Pamukkale Univ Muh Bilim Derg, 28(5), 737-747, 2022

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Evaluation of air temperature with machine learning regression methods using Seoul City meteorological data

Seul Şehri meteorolojik verileri kullanılarak makine öğrenmesi regresyon yöntemleri ile hava sıcaklığının değerlendirilmesi

Merve APAYDIN[1], *Mehmethan YUMUŞ*[1], *Ali DEĞİRMENCİ*[1], *Ömer KARAL*[1*]

[1]Department of Electrical and Electronics, School of Engineering and Natural Sciences, Ankara Yildirim Beyazit University, Turkey.
merveapydn@gmail.com, mehmethangumuss@gmail.com, adegirmenci@ybu.edu.tr, karal@ybu.edu.tr

Received/Geliş Tarihi: 25.08.2021 Revision/Düzeltme Tarihi: 17.02.2022 doi: 10.5505/pajes.2022.66915
Accepted/Kabul Tarihi: 14.03.2022 Research Article/Araştırma Makalesi

Abstract

Weather has a significant impact on human life and activities. As abrupt changes in air temperature negatively affect daily life and various industries, the importance of weather forecast accuracy is increasing day by day. Current weather forecasting methods can be divided into two main groups: numerical-based and machine learning-based approaches. Numerical-based weather forecasting methods use complex mathematical formulas that significantly increase the computational cost. On the other hand, machine learning-based methods have been preferred more in recent years due to their lower computational costs. In this study, the next day's maximum and minimum air temperature are estimated for Seoul, South Korea by using 12 different regression methods together with the boosting-based machine learning algorithms developed in recent years, as well as traditional machine learning methods. Furthermore, since tuning of hyperparameters affects the process time and performance of machine learning algorithms, all 12 methods have been extensively studied in terms of time and hyperparameters. The square correlation coefficient (R^2), which is frequently adopted in the literature, is used to compare the performances of the methods. According to the observed results, the boosting-based XGBoost and LightGBM methods are the most successful machine learning algorithms in predicting the maximum and minimum air temperature for all years with both statistical test analysis and the highest R^2 score.

Keywords: Machine learning, Daily weather forecasting, Regression method, Seoul.

Öz

Havanın insan yaşamı ve faaliyetleri üzerinde önemli bir etkisi vardır. Hava sıcaklığındaki ani değişimler günlük yaşamı ve çeşitli endüstrileri olumsuz etkilendiğinden hava tahmini doğruluğunun önemini günden güne artırmaktadır. Mevcut hava tahmin yöntemleri iki ana gruba ayrılabilir: sayısal tabanlı ve makine öğrenim tabanlı yaklaşımlar. Sayısal tabanlı hava tahmin yöntemleri, hesaplama maliyetini önemli ölçüde artıran karmaşık matematiksel formüller kullanır. Buna karşın, makine öğrenim tabanlı yöntemler ise düşük işlem maliyetleri nedeniyle son yıllarda daha çok tercih edilir. Bu çalışmada, geleneksel makine öğrenmesi yöntemlerinin yanı sıra son yıllarda geliştirilen yükseltme tabanlı makine öğrenmesi algoritmaları ile birlikte 12 farklı regresyon yöntemi kullanılarak Güney Kore Seul için bir sonraki günün maksimum ve minimum hava sıcaklığı tahmin edilmektedir. Ayrıca, hiperparametrelerin ayarlanması, makine öğrenmesi algoritmalarının işlem süresini ve performansını etkilediğinden, 12 yöntemin tümü zaman ve hiperparametreler açısından kapsamlı bir şekilde çalışılmıştır. Yöntemlerin performanslarının karşılaştırılmasında literatürde sıklıkla tercih edilen kare korelasyon katsayısı (R^2) kullanılmaktadır. Gözlemlenen sonuçlara göre, yükseltme tabanlı XGBoost ve LightGBM yöntemleri, hem istatistiksel test analizi hem de en yüksek R^2 puanı ile tüm yıllar için maksimum ve minimum hava sıcaklığını tahmin etmede en başarılı makine öğrenmesi algoritmalarıdır.

Anahtar kelimeler: Makine öğrenmesi, Günlük hava durumu tahmini, Regresyon yöntemi, Seul.

1 Introduction

Weather prediction is one of the popular research topics as it has a significant effect on people's daily lives as well as various industry sectors. Reliable and accurate weather information is needed in agriculture as temperature and humidity forecasting helps farmers take the necessary actions to increase their yields. In addition, weather data is constantly needed at the airport or in the marine system to alert them to sudden changes in climatic conditions. Moreover, in mining industries, weather information is generally needed to monitor the condition of the earth's crust. People need weather data to plan their future activities. As can be seen from the various examples above, from industry to agriculture, from daily commute to travel, we are heavily dependent on weather forecasts. Therefore, it is very important to predict the weather reliably and accurately.

The date of the weather forecast is based on the years before Christ. In 650 BC, Babylonians observed clouds to predict short-term weather conditions. In later years (340 BC), there were theories about how weather events occur in Aristotle's “Meteorologica” book. People have assumed these theories were correct for years and made their weather predictions based on them. However, as a result of the observations made with the invented measuring instruments after the 1600s, it has been observed that most of Aristotle's theories are wrong. Today, short- and long-term weather forecasts can be made using the data obtained from ground observations, ship observations, high atmosphere observations, satellite images and meteorology radars. Therefore, the complexity of the forecast is highly dependent on the nature of the weather data to be studied.

A wide variety of research has been carried out in recent years to reduce complexity and increase the efficiency of prediction

[*]Corresponding author/Yazışılan Yazar

models. Machine Learning (ML) techniques in weather forecasting have attracted a lot of attention recently, as they are insensitive to the multicollinearity of the input variables and therefore can deal with many input variables. However, it is a very important issue which ML technique should be used for weather forecasting and what the forecasting ability will be. This study analyzes 12 different ML methods, together with boosting-based machine learning algorithms developed in recent years, to predict the weather data of Seoul, South Korea, and provides an insight into the forecasting capability of each of these forecasting methods. Additionally, since tuning hyperparameters affects the processing time and performance of machine learning algorithms, all 12 methods are thoroughly examined in terms of time and hyperparameters. According to the experimental results, it is observed that the boosting based XGBoost and LightGBM methods are the most successful ML methods in terms of R^2 scores in predicting the maximum and minimum air temperature for all years.

This study is organized as follows. Section 2 provides an overview of machine learning models in weather forecasting, as well as the related works. In Section 3, the data set and regression techniques used are mentioned. Section 4 presents the metrics used to evaluate the performance of the models and their performance. Concluding remarks are presented in Section 5.

2 Literature Review

Existing weather forecasting approaches can be divided into two main categories: numerical-based and machine learning-based approaches. Numerical-based Weather Prediction (NWP) models based on physical relationships of parameters and mechanisms of atmospheric dynamics, uncertain physical parameterization and inaccurate initial/boundary conditions can cause model bias in air temperature prediction. Post-processing of the model output may be required to reduce bias and operational use of the models. This causes an increase in computational cost due to the use of complex mathematical formulas. On the other hand, ML-based methods have been preferred more in recent years due to their lower computational costs and insensitive to the multicollinearity of the input variables. Within the scope of the study, some new studies using ML-based methods for weather forecasting are summarized in Table 1.

Bushara and Abraham compared the success of 12 different ML methods such as, linear regression, regression bagging, multi-layer perceptron (MLP), IBk, Gaussian Processes, Kstar Additive, decision table, random subspace, regression by discretization, M5 rules, REPTree, M5P, and user classifier methods to determine the most accurate rainfall estimates using monthly meteorological data from 24 weather stations in Sudan between 2000 and 2012 [1]. The most accurate rainfall forecast was obtained with the IBk method with an average of 0.0905 mean absolute error (MAE).

