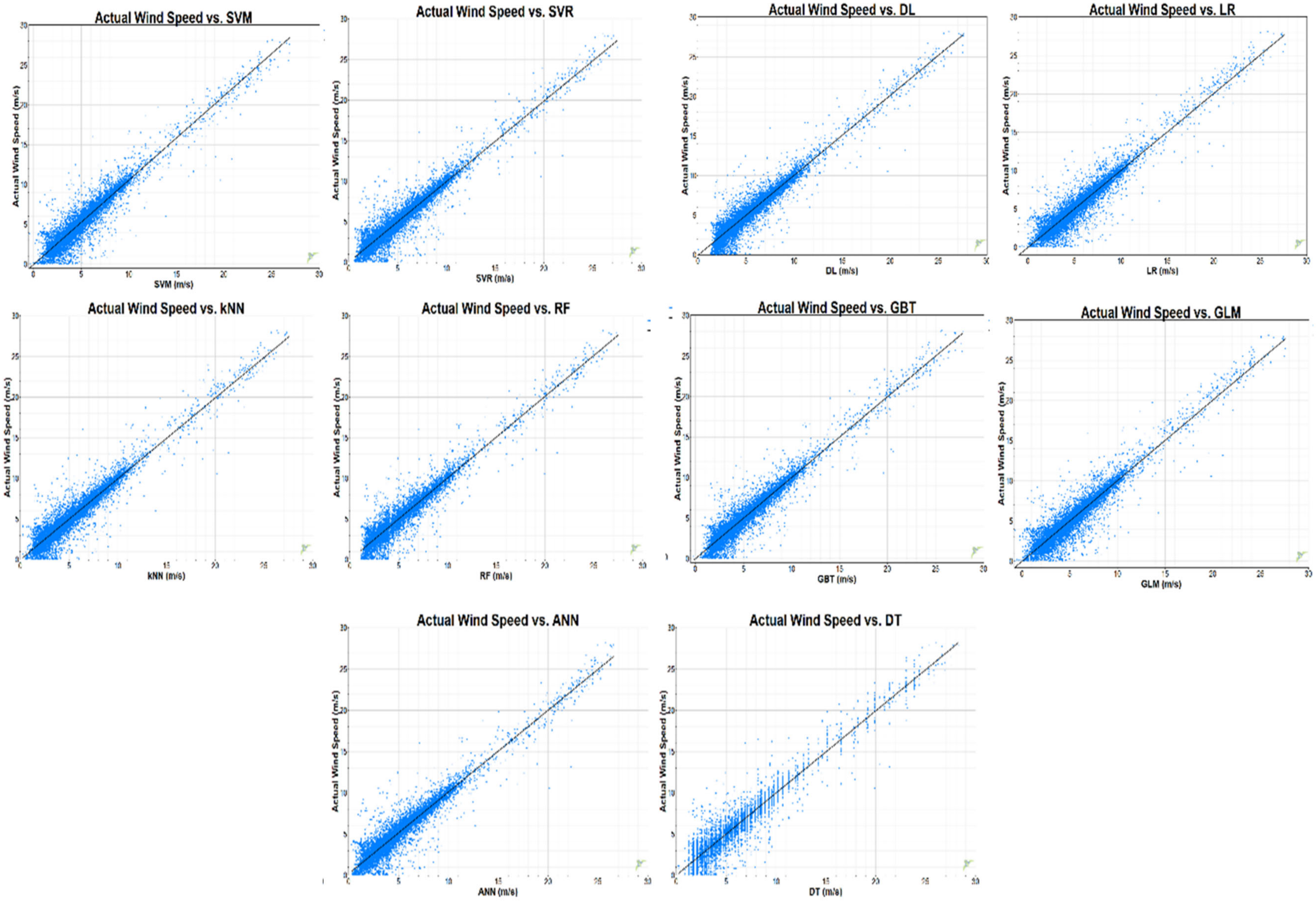

Fig. 5 Wind speed prediction charts (the predictions vs. the actual values) of ML models

In addition, both models had a better RMSE error rate. However, RF training requires more time and memory than GBT. Hence, GBT is recommended for applications in the real world. Hyperparameter adjustment is required to maximize the full potential of ML models. The more robust models (e.g., GBT, LR) are correct even without training, but others (e.g., ANN, SVM, SVR) have significant errors with their default parameters and require training.

