Knowledge-Based Systems 247 (2022) 108682

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Intelligent cost-effective winter road maintenance by predicting road surface temperature using machine learning techniques

Mahshid Hatamzad [a,*], Geanette Cleotilde Polanco Pinerez [a], Johan Casselgren [b]

[a] Department of Industrial Engineering, UiT/The Arctic University of Norway, Narvik, 8514 Nordland, Norway
[b] Department of Engineering Sciences and Mathematics, Luleå University of Technology, 971 87 Luleå, Sweden

ARTICLE INFO

Article history:
Received 25 August 2020
Received in revised form 11 January 2022
Accepted 25 March 2022
Available online 6 April 2022

Keywords:
Decision-making units
Decision support systems
Machine learning techniques
Road surface temperature
Winter road maintenance

ABSTRACT

Since Winter Road Maintenance (WRM) is an important activity in Nordic countries, accurate intelligent cost-effective WRM can create precise advance plans for developing decision support systems to improve traffic safety on the roads, while reducing cost and negative environmental impacts. Lack of comprehensive knowledge and inaccurate WRM information would lead to a certain loss of WRM budget, safety reduction, and irreparable environmental damage. This study proposes an intelligent methodology that uses data envelopment analysis and machine learning techniques. In the proposed methodology, WRM efficiency is calculated by data envelopment analysis for different decision-making units (roads), and inefficient units need to be considered for further assessments. Therefore, road surface temperature is predicted by means of machine learning methods, in order to achieve efficient and effective WRM on the roads during winter in cold regions. In total, four different methods have been used to predict road surface temperature on an inefficient road. One of these is linear regression, which is a classical statistical regression technique (ordinary least square regression); the other three methods are machine-learning techniques, including support vector regression, multilayer perceptron artificial neural network, and random forest regression. Graphical and numerical results indicate that support vector regression is the most accurate method.

1. Introduction

1.1. Motivation

Due to the recent surge in human's dependency on using car in daily life, the number of vehicles on the roads has increased, and there have been rapid changes in traffic conditions [1]; the demand to provide drivers with real-time traffic information is growing, in order to improve traffic efficiency management and safety on the roads [2]. Congested roads can reduce transportation efficiency and cause a detrimental impact on road safety and the environment [3]. A reduction in transportation efficiency can lead to a growth in the number of crashes, especially in wintertime when traffic conditions are challenging. Road collision is a dangerous problem in societies and can influence communities and people, resulting in economic losses, health issues, and fatalities [4]. In transportation, the robust and accurate prediction of traffic parameters (e.g. flow, speed, occupancy, and travel time) and non-traffic parameters (e.g. traffic events and weather) can lead to efficient traffic management, such as a faster and safer path for transporting goods and avoiding congestion [5,6]. Severe weather conditions such as snowstorm reduce transportation quality, due to low visibility and slippery road surface. One of the major tasks regarding safe transportation in winter in the Nordic countries (Denmark, Finland, Iceland, Norway, and Sweden) is Winter Road Maintenance (WRM). The need for WRM has increased rapidly due to the rise in extreme weather conditions. Slippery road surface can cause severe traffic accidents (especially for old vehicles). WRM is divided into two categories: (i) reactive activities are carried out after adverse weather events, while (ii) proactive activities are carried out before adverse weather conditions occur [7]. There are different WRM techniques, including anti-icing and de-icing. The former involves using chemicals to prevent bonds between road surfaces and ice crystals, while using chemicals to melt the formed ice on road surfaces is known as de-icing [8]. The chemicals have a negative impact on the environment and could damage vegetation, animals, and aquatic species [9]. Salt is the most frequently used material for anti-icing and de-icing. Reducing the use of salt on road surfaces could cause a reduction in cost and fewer negative impacts on the environment. In order to minimize the amount of salt on the ground during winter, it is necessary to develop a prediction model for surface temperature, which is able to improve the

* Corresponding author.
E-mail addresses: Mahshid.hatamzad@uit.no (M. Hatamzad), geanette.polanco@uit.no (G.C.P. Pinerez), johan.casselgren@ltu.se (J. Casselgren).

https://doi.org/10.1016/j.knosys.2022.108682

accuracy in grip calculation and consequently result in optimizing the salt quantity.

1.2. Literature review

Prediction is the procedure of projecting future performance according to historical data. Highly accurate prediction can help in decision-making and planning for the future [10]. There are two main types of methods for predicting road surface temperature: (i) numerical methods and (ii) statistical methods, which are classified into traditional and modern methods. In fact, most prediction models for road surface temperature use numerical methods, which involve a combination of mathematics and physics to create an equation to estimate the road surface temperature. These methods are not based on historical data, and it is hard to determine the parameters and solve the equations. Statistical methods establish a model according to observed data and sometimes are not difficult to implement. However, traditional statistical techniques are incapable of producing the desired accuracy, due to the low amount of data [11]. This means that using a large amount of data in traditional statistical methods can be helpful for achieving a high degree of accuracy, but, of course, this is hard and time-consuming. These issues have been addressed by modern statistical methods named Machine Learning (ML) techniques, and information can be extracted from distributed sensors to collect big data, which are being used extensively in the modern world [12]. Due to their fast speed of learning, accurate results, easiness to implement and determine parameters, as well as good performance as regards generalization, ML techniques have improved rapidly, in order to handle projects involving a large amount of historical data. Diverse ML techniques have been used in transportation and specifically in WRM for different targets. Ahabchane et al. [13] used Geographic Information Systems (GIS) to extract the characteristics of street-networks, which were excluded by former prediction models. This study considered truck telemetry, geomatics, and weather data, to build a model to predict the amount of salt and abrasive on the road segments. Roychowdhury et al. [14] proposed the two-stage model to classify road surface conditions by using a convolutional neural network to estimate friction by a rule-based model. Panahandeh et al. [15] used logistic regression, a Support Vector Machine (SVM), and an Artificial Neural Network (ANN) for classification to estimate road friction for connected vehicles. Ye et al. [16] proposed a methodology which combines sensitivity analysis and neural networks to evaluate the impacts of accurate weather information on WRM cost. Xu et al. [17] introduced an algorithm based on static and dynamic prediction, through an improvement of a back propagation neural network, to predict pavement temperature.

1.3. Contribution

Reviewing previous studies has shown that WRM has been generally excluded from using ML regression algorithms to predict Road Surface Temperature (RST) in winter using RWS and three different types of sensors. To address this issue, this paper combines a Data Envelopment Analysis (DEA) model and ML techniques, through following steps:

(i) To measure WRM efficiency for Decision-Making Units (DMUs) or roads by a DEA model. In fact, effective WRM does not necessarily mean that WRM is efficient, since it is possible to be effective but inefficient as a result of an unrealistic WRM budget [18]. That is why, in order to achieve cost-effective WRM, it must first be efficient (i.e. using the minimum amount of resources to maximize the safety on roadways).

(ii) To analyze data achieved by optical and road-mounted sensors and use ML regression techniques to predict RST in winter. RST is a major factor for discovering whether ice has formed on the ground or not [17]. Inaccurate prediction causes inaccurate advance planning, which results in less efficient or inefficient WRM and ultimately can lead to a notable increase in WRM budgets [7].

1.4. Purpose

In fact, accurate surface temperature prediction for WRM plays a crucial role in traffic safety, cost and environmental impacts. If the surface temperature is predicted precisely, it will lead to optimization of the amount of chemicals which need to be used on the ground in order to have an acceptable friction between tires and road surface. Hence, it is important to improve the prediction of surface temperature in wintertime in cold regions. Thus, this study introduces the prediction methodology for intelligent cost-effective WRM (ICWRM), through the combination of DEA and ML techniques to improve decision support systems (DSS) for DMUs (roads), and specifically implements an overview of data analysis in the early stage of an ML model. ICWRM uses a large amount of available high-quality real-time data, since it can develop reliability in WRM DSS in different traffic (flow) situations. ICWRM can predict the RST for an inefficient WRM, based on selected features to detect severe surface conditions, and transmit them to the close stations. It is important to mention that the DEA stage of this methodology is discussed in detail in another paper [19]. The ML regression models used in this paper are support vector regression (SVR), multilayer perceptron artificial neural networks (MLP-ANN), and random forest (RF). Furthermore, linear regression (LR) is also used as a classical statistical regression technique (ordinary least square regression).

1.5. Achievement

The most important achievement in this research study is to predict RST with high accuracy to improve decision support systems for decision-making units (roads) in order to reach efficient and effective WRM.

1.6. Outline of paper

The remainder of this paper is organized as follows. Section 2 introduces problem definition. The DEA model is explained in Section 3. In Section 4 we give a summary of the LR model. Section 5 details the ML models. The methodology is formulated in Section 5. Finally, in Section 6, we present conclusions.

2. Problem definition

Countries with a cold climate face challenges in terms of the management and prediction of both WRM quality and quantity [20]. For decades, salt has been used for both anti-icing and de-icing to maintain the roads in the winter season. Salt melts the ice and snow from the ground and increases the friction between tires and road surface to improve the surface's transportation performance [21]. High surface transportation performance can support the safety of drivers on the roads, especially in adverse weather conditions. Although salting improves the safety on the roads in wintertime, it also has negative impacts on the environment. Excess use of salt demands a large budget. In order to minimize salt expenses, it is important to accurately predict the RST, since it affects the grip conditions on the road surface and can determine the optimal salt quantity. RST prediction is dependent on the presence of historical data for different factors,

summarized as weather variables and road surface condition variables. Historical data of weather variables can be collected by Road Weather Stations (RWS) and include air temperature, precipitation, dew point, humidity, wind speed, wind direction, visibility, solar radiation, and pressure [13]. Historical data of road surface condition variables can be collected by different sensors and include surface state, level of grip, water/snow/ice layer, alarm status, amount of chemical (used or present) on the road surface, base temperature, and ground temperature. However, using all these variables makes the model complex, and we need to find the most predictive variables by means of feature selection methods.

3. Data envelopment analysis

DEA has been widely used in different areas since 1987; it works according to linear programming, using multi-inputs and outputs [22]. DEA is an optimization technique to compute input weights and output weights in order to measure the efficiencies for different DMUs. Based on this method, DMUs are divided into efficient DMUs and inefficient DMUs [23]. DEA is classified into two major types; Charnes, Cooper, and Rhodes presented the model named CCR in 1978, and Banker, Charnes, and Cooper presented the model named BCC in 1984. CCR is based on constant returns to scale, whereas BCC works according to variable returns to scale. As a matter of fact, the feasible zone in the BCC model is smaller than in the CCR model, due to one more free variable and the convexity restrictions, which can influence the efficiency score and can increase the quantity of efficient DMUs in the BCC model [19,24]. If DMU_p is efficient in the CCR model, DMU_p will definitely be efficient in the BCC model. If DMU_p is efficient in the BCC model, DMU_p can be either efficient or inefficient in the CCR model. The mathematical optimization DEA-CCR problem can be defined as the following procedure [25]:

Consider DMU_j ($j = 1, 2, \ldots, n$), where each DMU includes m diverse inputs (x_{ij}) to generate s outputs (y_{rj}) for DMU_{ij} ($i = 1, 2, \ldots, m$). Additionally, u_r ($r = 1, 2, \ldots, s$) represents input weights and v_i indicates output weights. x_{ij}, y_{rj}, u_r and v_i must be greater than zero. A DEA linear optimization problem attempts to maximize the efficiency score for each DMU subject to some restrictions. For instance, for DMU_p, the original linear divisive DEA-CCR problem to maximize the efficiency score (Z_p) can be written as:

$$Max\ Z_p = \frac{\sum_{r=1}^{s} u_r y_{rp}}{\sum_{i=1}^{m} v_i x_{ip}} \tag{1}$$

subject to:

$$\frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leq 1$$

$$u_r \geq 0\ ,\ \ v_i \geq 0$$

$$j = 1, \ldots, n \quad r = 1, \ldots, s \quad i = 1, \ldots, m$$

It is also possible to convert it to linear programming as:

$$Max\ Z_p = \sum_{r=1}^{s} u_r y_{rp} \tag{2}$$

subject to:

$$\sum_{i=1}^{m} v_i x_{ip} = 1$$

$$\sum_{r=1}^{s} u_r y_{rj} \leq \sum_{i=1}^{m} v_i x_{ij}$$

$$u_r \geq 0\ ,\ \ v_i \geq 0$$

$$j = 1, \ldots, n \quad r = 1, \ldots, s \quad i = 1, \ldots, m$$

This model needs to be solved for all DMUs [24], to obtain the final results which are the maximum possible efficiency score for each DMU and the numerical value of decision variables (u_r and v_i) [19].