Holmström et al. used linear and functional regression models to estimate the maximum and minimum temperatures for the next seven days with the data of the last two days from the weather data from the city of Stanford between 2011-2015 [2]. According to the root mean squared error (RMSE) metric, linear regression outperformed functional regression as a low-bias, high-variance model.

Saba et al. predicted whether the weather would be dry or rainy for the next two days using the individual Radial Basis Function and Multi Layer Perceptron (MLP) methods, as well as the hybrid model created by combining both [3]. The best performance is obtained by the hybrid model used (1^{st} day R^2 = 0.95, 2^{nd} day R^2 = 0.93).

Sharaff et al. analyzed the performance of linear regression, regression trees and artificial neural networks (ANN) methods on daily temperature using data collected from Mumbai Chhatrapati Shivaji Airport between 2001-2016 [4]. The results are compared using the mean squared error (MSE) metric, and the ANN (MSE = 3.46) method shows the best performance.

Table 1. Summary of the literature review.

Similar Studies	City/Country	Information of Data	Prominent Method	Performance
[1]	Sudan	2000-2012 years from 24 Weather Stations	IBk	MAE = 0.0905
[2]	Stanford	2011-2015 years	Lineer Regression	RMSE = 5.039-5.642
[3]	Saudi Arabia	5 years	RBF and MLP Hybrid Model	1^{st} day R^2 = 0.95, 2^{nd} day R^2 = 0.93
[4]	Mumbai Chhatrapati Shivaji Airport	2001-2016 years	ANN	MSE = 3.46
[5]	London's Integrated Data Archive System	2006 and 2017 from 29 weather stations	GBM	R^2 = 0.68 MAE = 1.60
[6]	Brazilian National Meteorological Institute	11.7 ± 1.34 years from 53 weather stations	XGBoost	R^2 = 0.87, RMSE = 0.56
[7]	Sea Surface Temperature Using NASA's Aqua Satellite Data	2002 to present at 4km horizontal resolution	GAM	MAE = 0.41, MAPE = 1.52
[8]	10 Years Weather Data from South Korea	2009 to 2018 years from 96 ASOS stations	CNN	MAE = 0.029 (Seoul), MAE = 0.019 (Daegwalleong), MAE = 0.025 (Seongsan)
[9]	Tennessee Region	9 weeks from 10 different cities	RF and Extra Tree Algorithm	RMSE Difference = 3.0
[10]	Antalya	2000-2016 years	ANN	R^2 = 0.99
[11]	Bingöl	2014-2020 years	LSTM and ARIMA	R^2 = 0.95 (LSTM), R^2 = 0.97 (ARIMA)
[12]	South Korea	2013-2017 years	MME	T_{min} R^2 = 0.90 (2016), T_{max} R^2 = 0.87 (2016)
Our Study	South Korea	2013-2017 years	XGBoost	T_{min} R^2 = 0.96 (2016), T_{max} R^2 = 0.97 (2016)

Santos et al. estimated air temperature using linear regression, random forest (RF), decision tree (DT), gradient boosting machine (GBM), ANN, and support vector regression (SVR) methods on daily maximum air temperature records obtained from London's Integrated Data Archive System [5]. Among these methods, the most successful result is obtained with the GBM (square correlation coefficient (R^2) = 0.68, MAE = 1.60, RMSE = 2.03).

Ferreira et al. estimated the water demand in nature based on weather reports, using hourly data collected from 53 automatic weather stations of the Brazilian National Meteorological Institute [6]. The performance of RF, extreme gradient boosting regression (XGBoost), ANN, and CNN are compared. The most successful method in daily estimation is XGBoost (R^2 = 0.87, Nash-Sutcliffe efficiency coefficient (NSE) = 0.79, RMSE = 0.56), while the most successful method in hourly estimation is CNN (R^2 = 0.91, NSE = 0.87 and RMSE = 0.45).

Wolff et al. estimated sea surface temperature using NASA's Aqua Satellite data using generalized additive model (GAM), RF, XGBoost, MLP, LSTM ML methods, and compared their results with the Physics-based European Center for Medium-Range Weather Forecasts (ECMWF) [7]. The closest result to the physics-based ECMWF is obtained from the GAM method with MAE = 0.41, mean absolute percent error (MAPE) =1.52.

Lee et al. compared the success of MLP, convolutional neural network (CNN) and long short-term memory (LSTM) methods on a data set collected by weather measurement over 10 years from three automated surface observing system stations in South Korea, metropolitan-Seoul, mountainous-Daegwalleong and coastal-Seongsan regions [8]. From the experimental results, the CNN algorithm using hourly input data with appropriate time length showed the best performance, depending on the target region.

Jakaria et al. estimated the next day's temperature by applying 5 different regression methods such as Ridge, SVR, MLP, RF, and Extra Tree (ET) Regressor on real weather data obtained from 10 different cities in the Tennessee region [9]. It has been observed that RF and ET regression methods give the best results in the performance evaluation made by considering the RMSE differences in the estimations of one city and 10 cities.

Akyüz et al. estimated the average monthly air pressure using ANN method on the real monthly average vapor pressure, monthly average relative humidity, related month and year data of Antalya between 2000 and 2016 [10]. According to the experimental results, it has been observed that the values predicted by the ANN model (R^2 = 0.99) are quite consistent with the real average air temperature values.

Sevinç and Kaya estimated the monthly average temperature using LSTM and ARIMA models on the meteorological data of Solhan district of Bingöl province, located in the Eastern Anatolia Region of Turkey, between 2014 and 2020 [11]. The R square score of the selected model was calculated as 0.95 in the LSTM model and 0.97 in the ARIMA model.

Cho et al. compared four different ML-based methods which are RF, SVR, ANN, and multi-model ensemble (MME), along with the LDAPS (Local Data Assimilation and Prediction System; a local NWP model over Korea) method to predict the next day's minimum (T_{min}) and maximum (T_{max}) air temperature in data obtained from Seoul, South Korea [12]. MME method is the most successful method for the years 2015 (T_{min} R^2 = 0.80, T_{max} R^2 = 0.68), 2016 (T_{min} R^2 = 0.90, T_{max} R^2 = 0.87) and 2017 (T_{min} R^2= 0.89, T_{max} R^2 = 0.74) in the data set.

In this study, performance analysis and comparison of 12 different ML-based regression methods for air temperature prediction are made using the same data set in [12]. Unlike the study in [12], the efficiency of boosting-based ML algorithms developed in recent years has also been investigated.

3 Material and methods

3.1 Data set

The data set named "Bias Correction of Numerical Prediction Model Temperature Forecast" used in the study is accessed from the UCI (University of California, Irvine) Machine Learning Repository website [12],[13]. Data are collected from 25 stations in Seoul, South Korea, in June, July and August between 2013 and 2017. As there is not enough space in the paper, only the graphical representation of the air temperature data (in terms of maximum and minimum temperature values) for 2017 in the form of time series is shown in Figure 1.

