creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학박사학위논문

Anomaly Handling of Observational Data Based on Machine Learning

기계 학습에 기반한 관측 자료의 이상 처리

2018년 8월

서울대학교 대학원
전기·컴퓨터공학부
이민기

Anomaly Handling of Observational Data Based on Machine Learning

by

Minki Lee

School of Computer Science & Engineering
Seoul National University
2018

Abstract

Observational data collected from automated observation system have played an important role in forecasting and analyzing a large variety of phenomena. However, abnormal values are abundant in observational data due to manifold faults in observation systems. It is significant to identify and manage abnormalities. One of the most representative and important observation data is meteorological data. In this thesis, we present novel methods based on machine learning for detecting and correcting abnormal values in observations, and we test them on various kind of real-world meteorological observations.

The process to find abnormalities is called quality control in meteorology. To correct abnormal values detected by quality control procedure, we propose three estimation models based on machine learning techniques. We compare them with traditional estimation methods, interpolations. Unlike

the interpolation methods, which use only the target attribute, the proposed models utilize the additional information consisting of the associated attributes of the target point and the relevant data of the neighboring observational points. The experimental results on real-world datasets collected from accredited agencies showed that the proposed approaches estimated target values better than the interpolation methods; reducing root mean square error (RMSE) by an average of 8.35%. In other words, our methods can provide more proper values to substitute for abnormal values than previous methods can.

We also present an improved quality control method determining abnormal values in observations from a spatial point of view. Support vector regression (SVR) is used to predict the observation value. The difference between the estimated value and the actual observed value determines if the observed value is abnormal or not. In addition, SVR input variables are deliberately selected to improve SVR performance and shorten computing time. In the selection process, a multi-objective genetic algorithm is used to optimize the two objective functions, similarity and spatial dispersion. In the experiments with real-world datasets, the proposed estimation method using SVR reduced the RMSE by an average of 45.44% compared to baseline estimators whilst maintaining competitive computing times.

Keywords : Observational data, meteorological data, anomaly detection, anomaly correction, machine learning

Student Number : 2008-30233

Contents

Abstract . . . . . **i**

Contents . . . . . **iv**

I. Introduction **1**

II. Preliminary **8**

2.1 Meteorological Data 8

2.1.1 Automatic Weather Station 8

2.1.2 Quality Control 9

2.2 Decision Tree Learning 10

2.3 Artificial Neural Networks 12

2.4 Support Vector Regression 14

2.5 Genetic Algorithm 18

III. Abnormal Data Correction **21**

3.1 Traditional Approaches 21

3.1.1 Linear Interpolation 21

3.1.2 Polynomial Interpolation 22

3.1.3 Spline Interpolation 23

3.2 Machine Learning Based Approaches 23

3.3 Datasets 25

3.4 Experimental Results 30

3.4.1 Preprocessing Data 31
3.4.2 Results 33

IV. Spatial Quality Control 39
4.1 Traditional Approaches 41
4.1.1 Cressman Method 41
4.1.2 Barnes Method 43
4.2 SVR-based Approach 44
4.3 Selecting Neighboring Stations 47
4.3.1 Similarity and Spatial Dispersion 47
4.3.2 Multi-Objective Genetic Algorithm 50
4.4 Datasets 57
4.5 Experimental Results 61
4.5.1 Representation of Wind Direction 63
4.5.2 Similarity Measure 64
4.5.3 Selecting Neighboring Stations 69
4.5.4 Comparison of Estimation Models 71
4.5.5 Size of Training Set 73
4.5.6 Result of Spatial Quality Control 74

V. Conclusions 77

List of Figures

Figure 1. Recursive partitioning algorithm 12

Figure 2. An example regression tree. The leaves are the regression values for the car price 13

Figure 3. Multilayer perceptron network 14

Figure 4. Backpropagation algorithm 15

Figure 5. Soft margin loss setting for support vector regression . 18

Figure 6. Input and output of the proposed machine learning model 25

Figure 7. Locations of the 692 AWSs in South Korea [SLK14] . 27

Figure 8. Calculation of the $Z_r(e)$, the estimated value of the grid point e in the Cressman method. Only observations of stations located within the effective radius r are used. In this example, z_1 and z_2 are used to calculate $Z_r(e)$, but z_3 is not used. 43