4. Linear regression

LR is an ordinary least squares (OLS) regression technique. In classical statistical regression, OLS regression is a method for estimating the equation that minimizes the summation of squared distances between actual values and predicted values (residuals) [26]. There are some assumptions in OLS regression that should be met: (i) the avoidance of multicollinearity, which means that the variable predictors need to be uncorrelated with each other, (ii) residuals need to be normally distributed, (iii) the variance of residuals is constant, (iv) residuals must be uncorrelated with predictor variables, v) there is no correlation between residuals (serial correlation), and (vi) regression coefficients need to be linear [27]. Regression analysis is a method for discovering the relationship between inputs (independent variables) and output (dependent variable). There are two types of LR: (i) simple LR and (ii) multiple LR [28,29]. Simple LR has one independent variable in the model, while multiple LR has two or more independent variables. The mathematical theory of LR can be defined as below:

If y is an output (dependent variable) and x_i shows inputs (independent variables), the simple LR model can be written as $y = b_0 + b_i x_i$ and $i = 1, \ldots, n$, where b_0 is an intercept and b_i is a regression coefficient for the variable x_i. Furthermore, we want to calculate the predicted output ($\widehat{y}$), which needs to be obtained by calculating the estimated intercept ($\hat{b}_0$) and the estimated regression coefficients ($\hat{b}_i$) plus the residuals (ε_i), i.e. $\hat{y} = \hat{b}_0 + \hat{b}_i x_i + \varepsilon_i$ that ε_i is independently and identically distributed [30]. Usually, in LR, the estimation of parameters is based on minimizing the summation of squared residuals [31].

5. Machine learning

An area of computer science, ML is considered a branch of artificial intelligence [32]. ML techniques have become one of the most frequently applied methods in technology, science, and commerce [33]. Due to the good performance of ML in various domains, its use has quickly increased [34]. ML methods can fit a suitable and flexible model, using the relationship between inputs and outputs in order to learn directly from the data [35], and it is able to estimate the unspecified dependencies from a dataset in order to predict new outputs [36]. Supervised ML methods learn from examples, meaning that they use a training set of data with corresponding targets (labeled data), in order to generalize the algorithm and predict outputs for all the possible unforeseen data, by finding the relationship between input variables (independent variables) and target variables (dependent variables) [37]. ML regression models are classified as supervised learning and work on a basis of continuous real values that belong to the output variable [38].

Data samples play a crucial role in ML techniques. Each sample is defined by certain features, and each feature includes various values. Data quality is dependent on the existence of missing data, outliers, noise, and categorical data that have to be treated with proper methods [36]. Feature Engineering (FE) and Feature Selections (FS) are the steps that help us to improve the data quality, which can lead to improving the analysis outputs. FE is the fundamental step in ML and can help us to improve the quality of data, so that they are ready to be used in the ML model. FE is a process of transforming the available observations (raw data) to create features that are used to describe data for

a prediction algorithm [39]. In fact, FE is a key to improvement in machine learning, and it demands considerable determined effort and time [13,39]. FS can select a relevant subset of variables (predictive features) without any extra transformation according to the analysis target, in order to build an ML algorithm [40].

Missing data (absence of data) occurs once there is no stored data for observation in a sample, and it happens for most of the datasets. Missing values influence the performance and the outcome's accuracy gained by the learning algorithm [41]. They require special attention; in some cases, it is possible to remove or drop these observations.

Categorical variables are the non-quantitative variables that contain explanatory characteristics instead of numbers. Categorical variables are classified into two types: (i) ordinal and (ii) nominal. Ordinal variables are variables that show intrinsic order, i.e. they are meaningful if ordered. For instance, on weekdays, Monday can be considered as 1 (first day of the week) and Sunday can be considered as 7 (seventh day of the week). Nominal variables do not have any intrinsic order (arbitrary names) such as gender. Categorical variables need to be transformed into numerical values in order to be understood by ML models [42].

Feature scaling changes the magnitude of the feature values from a diverse dynamic range into a particular range [43]. Regression coefficients can be directly affected by feature scaling. Without feature scaling, feature values with a small range will be dominated by feature values with a bigger range.

5.1. Support vector regression

SVR was introduced by Vapnik et al. in 1997 [44]. Its main concept is in accordance with the computing of linear regression [45]. SVR is rooted in the support vector machine that works on the basis of statistical learning theory [46]. SVM has two major categories: (i) Support Vector Classification (SVC) and (ii) support vector regression. SVR has an additional parameter compared to SVC called ε-tube (precision parameter), which represents an insensitive zone (radius) all over the regression function, in order to evaluate empirical error [45,47,48]. SVR is an effective and powerful ML method for solving several problems involved with the prediction for real cases [49]. The major advantage of SVR is that, regardless of having complex computations, it is not sensitive to the high dimensionality of variable space. In addition, SVR does not attempt to minimize the training error; it tries to minimize the error for generalization, so it is capable of achieving excellent performance in generalization [45,46]. The mathematical theory of SVR is defined as [47]:

Consider $(x_i, y_i)_{i=1}^{m}$ as a dataset, where x_i shows input vectors (independent variables), which have n dimensions, y_i shows corresponding real values or outputs (dependent variables) and m shows the number of observations. The goal of regression analysis is to discover $f(x)$, known as the regression function, which can predict the outputs as accurately as possible. The SVR formulation uses $f(x) = w.\phi(\mathrm{x}) + \mathrm{b}$ as a linear estimation function, where x, w, ϕ and b are an input vector, weight vector, nonlinear map, and constant, respectively. As SVR needs to solve a problem (nonlinear regression), it tries to nonlinearly map the input vector into a feature space with a high dimension and then correlate it with the output linearly. The following formula is a cost function (L_ε), which uses the ε-insensitive loss function.

$$L_\varepsilon(f(x), y) = \begin{cases} 0 & if \ \ |f(x) - y| < \varepsilon \\ |f(x) - y| - \varepsilon & otherwise \end{cases} \tag{3}$$

w *and* b can be calculated by minimizing the following function, named the regularized risk function:

$$R(c) = \frac{1}{2}\|w\|^2 + \mathrm{C}\frac{1}{m}\sum_{\mathrm{i}=1}^{\mathrm{m}} L_\varepsilon(f(x), y) \tag{4}$$

where $\frac{1}{2}\|w\|^2$ is known as the regularization term, which measures the function's flatness. C is a parameter called the regularization constant, which specifies the trade-off between the regularization term (flatness of the model) and the empirical risk [48]. To estimate how much $y_i - f(x_i)$ deviates from the ε-insensitive region, two positive variables can be applied as slacks, presented by ξ_i and $\hat{\xi}_i$. By using the slacks, Eq. (4) is changed into the following optimization problem, i.e. Eq. (5), with one objective and three constraints:

$$\min \frac{1}{2}\|w\|^2 + \mathrm{C}\sum_{\mathrm{i}=1}^{\mathrm{m}}(\xi_\mathrm{i} + \hat{\xi}_i) \tag{5}$$

subject to:

$$(w.\phi(\mathrm{x}_i)) + b - y_i \geq -(\varepsilon + \xi_\mathrm{i})$$

$$(w.\phi(\mathrm{x}_i)) + b - y_i \leq \varepsilon + \hat{\xi}_i$$

$$\xi_\mathrm{i}, \hat{\xi}_i \geq 0$$

It is possible to write the dual form of this mathematical optimization problem by using Karush–Kuhn–Tucker (KKT) conditions and Lagrange multipliers:

$$\max_{\alpha,\hat{\alpha}} \mathrm{W}(\alpha, \hat{\alpha}) = \tag{6}$$

$$\sum_{i=1}^{m}(y_i\,\hat{\alpha}_i - \alpha_i) - \varepsilon\sum_{i=1}^{m}(\hat{\alpha}_i + \alpha_i) - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_i - \alpha_j)K(x_i, x_j)$$

subject to:

$$\sum_{i=1}^{m}(\hat{\alpha}_i - \alpha_i) = 0$$

$$0 \leq \alpha_i, \hat{\alpha}_i = \mathrm{C}$$

where α_i and $\hat{\alpha}_i$ are Lagrange multipliers, which need to satisfy $\alpha_i\hat{\alpha}_i = 0$ condition. Therefore, the general form of the SVR function is written as:

$$f(x) = \sum_{i=1}^{n}(\hat{\alpha}_i - \alpha_i)\,k(x_i, x) + b \tag{7}$$

where $K(x_i, x_j)$ shows the kernel function and is equal to $\phi(\mathrm{x}_i)\,\phi(\mathrm{x}_j)$. Achieving an optimal solution in SVR depends on the type of kernel function used. Several kernels can be used in SVR, such as sigmoid, linear, polynomial, and radial basis functions (RBFs). There are two reasons why RBFs have been the most frequently used kernel: (i) they have few parameters to be set, and (ii) they are capable of classifying the datasets with multi-dimensions. Hence, the RBF kernel is used in this study to achieve an optimal solution. Eq. (8) shows the RBF function:

$$K(x_i, x_j) = \exp\left(-\gamma\|x_i, -x_j\|^2\right) \tag{8}$$

where γ is the parameter of the kernel.

5.2. Random forest

Random forest (RF) is an ensemble learning of decision trees, and it is one of the most used algorithms in many applications for both regression and classification [50]. RF combines tree predictors such that a single tree is dependent on the random vector values that are sampled independently with a similar distribution for all the trees in the forest [51]. There are three steps to constructing the RF: (i) to produce many different subsets by adopting bagging [52] on the training dataset, (ii) a decision tree is constructed by using each subset, and each node is split based on the randomly chosen group of candidates to grow the tree; (iii) all these trees are integrated [53].

5.3. Multilayer perceptron artificial neural network

An ANN consists of many interconnected processors named neurons that are similar to biological neurons located in the brain. These neurons are connected through numerical weights that can send signals to each other. These weights indicate how important each neuron input is. The learning process in an ANN is based on how to adjust these weights repeatedly. These networks collect the information achieved in the training process and respond to new cases in the proper manner. This study uses a typical ANN model called an MLP. In this model, the input layer receives input signals and sends them to the neurons of the hidden layer. The hidden layer recognizes the features in the input through the neuron's weights. The output layer builds the pattern for the output in the whole network [54].

5.4. Cross validation

Cross validation is a statistical technique to assess learning algorithms in order to evaluate the correct prediction error for independent observations/dataset (testing set) for generalization [55,56].

K-fold cross validation is the most frequently used method for cross validation. In this method, observations are divided into k equal subsets (folds), with one subset being kept for validation and $k-1$ subsets being used for training. This procedure iterates k times [13,55]. The training dataset is split k times by the generator of cross validation. The estimator is trained by different training sizes, and each training size and validation set are scored; then, the mean value of the score is calculated based on k iterations [57].

5.5. Bias variance trade-off and learning curves

Bias variance trade-off is a key part that can help us to understand the performance of the ML model. Discovering the correct balance between the bias of the model and the variance of the model, named bias variance trade-off, is a fundamental way to understand the performance of the model (i.e., the ML model is neither overfitted nor underfitted) [58].

Bias error is the difference between the predicted output by the model and the actual value [59]. Bias is high when the learning algorithm misses the main pattern amongst input variables and targets. In a high bias situation, the model includes quite simple assumptions to find the relationship between variables, and it leads to underfitting [58,59].

A variance error occurs when the model is not able to show as high a level of performance as it has in training data. A high variance occurs when the model is trained by lots of redundant data [59]. In a high variance situation, the model cannot have satisfactory results on unseen observations, so it suffers from a loss of generalization, leading to overfitting. Fig. 1 shows a graphical illustration of bias and variance. As shown, there are four alternatives regarding bias and variance [58]: (i) low bias and low variance (ideal option), which shows a consistent and accurate model; (ii) low bias and high variance, which shows an inconsistent but moderately accurate model; (iii) high bias and low variance, which shows a consistent and inaccurate model; (iv) high bias and high variance, which shows an inconsistent and inaccurate model. The final error achieved by the ML model is the summation of three different types of error: (i) bias error, (ii) variance error, and (iii) irreducible error, which occurs due to noises and which it is impossible to reduce [59]. The mathematical definition of this concept is as below [60]:

Consider Y as a target and X as an input (variable). Assume the relationship between X and Y is $Y = f(X) + \epsilon$, where ϵ shows the error term and it has a normal distribution with an expected value of zero ($E(\epsilon) = 0$) and variance of σ^2 ($var(\epsilon) = \sigma^2$). The objective is to estimate $f(X)$ by using the ML model, which is shown by $\hat{f}(X)$. So, the square of the expected prediction error for x is:

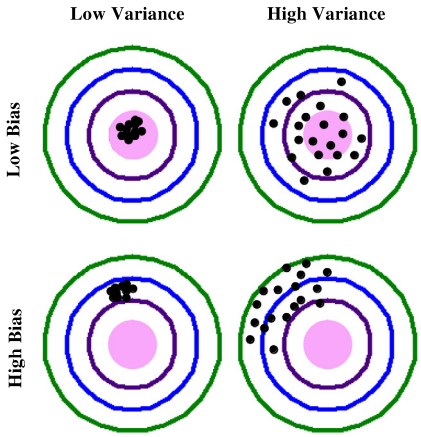

Fig. 1. Bias and variance schematic illustration [60].

$$Err(x) = E\left[(Y - \hat{f}(x))^2\right] \tag{9}$$

This error can be broken down into two parts: (i) bias and (ii) variance (Eq. (10)):

$$Err(x) = \left(E\left[\hat{f}(x)\right] - f(x)\right)^2 + E\left[\hat{f}(x) - E[\hat{f}(x)]^2\right] + \sigma_e^2 \tag{10}$$

which can be written as Eq. (11):

$$Err(x) = Bias^2 + Variance + Irreducible\ Error \tag{11}$$

where the third term shows the irreducible error.