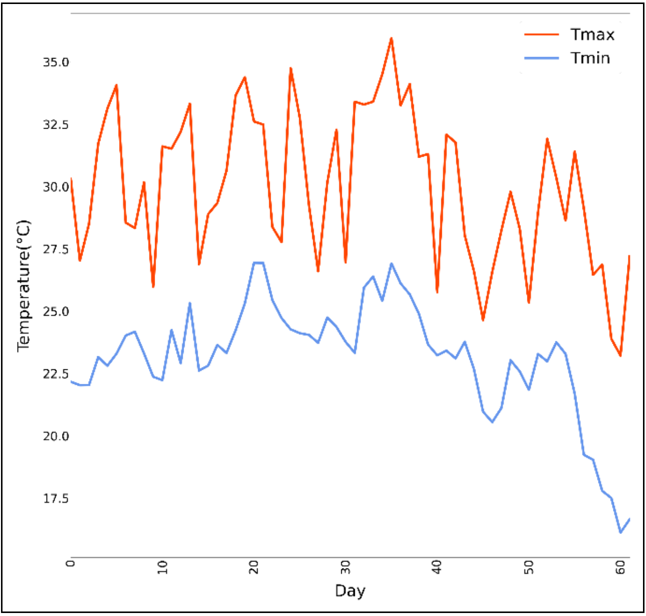


Figure 1. Maximum and minimum temperatures for 2017.

Since the data set is manually divided into five separate sets by years, the date feature is not taken into account in the study. In addition, it is observed that the number of stations, latitude and longitude features has almost no effect on the performance of the model. For these reasons, the four features mentioned above (date, station number, latitude, and longitude) are removed in the training of the models. Details of the remaining 21 features are shown in Table 2. These features consist of the location and time information of the stations as well as the forecast values of the LDAPS model. The target features are the minimum and maximum temperature values for the next day.

3.2 Regression techniques

Twelve different ML-based regression methods, such as ridge, lasso, elastic net, SGD, KNN, DT, RF, ANN, SVR, AdaBoost, XGBoost and LightGBM, used in the study are explained in the following subsections, respectively.

Table 2. Details of data set features.

Feature Name	Max.	Min.	Mean	Std.
Maximum air temperature	37.6	20.0	29.77	2.97
Minimum air temperature	29.9	11.3	23.23	2.41
LDAPS model next-day minimum relative humidity	98.52	19.79	56.76	14.67
LDAPS model forecast of next-day maximum relative humidity	100.00	58.94	88.37	7.19
LDAPS model forecast of next-day maximum air temperature applied lapse rate	38.54	17.62	29.61	2.95
LDAPS model forecast of next-day minimum air temperature applied lapse rate	29.62	14.27	23.51	2.35
LDAPS model forecast of next-day average wind speed (m/s)	21.86	2.88	7.09	2.18
LDAPS model forecast of next-day average latent heat flux (W/m^2)	213.41	-13.60	62.51	33.73
LDAPS model forecast of next-day 1st 6-hour split average cloud cover (0-5 h) (%)	0.97	0	0.37	0.26
LDAPS model forecast of next-day 2nd 6-hour split average cloud cover (6-11 h) (%)	0.97	0	0.36	0.26
LDAPS model forecast of next-day 3rd 6-hour split average cloud cover (12-17 h) (%)	0.98	0	0.32	0.25
LDAPS model forecast of next-day 4th 6-hour split average cloud cover (18-23 h) (%)	0.97	0	0.29	0.25
LDAPS model forecast of next-day 1st 6-hour split average precipitation (0-5 h) (%)	23.70	0	0.59	1.95
LDAPS model forecast of next-day 2nd 6-hour split average precipitation (6-11 h) (%)	21.62	0	0.49	1.76
LDAPS model forecast of next-day 3rd 6-hour split average precipitation (12-17 h) (%)	15.84	0	0.28	1.16
LDAPS model forecast of next-day 4th 6-hour split average precipitation (18-23 h) (%)	16.65	0	0.27	1.21
Elevation (m)	212.34	12.37	61.87	54.28
Slope (Â°)	5.18	0.09	1.26	1.37
Daily incoming solar radiation (wh/m^2)	5992.89	4329.52	5341.50	429.13
The next-day maximum air temperature (Â°C)	38.9	17.4	30.27	3.12
The next-day minimum air temperature (Â°C)	29.8	11.3	22.93	2.49

3.2.1 Ridge regression

The cost equation for ridge regression is as in Equation (1).

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{i=1}^{P}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{P}\beta_j^2 \tag{1}$$

where, y represents the dependent variables, x is the independent variables, λ controls the penalty amount, β is the size of the coefficients, n is the number of samples, and p is the number of features [14].

3.2.2 Lasso regression

Lasso regression is similar to the ridge regression model. The difference with ridge regression is that it takes the absolute value of the feature coefficients instead of squares. In this way, the data set becomes simpler, and the complexity of the model is reduced [15].

The cost equation for lasso regression is given in Equation (2).

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{i=1}^{P}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{P}\left|\beta_j\right| \tag{2}$$

3.2.3 Elastic net regression

Elastic net method combines the desired properties of ridge and lasso regression methods, which are de-correlation and feature selection, respectively [16]. Cost equation of the elastic net is expressed as in Equation (3).

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda_1\sum_{j=1}^{P}\beta_j^2 + \lambda_2\sum_{j=1}^{P}\left|\beta_j\right| \tag{3}$$

where, λ_1 and λ_2 are tuning parameters.

3.2.4 Stochastic gradient descent (SGD) regression

The SGD regression creates a model by minimizing the gradient of the objective function. The algorithm uses the gradient descent technique in the training process. Initially, a random starting point is determined from the training data. The algorithm then penalizes the model in the search process and finds the lowest point of cost. The regularization term is used in the punishment process, as in the ridge regression. Penalization is applied by adding bias to the cost function of the model until the optimum result is obtained. In this way, the constructed model finds the appropriate point between generalization and overfitting.

SGD regression selects only one random training sample at a time to evaluate gradients. Then it calculates the gradient for the cluster it chooses [17]. With this technique, it works much faster than the gradient descent algorithm in large data sets [18]. The general expression of the SGD regression algorithm is as shown in Equation (4).

$$x_{t+1} = x_t - n_t \nabla f_{t(x_t)} \tag{4}$$

where, n_t is a learning rate, x_t is vector at time step t, f_t is a loss function and $\nabla f_{t(x_t)}$ is gradient information.

3.2.5 K nearest neighbors (KNN)

KNN is initially developed for the classification, but it can also be used in the regression problems. In the KNN regression, the target variable is predicted by averaging the values of the nearest neighboring samples and it is defined as in Equation (5).

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i(x) \tag{5}$$

where, y_i is the value of the i^{th} nearest neighbor of the query instance x and k is the number of nearest neighbors. The main factor affecting the performance of the KNN method is the determination of the number of k nearest neighborhoods. If the k is set to lower values, the probability of the model being overfit will be very high. On the other hand, when larger values of k are selected, it results in underfitting and the performance of the model decreases because of outliers on the prediction. Considering those mentioned above, the k needs to be adjusted adaptively for different data sets [19], [20].

To find the optimal k, the errors of k values in a given range are computed sequentially. Then, the k value with the lowest error value is selected and the estimation is made.

3.2.6 Decision tree regression (DT)

DT is a tree-based ML algorithm with a top-down layer approach. DT structure consists of a root node, internal nodes (branches) and terminal nodes (leaves) [21]. This structure starts from the root node, continues with branching, and ends at the leaves. Branches are made according to the error criteria determined by the user. The feature that best divides the data set is chosen as the root node. Then the branching continues in this way until the error reaches an acceptable level. The parts of the leaves where the branching ends are used to determine the result of the prediction. Subsequently, the estimation of the new sample is computed by the average values of the samples in the corresponding leaf.

3.2.7 Random forest (RF)

In the RF method, multiple DTs working independently and in parallel are created by using the bagging technique. With multiple tree models, it is tried to obtain maximum efficiency from each sample of the data set [22]. Additionally, the method is aimed to give more robust results thanks to the bagging technique.