The application-ready ML models from a high-level program library described in this article could assist researchers and practitioners in offering relatively accurate solutions for operational wind speed predictions to attain maximum power. Nonetheless, there is room for improvement in fine-tuning various combinations of hyperparameters to estimate wind speed better precisely. Also, it would be intriguing to apply the models to additional datasets from different regions. It would enable us to determine the extent to which site location influences model performance. In addition, it is challanging to investigate the integration of potentially utilizing the physical process model data as features for the machine learning model or adopting dynamic weighting depending on forecast uncertainty. However, there is still an extent of improvement in fine-tuning other combinations of hyperparameters to predict wind speed more accurately. Furthermore, applying the models to other datasets from different locations would be interesting. It would allow us to study how much a particular site location affects model performance. To improve the accuracy of wind forecasting, more research into the hybridization of physical and ML models is another option.

Acknowledgments

This paper is part of the funding house program of the National Research and Innovation (BRIN), Republic of Indonesia. The authors thank The Baron Techno Park for the data provided and the Directorate of Repositories, Multimedia and Scientific Publishing for library support provided via e-lawas BRIN.

Abbreviation

ANN	Artificial Neural Network
ARD	Automatic Relevance Determination
BNN	Bayesian Neural Network
BR	Bayesian Regularization
CC	Correlation Coefficient
CNN	Convolutional neural networks
CSO	Cuckoo Search Algorithm
DJF	December, January, February
DL	Deep Learning
DNN	Dynamic Neural Networks
DT	Decision Tree
ELM	Extreme Learning Machines
ENS	Nash-Sutcliffe coefficient
ESN	Echo State Networks
GAM	Generalized Additive Models
GBM	Gradient Boosting Machine
GBT	Gradient Boosted Tree
GLM	Generalized Linear Model
GFS	Global Forecast System
JJA	Juni, July, August
kNN	K-Nearest Neighbor
LM	Linear Model
LSSVM	Least Square Support Vector Machines
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAM	March, April, May
MAPE	Mean Absolut Percentage error
MedAE	Median Absolute Error
ML	Machine Learning
MLP	Multilayer perceptron
MSSO	Multi-objective Salp Swarm Optimizer
NARX	Nonlinear Autoregressive Exogenous
NWP	Numerical Weather Prediction
RAM	Random Access Memory
RF	Random Forest
RMSE	Root-Mean-Square Error
RNN	Recurrent Neural Network
SCG	Scaled Conjugate Gradient
SGB	Stochastic Gradient Boosting
SON	September, October, November
SVM	Support Vector Machine
SVR	Support Vector Regression
TSVR	Twin Support vector regression
WI	Willmott's Index

Symbols

O_i	the actual value
F_i	forecasted value
$\bar{O}$	mean of actual
$\bar{F}$	mean of forecasted
n	number of data samples.

References

1) T.C. Rudien, D.H. Didane, M.F.M. Batcha, K. Abdullah, S. Mohd, B. Manshoor, and S. Al-Alimi, “Technical feasibility analysis of wind energy potentials in two sites of east malaysia: santubong and kudat,” Evergreen, 8 (2) 271–279 (2021). doi:10.5109/4480703.
2) A. Fadhila Pachman, D. Hissein Didane, M. Al-Ghriybah, N. Fitriah Nasir, S. Al-Alimi, and B. Manshoor, “A Study of Global Solar Radiations Measurement in Java Island, Indonesia,” 2023.
3) M. Muslihudin, W.R. Adawiyah, E. Hendarto, R.D. Megasari, and M.F. Ramadhan, “Environmental constraints in building process a sustainable geothermal power plant on the slopes of slamet mount, central java, indonesia,” Evergreen, 9 (2) 300–309 (2022). doi:10.5109/4793669.
4) M. Al-Ghriybah, “Assessment of wind energy

potentiality at ajloun, jordan using weibull distribution function," Evergreen, 9 (1) 10–16 (2022). doi:10.5109/4774211.

5) A.G. Abdullah, A.H. Setiorini, N.A. Dwitasari, D.L. Hakim, and M. Aziz, "Location suitability analysis for wind farm exploitation using fuzzy analytic hierarchy process," Indonesian Journal of Science and Technology, 6 (3) 523–534 (2021). doi:10.17509/ijost.v6i3.38957.

6) M. Al-Ghriybah, "Performance analysis of a modified savonius rotor using a variable blade thickness," Evergreen, 9 (3) 645–653 (2022). doi.org/10.5109/4842522.