Learning curves have been extensively used in ML. They consist of a training curve and a validation curve, which show the training error and the validation error, respectively, according to training size (Fig. 2). The training curve shows the process of learning based on the training dataset, while the validation curve shows the process of generalization based on the validation dataset. Therefore, learning curves can be used to illustrate the model's performance which refers to whether the model is overfitting, underfitting, or a good fit [61].

Fig. 3 graphically illustrates the training error curve and the testing error curve, based on the complexity of the model. On the left side of the graph, both the training error and the validation error are high (high bias area), whereas, on the right side of the graph, the training error is not high, but the validation error is high (high variance area). The middle part of the graph is the bias variance trade-off spot [59].

5.6. Evaluation metrics

The Mean Square Error (MSE) and the Root Mean Square Error (RMSE) are two standard statistical metrics for evaluating the model's performance. One of the most preferred evaluation metrics for regression models, the MSE is calculated by the average of squaring the difference between actual observations and predicted values. This squared difference penalizes even small errors that can overestimate how unfit the model is. The MSE is chosen more than other performance metrics since it is differentiable and thus able to be better optimized [64]. The RMSE is the square root of the MSE and shows the sample standard deviation of the residuals (the difference between original values and predicted

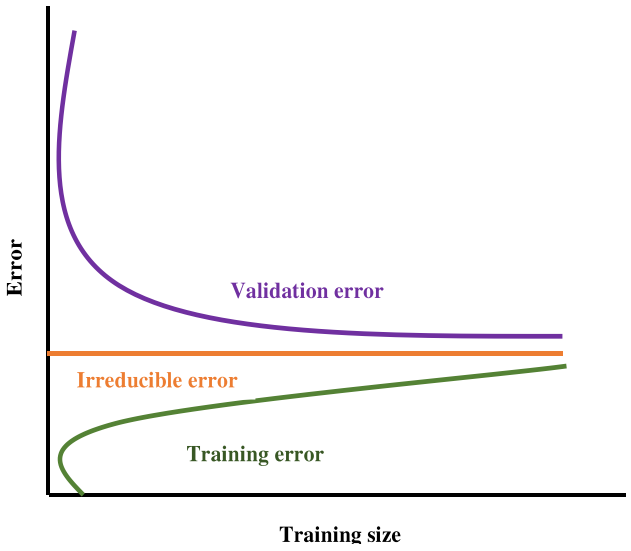

Fig. 2. Learning curve (training error, validation error, and irreducible error) [62].

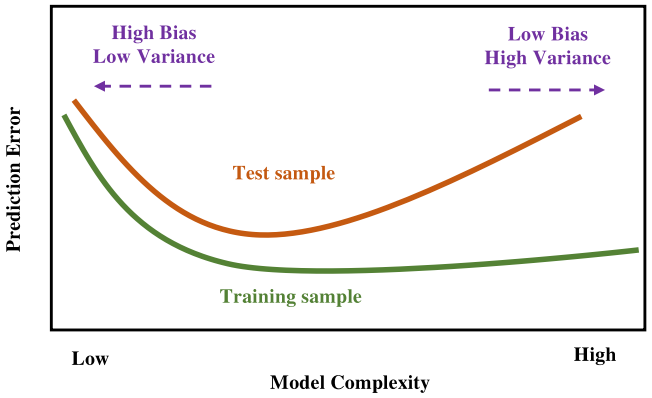

Fig. 3. Training error curve and testing error curve based on the complexity of the model [63].

values). The RMSE is the most widely used evaluation metric because of squaring the errors before averaging; the RMSE imposes a relatively heavy penalty on high errors. This indicates that the RMSE can be more helpful when high errors are not desirable. Moreover, the unit of the RMSE is the same as the unit of output and hence it is easier to be interpreted than the MSE [64,65]. Eqs. (12) and (13) show the mathematical calculation of the MSE and RMSE [32].

$$MSE = \frac{1}{N} \times \sum_{i=1}^{N} (y_i - f(x_i))^2 \tag{12}$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \times \sum_{i=1}^{N} (y_i - f(x_i))^2} \tag{13}$$

where y are actual values and $f(x_i)$ are values predicted by the model.

The other evaluation metric for the regression model is called the explained variance score (R^2), which shows how much variance the regression model explains. The best possible R^2 score is 1 [66].

6. Methodology

This article presents a five-stage framework for intelligent cost-effective WRM prediction. As shown in Fig. 4, the first stage is to calculate the efficiency of WRM based on the DEA-CCR model, which is a topic of another paper [19]. If the DMU is efficient, there is no need for further evaluation. However, if the DMU is not efficient, the DMU needs to be considered for more evaluation. The second stage is to collect the data for inefficient DMU. Here, the E18 road in Sweden was selected as an inefficient DMU. In order to enhance the WRM efficiency in this road, several influential variables were chosen to be observed every 10 min in February 2019, using three types of sensors and RWS. The first sensor was mounted in the wheel track named DRS511 1. The second sensor was mounted in the middle of the driveway named DRS511 2 and the third sensor was an optical sensor named DST. Forty-five variables were selected as inputs for observations (Table 3) to predict the RST on this road. The observations were analyzed to understand the nature of the data, which can be helpful for the next stage. In the third stage (FE), first the dataset was split into main training dataset and testing dataset. Then, rare label encoding was performed for categorical observations, and then categorical observations were transformed into numerical data to be understandable for ML; after that, all the data were scaled to be in the same range by robust scaler technique. In the fourth stage (FS), the most predictive features were selected by four different filter methods. Filter methods are the procedures for selecting features only based on feature information. Four filter methods were considered in this study: (i) constant features, (ii) quasi-constant features, (iii) correlation, and (iv) statistical measure. In the fifth stage, the main training dataset was divided into learning (training) and validation sets, to evaluate the generalization performance of the learning algorithms (SVR, MLP, LR, and RF), using the learning curves. The algorithms (SVR, MLP, LR, and RF) learn the patterns from the main training dataset, then the generalization error is estimated by the testing set. If the algorithm (SVR, MLP, LR, and RF) performs well, based on graphical and numerical results, ICWRM is ready to predict outputs; otherwise, the process needs to return to the FE stage. This methodology was applied in Python 3 with several libraries, including Numpy [67], Pandas [68], Matplotlib [69], Seaborn [70], Cufflinks [71], and Scikit-learn [72]. The data are from the Swedish Transport Administration's RWIS station at Test site E18 [73]. Dataset and all coding stages can be found on GitHub [74].

6.1. Stage 1- DEA

This study measures the efficiency of salting as a WRM technique, by using the DEA model in the Arctic region. Each road was considered as one DMU. Input and output variables were selected according to the relationship between these variables and the goal of this study. There were three input variables: WRM cost, traffic flow and road area. WRM cost was calculated by the summation of equipment cost, material/salt cost and labor cost. Traffic flow was computed based on average daily traffic (ADT). The road area was estimated by the length times the width of the road ($length \times width$). There were two output variables: environmental impacts and the safety level in each road after salting. The safety level is classified into three classes: high, medium and low, which are described in Table 1. Environmental impact is divided into three levels: high, medium and low, as interpreted in Table 2. Both outputs are qualitative and need to be quantified in order to be used in the mathematical optimization problem. This stage is fully discussed in another article with the topic of non-parametric linear technique for measuring the efficiency of winter road maintenance in the Arctic area [19].

Table 1
Description of safety levels.

Safety level	Description
High	No accidents on the road during a specific period
Medium	Accidents without any severe consequences on the road during a specific period
Low	Accident with casualties and severe injuries on the road during a specific period

Table 2
Description of different levels of environmental impacts.

Environmental impact	Description
High	The road is close to vegetation and water
Medium	The road is close to either vegetation or water
Low	The road is close to neither vegetation nor water

6.2. Stage 2- Data acquisition and data analysis

Several features were determined, which influence RST prediction. Table 3 shows all the variables and their abbreviations used in this study. The observations were collected by reading the data from RWS and three different types of sensors: (i) mounted sensor in the wheel track named DRS511 1, (ii) mounted sensor in the middle of the driveway named DRS511, and 2 (iii) optical sensor, referred to as DST. Analyzing the observations provided us with the type of variables (Fig. 5) and their characteristics.

6.2.1. Description of features

Surface temperature was measured by three different sensors, two mounted in the road and one optical sensor (DST). The observation extracted by an optical sensor for surface temperature was selected as an output (Surface_temp). The surface state was also measured by three sensors (road mounted sensors and optical sensor); it included dry, moist, wet, slushy, snowy, and icy surface conditions. Air temperature, dew point temperature, percentage of humidity, rain state (none, light, medium, snow), rain intensity, wind speed, wind direction, visibility, present weather (none, snow, rain, dry), precipitation, barometric pressure, rain on/off and maximum wind speed were collected by the RWS. The grip, water layer, ice layer, and snow layer were measured by the optical sensor. Three sensors measured the alarm status, which included none, frost warning, rain warning, snow warning, and ice warning. There were two different observations for visibility status and general status: either ok or not ok. Snow height and base temperature were measured by the sensor mounted in the middle of the driveway. Liquid freezing temperature, ground temperature, freezing temperature, and water thickness were measured by two different sensors mounted in the road. The concentration, conductivity, and amount of chemicals was measured by the DRS511 mounted on the road, giving the amount of salt present on the road in different units: g/l, mS/cm, and g/m^2, respectively. The relay state is configurable rules to control relays, which in this case is constant 0. The battery voltage parameter shows the power of the backup battery.

Each type of variable can be divided into different types. Numerical variables are those whose values are numbers, and they are divided into two types: (i) discrete (whole numbers), such as the number of students in the classroom, and (ii) continuous variables, including any values in a different range such as vehicle prices. Categorical variables were previously mentioned in Section 5. Date/time variables are those which contain time and/or date. Mixed variables are those which have both numbers and categories. Other variables can be text or images [75]. Table 4 shows different variable types in our case study, which consists of 35 numerical variables (4 discrete and 31 continuous), 11 nominal categorical variables, and one date–time variable.

Table 3
Features and their abbreviations.

Features	Abbreviation
Surface temperature (°C)\nDRS511 1	Surface_temp1
Surface temperature (°C)\nDRS511 2	Surface_temp2
Surface temperature (°C)\nDSC/DST (output)	Surface_temp
Surface state\nDRS511 1	Surface_state1
Surface state\nDRS511 2	Surface_state2
Surface state\nDSC/DST	Surface_state
Air temperature (°C)\nAtmospheric site	Air_temp
Dew point temperature (°C)\nAtmospheric site	Dew_point
Level of grip\nDSC/DST	Grip
Water layer (mm)\nDSC/DST	Water_L
Ice layer (mm)\nDSC/DST	Ice_L
Snow layer (water equivalent) (mm)\nDSC/DST	Snow_L
Relative humidity (%)\nAtmospheric site	Humidity
Rain state\nAtmospheric site	Rain_state
Rain intensity (mm/h)\nAtmospheric site	Rain_int
Wind speed (m/s)\nAtmospheric site	Wind_S
Wind direction (°)\nAtmospheric site	Wind_D
Visibility (m)\nAtmospheric site	Visibility
Present weather\nAtmospheric site	Present_weather
Precipitation total, past 24 h (mm)\nAtmospheric site	Precipitation_24
Alarm status\nDRS511 1	Alarm1
Alarm status\nDRS511 2	Alarm2
Alarm status\nDSC/DST	Alarm
Battery voltage (V)\nAtmospheric site	Battery_voltage
Concentration (g/l)\nDRS511 1	Concentration1
Concentration (g/l)\nDRS511 2	Concentration2
Conductivity\nDRS511 1	Conductivity1
Conductivity\nDRS511 2	Conductivity2
Visibility sensor status\nAtmospheric site	Visibility_status
Amount of chemical (g/m^2)\nDRS511 1	Chemical1
Amount of chemical (g/m^2)\nDRS511 2	Chemical2
General status\nAtmospheric site	General_status
Barometric pressure (hPa)\nAtmospheric site	Pressure
Rain on/off\nAtmospheric site	Rain_nf
Relay states\nAtmospheric site	Relay_state
Snow height (mm)\nDRS511 1	Snow_h
Base temperature (°C)\nDRS511 1	Base_temp
Liquid freezing temperature (°C)\nDRS511 1	Lfreezing1
Liquid freezing temperature (°C)\nDRS511 2	Lfreezing2
Ground temperature (°C)\nDRS511 1	Ground_temp1
Ground temperature (°C)\nDRS511 2	Ground_temp2
Freezing temperature (°C)\nDRS511 1	Freezing_temp1
Freezing temperature (°C)\nDRS511 2	Freezing_temp2
Max wind speed (m/s)\nAtmospheric site	Max_windS
Water thickness (mm)\nDRS511 1	Water_t1
Water thickness (mm)\nDRS511 2	Water_t2

As previously mentioned, data analysis can also give us information about the characteristics of variables, such as data shape, statistical description of data, detecting the missing values, data description, distributions of variables (normal or skewed), scale of different features (magnitude of features), outliers or unusual values, etc. The data shape for our case study is (3847, 46), which means that there are 46 variables (45 input variables and 1 output variable) and 3847 observations. The date–time variable is used as an index in this study. Tables 5 and 6 show a statistical description of numerical data, excluding missing values.