The predicted value in RF is determined by computing the average of created tree models. The estimation result in the RF model consisting of N decision trees is computed as in Equation (6) by taking the average of the results obtained after each DTs $\{T(x)\}_1^N$ is grown.

$$\hat{f}_{rf}^{N} = \frac{1}{N}\sum_{n=1}^{N} T(x) \tag{6}$$

3.2.8 Artificial neural networks (ANN)

ANN consists of three basic layers, which are input layer, hidden layer, and output layer. Layers contain basic structures called artificial neurons, and artificial neurons connect these layers by weight. Weight adjustment is important to increase the reliability of the model. Although different methods are suggested for weight adjustment, back propagation is the commonly used algorithm for training feed forward neural networks [23]. The regression calculation of ANN is given in Equation (7).

$$y_t = \alpha_0 + \sum_{j=1}^{n} \alpha_j f\left(\sum_{i=1}^{m} \beta_{ij} y_{t-i} + \beta_{0j}\right) + \varepsilon_t \tag{7}$$

where, m is the number of input nodes, n is the number of hidden nodes, *f(.)* is the activation function, α_0 and β_{0j} are the weights of the arcs whose values are always equal to 1 and originating from the bias terms.

3.2.9 Support vector regression (SVR)

Support vector machines (SVM) is based on statistical learning theory and is used in both classification and regression (SVR) applications [24]-[26]. The success of SVR in application is largely dependent on the choice of the proper kernel function. Commonly used kernel functions are linear, polynomial, sigmoid, and RBF.. The most preferred kernel function is RBF due to its easy design, good generalization ability and strong tolerance to noise.

SVR makes adjustments to fit the best curve to the data set. To train this model, the optimization problem in Equation (8) needs to be solved according to the constraints given in Equation (9)-(11). While making this correction, the error term specified as the maximum error and indicated by the epsilon (ε) value is computed. Then, constraints that are less than or equal to a certain margin are obtained. Regularization parameter (C) plays a significant role in the performance of the SVR algorithm. It controls the penalty applied to observations outside the specified epsilon margins and allows adjustment between overfitting and generalization.

$$min\left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right) \tag{8}$$

subject to

$$y_i - w^T\phi(x_i) - b \le \varepsilon + \xi_i, i = 1, ..., l \tag{9}$$

$$-y_i + w^T\phi(x_i) + b \le \varepsilon + \xi_i^*, i = 1, ..., l \tag{10}$$

$$\xi_i, \xi_i^* \ge 0, i = 1, ..., l \tag{11}$$

where, $\boldsymbol{w}$ controls the smoothness of the model, ξ_i and ξ_i^* are slack variables, $\phi(\boldsymbol{x})$ is a function of projection of the input space to the feature space, b is a parameter of bias, y_i is the output value to be estimated.

3.2.10 Adaptive boosting regression (AdaBoost)

AdaBoost algorithm searches the sample feature space iteratively and finds training feature weights. In the iterative process, the feature weights of the training samples are constantly adjusted [27]. The most basic feature of this algorithm is to create the most successful model by taking the good sides of the weak models.

In AdaBoost, multiple models are created, and the strengths of each model is combined. The algorithm then tries to obtain a successful model from these models. The AdaBoost algorithm combines the weak models and produces the output for the most successful model as in Equation (13).

$$h: x \rightarrow \{-1,1\} \tag{12}$$

$$h_f(x) = \inf\{y \in Y: \sum_{t: h_{t(x)} \le y} \log(1/\alpha_t) \ge \frac{1}{2}\sum_t \log(1/\alpha_t)\} \tag{13}$$

where, $h_{t(x)}$ represents weak learners and α_t is model's weight.

3.2.11 Extreme gradient boosting regression (XGBoost)

The XGBoost algorithm uses the gradient enhancement technique. It firstly makes predictions by creating many different models. Then, based on the remains of these models, it creates the main model. A new tree is added in each iteration and the learning process takes place by reflecting the model's errors as punishment Equation (14) [28].

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{14}$$

where, $\hat{y}_i^{(t)}$ represents the prediction result at time *t*.

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(yi, \hat{y}i^{(t-1)}) \tag{15}$$

$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(yi, \hat{y}i^{(t-1)}) \tag{16}$$

where, g_i and h_i are calculated using the loss function *l*.

In the XGBoost algorithm, the information gain of the objective function after each split is obtained as in Equation (17).

$$G = \frac{1}{2}\left[\frac{(\Sigma_{i\in I_L} g_i)^2}{\Sigma_{i\in I_L} h_i + \lambda} + \frac{(\Sigma_{i\in I_R} g_i)^2}{\Sigma_{i\in I_R} h_i + \lambda} + \frac{(\Sigma_{i\in I} g_i)^2}{\Sigma_{i\in I} h_i + \lambda}\right] - \gamma \tag{17}$$

where, g_i and h_i are the first and second order gradient statistics of the loss function, I_L and I_R are the sample sets of the left and right branches, γ is a splitting threshold. Leaf node splits are not allowed when the information gain is less than γ.

3.2.12 Light gradient boosting machine (LightGBM)

LightGBM is a gradient boosting decision tree (GBDT) algorithm developed by Microsoft in 2017 [29]. The LightGBM method uses two techniques. The first is the gradient-based one-side sampling (GOSS) method. It randomly selects samples with small gradients and gives them constant weight while holding samples with large gradients. In this way, the model focuses on more trained samples without changing the data distribution. The second is exclusive feature bundling (EFB). With the EFB technique, histograms are extracted from the relationships between features in high dimensional data sets. Therefore, the complexity of the model to be created with these histograms is reduced. The objective function of the LightGBM method is given in Equation (18).

$$G = \frac{1}{2}\left[\frac{(\Sigma_{i\in I_L} g_i)^2}{\Sigma_{i\in I_L} h_i + \lambda} + \frac{(\Sigma_{i\in I_R} g_i)^2}{\Sigma_{i\in I_R} h_i + \lambda} - \frac{(\Sigma_{i\in I} g_i)^2}{\Sigma_{i\in I} h_i + \lambda}\right] \tag{18}$$

4 Result and discussion

4.1 Evaluation criteria

The R^2 metric, which is widely used in the literature [30], is used to evaluate the performance of benchmark regression methods and is defined as:

$$R^2 = 1 - \frac{\Sigma(y_i - \hat{y})^2}{\Sigma(y_i - \bar{y})^2} \tag{19}$$

where, y_i is original value, $\bar{y}$ is mean value, $\hat{y}$ is predicted value, and N is the number of observations.

K-fold cross validation is a resampling technique used to evaluate ML-based regression methods' performances independently from the data set. The number of k should be chosen carefully, as poor selection may result in high bias or high variance. In the literature, k = 5 or k = 10 values are frequently used, it has been experimentally observed that these values do not cause very high variance or very high bias [31].

4.2 Experimental results

The data set is divided into 5 parts according to years (2013-2017) and the missing values in each data set are filled with the mean imputation. Maximum and minimum air temperature of the next day is estimated with ridge, lasso, elastic net, SGD, KNN, DT, RF, ANN, SVR, AdaBoost, XGBoost, and LightGBM methods. As is known, the performance of benchmarking methods depends on the setting of the hyperparameters which differs based on the regression method. Therefore, a grid search technique is used to adjust the hyperparameters. Parameters of each comparison method and their value ranges are given in Table 3. In addition, k = 10 is chosen in the k-fold cross validation technique and it is repeated 10 times for each k value to increase reliability.