7) Y.D. Herlambang, Supriyo, B. Prasetiyo, A.S. Alfauzi, T. Prasetyo, Marliyati, and F. Arifin, "Experimental and simulation investigation on savonius turbine: influence of inlet-outlet ratio using a modified blade shaped to improve performance," Evergreen, 9 (2) 457–464 (2022). doi:10.5109/4794172.

8) P.S.G. de Mattos Neto, J.F.L. de Oliveira, D.S. Domingos, H.V. Siqueira, M.H.N. Marinho, and F. Madeiro, "An adaptive hybrid system using deep learning for wind speed forecasting," Inf Sci (N Y), 581 495–514 (2021). doi:10.1016/j.ins.2021.09.054.

9) G.D. Nugraha, B. Sudiarto, and K. Ramli, "Machine learning-based energy management system for prosumer," Evergreen, 7 (2) 309–313 (2020). doi:10.5109/4055238.

10) S.A. Shaedi, N. Mohd-Ghazali, J.T. Oh, R. Ahmad, and Y. Mohd-Yunos, "Entropy generation minimization of two-phase flow in a mini channel with genetic algorithm," Evergreen, 6 (1) 39–43 (2019). doi:10.5109/2321004.

11) T.N. Dief, and S. Yoshida, "System identification for quad-rotor parameters using neural network," Evergreen, 3 (1) 6–11 (2016). doi:10.5109/1657380.

12) M. Yesilbudak, S. Sagiroglu, and I. Colak, "A novel implementation of knn classifier based on multi-tupled meteorological input data for wind power prediction," Energy Convers Manag, 135 434–444 (2017). doi:10.1016/j.enconman.2016.12.094.

13) R.E. Caraka, R.C. Chen, H. Yasin, S. Suhartono, Y. Lee, and B. Pardamean, "Hybrid vector autoregression feedforward neural network with genetic algorithm model for forecasting space-time pollution data," Indonesian Journal of Science and Technology, 6 (1) 243–266 (2021).

14) A. Utku, and S.K. Kaya, "Deep learning based a comprehensive analysis for waste prediction," Operational Research in Engineering Sciences: Theory and Applications, 5 (2) 176–189 (2022).

15) N. Weake, M. Pant, A. Sheoran, A. Haleem, and H. Kumar, "Optimising parameters of fused filament fabrication process to achieve optimum tensile strength using artificial neural network," Evergreen, 7 (3) 373–381 (2020). doi:10.5109/4068614.

16) A. Bhattacharya, A. Dutta, and S. Kar, "Does demographics influence the risk behaviour of urban investors? a machine learning model based approach," Operational Research in Engineering Sciences: Theory and Applications, (2022).

17) H. Furqon, R. Setiawan, R. Dhelika, A. Surjosatyo, and H. Dafiqurrohman, "Investigation of relative influence of process variables in a 10-kw downdraft fixed-bed gasifier with ann models," Journal of Applied Engineering Science, 20 (3) 971–977 (2022). doi:10.5937/jaes0-34344.

18) D. Mladenović, S. Janković, S. Zdravković, S. Mladenović, and A. Uzelac, "Night traffic flow prediction using k-nearest neighbors algorithm," Operational Research in Engineering Sciences: Theory and Applications, 5 (1) 152–168 (2022).

19) J.F. Fatriansyah, S.N. Surip, and F. Hartoyo, "Mechanical property prediction of poly(lactic acid) blends using deep neural network," Evergreen, 9 (1) 141–144 (2022). doi:10.5109/4774229.

20) G. Martínez-Arellano, L. Nolle, R. Cant, A. Lotfi, and C. Windmill, "Characterisation of large changes in wind power for the day-ahead market using a fuzzy logic approach," KI-Künstliche Intelligenz, 28 (4) 239–253 (2014).

21) A. Daraeepour, and D.P. Echeverri, "Day-ahead wind speed prediction by a Neural Network-based model," in: ISGT 2014, IEEE, 2014: pp. 1–5.

22) H. Zhou, J.X. Jiang, and M. Huang, "Short-term wind power prediction based on statistical clustering," in: 2011 IEEE Power and Energy Society General Meeting, IEEE, 2011: pp. 1–7.

23) A.S. Qureshi, A. Khan, A. Zameer, and A. Usman, "Wind power prediction using deep neural network based meta regression and transfer learning," Appl Soft Comput, 58 742–755 (2017).