6.2.2. Removing missing values

The second step is to remove missing values. The first column in Tables 5 and 6 ('Count') demonstrates the number of observations for different variables. As can be seen, the numbers in this column are not similar for all features; for some, this number is less than 3847. The difference between 3847 and that number shows the number of missing values in the dataset. In our case, the number of missing values is not high. Therefore, due to the low number of missing values (maximum is 16 out of 3847), they have been dropped from the observations. Fig. 6 presents

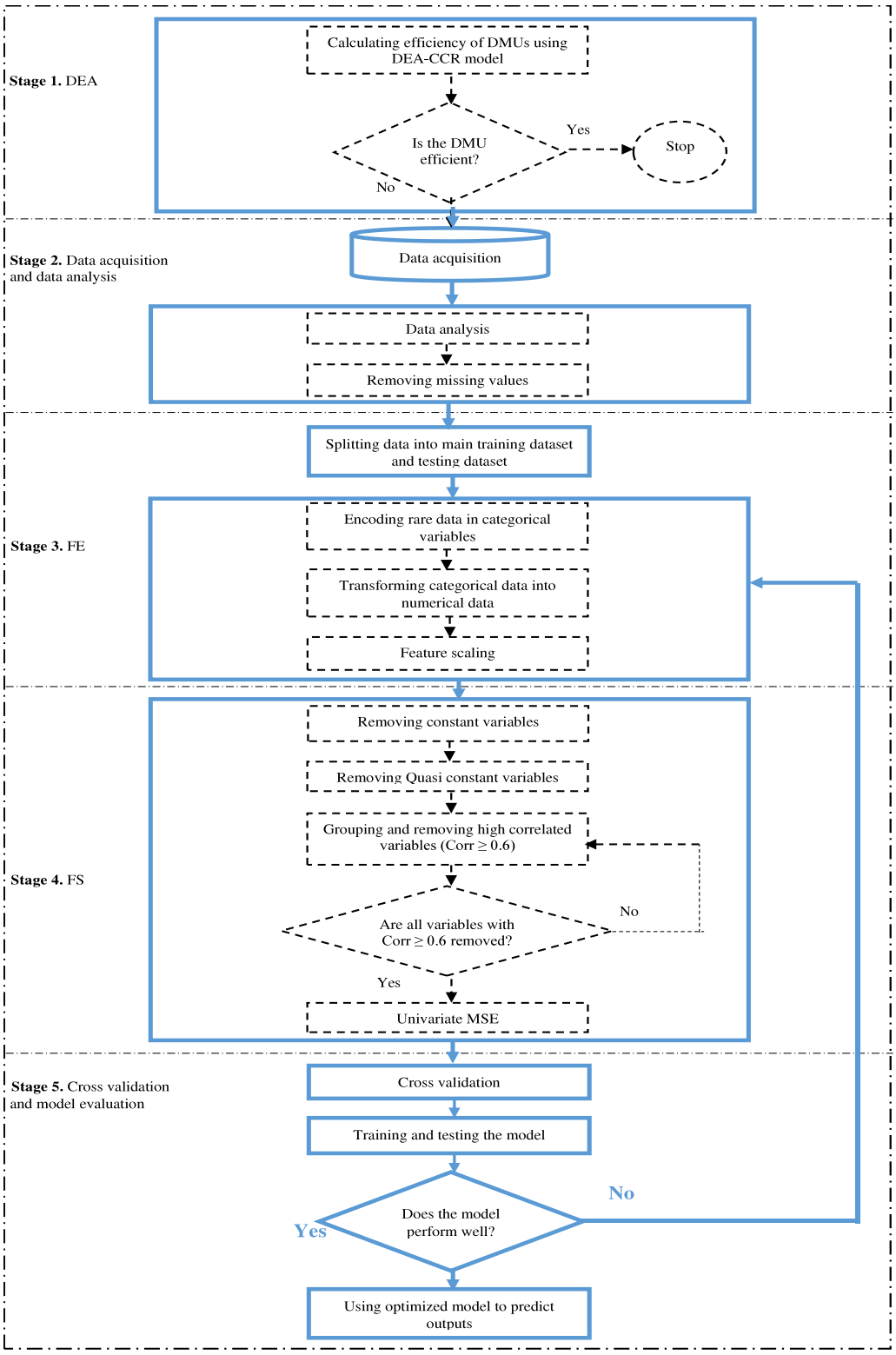

Fig. 4. Intelligent cost-effective winter road maintenance (ICWRM) framework.

Table 4
Number of different types of variables in our case study.

Data type	Numerical		Categorical		Date–time		Mixed	Other
	Discrete	Continuous	Nominal	Ordinal	Date	Time		
	✓	✓	✓	×	✓	✓	×	×
Number of variables	4	31	11	_	1	1	_	_

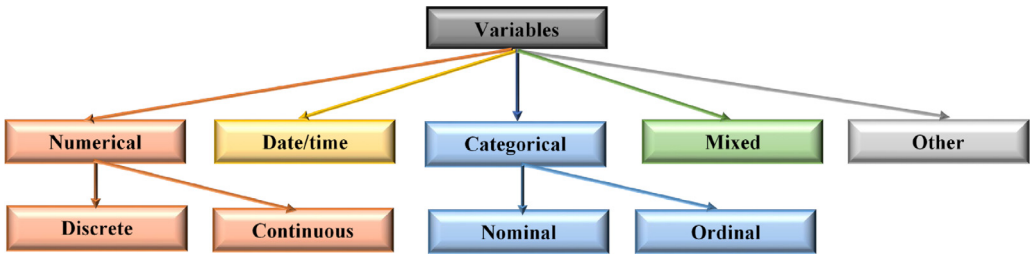

Fig. 5. Different types of variables.

Table 5
Characteristics of discrete numerical variables.

Discrete numerical variables	Count	Mean	Std	Min	25%	50%	75%	Max
Wind_D	3831	207.15	83.73	1	151	226	269	360
Visibility	3847	17 871.16	4653	370	20 000	20 000	20 000	20 000
Relay_states	3845	0	0	0	0	0	0	0
Snow_h	3845	2.56	5.33	0	0	0	2	47

Table 6
Characteristics of continuous numerical variables.

Continuous numerical values	Count	Mean	Std	Min	25%	50%	75%	Max
Surface_temp1	3845	0.61	4.63	−14.6	−1.5	1.1	3.3	14.2
Surface_temp2	3845	0.67	4.72	−14.5	−1.5	1	3.4	15.2
Surface_temp (output)	3845	0.23	4.27	−14.2	−1.5	0.7	2.7	11.2
Air_temp	3847	0.81	4.98	−20	−0.9	1.9	3.8	10.4
Dew_point	3847	−2.4	4.88	−21.9	−4.2	−0.9	1.2	3.7
Grip	3845	0.75	0.15	0.11	0.78	0.81	0.82	0.82
Water_L	3845	0.081	0.12	0	0	0.05	0.11	1.81
Ice_L	3845	0.019	0.06	0	0	0	0	0.51
Snow_L	3845	0.037	0.15	0	0	0	0	1.04
Humidity	3847	80.54	14.19	39	71	85	91	99
Rain_int	3847	0.045	0.23	0	0	0	0	10
Wind_S	3831	3.62	1.69	0.2	2.4	3.7	4.8	8.5
Precipitation_24	3847	0.45	0.73	0	0	0	0.6	5.8
Battery_voltage	3847	13.83	0.1	13.6	13.8	13.8	13.8	14.2
Concentration1	3845	25.38	77.32	0	0	1.5	7.3	352.7
Concentration2	3845	120.48	153.23	0	0	6.3	321.2	357.6
Conductivity1	3845	1.94	1.76	0	0	1.9	3.5	8.6
Conductivity2	3845	1.39	1.75	0	0	0.3	2.7	9.2
Chemical1	3845	0.24	0.9	0	0	0.1	0.2	16.9
Chemical2	3845	0.38	1.47	0	0	0.1	0.4	30.8
Pressure	3847	1010.87	12.71	981.6	1004.1	1011.3	1018.9	1040.6
Base_temp	3847	−0.04	2.08	−4.6	−1.5	−0.1	1.4	4.6
Lfreezing1	3845	−0.22	0.48	−8.3	−0.3	−0.1	0	0
Lfreezing2	3845	−0.75	1.31	−21.1	−1	−0.4	−0.1	0
Ground_temp1	3845	0.6	3.76	−11.3	−1.2	0.8	3	10.3
Ground_temp2	3845	0.57	3.7	−11.3	−1.2	0.8	2.9	10.2
Freezing_temp1	3845	−0.46	1.52	−21.1	−0.5	−0.1	0	0
Freezing_temp2	3845	−0.93	1.86	−21.1	−1.1	−0.5	−0.1	0
Max_windS	3831	5.78	2.64	0.5	3.7	5.7	7.6	15.4
Water_t1	3845	0.06	0.13	0	0	0.03	0.06	1.88
Water_t2	3845	0.05	0.22	0	0	0	0.01	3.1

histogram and density plots for the variable Surface_temp1. Histogram and density plots can help us to understand how much our observations are normally distributed. Furthermore, red sections in the histogram and density plots demonstrate those parts removed from the original dataset due to missing values; it is clear that this does not change the distributions or characteristics of the data. This procedure is considered for the rest of the numerical data.

In order to show that removing missing values does not affect the characteristics of categorical variables (Surface_state1, Surface_state2, Surface_state, Rain_ state, Present_weather, Alarm1, Alarm2, Alarm, Visibility_status, General_status and Rain_state), we calculated the percentage of various observations in each category before and after removing missing values. It is also possible to plot categorical distributions based on these percentages. Fig. 7 shows that the distribution plots for the surface_state variable remain almost the same, which means that the difference in percentages before and after deleting missing data is almost zero. This procedure is the same for other categorical variables.

6.3. Stage 3- Feature engineering

In the FE stage, the first step is to split the dataset into main training dataset and testing dataset. We decided to select 70% of the whole data for the main training set and 30% for the testing set, since the testing set needs to be big enough to lead to meaningful statistical results, and both main training and testing sets have the same characteristics. In addition, Surface_temp was selected as an output and the rest of the variables as the inputs. Main training and testing data shapes are (2678, 45), (1149, 45) respectively, where 2687 and 1149 illustrate the number of observations, and 45 is the number of input features.

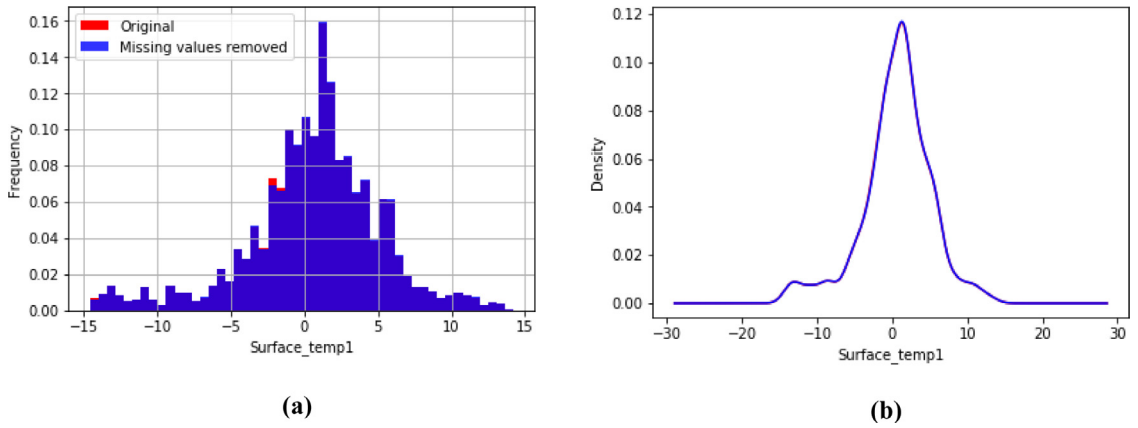

Fig. 6. (a) Histogram plot before and after removing missing values for Surface_temp1, (b) Density plot before and after removing missing values for surface_temp1.

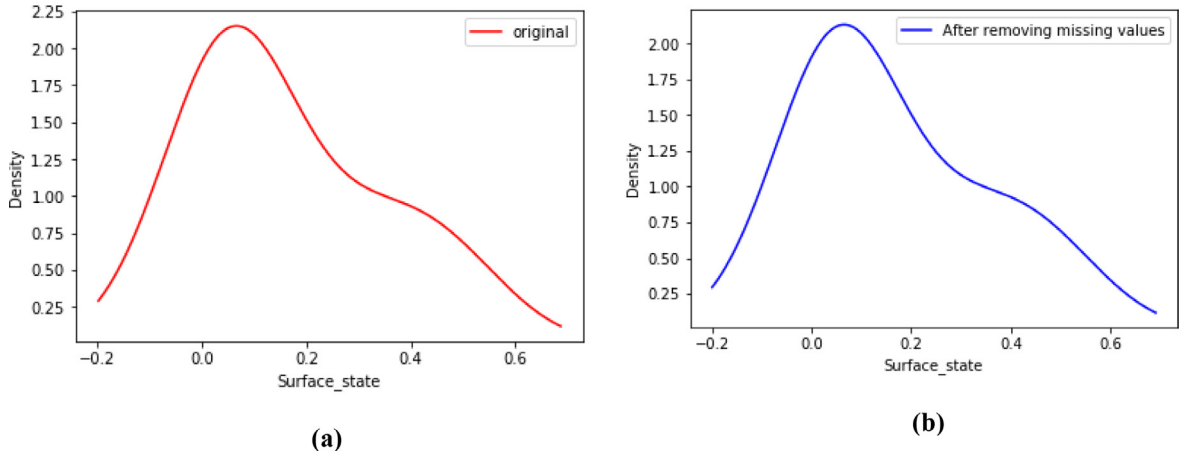

Fig. 7. Density plot for Surface_state (a) before dropping missing values, (b) after removing missing values.