The maximum and minimum temperature prediction obtained from 12 different ML algorithms for the years 2013-2017 along with the real temperature values are given in Figure 2 comparatively. To visualize the algorithm predictions, the data set is randomly divided 90/10 into the training and test data. As seen in Figure 2 (a-j), the regression curves most compatible with the real values are obtained by XGBoost and LightGBM methods, while the least compatible curves are obtained by KNN and SGD methods.

Table 4 shows the R^2 results for T_{max} and T_{min} estimation on meteorological data between 2013 and 2017 of 12 different ML-based regression methods compared. As can be seen from the Table 4, the XGBoost and LightGBM methods show the highest performance with the highest R^2 scores for T_{max} and T_{min} in all years. As seen in the 2013 and 2017 columns of Table 4, XGBoost and LightGBM methods show the same performance in estimating T_{max} and T_{min} values. For 2013, the R^2 scores for LightGBM and XGBoost T_{max} and T_{min} are 0.93 and 0.91, respectively, while for 2017 they are 0.94 and 0.93. In other years, the performance of both methods is almost the same. As can be seen in 11th and 12th rows of Table 4, XGBoost and LightGBM methods are very close to each other and most successful in estimating T_{max} and T_{min} values for 2014, 2015, and 2016 years. For 2014, LightGBM's T_{max} and T_{min} R^2 scores are 0.91 and 0.94, respectively, while XGBoost has 0.91 and 0.93. Similarly, LightGBM's T_{max} and T_{min} R^2 scores for 2015 are 0.93 and 0.92, respectively, while 0.92 and 0.91 for XGBoost. Finally, for 2016, LightGBM's T_{max} and T_{min} R^2 scores are 0.95 and 0.95, respectively, while XGBoost has 0.96 and 0.97.

There is only 0.01 difference between the R^2 scores. On the other hand, as can be seen from 7th and 8th rows of Table 4, ANN and RF methods perform quite similarly to XGBoost and LightGBM methods in estimating T_{max} and T_{min} values. For example, the R^2 scores of the RF method (T_{max}: 0.91, T_{min}: 0.91) and the ANN method (T_{max}: 0.92, T_{min}: 0.91) for 2017, are very close to the R^2 scores of XGBoost and LightGBM methods (T_{max}: 0.94, T_{min}: 0.93).

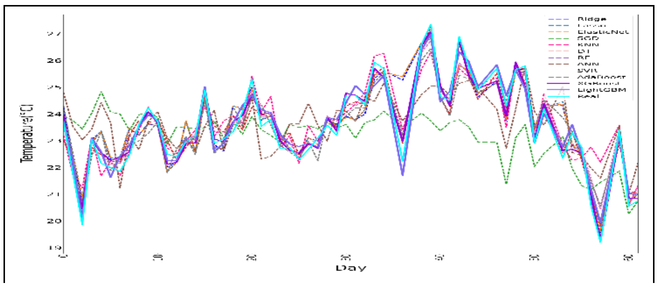
Considering the R^2 values of T_{max} and T_{min} of the comparison methods in Table 4 and Figure 2 for the years 2013-2017, the KNN method gives the worst T_{max} performance for all years except 2013 and the worst T_{min} performance for all years. The ridge, lasso, and elastic net methods show approximately the same R^2 performance for T_{max} and T_{min} in all years.

Table 3. Regression techniques parameters and range of values used in the study.

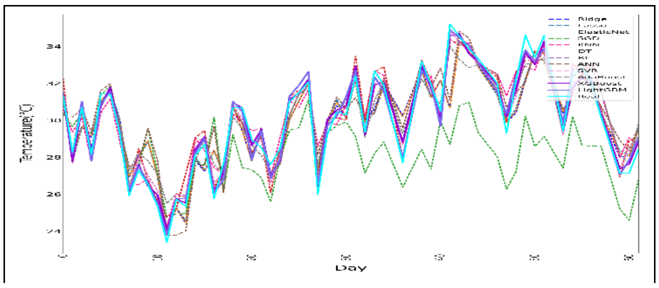
Regression Techniques	Parameters	Value
Ridge	alpha	0,1,2,3...1000
Lasso	alpha	0,1,2,3...1000
	max_iter	100,1000,10000,100000
Elastic net	alpha	0,1,2,3...1000
	max_iter	100,1000,10000,100000
SGD	loss	“squared_loss”, “huber”, ”epsilon_insensitive”, ‘squared_epsilon_insensitive’
	penalty	“l1”, “l2”, “elasticnet”
	alpha	0,1,2,3...1000
	max_iter	100,1000,10000,100000
KNN	n_neighbors	1,2,3,...,30
	weights	“uniform”, “distance”
	algorithm	“auto”, “ball_tree”, “kd_tree”, “brute"
DT	max_lelaf_nodes	10,20,30,40,60
	max_depth	2,3,4,5,10,20
	min_samples_split	2,5,10,20,30
	min_samples_leaf	1,5,10,20,25
RF	max_depth	10,20,30,50,90
	max_features	“auto”, “sqrt”
	n_estimators	200,500,800,1000,1400,1800,2000
	min_samples_split	2,5
	min_samples_leaf	1,2
ANN	hidden_layer_sizes	(30,60), (40,60), (20,40)
	alpha	0,0001, 0.001, 0.01, 0.02, 0.1
SVR	cost	$2^{-4}, 2^{-3}, 2^{-2},..., 2^{14}$
	gamma	$2^{-10}, 2^{-9}, 2^{-8},..., 2^{5}$
	epsilon	0.01,0.02,....,1
AdaBoost	learning_rate	0.001, 0.01, 0.1, 0.5
	n_estimators	100, 200, 500, 1000
	loss	“linear”, “square”, “exponential”
XGBoost	learning_rate	0.001, 0.01, 0.1, 0.5
	max_depth	2,3,5,8,10
	n_estimators	100,200,500,1000
	colsample_bytree	0.4, 0.7, 0.8, 1
LightGBM	learning_rate	0.001, 0.01, 0.1, 0.5
	max_depth	3,5,8,10
	n_estimators	100,200,500,1000
	colsample_bytree	0.4, 0.7, 0.8, 1

Table 4. R^2 results for 2013-2017.

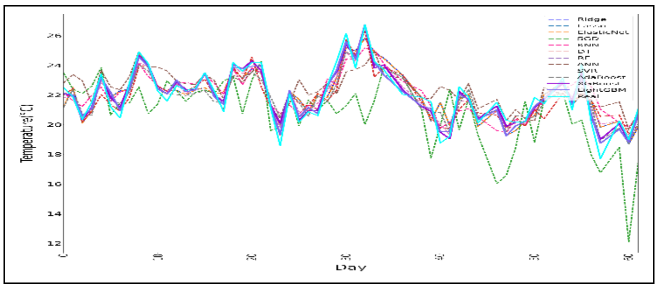
	2013		2014		2015		2016		2017	
Method	T_{min}	T_{max}	T_{min}	T_{max}	T_{min}	T_{max}	T_{min}	T_{max}	T_{min}	T_{max}
Ridge	0.77	0.76	0.70	0.77	0.81	0.76	0.88	0.88	0.87	0.75
Lasso	0.76	0.76	0.70	0.77	0.81	0.76	0.88	0.88	0.85	0.72
Elastic net	0.76	0.76	0.69	0.77	0.81	0.76	0.88	0.88	0.86	0.72
SGD	0.72	0.74	0.62	0.67	0.80	0.75	0.88	0.89	0.85	0.75
KNN	0.67	0.79	0.62	0.64	0.72	0.64	0.83	0.84	0.81	0.66
DT	0.74	0.84	0.70	0.81	0.73	0.68	0.88	0.90	0.81	0.70
RF	0.88	0.92	0.85	0.91	0.89	0.89	0.94	0.96	0.91	0.91
ANN	0.87	0.91	0.87	0.93	0.90	0.89	0.94	0.95	0.92	0.91
SVR	0.75	0.78	0.86	0.92	0.75	0.77	0.89	0.89	0.86	0.76
AdaBoost	0.78	0.83	0.73	0.84	0.82	0.80	0.87	0.92	0.86	0.80
XGBoost	0.91	0.93	0.91	0.93	0.92	0.91	0.96	0.97	0.94	0.93
LightGBM	0.91	0.93	0.91	0.94	0.93	0.92	0.95	0.95	0.94	0.93