24) N.A. Treiber, J. Heinermann, and O. Kramer, "Wind power prediction with machine learning," in: Computational Sustainability, Springer, 2016: pp. 13–29.

25) M.G. De Giorgi, S. Campilongo, A. Ficarella, and P.M. Congedo, "Comparison between wind power prediction models based on wavelet decomposition with least-squares support vector machine (ls-svm) and artificial neural network (ann)," Energies (Basel), 7 (8) 5251–5272 (2014).

26) Q. Hu, S. Zhang, M. Yu, and Z. Xie, "Short-term wind speed or power forecasting with heteroscedastic support vector regression," IEEE Trans Sustain Energy, 7 (1) 241–249 (2015).

27) H. Demolli, A.S. Dokuz, A. Ecemis, M. %J E.C. Gokcek, and Management, "Wind power forecasting based on daily wind speed data using machine learning algorithms," Energy Conversion, 198 111823 (2019).

28) H. do Nascimento Camelo, P.S. Lucio, J.B.V.L. Junior, and P.C.M. de Carvalho, "A hybrid model based on time series models and neural network for forecasting

wind speed in the brazilian northeast region," Sustainable Energy Technologies and Assessments, 28 65–72 (2018). doi:10.1016/j.seta.2018.06.009.

29) J. Belotti, H. Siqueira, L. Araujo, S.L. Stevan, P.S.G. de Mattos Neto, M.H.N. Marinho, J.F.L. de Oliveira, F. Usberti, M. de Almeida Leone Filho, A. Converti, and L.A. Sarubbo, "Neural-based ensembles and unorganized machines to predict streamflow series from hydroelectric plants," Energies (Basel), 13 (18) (2020). doi:10.3390/en13184769.

30) F. Hutter, L. Kotthoff, and J. Vanschoren, "Automated machine learning: methods, systems, challenges," Springer Nature, 2019.

31) P. Probst, M.N. Wright, and A. Boulesteix, "Hyperparameters and tuning strategies for random forest," Wiley Interdiscip Rev Data Min Knowl Discov, 9 (3) e1301 (2019).

32) M.J. Mayer, and G. Gróf, "Extensive comparison of physical models for photovoltaic power forecasting," Appl Energy, 283 116239 (2021).

33) Z. Cheng, and J. Wang, "A new combined model based on multi-objective salp swarm optimization for wind speed forecasting," Applied Soft Computing Journal, 92 (2020). doi:10.1016/j.asoc.2020.106294.

34) T. Fu, and C. Wang, "A hybrid wind speed forecasting method and wind energy resource analysis based on a swarm intelligence optimization algorithm and an artificial intelligence model," Sustainability (Switzerland), 10 (11) (2018). doi:10.3390/su10113913.

35) X. Zhang, "Developing a hybrid probabilistic model for short-term wind speed forecasting," Applied Intelligence, 53 728–745 (2023). doi:10.1007/s10489-022-03644-8/Published.

36) C. He, Q. Chen, X. Fang, Y. Zhou, R. Fu, and W. Jin, "Wind speed forecasting in fishing harbor anchorage using a novel deep convolutional neural network," Front Earth Sci (Lausanne), 1043 (2021).

37) D. Said, "Intelligent photovoltaic power forecasting methods for a sustainable electricity market of smart micro-grid," IEEE Communications Magazine, 59 (7) 122–128 (2021). doi:10.1109/MCOM.001.2001140.

38) N.W. Hesty, D.G. Cendrawati, R. Nepal, and M.I. Al Irsyad, "Wind Energy Potential Assessment Based-on WRF Four-Dimensional Data Assimilation System and Cross-Calibrated Multi-Platform Dataset," in: IOP Conf Ser Earth Environ Sci, IOP Publishing Ltd, 2021. doi:10.1088/1755-1315/897/1/012004.

39) N. Widya Hesty, D. Galuh Cendrawati, R. Nepal, and M. Indra al Irsyad, "Crawford School of Public Policy CAMA Centre for Applied Macroeconomic Analysis Energy Potential Assessments and Investment Opportunities for Wind Energy in Indonesia," n.d.

40) H. Halidah, N. Hesty, P. Aji, D. Amelia, and K. Akhmad, "Short-Term Wind Forecasting with Weather Data using Deep Learning-Case Study in Baron Techno Park," 2023.