Table 7
Frequent and infrequent labels in categorical variables.

Categorical variables	Infrequent categories	Frequent categories
Surface_state	[icy, snowy]	[dry, moist, slushy, wet]

6.3.1. Encoding rare labels in categorical variables

The next step is to encode rare labels in categorical variables. Rare data points in categorical variables were encoded after splitting the dataset into main training dataset and testing dataset and it applied in both sets. Rare labels are infrequent (rare) observations in the dataset. All categorical variables in our dataset included a minimum of one category with a tiny proportion (less than 5%). As an example, Table 7 shows frequent and infrequent labels for the variable Surface_state. Furthermore, Fig. 8 shows the frequency of different labels in this categorical variable before and after rare label encoding; the red line in the charts helps us to visualize the rare labels.

6.3.2. Transforming categorical values into numerical values

In the next step, categorical variables have been changed into numerical values to be understandable for the machine learning algorithm. Categorical data are transformed into numerical data after splitting the dataset into main training dataset and test dataset and it applied for both datasets. Ordinal or label encoding is the method used for this transformation. Ordinal encoding arbitrarily assigns an integer to categories in the main training set and applies those mappings into the testing set. This method does not add new variables to the dataset [76].

6.3.3. Feature scaling

The last step in this stage is feature scaling for inputs. Training the ML models by scaled data can lead to the model performing significantly better than models trained by unscaled data [77]. We have taken the scaler and fit it just to the main training dataset because in real life we are going to be able to scale to our current known data and then predicting in the future. We are not actually going to know the scale of that data. So, that is why this scale is only being fit on the main training dataset. After that we transform the main training data and testing data. Robust scaler is the method used in this research, as it is robust towards outliers. Sometimes, outliers can have a negative impact on the mean and variance of the sample. In this situation, the interquartile quantile range (IQR) and median can result in better outcomes. This scaling method removes the median from the observations and scales the data between the 25th quantile (1st quantile) and the 75th quantile (3rd quantile) or in the IQR [78]. For each feature, scaling and centering are calculated independently, according to the computation of the relevant sample's statistics in the main training set. As is clear, this step occurs after splitting data into main training dataset and testing dataset, since scalers require to learn the IQR and median values of the variables, based

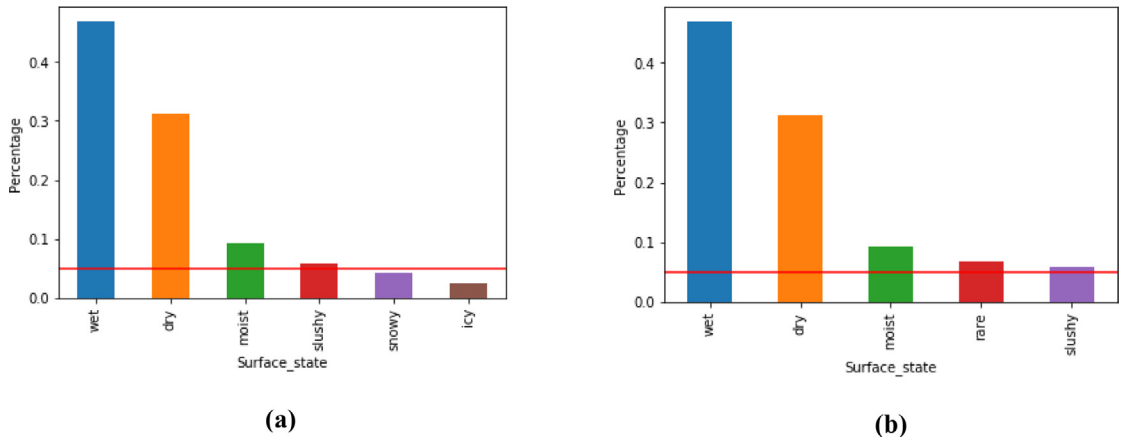

Fig. 8. Frequency of different labels in Surface_state (a) before rare label encoding, (b) after rare label encoding.

Table 8
Median values of the original observations saved by scaler.

Variable	Median	Variable	Median	Variable	Median	Variable	Median	Variable	Median
Surface_temp1	1.1	Ice_L	0	Precipitation_24	0	Visibility_status	0	L_freezing1	−0.1
Surface_temp2	1	Snow_L	0	Alarm1	2	Chemical1	0.1	L_freezing2	−0.4
Surface_state1	3	Humidity	85	Alarm2	2	Chemical2	0.1	Ground_temp1	0.9
Surface_state2	3	Rain_state	1	Alarm	2	General_status	1	Ground temp2	0.8
Surface_state	2	Rain_int	0	Battery_voltage	13.8	pressure	1011.4	Freezing_temp1	−0.1
Air_temp	2	Wind_S	3.7	Concentration1	1.7	Rain_nf	1	Freezing_temp2	−0.5
Dew_point	−0.8	Wind_D	226	Concentration2	8.35	Relay_state	0	Max_windS	5.7
Grip	0.81	Visibility	2×10^4	Conductivity1	2	Snow_h	0	Water_t1	0.03
Water_L	0.05	Present_weather	1	Conductivity2	0.4	Base_temp	0	Water_t2	0

Table 9
IQR values of the original observations saved by scaler.

Variable	IQR	Variable	IQR	Variable	IQR	Variable	IQR	Variable	IQR
Surface_temp1	4.7	Ice_L	1	Precipitation_24	o.6	Visibility_status	1	L_freezing1	0.3
Surface_temp2	4.8	Snow_L	1	Alarm1	1	Chemical1	0.275	L_freezing2	0.9
Surface_state1	3	Humidity	20	Alarm2	1	Chemical2	0.4	Ground_temp1	4
Surface_state2	3	Rain_state	1	Alarm	1	General_status	1	Ground temp2	4
Surface_state	2	Rain_int	1	Battery_voltage	1	pressure	14.7	Freezing_temp1	0.5
Air_temp	4.6	Wind_S	2.5	Concentration1	7.3	Rain_nf	1	Freezing_temp2	1.1
Dew_point	5.5	Wind_D	119	Concentration2	321.9	Relay_state	1	Max_windS	3.9
Grip	0.04	Visibility	1	Conductivity1	3.5	Snow_h	2	Water_t1	0.06
Water_L	0.11	Present_weather	1	Conductivity2	2.975	Base_temp	2.9	Water_t2	0.01

on the main training set, to store and use them on later data by means of the *transform* method [79]. Tables 8 and 9 show the stored median and IQR values of the original observations, respectively.

6.4. Stage 4- Feature selection

The features should be selected after the feature scaling (FE stage), using the main training set to avoid the ML model from overfitting. Filter methods were applied for FS in this study. Filter methods remove redundant features independently and quickly, based on the characteristics of the features in the dataset [80]. Filter methods include constant features, quasi-constant features, correlation and statistical measures [81].

6.4.1. Removing constant variables

Constant variables show the same value for all the observations in the dataset. In this paper, constant features were identified by selecting the variance threshold, using Scikit-learn in Python. The variance threshold was set to be zero. Visibility_status and Relay_state are two constant features that were removed from the dataset.

6.4.2. Removing quasi-constant variables

Quasi-constant variables show a similar value for the majority of observations. In this study, quasi-constant features were identified by selecting a variance threshold equal to 0.01 (0.01 illustrates that proportion of a single observation in one feature is around 0.99). In this step, Ice_L and General_status are two quasi-constant features that were removed from the dataset.

6.4.3. Removing high correlated features

Correlation calculates the linear relationship between two or more variables. Correlated features do not give us any information. Therefore, features should be uncorrelated with each other, but they should correlate with the output [82]. When two features have a high correlation with each other, the second feature is not able to add much information over the first feature, and it can help us to reduce features. Pearson's correlation coefficient has been used in this study. It calculates the strength of linear relationships among variables. This value varies between −1 (negative linear relationship) and +1 (positive linear relationship). If there is a positive relationship, increasing one variable causes an increase in another variable. If there is a negative relationship, an increase in one variable results in a reduction in

Table 10
Description of Pearson's correlation coefficient.

Pearson's correlation coefficient	Description
$0.0 \leq r \leq 0.19$	Weak
$0.2 \leq r \leq 0.39$	Mild
$0.4 \leq r \leq 0.59$	Moderate
$0.6 \leq r \leq 0.79$	Moderately strong
$0.8 \leq r \leq 1.0$	Strong

Table 11
Highly correlated feature groups in the first repetition (group 1).

Feature 1	Feature 2	Corr
Ground_temp1	Ground_temp2	0.999
Ground_temp1	Surface_temp1	0.960
Ground_temp1	Surface_temp2	0.949
Ground_temp1	Air_temp	0.936
Ground_temp1	Battery_voltage	0.882
Ground_temp1	Base_temp	0.740
Ground_temp1	Dew_point	0.722
Ground_temp1	Surface_state1	0.601

Table 12
Highly correlated feature groups in the first repetition (group 2).

Feature 1	Feature 2	Corr
Chemical1	Freezing_temp1	0.976
Chemical1	Chemical2	0.926
Chemical1	Freezing_temp2	0.860
Chemical1	Lfreezing1	0.783
Chemical1	Lfreezing2	0.701

Table 13
Highly correlated feature groups in the first repetition (group 3).

Feature 1	Feature 2	Corr
Max_windS	Wind_S	0.970

Table 14
Highly correlated feature groups in the first repetition (group 4).

Feature 1	Feature 2	Corr
Alarm2	Alarm1	0.876
Alarm2	Grip	0.619

another variable. If the value is -1 and $+1$, there is a strong linear relationship (manner) between two variables and, if this value is zero, there is no linear relationship between two variables [83]. The general rule (rule of thumb) for interpreting the strength of linear relationships between variables (absolute value) is based on Table 10 [84]:

To find the highly correlated features, we decided to follow the procedure that can discover the different groups of features [85] with a correlation coefficient $\geq$0.6. This procedure gives us more insight into which features we should keep or ignore. This process repeats until all correlated features (Pearson's correlation coefficient $\geq$0.6) are removed. Removing the features is based on the decision maker's choice (i.e. which feature they decide to keep, according to its importance).

In the first repetition, 11 correlated groups were found out of 45 total features. The results are shown in Tables 11–21. We decided to remove Ground_temp1, Chemical1, Max_windS, Alarm2, Alarm, Conductivity, Snow_h, Surface_state, Water_t1, Surface_state2, and Present_weather from the main training set and testing set.

In the second repetition, six correlated groups were found. The results are shown in Tables 22–27. We decided to remove Surface_temp1, Chemical2, Lfreezing2, Base_temp, grip, and Conductivity1 from the main training set and testing set.

In the third repetition, three correlated groups were found. The results are shown in Tables 28–30. We decided to remove Surface_temp2, Freezing_temp1, and Snow_L from the main training set and testing set.

Table 15
Highly correlated feature groups in the first repetition (group 5).

Feature 1	Feature 2	Corr
Alarm	Grip	0.822
Alarm	Snow_L	0.615

Table 16
Highly correlated feature groups in the first repetition (group 6).

Feature 1	Feature 2	Corr
Conductivity2	Conductivity1	0.790

Table 17
Highly correlated feature groups in the first repetition (group 7).

Feature 1	Feature 2	Corr
Snow_h	Precipitation_24	0.767
Snow_h	Grip	0.618

Table 18
Highly correlated feature groups in the first repetition (group 8).

Feature 1	Feature 2	Corr
Surface_state	Conductivity1	0.723
Surface_state	Humidity	0.707

Table 19
Highly correlated feature groups in the first repetition (group 9).

Feature 1	Feature 2	corr
Water_t1	Snow_L	0.710
Water_t1	Grip	0.631
Water_t1	Water_t2	0.625

Table 20
Highly correlated feature groups in the first repetition (group 10).

Feature 1	Feature 2	Corr
Surface_state2	Surface_state1	0.667

Table 21
Highly correlated feature groups in the first repetition (group 11).

Feature 1	Feature 2	Corr
Present_weather	Rain_state	0.653

Table 22
Highly correlated feature groups in the second repetition (group 1).

Feature 1	Feature 2	Corr
Surface_temp1	Surface_temp2	0.997
Surface_temp1	Ground_temp2	0.956
Surface_temp1	Air_temp	0.943
Surface_temp1	Battery_voltage	0.908
Surface_temp1	Dew_point	0.747
Surface_temp1	Surface_state1	0.608

In the fourth repetition, three correlated groups were found. The results are shown in Tables 31–33. We decided to remove Battery_voltage, Freezing_temp2, and Ground_temp2 from the main training set and testing set.

In the last repetition, one correlated group was found. The results are shown in Table 34. We decided to remove Dew_point from main training set and testing set

Thus, 17 features were selected after these five iterations. These features are: Surface_state1, Air_temp, Water_L, Humidity, Rain_state, Rain_int, Wind_S, Wind_D, Visibility, Precipitation_24, Alarm1, Concentration1, Concentration2, Pressure, Rain_nf, Lfreezing1, and Water_t2.