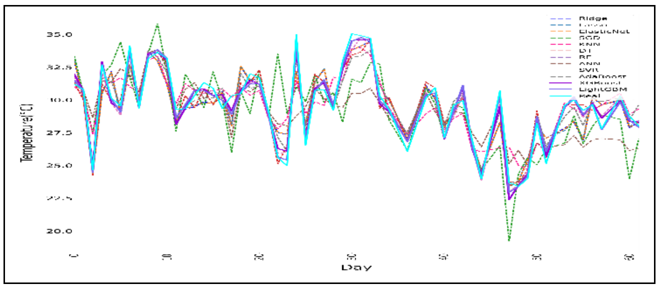
(a)



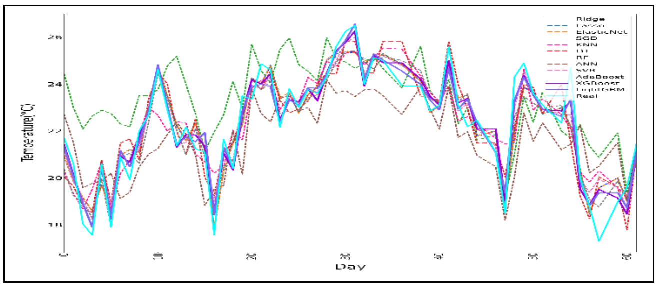
(b)



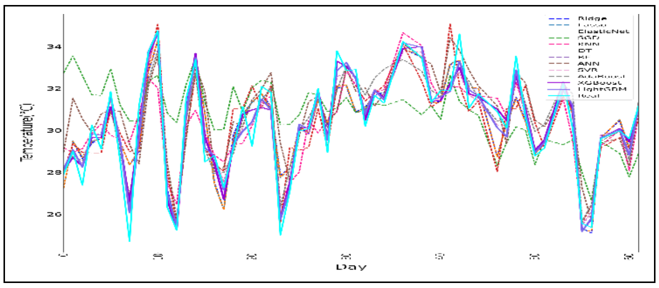
(c)



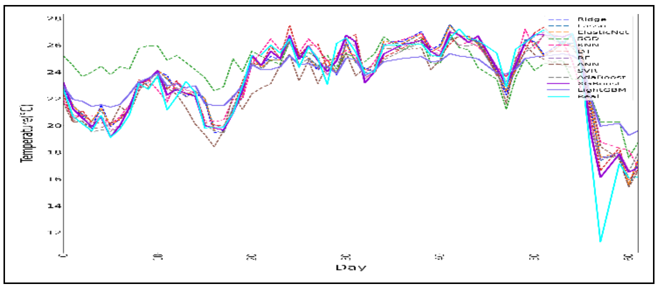
(d)



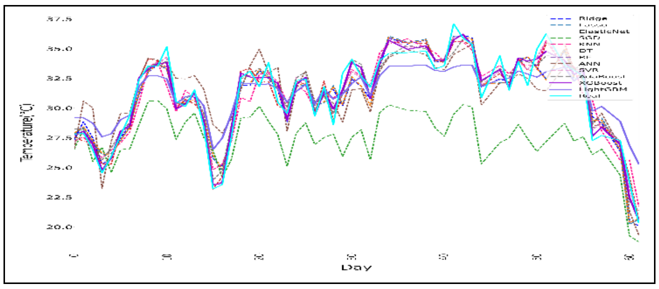
(e)



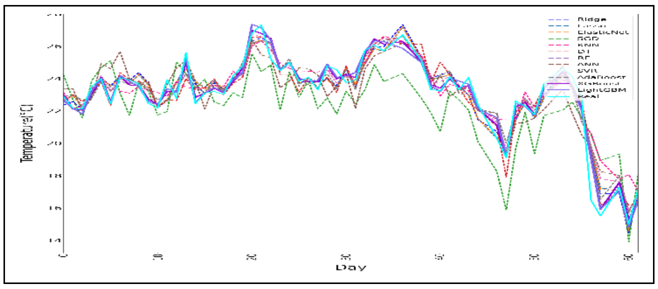
(f)



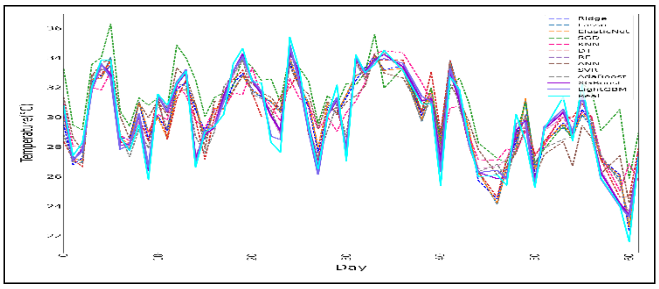
(g)



(h)



(i)



(j)

Note: MinTP and MaxTP denote mininimum temperature prediction and maximum temperature prediction respectively.

Figure 2. Minimum and maximum temperature prediction for 2013-2017. (a): MinTP for 2013, (b): MaxTP for 2013, (c): MinTP for 2014, (d): MaxTP for 2014, (e): MinTP for 2015, (f): MaxTP for 2015, (g): MinTP for 2016, (h): MaxTP for 2016, (i): MinTP for 2017, (j): MaxTP for 2017

4.3 Statistical test analysis

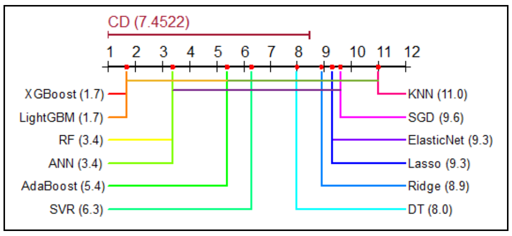
Statistical tests are applied to determine the significance of the outputs of machine learning algorithms in all data sets. Friedman test is a widely used nonparametric test compares the average ranks of the benchmarking models. In the Friedman test, which is a two-way statistical evaluation of rank variance, it is stated that the performance of all ML algorithms in the null hypothesis (H_0) is the same in terms of the given measure (such as R^2), and at least one approach is significantly different in the alternative hypothesis (H_A). In this study, the average ranking of the ML models based on the Friedman's aligned rank test (from the best 1 to the worst 12) across all datasets. Then, Friedman test with the Iman–Davenport correction is used test the null hypothesis [32]. Statistical test results on R^2 indicator is given in Table 5.

Table 5. Results of Friedman tests (12 benchmarking methods on 5 years).

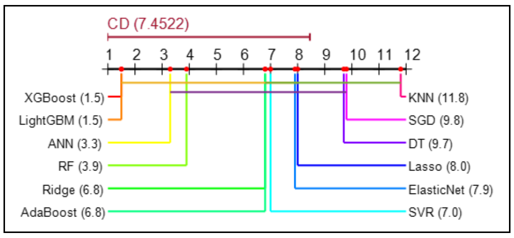
	Critical Value	p Value	F_F Value	Null Hypothesis
T_{min}	2.014	1.53568E-06	21.8355	Reject
T_{max}	2.014	2.13803E-06	20.9871	Reject

According to the test results in Table 5, the Friedman test rejects the null hypothesis for both T_{min} and T_{max}, as both p values are less than the significance level (α=0.05) and F_Fs are greater than the critical value. Hence, the null hypothesis of the Friedman is rejected. This demonstrates that there is a significant difference between the compared models.