41) L. Donadio, J. Fang, and F. Porté-Agel, "Numerical weather prediction and artificial neural network coupling for wind energy forecast," Energies (Basel), 14 (2) 338 (2021).

42) R. Mbuvha, W.T. Mongwe, and T. Marwala, "Separable shadow hamiltonian hybrid monte carlo for bayesian neural network inference in wind speed forecasting," Energy and AI, 6 100108 (2021).

43) A.T. Peiris, J. Jayasinghe, and U. Rathnayake, "Forecasting wind power generation using artificial neural network:'pawan danawi'—a case study from sri lanka," Journal of Electrical Computer Engineering, 2021 (2021).

44) T. Liu, S. Liu, J. Heng, and Y. Gao, "A new hybrid approach for wind speed forecasting applying support vector machine with ensemble empirical mode decomposition and cuckoo search algorithm," Applied Sciences, 8 (10) 1754 (2018).

45) A.O. Sarp, E.C. Mengüç, M. Peker, and B.Ç. Güvenç, "Data-adaptive censoring for short-term wind speed predictors based on mlp, rnn, and svm," IEEE Syst J, (2022).

46) C. He, Q. Chen, X. Fang, Y. Zhou, R. Fu, and W. Jin, "Wind speed forecasting in fishing harbor anchorage using a novel deep convolutional neural network," Front Earth Sci (Lausanne), 9 1043 (2021). doi:10.3389/feart.2021.731803.

47) H.S. Dhiman, and D. %J I.T. on E.E.S. Deb, "Machine intelligent and deep learning techniques for large training data in short-term wind speed and ramp event forecasting," International Transactions on Electrical Energy Systems, 31 (9) e12818 (2021).

48) K. Shi, Y. Qiao, W. Zhao, Q. Wang, M. Liu, and Z. Lu, "An improved random forest model of short-term wind-power forecasting to enhance accuracy, efficiency, and robustness," Wind Energy, 21 (12) 1383–1394 (2018).

49) S. Zhang, C. Liu, W. Wang, and B. Chang, "Twin least square support vector regression model based on gauss-laplace mixed noise feature with its application in wind speed prediction," Entropy, 22 (10) 1102 (2020).

50) P. Meesad, S. Sodsee, and H. Unger, "Recent advances in information and communication technology 2017," Proceedings of the 13th International Conference on Computing and Information Technology (IC2IT), (2017).

51) L.O. Daniel, C. Sigauke, C. Chibaya, and R. Mbuvha, "Short-term wind speed forecasting using statistical and machine learning methods," (n.d.). doi:10.3390/a13060132.

52) T. Brahimi, "Using artificial intelligence to predict wind speed for energy application in saudi arabia," Energies (Basel), 12 (24) 4669 (2019).

53) R. Meenal, K. Kailash, P.A. Michael, J.J. Joseph, F.T. Josh, and E. Rajasekaran, "Machine learning based

smart weather prediction," Indonesian Journal of Electrical Engineering and Computer Science, 28 (1) 508–515 (2022). doi:10.11591/ijeecs.v28.i1.pp508-515.

54) U. Singh, M. Rizwan, M. Alaraj, and I. Alsaidan, "A machine learning-based gradient boosting regression approach for wind power production forecasting: a step towards smart grid environments," Energies (Basel), 14 (16) (2021). doi:10.3390/en14165196.

55) G. López, and P. Arboleya, "Short-term wind speed forecasting over complex terrain using linear regression models and multivariable lstm and narx networks in the andes mountains, ecuador," Renew Energy, 183 351–368 (2022).

56) P. Almeida, and J. Bernardino, "A survey on open source Data Mining Tools for SMEs," in: New Advances in Information Systems and Technologies, Springer, 2016: pp. 253–262.

57) RapidMiner, "Data science platform | machine learning | rapidminer," (2017).

58) C. Idoine, P. Krensky, E. Brethenoux, J. Hare, S. Sicular, and S. Vashisth, "Magic quadrant for data science and machine-learning platforms," Gartner, Inc, 13 (2018).

59) S. Singh, Y. Liu, W. Ding, and Z. Li, "Evaluation of data mining tools for Telecommunication Monitoring Data using design of experiment," in: 2016 IEEE International Congress on Big Data (BigData Congress), IEEE, 2016: pp. 283–290.

60) R.I. Rasel, N. Sultana, and P. Meesad, "An efficient modelling approach for forecasting financial time series data using support vector regression and windowing operators," International Journal of Computational Intelligence Studies, 4 (2) 134–150 (2015).