Table 23
Highly correlated feature groups in the second repetition (group 2).

Feature 1	Feature 2	Corr
Chemical2	Freezing_temp1	0.897
Chemical2	Freezing_temp2	0.883
Chemical2	Lfreezing1	0.627

Table 24
Highly correlated feature groups in the second repetition (group 3).

Feature 1	Feature 2	Corr
Lfreezing2	Freezing_temp2	0.830
Lfreezing2	Lfreezing1	0.804
Lfreezing2	Freezing_temp1	0.750

Table 25
Highly correlated feature groups in the second repetition (group 4).

Feature 1	Feature 2	Corr
Base_temp	Ground_temp2	0.744
Base_temp	Air_temp	0.602

Table 26
Highly correlated feature groups in the second repetition (group 5).

Feature 1	Feature 2	Corr
Grip	Snow_L	0.694
Grip	Alarm1	0.638

Table 27
Highly correlated feature groups in the second repetition (group 6).

Feature 1	Feature 2	Corr
Conductivity1	Surface_state1	0.636

Table 28
Highly correlated feature groups in the third repetition (group 1).

Feature 1	Feature 2	Corr
Surface_temp2	Ground_temp2	0.946
Surface_temp2	Air_temp	0.929
Surface_temp2	Battery_voltage	0.901
Surface_temp2	Dew_point	0.726
Surface_temp2	Surface_state1	0.609

Table 29
Highly correlated feature groups in the third repetition (group 2).

Feature 1	Feature 2	Corr
Freezing_temp1	Freezing_temp2	0.892
Freezing_temp1	Lfreezing1	0.840

Table 30
Highly correlated feature groups in the third repetition (group 3).

Feature 1	Feature 2	Corr
Snow_L	Alarm1	0.616

6.4.4. Statistical measures

In the next step, a statistical measure, called the univariate approach, was used to build a decision tree regression, based on each variable in the main training set, to predict the output. This method ranks variables based on the MSE and then selects the features with the highest ranks. The lower MSE shows the better performance of the ML model [85]. MSE values calculated based on this method showed 11 features with a high predictive performance. Therefore, six low predictive variables were removed in this procedure, including Visibility, Rain_nf, Concentration1, Rain_int, Rain_state, and Water_t2. Consequently, 11 features were selected as predictors (Surface_state1, Air_temp, Water_L, Humidity, Wind_S, Wind_D, Precipitation_24, Alarm1,

Table 31
Highly correlated feature groups in the fourth repetition (group 1).

Feature 1	Feature 2	Corr
Battery_voltage	Air_temp	0.938
Battery_voltage	Ground_temp2	0.876
Battery_voltage	Dew_point	0.767

Table 32
Highly correlated feature groups in the fourth repetition (group 2).

Feature 1	Feature 2	Corr
Freezing_temp2	Lfreezing1	0.754

Table 33
Highly correlated feature groups in the fourth repetition (group 3).

Feature 1	Feature 2	Corr
Ground_temp2	Surface_state1	0.602

Table 34
The highly correlated feature group in the fifth repetition.

Feature 1	Feature 2	Corr
Dew_point	Air_temp	0.847

Concentration2, Pressure, LFreezing1), ready for use in ML models to generate predicted values. The number of observations in the main training dataset and testing dataset is 2678 and 1149, respectively.

6.5. Stage 5- Cross validation and model evaluation

An attempt has made to build four different ML models (SVR, MLP, LR, and RF) [74] to predict the surface temperature every 10 min, based on the testing set.

6.5.1. Cross validation

Five-fold ($k = 5$) cross validation is chosen to validate the ML models. This method helps us to generate the learning curves. Plotting the learning curves requires different training sizes. The lowest number of training sizes is one. The maximum number of training sizes depends on the ratio of the learning (training) set to the validation set. The main training dataset have been split into learning (training) and validation sets with the ratio of 80:20. This ratio means that 80% of instances are in the learning (training) set and 20% put aside for the validation set. Therefore, the learning (training) set has 2142 instances ($2678 \times 0.80 = 2142$) and the validation set has 536 instances ($2678 \times 0.20 = 536$). That is why the maximum number of training sizes is 2142. For this case study, six sizes have been used: [1, 100, 500, 1000, 1500, 2142].

6.5.2. Model evaluation and experimental results

Table 35 depicts the MSE values for learning (training) and validation sets, which shows by increasing the number of data points, the MSE value decreases in the validation sets until they converge to the irreducible errors. The irreducible errors for SVR, MLP, LR, and RF are approximately 0.406, 0.489, 1.011, and 1.047, respectively. Moreover, the training speeds of all the models are fast and they are easy to implement. Therefore, cross validation provided us with an indication for the performance of the different regression algorithms used in this study. Graphical results obtained from the applied methods are shown in Figs. 9, 10, and 11. Fig. 9 displays the learning curves for the learning (training) and validation sets. Fig. 10 shows the scatter plots of true values over predicted values, while Fig. 11 displays the residual plots, which mark the difference between the actual observations and the predicted values. Numerical results obtained from the applied methods are illustrated in Table 36, which includes the MSE, RMSE, and R^2 for the testing sets.

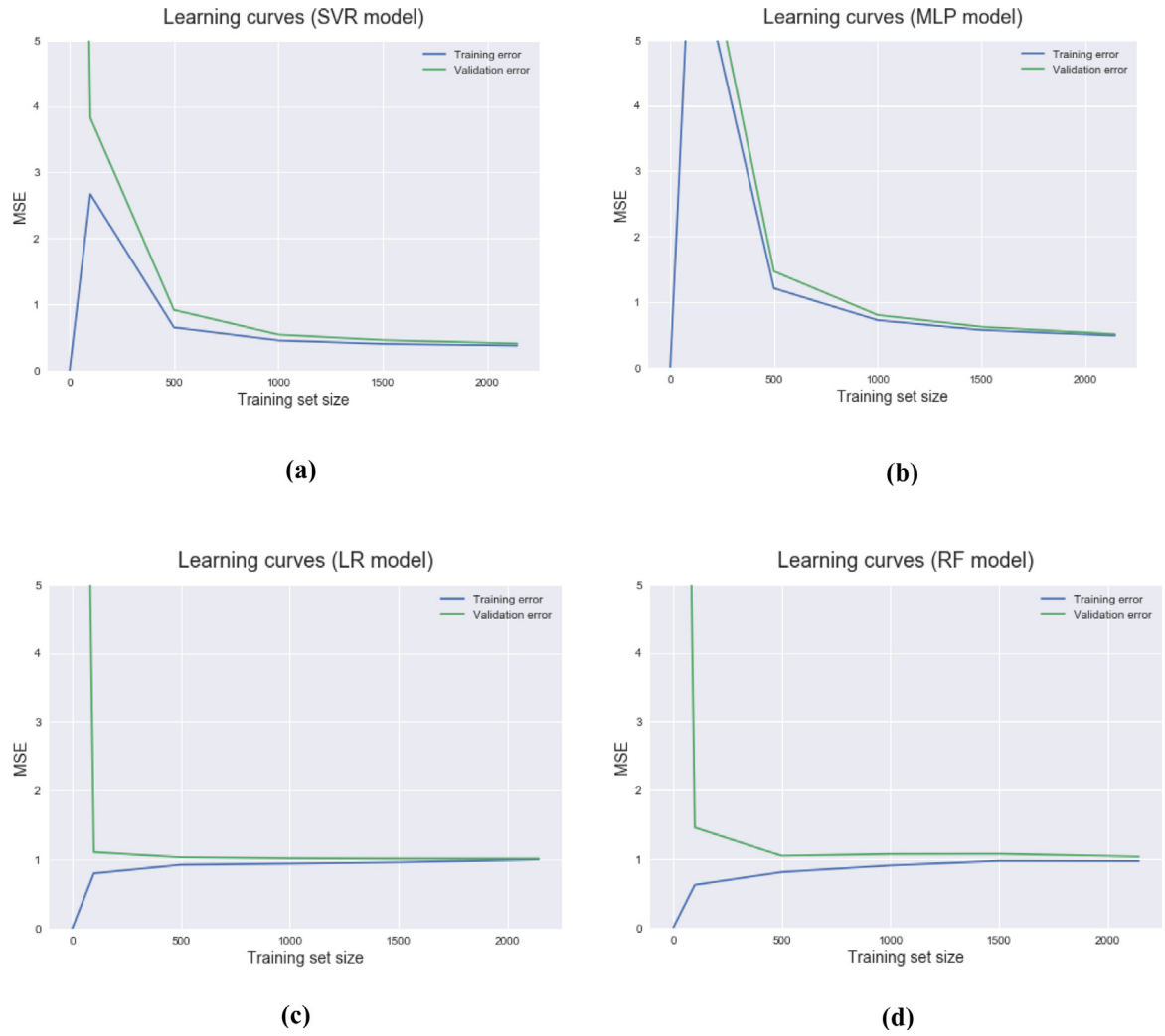

Fig. 9. Learning curves achieved from (a) SVR model, (b) MLP model, (c) LR model, (d) RF model.

Table 35
MSE values for learning (training) and validation sets based on diverse training sizes for the SVR, MLP, LR, and RFC models.

Training size	Mean learning (training) scores				Mean validation scores			
	SVR	MLP	LR	RF	SVR	MLP	LR	RF
1	0.00	0.434	0.00	0.00	24.087	17.462	24.087	24.087
100	2.671	6.711	0.799	0.608	3.826	7.167	1.106	1.406
500	0.650	1.099	0.925	0.802	0.914	1.294	1.032	1.045
1000	0.453	0.707	0.941	0.902	0.542	0.789	1.016	1.060
1500	0.400	0.576	0.960	0.921	0.459	0.625	1.012	1.029
2142	0.373	0.468	0.999	0.982	0.406	0.489	1.011	1.047

Table 36
Evaluation metrics for SVR, MLP, LR, and RF models.

	SVR	MLP	LR	RF
Model evaluation	MSE: 0.395 RMSE: 0.628 R^2: 0.976	MSE: 0.467 RMSE: 0.683 R^2: 0.972	MSE: 0.955 RMSE: 0.977 R^2: 0.943	MSE: 1.038 RMSE: 1.019 R^2: 0.938

6.5.2.1. Learning Curves. Some information can be extracted from the learning curves (Fig. 9). The gap between the two curves shows the variance. In the starting point, when the training size is small, this gap is wide (high variance) and adding more training data is more likely to be helpful. Hence, as the number of training data points is increasing, the gap between the two curves is becoming narrower. Here, we explain the learning curve for the SVR model. As is clear in the plot, the MSE for the training curve is zero once the training size is one. This demonstrates a normal manner because the model does not face any problem in fitting one data point as perfectly as possible. However, the value of the MSE for the validation set in this situation is drastically high (24.087). We change the y-axis limitation to be between 0 and 5, to see the graph clearly. In fact, this amount of high error is not strange, as the model is only trained by one single data point, and it is unable to generalize precisely for unseen instances. Moreover,

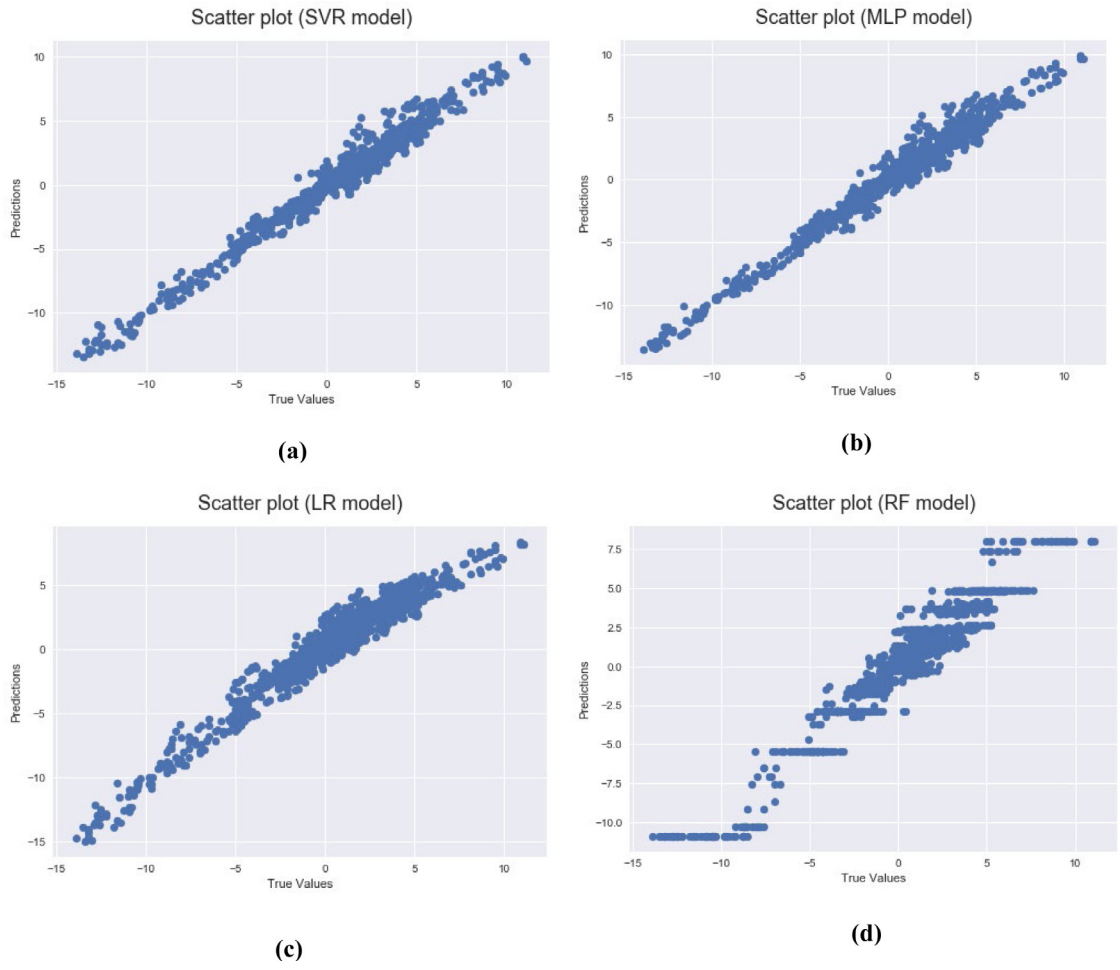

Fig. 10. Scatter plot achieved from (a) SVR model, (b) MLP model, (c) LR model, (d) RF model.

as the training size reaches 100, the MSE for the training curve increases to approximately 2.671, whereas there is a reduction in the MSE for the validation set (it decreases to 3.826). Therefore, the model (here, it is SVR) cannot predict all 100 data points in the training set accurately. Nevertheless, the model performance is better for the validation set, due to the growth in the number of data points. When the training size increases to 500, the MSE for the training set decreases to 0.65, while the MSE for the validation set drops to 0.914. After this trend (1000 and more) two curves are converged (low bias and low variance). The converged point shows the irreducible error [62].