The Nemenyi post hoc test, applied after the null hypothesis of the Friedman test is rejected, allows pairwise comparisons between ML models. In Nemenyi test, if the difference between the average ranks of the models is greater than critical difference (CD), the performance of the compared models is significantly different. For 12 different ML models over 5 years of data at a significance level of α=0.05, CD=7.4522. The CD diagram of the Nemenyi test is demonstrated in Figure 3 for both T_{max} and T_{min}. In Figure 3(a) and (b), groups of models that are not significantly different (at p = 0.05) are connected.



(a)



(b)

Figure 3. CD diagram of the average performance rankings. (a): T_{max}, (b): T_{min}.

The diagram shows that the 12 ML models are divided into two categories in terms of average ranking. With regard to T_{max}, the category of best-performing methods consists of XGBoost, LightGBM, RF, ANN, AdaBoost, and SVR. The other category contains DT, Ridge, Lasso, ElasticNet, SGD, and KNN. With respect to T_{min}, the category of best-performing methods consists of XGBoost, LightGBM, ANN, RF, Ridge, and AdaBoost. The other category contains SVR, ElasticNet, Lasso, DT, SGD, and KNN.

4.4 Process time analysis

In some applications, the processing time is as important as the accuracy of the methods because they may require faster response time. For example, in constantly changing training data such as weather conditions, execution time gains importance in training and estimating of the model. For this reason, the processing times of the benchmarking methods are also analyzed.

Table 6 shows the grid search performed to determine the optimum parameters of the ML-based regression methods employed in the study and the estimation time in seconds after the parameters are specified. All experiments are performed on a computer with Intel core i7@2.6 GHz CPU, 16 GB RAM, NVIDIA GeForce GTX 960M GPU and Windows 10 operating system.

Table 6. Time (sec) for grid search and estimation.

Method	Grid Search Time	Estimation Time
Ridge	4.7	0.003
Lasso	22.48	0.004
Elastic Net	21.68	0.004
SGD	3.57	0.006
KNN	4.38	0.004
DT	10.82	0.013
RF	2128.29	1.884
ANN	21.82	1.552
SVR	1951.06	0.159
AdaBoost	110.70	1.073
XGBoost	235.46	0.853
LightGBM	169.18	0.279

As can be seen from Table 6, the KNN method, which gives the worst R^2 result for T_{max} and T_{min}, requires almost the least grid search and estimation time. On the other hand, the grid search and estimation times of XGBoost and LightGBM methods, which yield the best R^2 results, are quite different from each other.

While the time spent to determine the optimal parameters of the XGBoost algorithm is 235 seconds, this time is only 169 seconds for the LightGBM method. Furthermore, XGBoost's prediction time is 0.8 seconds, about 4 times that of LightGBM (0.2 seconds). The ridge, lasso, elastic net, and SGD methods in Table 6 provide very fast and fairly close estimates, but not a high success rate. When the times of the bagging and boosting algorithms (RF, AdaBoost, XGBoost, and LightGBM) are compared, it is clear that LightGBM is the fastest method among them.

4.5 Discussion

The state-of-the-art study of this article is "Comparative Assessment of Various Machine Learning-Based Bias Correction Methods for Numerical Weather Prediction Model Forecasts of Extreme Air Temperatures in Urban Areas" [12].

The results of machine learning-based RF, SVR, ANN, and Multi Model Ensemble (MME) algorithms are compared. Achievements by years are evaluated with R^2 and RMSE performance metrics. The maximum air temperature prediction successes (in R^2) of each method for 2015, 2016, and 2017 are as follows, respectively: 0.62, 0.85, and 0.70 for RF; 0.67, 0.87, and 0.74 for SVR; 0.68, 0.85, and 0.74 for ANN; and 0.68, 0.87, and 0.74 for MME. On the other hand, the minimum air temperature prediction successes (in R^2) of each method by years are as follows, respectively: 0.79, 0.89, and 0.89 for RF; 0.80, 0.89, and 0.87 for SVR; 0.78, 0.89, and 0.88 for ANN; and 0.80, 0.90, and 0.89 for MME [12].

In our study, 12 different machine learning methods are used. These methods are ridge, lasso, elastic net, SGD, KNN, DT, RF, ANN, SVR, AdaBoost, XGBoost, and LightGBM. Our difference with the compared article [12] is that besides traditional machine learning methods, boosting algorithms, which have become popular today, are also used. In addition, k-fold cross validation was used validation of our study, and the results were obtained by averaging the values calculated as a result of too many repetitions. Finally, while the study [12] compared ML methods for only 3 years (2015, 2016, and 2017), we do so for 5 years (2013, 2014, 2015, 2016, and 2017).

When the study [12] is compared with this study (for 2015, 2016, and 2017), it is clearly seen that the performance of the ML-based methods used in this study is higher. The R^2 results of the methods used in our study for the estimation of the minimum air temperature for the years 2015, 2016, and 2017 are as follows, respectively: 0.89, 0.94, and 0.91 for RF; 0.75, 0.89, and 0.86 for SVR; 0.90, 0.94, and 0.92 for ANN; 0.92, 0.96, and 0.94 for XGBoost; 0.93, 0.95, and 0.94 for LightGBM.

Furthermore, the maximum air temperature prediction successes (in R^2) of each method by years are as follows, respectively: 0.89, 0.96, and 0.91 for RF; 0.77, 0.89, and 0.76 for SVR; 0.89, 0.95, and 0.91 for ANN; 0.91, 0.97, and 0.93 for XGBoost; 0.92, 0.95, and 0.93 for LightGBM.

In summary, RF, SVR and ANN ML-based methods were used for the same data set in both this study and [12]. Comparing the results of both studies, this study performed better than [12] in the same years. There could be several reasons for this situation. First, parameter optimization is applied to each model so that the ML-based models used in this study yields better results during the training phase. Models with optimum parameters enable us to reach more compatible curves during the learning phase. The latter may be due to the different validation stages used in both studies. While k-fold cross validation is used in this study, leave-one-out cross validation is used in [12]. Through k-fold cross validation, it is aimed to learn and make predictions by making the method independent from each sample of the data set. In this way, the averages obtained from many results may have been more successful. Finally, [12] compares the performance of 4 ML-based models which averaged across days with non-missing in-situ observations at all stations (i.e., 135 days) during July and August from 2015 to 2017. Unlike [12], this study also uses missing in situ observations at all stations from 2013 to 2014. Missing observations are filled in with the mean imputation technique.

5 Conclusion

Accurate forecasting of the weather is very important in many critical areas such as aviation, agriculture, transportation, and the military. Therefore, many different methods have been developed to improve the accuracy of weather forecasting. The use of ML-based regression methods with low operating cost and high computational power has increased considerably in recent years. In this study, the next day's maximum and minimum air temperature are predicted for Seoul, South Korea, using boosting-based ML algorithms (AdaBoost, XGBoost, and LightGBM) and traditional ones (ridge, lasso, elastic net, SGD, KNN, DT, RF, ANN, and SVR). The performance of 12 different ML-based models is compared on meteorological data collected from 2013 to 2017. As can be seen from the Table 4 and the Figure 3, XGBoost and LightGBM methods show the highest performance for T_{max} and T_{min} in all years, with both statistical test analysis and the highest R^2 score. Moreover, according to the processing time analysis performed, the LightGBM method is more successful than XGBoost in terms of the time required to optimally determine the parameters and estimate the test sample.