Figure 9. Input and output of the proposed support vector regression model 45

Figure 10. Examples of neighbor selection 48

Figure 11. Illustration of Pareto-optimal solutions and Pareto Front in a 2-objective problem 51

Figure 12. The framework of our hybrid multi-objective genetic algorithm 52

Figure 13. An example of representation of the solution 53

Figure 14. Two-point crossover 55

Figure 15. The proposed spatial quality control process 56
Figure 16. Locations of the 572 automatic weather stations (AWSs) in South Korea [SLK14] 58
Figure 17. Similarity map for different meteorological elements . 65
Figure 17. Similarity map for different meteorological elements (cont.) 66
Figure 17. Similarity map for different meteorological elements (cont.) 67
Figure 17. Similarity map for different meteorological elements (cont.) 68
Figure 18. Accuracy of estimates according to the number of selected neighboring stations 70
Figure 19. Average time spent on learning one model depending on the size of the training set 75
Figure 20. Time spent on determining one value depending on the size of the training set 75

List of Tables

Table 1. Units of AWS data 26
Table 2. Limits and results of absence & physical limit test 28
Table 3. Critical points and results of step test 28
Table 4. Critical points and results of persistence test 29
Table 5. Results of internal consistency test 29
Table 6. Overall proportions of abnormal values 30
Table 7. Average running time of the estimation models 34
Table 8. Comparison of wind direction representations 34
Table 9. Performances of interpolation methods 35
Table 10.Performances of ML-based models using time-series data 36
Table 11.Performances of ML-based models using time-series data and the other elements 36
Table 12.Performances of ML-based models using time-series data, the other elements, and three neighbor station data 37
Table 13.Performances of ML-based models using time-series data, the other elements, and five neighbor station data 38
Table 14.Meteorological elements in automatic weather station (AWS) data 57
Table 15.Limits for physical limit test 59
Table 16.Maximum amount of change for step test 60
Table 17.Minimum amount of change for persistence test 60
Table 18.Results of basic quality control 62

Table 19.Accuracy of estimates according to wind direction representation 63
Table 20.Accuracy of estimates for each similarity measure 64
Table 21.Optimal number of neighboring stations per meteorological element 69
Table 22.Comparison of SVR estimation accuracy with neighboring stations selected randomly or by MOGA 71
Table 23.Comparison of estimation accuracy based on estimation model 72
Table 24.Execution time for spatial quality control based on estimation model 73
Table 25.Accuracy of estimated values based on the size of the training set 74
Table 26.Results of the proposed spatial quality control method . . 76

Chapter 1

Introduction

Automated observations of various status of nature have increased due to advances of measuring device and data processing. Large amount of collected observations enable us to establish policies and to prevent expected loss. However, data from automated observation system are not fully credible. Although the reliability of equipments has been improved, still many problems including sensor malfunction, power failure, and wire oxidation occur. Consequently, there are unignorable number of abnormal values such as missing values and inaccurate values in observed data.

Meteorological data is a prominent example of automated observation data. The collection of meteorological information, which was previously done manually, has been automated in line with computational advances. Large growth of automatic weather stations (AWSs) in last decades enabled us to get near real-time weather observation data. Meteorological data collected from weather stations can be used in many applications. For example, data collected from AWSs were used to analyze energy balance of a glacier surface in Switzerland [OK02] and to explain surface mass-balance anomalies near West Greenland [vdWGvdB$^+$05]. Meteorological observations play an important role in weather forecasting, disaster warning, and policy formulation in agriculture and various industries [SK12, KY16, CM07, SH03]. In addition, meteorological observations are

used for efficient operation of alternative energy sources such as solar power, hydropower, and wind power [KCBH13, YLB03, Kal00]. In recent years, as climate change due to global warming has accelerated, the extent of damage due to abnormal weather phenomena is increasing and becoming more difficult to predict. Therefore, there is a greater need for accurate and quantitative weather data based on meteorological observations.

However, meteorological data gathered by AWS often includes errors, and unusual metrics can be observed for a variety of reasons. Causes of unusual values include sensor malfunction, hardware error, power supply error, ambient environment change, and in some rare circumstances, abnormal weather phenomena. Quality control is achieved by several methods, ranging from simple discrimination using criteria related to physical limits to relatively complex discrimination related to spatio-temporal relationships with other observations [EGG11, Zah04, SBS$^+$04, GDE04, Gan88, FHQ04]. As the installation of AWS is expanding, and the amount and types of collected data are increasing, a fast and reliable quality control algorithm must be developed. Abnormal data identified during the quality control process are examined thoroughly by an expert and may become the subject of further research.