6.5.2.2. Scatter plots. With the exception of RF, scatter plots for SVR, MLP and LR are straight lines. However, the flawless straight line of the scatter plot for the SVR and the MLP models show that they are the perfect prediction models.

6.5.2.3. Residual plots. Although the residual plots for all the models are normally distributed, the SVR and MLP residual plots show better performance. This means that SVR and MLP are the correct selections for our dataset (observed data). If they were not normally distributed and there was some strange behavior in the residual plots, they would not be correct choices, due to the characteristics of the dataset.

6.5.2.4. Numerical results (MSE, RMSE, and R^2). In addition, the MSE values for SVR, MLP, LR and RF are 0.395, 0.467, 0.955, and 1.038, respectively, while the RMSE values for SVR, MLP, LR, and RF are 0.628, 0.683, 0.977, and 1.019, respectively. Therefore, both the MSE and RMSE values are the lowest values in the SVR model, and we can explain around 98% of the variance ($R^2 = 0.98$) based on the SVR model, while this number is almost 97% for the MLP model and 94% for both the LR and RF models. Thus, SVR is the most accurate model and can be selected as the better estimator, due to good generalization performance. Furthermore, the MLP model has shown tiny differences in both graphical and numerical results and has performed much better than LR and RF.

It is good to mention that, in the LR model, the estimated intercept ($\hat{b}_0$) is 1.301, and the estimated coefficients ($\widehat{b_i}$) are shown in Table 37.

Table 37
Estimated coefficients for LR model.

Variable	Estimated coefficients ($\widehat{b_i}$)
Surface_state1	0.614
Air_temp	3.637
Water_L	−0.145
Humidity	−0.088
Wind_S	0.097
Wind_D	−0.171
Precipitation_24	0.110
Alarm1	0.382
Concentration2	−0.109
Pressure	0.242
Lfreezing1	0.0258

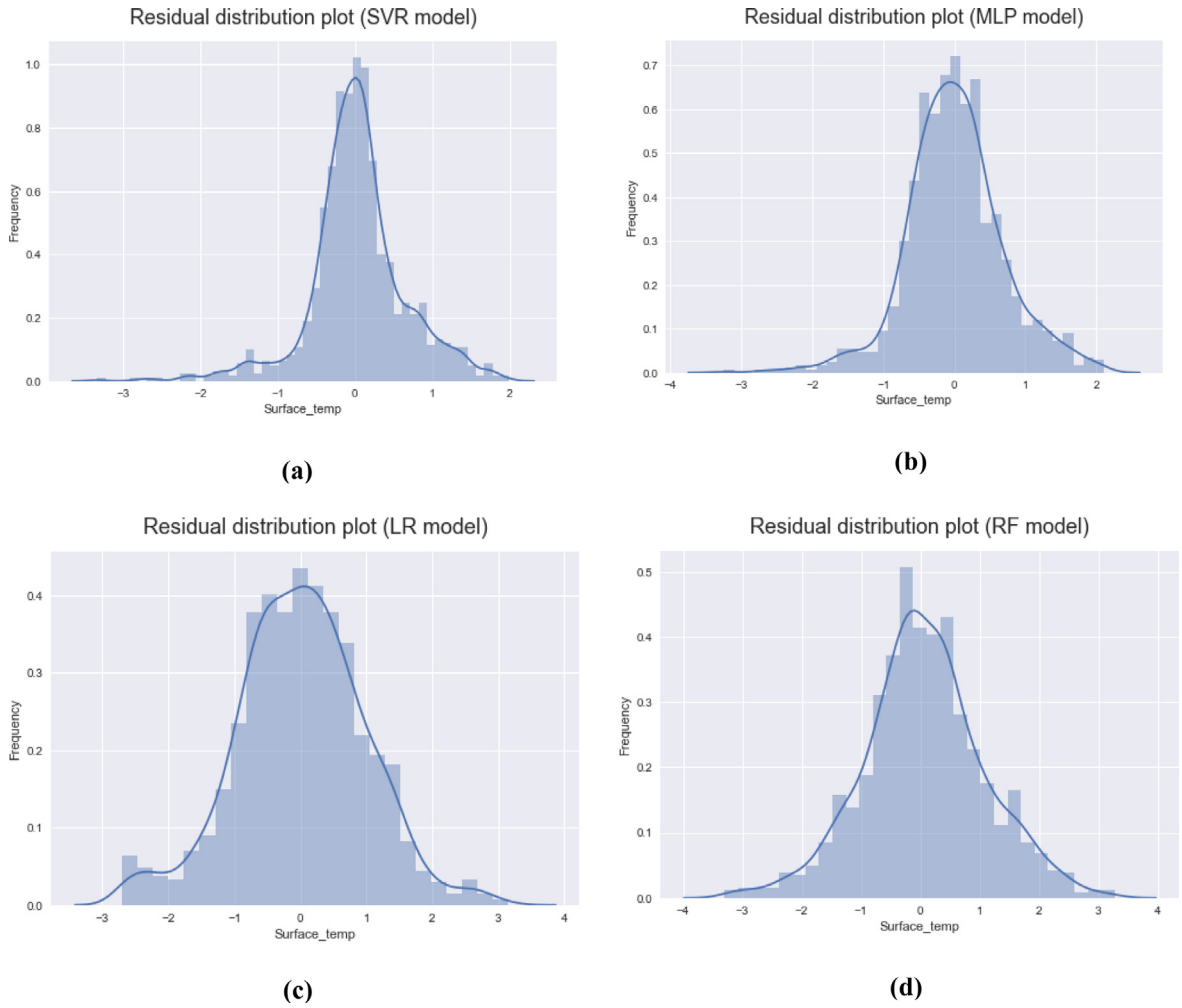

Fig. 11. Residual plot achieved from (a) SVR model, (b) MLP model, (c) LR model, (d) RF model.

7. Conclusions

This study presents a five-stage framework for intelligent predictive winter road maintenance, by combining data envelopment analysis and machine learning techniques to develop decision support systems for decision-making units (roads). In the first stage, the efficiency of DMUs is measured by the DEA-CCR model for different units. If the unit is efficient, there is no need for further evaluation because it can save time and cost. However, inefficient units must be taken into consideration for further assessment. In the second stage, the related data are collected and analyzed to provide us with statistical information for different variables. In the third stage (feature engineering), the dataset is split into main training dataset and testing dataset, and then rare categorical variable labels are encoded; after that, categorical observations are quantified for the sake of being readable for machine learning techniques. Lastly, input observations are scaled through the robust scaler method. In the next stage (feature selection), unrelated features are filtered out to retain relevant features that have a significant impact on the output (road surface temperature). Finally, in the cross validation and model evaluation stage, validation set scores and learning (training) set scores are estimated, based on different training sizes and the k-fold cross validation method. Then the SVR, MLP, LR, and RF models are trained to see whether the models can generate predicted road surface temperature values accurately or not. The graphical results achieved by learning curves have shown that the applied models avoid overfitting and underfitting. The scatter plots for the SVR, MLP and LR models fulfill the requirement of the straight line. Moreover, the residual plots for the applied models are normally distributed. Nonetheless, these three graphs show that SVR and MLP models have the best visual performance. The numerical outcomes indicate that SVR has achieved the lowest mean square error and root mean square error values, whereas RF has obtained the biggest error values. Moreover, the variance explained by SVR is better than in the other models. However, the difference in numerical values between SVR and MLP is small. Hence, both SVR and MLP models have shown good performance, but SVR is more accurate than MLP, according to the evaluation metrics, and the irreducible error in the learning curves.

The proposed methodology is not limited to predicting surface temperature in winter. It can be applied in other prediction applications. ICWRM is easy to implement, with a fast training process, and it has high prediction accuracy. It is also possible, to combine the machine learning techniques with optimization algorithms, to find the best network parameters (i.e. number of neurons or hidden layers of MLP), in order to enhance the accuracy of prediction. However, this methodology is highly dependent on the data quality, the data quantity, and the adjustment of different parameters in the models. Therefore, massive data is required to achieve accurate cost-effective WRM. For future work, alternative methods with lower dependency on data can be found. Moreover, vehicle routing problem can be formulated to increase the WRM efficiency in a road, because in different parts of the road, surface

Table 38
List of abbreviations.

Abbreviation	Explanation
ADT	Average Daily Traffic
ANN	Artificial Neural Network
BCC	Banker, Charnes, and Cooper
CCR	Charnes, Cooper, and Rhodes
Corr	Correlation
DEA	Data Envelopment Analysis
DMUs	Decision Making Units
DSS	Decision Support Systems
FE	Feature Engineering
FS	Feature Selection
GIS	Geographic Information Systems
ICWRM	Intelligent Cost-effective Winter Road Maintenance
IQR	Interquartile Quantile Range
KKT	Karush–Kuhn–Tucker
LR	Linear Regression
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Square Error
OLS	Ordinary Least Square
RBFs	Radial Basis Functions
RF	Random Forest
RMSE	Root Mean Square Error
RST	Road Surface Temperature
RWS	Road Weather Stations
SVC	Support Vector Classification
SVM	Support Vector Machine
SVR	Support Vector Regression
WRM	Winter Road Maintenance

temperature can change due to different road geometry and average daily traffic.

List of abbreviations

Table 38 shows all the abbreviations used in this research article.

CRediT authorship contribution statement

Mahshid Hatamzad: Visualization, Analysis, Methodology, Software, Technical results, Validation, Investigation, Writing – original draft, Writing – review & editing. **Geanette Cleotilde Polanco Pinerez:** Supervision, Project administration, Writing – review & editing. **Johan Casselgren:** Supervision, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would specially like to express their sincere appreciation to anonymous reviewers for their insightful comments that extremely helped us to improve the quality of our research article. In addition, the publication charges for this article have been funded by a grant from the publication fund of UiT/ The Arctic University of Norway.