61) S.M. Malakouti, "Babysitting hyperparameter optimization and 10-fold-cross-validation to enhance the performance of ml methods in predicting wind speed and energy generation," Intelligent Systems with Applications, 19 (2023). doi:10.1016/j.iswa.2023.200248.

62) A. Lindholm, N. Wahlström, F. Lindsten, and T.B. Schön, "Machine Learning: A First Course for Engineers and Scientists," Cambridge University Press, 2022.

63) U. Singh, M. Rizwan, M. Alaraj, and I. Alsaidan, "A machine learning-based gradient boosting regression approach for wind power production forecasting: a step towards smart grid environments," Energies (Basel), 14 (16) 5196 (2021).

64) L. Silva, "A feature engineering approach to wind power forecasting: gefcom 2012," Int J Forecast, 30 (2) 395–401 (2014).

65) M. Ebrahimi, M. Mohammadi-Dehcheshmeh, E. Ebrahimie, and K.R. Petrovski, "Comprehensive analysis of machine learning models for prediction of sub-clinical mastitis: deep learning and gradient-boosted trees outperform other models," Comput Biol Med, 114 (2019). doi:10.1016/j.compbiomed.2019.103456.

66) L.O. Daniel, C. Sigauke, C. Chibaya, and R. Mbuvha, "Short-term wind speed forecasting using statistical and machine learning methods," Algorithms, 13 (6) (2020). doi:10.3390/A13060132.

67) P. Gijsbers, E. LeDell, J. Thomas, S. Poirier, B. Bischl, and J. Vanschoren, "An open source automl benchmark," ArXiv Preprint ArXiv:1907.00909, (2019).

68) A. Khosravi, L. Machado, and R.O. Nunes, "Time-series prediction of wind speed using machine learning algorithms: a case study osorio wind farm, brazil," Appl Energy, 224 550–566 (2018). doi:10.1016/j.apenergy.2018.05.043.

69) H. Toylan, "Application of prediction of solar irradiance using bagging decision tree-based machine learning," Kırklareli Üniversitesi Mühendislik ve Fen Bilimleri Dergisi, 8 (1) 15–24 (2022).

70) Y.L. Suen, P. Melville, and R.J. Mooney, "Combining bias and variance reduction techniques for regression trees," in: Springer, 2005: pp. 741–749.

Supplement Table.

Summary of the metrics ML model.

Tuning	RMSE	MAE	Corr.
			DT
Default	1.07	0.65	0.94
Maximal Depth			
2	2.43	1.60	0.71
10	0.94	0.57	0.96
50	1.00	0.60	0.96
100	1.00	0.60	0.96
500	1.00	0.60	0.96
Minimal Gain			
0.0001	0.94	0.57	0.96
0.001	0.94	0.57	0.96
0.01	0.94	0.57	0.96
0.1	0.97	0.62	0.96
Minimal Leaf			
1	0.96	0.57	0.96
2	0.96	0.57	0.96
3	0.94	0.57	0.96
4	0.95	0.57	0.96

Tuning	RMSE	MAE	Corr.
			DL
Default	1.01	0.67	0.95
Activation			
Rectifier	0.98	0.62	0.96
Tanh	0.94	0.59	0.96
Maxout	0.94	0.60	0.96
Exp Rectifier	0.94	0.60	0.96
Epoch			
20	0.97	0.63	0.96
50	0.93	0.59	0.96
100	0.92	0.57	0.96
200	0.91	0.55	0.96
Hidden Layer Size			
HL1: 10; HL2: 10	0.92	0.57	0.96
HL1: 50; HL2: 50	0.91	0.55	0.96
HL1: 100; HL2: 100	0.92	0.57	0.96

Tuning	RMSE	MAE	Corr.
			ANN
Default	1.05	0.68	0.94
Training Cycles			
200	0.98	0.62	0.96
400	0.94	0.59	0.96
600	0.94	0.60	0.96
1000	0.94	0.60	0.96
1500	0.94	0.60	0.96
Hidden Layer			
HL: 2	0.98	0.62	0.96

Tuning	RMSE	MAE	Corr.
HL1: 2; HL2: 2	1.11	0.80	0.96
HL1: 2; HL2; HL3:2	1.01	0.66	0.96
Learning Rate			
0.001	1.00	0.64	0.96
0.01	0.98	0.62	0.96
0.1	1.16	0.84	0.96