Regarding the limitations of this study, the forecast results are based on weather-relevant data and location data are not considered. In future studies, more concrete results can be obtained by observing the effect of location data. The generalization capabilities of ML-based models can be enhanced by increasing the number of observations in the data set. Furthermore, high-impact features can be determined by applying feature selection techniques, thus providing both time and memory efficiency to methods with high performance but longer prediction time. In addition, only ML-based regression algorithms are analysis for this study; no other time series algorithms have been studied. In the future, analysis of other time series algorithms such as ARIMA, LSTM can be adopted. Moreover, more up-to-date studies can be done by following the newly made NWP-based weather forecasting studies.

6 Author contribution statements

In the scope of this study, Merve APAYDIN, Mehmethan YUMUŞ, Ali DEĞİRMENCİ, and Ömer KARAL contributed equally to the creation of the idea, its design, the literature review, the evaluation of the results obtained, the analysis of the results, and the checking of the spelling and content of the article.

7 Ethics committee approval and conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared. There is no conflict of interest with any person/institution in the article prepared.

8 References

[1] Bushara NO, Abraham A. “Weather forecasting in Sudan using machine learning schemes”. *Journal of Network and Innovative Computing*, 2(1), 309-317, 2014.

[2] Holmstrom M, Liu D, Vo C. “Machine learning applied to weather forecasting”. *Meteorological Applications*, 10, 1-5, 2016.

[3] Saba T, Rehman A, AlGhamdi JS. "Weather forecasting based on hybrid neural model". *Applied Water Science*, 7(7), 3869-3874, 2017.

[4] Sharaff A, Roy SR. "Comparative analysis of temperature prediction using regression methods and back propagation neural network". *IEEE 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 11-12 May 2018.

[5] dos Santos RS. "Estimating spatio-temporal air temperature in London (UK) using machine learning and earth observation satellite data". *International Journal of Applied Earth Observation and Geoinformation*, 2020. https://doi.org/10.1016/j.jag.2020.102066

[6] Ferreira LB, da Cunha FF. "New approach to estimate daily reference evapotranspiration based on hourly temperature and relative humidity using machine learning and deep learning". *Agricultural Water Management*, 2020. https://doi.org/10.1016/j.agwat.2020.106113

[7] Wolff S, O'Donncha F, Chen B. "Statistical and machine learning ensemble modelling to forecast sea surface temperature". *Journal of Marine Systems*, 2020. https://doi.org/10.1016/j.jmarsys.2020.103347

[8] Lee S, Lee YS, Son Y. "Forecasting daily temperatures with different time interval data using deep neural networks". *Applied Sciences*, 2020. https://doi.org/10.3390/app10051609

[9] Jakaria AHM., Hossain MM., Rahman MA. "Smart weather forecasting using machine learning: a case study in Tennessee". *arXiv*, 2020. https://arxiv.org/pdf/2008.10789.pdf

[10] Akyüz AÖ, Kumaş K, Ayan M, Güngör A. "Antalya ili meteorolojik verileri yardımıyla hava sıcaklığının yapay sinir ağları metodu ile tahmini". *Gümüşhane Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 10(1), 146-154, 2020.

[11] Sevinç A, Kaya B. "Derin Öğrenme ve İstatistiksel Modelleme Yöntemiyle Sıcaklık Tahmini ve Karşılaştırılması". *Avrupa Bilim ve Teknoloji Dergisi*, (28), 1222-1228, 2021.

[12] Cho D, Yoo C, Im J, Cha DH. "Comparative assessment of various machine learning-based bias correction methods for numerical weather prediction model forecasts of extreme air temperatures in urban areas". *Earth and Space Science*, 2020. https://doi.org/10.1029/2019EA000740

[13] Duo D, Graff C. "UCI Machine Learning Repository". https://archive.ics.uci.edu/ml (15.01.2021).

[14] Hoerl AE, Kennard RW. "Ridge regression: biased estimation for nonorthogonal problems". *Technometrics,* 12(1), 55-67, 1970.

[15] Muthukrishnan R, Rohini R. "LASSO: A feature selection technique in predictive modeling for machine learning". *IEEE 2016 International Conference on Advances in Computer Applications (ICACA)*, Coimbatore, India, 24-24 October 2016.

[16] Münch MM, Peeters CF, Van Der Vaart AW, Van De Wiel MA. "Adaptive group-regularized logistic elastic net regression". *Biostatistics,* 2018. https://doi.org/10.1093/biostatistics/kxz062

[17] González-Briones A, Hernández G, Pinto T, Vale Z, Corchado JM. "A review of the main machine learning methods for predicting residential energy consumption". *IEEE 2019 16th International Conference on the European Energy Market (EEM),* Ljubljana, Slovenia, 18-20 September 2019.

[18] Bottou L. "Large-scale machine learning with stochastic gradient descent". *Proceedings of COMPSTAT'2010,* 2010. https://doi.org/10.1007/978-3-7908-2604-3_16

[19] Imandoust SB, Bolandraftar M. "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background". International Journal of Engineering Research and Applications, 3(5), 605-610, 2013.

[20] Rhys HI. *Machine Learning with R, the Tidyverse, and MLR*. 1st ed. New York, USA, Manning, 2020.

[21] Yumus M, Apaydin M, Degirmenci A, Karal O. "Missing data imputation using machine learning based methods to improve HCC survival prediction". *IEEE 2020 28th Signal Processing and Communications Applications Conference (SIU),* Gaziantep, Turkey, 5-7 October 2020.

[22] Karasu S, Altan A. "Recognition model for solar radiation time series based on random forest with feature selection approach". In *2019 11th International Conference on Electrical and Electronics Engineering (ELECO) IEEE*, Bursa, Turkey, 28-30 November 2019.

[23] Vamsidhar E, Varma KVSRP, Rao PS, Satapati R. "Prediction of rainfall using backpropagation neural network model." *International Journal on Computer Science and Engineering,* 2(4), 1119-1121, 2010.

[24] Karal O. "Maximum likelihood optimal and robust Support Vector Regression with lncosh loss function". *Neural Networks,* 94, 1-12, 2017.

[25] Karal O. "EKG verilerinin destek vektör regresyon yöntemiyle sıkıştırılması". *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi,* 33(2), 743-756, 2018.

[26] Degirmenci A, Karal O. "Evaluation of kernel effects on svm classification in the success of wart treatment methods". *American Journal of Engineering Research*, *7*, 238-244, 2018.

[27] Freund Y, Schapire RE. "A decision-theoretic generalization of on-line learning and an application to boosting". *Journal of Computer and System Sciences*, 55(1), 119-139, 1997.

[28] Ogunleye A, Wang QG. "XGBoost model for chronic kidney disease diagnosis". *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* 17(6), 2131-2140, 2019.

[29] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY. "Lightgbm: A highly efficient gradient boosting decision tree". *Advances in Neural Information Processing Systems*, 30, 3146-3154, 2017.

[30] Hacioğlu R. "Prediction of solar radiation based on machine learning methods". *The Journal of Cognitive Systems*, 2(1), 16-20, 2017.

[31] Karal O. "Performance comparison of different kernel functions in SVM for different k value in k-fold cross-validation". *IEEE 2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Istanbul, Turkey, 15-17 October 2020.

[32] García S, Fernández A, Luengo, J, Herrera, F. "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power". *Information Sciences*, 180(10), 2044-2064, 2010.