References

[1] J. Xiao, Z. Xiao, D. Wang, J. Bai, V. Havyarimana, F. Zeng, Short-term traffic volume prediction by ensemble learning in concept drifting environments, Knowl.-Based Syst. 164 (2019) 213–225.
[2] H.-Y. Cheng, V. Gau, C.-W. Huang, J.-N. Hwang, Advanced formation and delivery of traffic information in intelligent transportation systems, Expert Syst. Appl. 39 (9) (2012) 8356–8368.
[3] K. Tang, S. Chen, A.J. Khattak, Personalized travel time estimation for urban road networks: A tensor-based context-aware approach, Expert Syst. Appl. 103 (2018) 118–132.
[4] Z.E. Abou Elassad, H. Mousannif, H. Al Moatassime, A proactive decision support system for predicting traffic crash events: A critical analysis of imbalanced class distribution, Knowl.-Based Syst. 205 (2020) 1–14.
[5] A. Bezuglov, G. Comert, Short-term freeway traffic parameter prediction: Application of grey system theory models, Expert Syst. Appl. 62 (2016) 284–292.
[6] W.-H. Lee, S.-S. Tseng, S.-H. Tsai, A knowledge based real-time travel time prediction system for urban network, Expert Syst. Appl. 36 (3) (2009) 4239–4247.
[7] K.C. Dey, A. Mishra, M. Chowdhury, Potential of intelligent transportation systems in mitigating adverse weather impacts on road mobility: a review, IEEE Trans. Intell. Transp. Syst. 16 (3) (2014) 1107–1119.
[8] V.J. Berrocal, A.E. Raftery, T. Gneiting, R.C. Steed, Probabilistic weather forecasting for winter road maintenance, J. Amer. Statist. Assoc. 105 (490) (2010) 522–537.
[9] V.H. Run, et al., A review of environmental impacts of winter road maintenance, Cold Reg. Sci. Technol. 158 (2019) 143–153.
[10] S. Asadi, E. Hadavandi, F. Mehmanpazir, M.M. Nakhostin, Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction, Knowl.-Based Syst. 35 (2012) 245–258.
[11] B. Liu, et al., Road surface temperature prediction based on gradient extreme learning machine boosting, Comput. Ind. 99 (2018) 294–302.
[12] A. Wibisono, W. Jatmiko, H.A. Wisesa, B. Hardjono, P. Mursanto, Traffic big data prediction and visualization using fast incremental model trees-drift detection (FIMT-DD), Knowl.-Based Syst. 93 (2016) 33–46.
[13] C. Ahabchane, M. Trépanier, A. Langevin, Street-segment-based salt and abrasive prediction for winter maintenance using machine learning and GIS, Willey Trans. GIS 23 (1) (2018) 48–69.
[14] S. Roychowdhury, M. Zhao, A. Wallin, N. Ohlsson, M. Jonasson, Machine learning models for road surface and friction estimation using front-camera images, in: International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.
[15] G. Panahandeh, E. Ek, N. Mohammadiha, Road friction estimation for connected vehicles using supervised machine learning, in: IEEE Intelligent Vehicles Symposium (IV), IEEE, 2017, pp. 1262–1267.
[16] Z. Ye, X. Shi, C.K. Strong, T.H. Greenfield, Evaluation of effects of weather information on winter maintenance costs, Transp. Res. Rec. 2107 (1) (2009) 104–110.
[17] B. Xu, H.-C. Dan, L. Li, Temperature prediction model of asphalt pavement in cold regions based on an improved BP neural network, Appl. Therm. Eng. 120 (2017) 568–580.
[18] M.E. Ozbek, Development of a Comprehensive Framework for the Efficiency Measurement of Road Maintenance Strategies using Data Envelopment Analysis (Ph.D. thesis), Department of Computer Science, Virginia Tech, Blacksburg, Virginia, 2007.
[19] M. Hatamzad, G. Polanco Pinerez, Non-parametric linear technique for measuring the efficiency of winter road maintenance in the arctic area, Int. J. Ind. Manuf. Eng. 13 (11) (2019) 678–683.
[20] W.R. Trenouth, B. Gharabaghi, N. Perera, Road salt application planning tool for winter de-icing operations, J. Hydrol. 524 (2015) 401–410.
[21] T. Kramberger, J. Žerovnik, Environment, A contribution to environmentally friendly winter road maintenance: Optimizing road de-icing, Transp. Res. D 13 (5) (2008) 340–346.
[22] M. Riehm, Measurements for Winter Road Maintenance (Ph.D thesis), Department of Land and Water Resources Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden, 2012.
[23] K. Vincova, Using DEA models to measure efficiency, Biatec 13 (8) (2005) 24–28.
[24] D.D. Wu, Z. Yang, L. Liang, Using DEA-neural network approach to evaluate branch efficiency of a large Canadian bank, Expert Syst. Appl. 31 (1) (2006) 108–115.
[25] A. Charnes, W.W. Cooper, E. Rhodes, Measuring the efficiency of decision making units, European J. Oper. Res. 2 (6) (1978) 429–444.
[26] D.C. Montgomery, E.A. Peck, G.G. Vining, Introduction to Linear Regression Analysis, John Wiley & Sons, 2012, pp. 12–15.
[27] H.A. Farahani, A. Rahiminezhad, L. Same, A comparison of Partial Least Squares (PLS) and Ordinary Least Squares (OLS) regressions in predicting of couples mental health based on their communicational patterns, Procedia-Soc. Behav. Sci. 5 (2010) 1459–1463.
[28] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, first ed., Springer, 2013, pp. 59–70.
[29] S. Rong, Z. Bao-wen, The research of regression model in machine learning field, in: MATEC Web of Conferences, Vol. 176, EDP Sciences, 2018, p. 01033.

[30] R. Decook, Simple linear regression and correlation, 2016, Retrieved from http://homepage.divms.uiowa.edu/~rdecook/stat2020/notes/ch11_pt1.pdf.
[31] D. Nguyen, N.A. Smith, C. Rose, Author age prediction from text using linear regression, in: Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, 2011, pp. 115–123.
[32] C. Voyant, et al., Machine learning methods for solar radiation forecasting: A review, Renew. Energy 105 (2017) 569–582.
[33] M. Belkin, D. Hsu, S. Ma, S. Mandal, Reconciling modern machine-learning practice and the classical bias–variance trade-off, Proc. Natl. Acad. Sci. 116 (32) (2019) 15849–15854.
[34] P. Domingos, A few useful things to know about machine learning, Commun. ACM 55 (10) (2012) 78–87.
[35] J. Verrelst, et al., Machine learning regression algorithms for biophysical parameter retrieval: Opportunities for Sentinel-2 and-3, Remote Sens. Environ. 118 (2012) 127–139.
[36] K. Kourou, T.P. Exarchos, K.P. Exarchos, M.V. Karamouzis, D.I. Fotiadis, Machine learning applications in cancer prognosis and prediction, Comput. Struct. Biotechnol. J. 13 (2015) 8–17.
[37] A.A. Soofi, A. Awan, Classification techniques in machine learning: applications and issues, J. Basic Appl. Sci. 13 (2017) 459–465.
[38] A. L'heureux, K. Grolinger, H.F. Elyamany, M.A. Capretz, Machine learning with big data: Challenges and approaches, IEEE Access 5 (2017) 7776–7797.
[39] T. Rawat, V. Khemchandani, Feature engineering (FE) tools and techniques for better classification performance, Int. J. Innov. Eng. Technol. (IJIET) 8 (2) (2017) 169–179.
[40] A. Jović, K. Brkić, N. Bogunović, A review of feature selection methods with applications, in: 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, 2015, pp. 1200–1205.
[41] N.C. Guan, M.S.B. Yusoff, Missing values in data analysis: Ignore or impute? Educ. Med. J. 3 (1) (2011) 6–11.
[42] H.J. Seltman, Experimental Design and Analysis, Department of Statistics at Carnegie Mellon (Online Only), 2009, pp. 15–16.
[43] I.B. Mohamad, D. Usman, Standardization and its effects on K-means clustering algorithm, Res. J. Appl. Sci. Eng. Technol. 6 (17) (2013) 3299–3303.
[44] V. Vapnik, S.E. Golowich, A.J. Smola, Support vector method for function approximation, regression estimation and signal processing, in: Advances in Neural Information Processing Systems, 1997, pp. 281–287.
[45] D. Basak, S. Pal, D.C. Patranabis, Support vector regression, Neural Inf. Process. – Lett. Rev. 11 (10) (2007) 203–224.
[46] M. Awad, R. Khanna, Support vector regression, in: Efficient Learning Machines, Apress, Springer, Berkeley, CA, 2015, pp. 67–80, http://dx.doi.org/10.1007/978-1-4302-5990-9_4.
[47] H. Son, C. Kim, C. Kim, Hybrid principal component analysis and support vector machine model for predicting the cost performance of commercial building projects using pre-project planning variables, Autom. Constr. 27 (2012) 60–66.
[48] J. Wang, L. Li, D. Niu, Z. Tan, An annual load forecasting model based on support vector regression with differential evolution algorithm, Appl. Energy 94 (2012) 65–70.
[49] M. Farahmand, M.I. Desa, M. Nilashi, A combined data envelopment analysis and support vector regression for efficiency evaluation of large decision making units, Int. J. Eng. Technol. (IJET) 6 (5) (2014) 2310–2321.
[50] Y. Wang, T. Shen, G. Yuan, J. Bian, X. Fu, Appearance-based gaze estimation using deep features and random forest regression, Knowl.-Based Syst. 110 (2016) 293–301.
[51] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[52] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[53] Q. Zhou, H. Zhou, T. Li, Cost-sensitive feature selection using random forest: Selecting low-cost subsets of informative features, Knowl.-Based Syst. 95 (2016) 1–11.
[54] L. Ekonomou, Greek long-term energy consumption prediction using artificial neural networks, Energy 35 (2) (2010) 512–517.
[55] P. Refaeilzadeh, L. Tang, L. Huan, Arizona State University Encyclopedia of Database Systems, Springer US, 2009.
[56] D. Berrar, Cross-validation, in: Encyclopedia of Bioinformatics and Computational Biology, 2019, pp. 542–545.
[57] Scikitlearn. validation curves: plotting scores to evaluate models. Retrieved from https://scikit-learn.org/stable/modules/learning_curve.html#learning-curve.
[58] M. Waseem, What is bias-variance in machine learning?, 2019, Retrieved from https://www.edureka.co/blog/bias--variance-in-machine-learning/.
[59] Z. Hasan, Bias variance trade-off and learning curve, 2020, Retrieved from https://zahidhasan.github.io/2020/10/13/bias--variance-trade-off-and-learning-curve.html.
[60] S. Fortmann-Roe, Understanding the bias-variance tradeoff, 2012, Retrieved from http://scott.fortmann-roe.com/docs/BiasVariance.html.
[61] J. Brownlee, How to use learning curves to diagnose machine learning model performance, 2019, Retrieved from https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/.
[62] A. Olteanu, Tutorial: Learning curves for machine learning in Python, 2018, Retrieved from https://www.dataquest.io/blog/learning-curves-machine-learning/.
[63] Trevor Hastie, Jerome Friedman, Robert Tibshirani, Overview of supervised learning, in: The Elements of Statistical Learning. Springer Series in Statistics, Springer, 2001, http://dx.doi.org/10.1007/978-0-387-21606-5_2.
[64] D. Mishra, Regression: An explanation of regression metrics and what can go wrong, 2019, Retrieved from https://towardsdatascience.com/regression-an-explanation-of-regression-metrics-and-what-can-go-wrong-a39a9793d914.
[65] JJ, MAE and RMSE — Which metric is better?, 2016, Retrieved from https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d.
[66] Scikitlearn. sklearn.metrics.explained_variance_score. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.explained_variance_score.html.
[67] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, Array programming with numpy, Nature 585 (7825) (2020) 357–362, http://dx.doi.org/10.1038/s41586-020-2649-2.
[68] W. McKinney, Data structures for statistical computing in Python, in: Proc. 9th Python Sci. Conf. 2010, pp. 51–56.
[69] J.D. Hunter, Matplotlib: A 2D graphics environment, Comput. Sci. Eng. 9 (3) (2007) 90–95, http://dx.doi.org/10.1109/MCSE.2007.55.
[70] M. Waskom, O. Botvinnik, D. Kane, P. Hobson, S. Lukauskas, D. Gemperline, et al., Mwaskom/Seaborn: V0.8.1 (September 2017), [Internet]. Zenodo, 2017, http://dx.doi.org/10.5281/zenodo.883859.
[71] C. Trapnell, Cufflinks. Retrieved from: https://github.com/cole-trapnell-lab/cufflinks.
[72] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[73] Traffikverket. Retrieved from: https://www.trafikverket.se/resa-och-trafik/forskning-och-innovation/aktuell-forskning/transport-pa-vag/testsite-e18--en-vagforskningsstation/.
[74] M. Hatamzad, Retrieved from: https://github.com/MahshidHatamzad/Regression_WRM.
[75] S. Galli, Feature engineering for machine learning, 2020, Retrieved from https://www.udemy.com/course/feature-engineering-for-machine-learning/learn/lecture/7631540#overview.
[76] K. Potdar, T.S. Pardawala, C.D. Pai, A comparative study of categorical variable encoding techniques for neural network classifiers, Int. J. Comput. Appl. 175 (4) (2017) 7–9.
[77] X.H. Cao, I. Stojkovic, Z. Obradovic, A robust data scaling algorithm to improve classification accuracies in biomedical data, BMC Bioinformatics 17 (1) (2016) 359.
[78] Dask-ml, Dask_ml.preprocessing. RobustScaler, 2017, Retrieved from https://ml.dask.org/modules/generated/dask_ml.preprocessing.RobustScaler.html.
[79] Kite. RobustScaler. Retrieved from https://kite.com/python/docs/sklearn.preprocessing.RobustScaler.
[80] N. Sánchez-Maroño, A. Alonso-Betanzos, M. Tombilla-Sanromán, Filter methods for feature selection–a comparative study, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2007, pp. 178–187.
[81] J. Xie, V. Rojkova, S. Pal, S. Coggeshall, A combination of boosting and bagging for kdd cup 2009-fast scoring on a large database, in: KDD-Cup 2009 Competition, 2009, pp. 35–43.
[82] M.A. Hall, Correlation-Based Feature Selection for Machine Learning (Ph.D thesis), Department of Computer Science, The University of Waikato, Hamilton, NewZealand, 1999.
[83] N. Gogtay, U. Thatte, Principles of correlation analysis, J. Assoc. Phys. India 65 (3) (2017) 78–81.
[84] D.D. Brewer, Regression and correlation, 2001, Retrieved from: http://faculty.washington.edu/ddbrewer/s231/s231regr.htm.
[85] S. Galli, Feature selection for machine learning, 2020, Retrieved from https://www.udemy.com/course/feature-selection-for-machine-learning/learn/lecture/9341746#overview.