Tuning	RMSE	MAE	Corr.
			KNN
Default	1.04	0.62	0.95
k			
K:1	5.80	4.38	0.16
K:2	1.04	0.62	0.95
K:5	0.97	0.60	0.96
K:10	0.94	0.58	0.96
K:50	0.95	0.57	0.96
K:100	0.96	0.58	0.96
Measured Types			
Mixed Measures	0.94	0.58	0.96
Numerical Measures	0.94	0.58	0.96
Bregman Divergences	0.95	0.59	0.96

Tuning	RMSE	MAE	Corr.
			RF
Default	0.93	0.58	0.96
Maximal Depth			
2	2.32	1.48	0.74
5	1.01	0.64	0.96
7	0.93	0.58	0.96
10	0.90	0.55	0.97
Number of Trees		MD:10	
50	0.90	0.55	0.97
100	0.90	0.55	0.97
200	0.90	0.55	0.97
500	0.90	0.55	0.97

Tuning	RMSE	MAE	Corr.
			GBT
Default	0.90	0.55	0.97
Maximal Depth			
2	0.93	0.57	0.96
10	0.90	0.55	0.97
50	0.93	0.56	0.96
100	0.93	0.57	0.96
Number of Trees		MD:10	
50	0.90	0.55	0.97

Tuning	RMSE	MAE	Corr.
100	0.90	0.55	0.97
200	0.90	0.55	0.97
500	0.90	0.55	0.97

Tuning	RMSE	MAE	Corr.
			LR
Default	1.01	0.63	0.96
Feature Selection			
M5 Prime	1.01	0.63	0.96
Greedy	1.01	0.63	0.96
T Test	1.01	0.63	0.96
Iterative T Test	1.01	0.63	0.96
Min Tolerance			
0.05	1.01	0.63	0.96
0.1	1.07	0.65	0.95
0.5	1.18	0.63	0.94
1	1.18	0.63	0.94
Ridge			
0.001	1.01	0.63	0.96
0.1	1.01	0.63	0.96
1	1.01	0.63	0.96
10	1.01	0.63	0.96

Tuning	RMSE	MAE	Corr.
			SVR
Default	1.07	0.60	0.95
Kernel Type			
RBF	1.09	0.66	0.95
Linear	1.07	0.60	0.95
Poly	2.45	1.81	0.75
Sigmoid	1.26	0.68	0.93
Gamma		Kernel: RBF	
0.001	1.00	0.59	0.96
0.01	0.92	0.55	0.96
0.1	0.97	0.56	0.96
0.5	1.41	0.67	0.92
C	Kernel: RBF, Gamma: 0.01		
0.01	1.41	0.73	0.92
0.1	0.97	0.57	0.96
1	0.92	0.55	0.96
5	0.91	0.54	0.96

Tuning	RMSE	MAE	Corr.
			SVM
Default	1.09	0.68	0.95
Kernel Type			
Dot	1.33	1.02	0.95
Radial	1.06	0.68	0.95
Polinomial	4.44	3.74	0.65
Anova	1.09	0.68	0.95
Gamma		Kernel: Radial	
0.001	1.06	0.68	0.95
0.01	1.01	0.67	0.96
0.1	1.14	0.87	0.95
0.5	1.17	0.87	0.94
1	1.12	0.81	0.95
C	Kernel: Radial, Gamma: 0.01		
0.001	3.34	2.38	0.93
0.01	2.48	1.77	0.93
0.1	1.21	0.80	0.939
0.5	1.07	0.68	0.95
1	1.06	0.68	0.953
5	1.05	0.71	0.956
10	1.23	0.91	0.96

Tuning	RMSE	MAE	Corr.
			GLM
Default	1.03	0.63	0.96
Family			
auto	1.03	0.63	0.96
gaussian	1.06	0.65	0.93
poisson	1.91	1.24	0.85
gamma	2.19	2.17	0.76
twedie	1.05	0.65	0.93
negative binomial	1.91	1.24	0.85
Lambda		family: Auto	
0.01	1.02	0.63	0.96
0.1	1.08	0.68	0.95
0.5	1.24	0.84	0.94
1	1.43	1.01	0.94
Alpha		family: Auto	
0	1.02	0.63	0.96
0.1	1.05	0.65	0.95
0.5	1.02	0.63	0.96
1	1.02	0.63	0.96

Note: Corr. : Correlation