Quality control procedure can be regarded as a anomaly detection. Anomaly detection is called in various terms as outlier detection, novelty detection, noise detection, deviation detection or exception mining. Although they mean the same thing in many cases, details of the approaches can be different. For example, Chandola et al. [CBK09] argued that anomaly detection was distinguished from noise detection in a regard that anomalous

values are of interest to researchers and novelty detection usually means one-class classification, in which a model is created to describe normal data. A definition of anomaly or outlier depends on researcher. For this thesis, we take the definition from Barnett and Lewis: “An observation which appears to be inconsistent with the remainder of that set of data” [BL94]. Anomaly detection methodologies can be categorized into the following: i) Detecting the anomalies without prior knowledge of the data. This approach is similar to unsupervised clustering. ii) Modeling both normal data and abnormal data. This approach is similar to supervised classification which needs pre-labelled data. iii) Modeling only normal data. This approach is similar to semi-supervised recognition [HA04]. Anomaly detection for meteorological data which is not labelled as normal or abnormal uses methodologies belong to category i or iii.

If a detected abnormality is due to an error in the measurement process, it is necessary to replace the observed value with a value which is thought to be accurate [LMKM14, KKY+15, KHY+16, HKI+18]. One of our goal is to estimate the values to substitute for the abnormal values. The most intuitive approaches for filling missing values in time-series data is interpolation [Slu09, BC06]. In this thesis, we present three regression algorithms based on machine learning (ML) techniques: decision tree, artificial neural network, and support vector regression. ML-based methods have the advantage that we can use information such as other climatic variables and data collected from neighbor weather stations besides time-series data. We conducted experiments on real meteorological data collected for 6 years from 692 AWSs in South Korea. Experimental results showed that ML-based

methods resulted in improved estimation performance compared to traditional interpolation approaches.

The other goal of this thesis is to develop a quality control procedure within the framework of spatiality following the result of ML-based correction. The spatial quality control processes are distinguished from the other quality controls in the way that they compares a observational point's data against neighboring observation points' [HGS+05, Hub01, RDO92, YHG08]. Daly et al. [DGD+04] performed quality control of meteorological data metrics using climate statistics and spatial interpolation, and Sciuto et al. [SBCR09] proposed a spatial quality control procedure for daily rainfall data using a neural network. We propose a spatial quality control method, which uses values obtained from observational points surrounding the target observational point to determine spatial compatibility and estimate the value of the observation point. It is possible to determine if an observed value is abnormal or not, based on differences with the estimated value. The developed spatial quality control method uses support vector regression and a genetic algorithm. It can be applied to a wide range of meteorological elements to reflect the geographic and climatic characteristics of observation points by studying past data through support vector regression. As meteorological data is not labelled, we uses semi-supervised learning which is category iii anomaly detection according to classification of Hodge. In semi-supervised learning, we assume abnormal values are rare and most of the data are normal. Therefore, we can consider all training samples are normal. Our method checks whether the observation falls within the confidence interval formed by support vector regression using observa-

tions from surrounding stations. This process can determine whether the test sample is generated with the same distribution as normal data are generated. During pre-processing of the support vector regression, input variables i.e. the surrounding observation points are selected according to two objective functions: similarity and spatial dispersion. Multi-objective optimization is required to simultaneously optimize the objective functions that could be dependent on each other. This is effectively performed by the genetic algorithm, which improves performance and reduces execution time in this study. To verify the performance of the proposed method, we applied it to observational data measured by the Korea Meteorological Administration (KMA) for one year in 2014. Experiments on real-world data sets show that the performance of the proposed method is superior to previous methods such as the Cressman method [Cre59] and the Barnes method [Bar64], which have previously been used for spatial quality control.

This thesis has following contributions:

- *We propose a correction scheme replacing abnormal values in observations with appropriate values.*

  The regression models based on machine learning techniques are used to correct abnormal values. We use values in time-series, those of other attributes and those of other observation points as input variables of the regression models. The added features than traditional methods increase the performances of the our correction algorithms.

- *We propose a quality control scheme to detect abnormal values in observations.*

In spatial quality control, we need two values, the predicted value and the difference threshold between predicted value and observed value. We use support vector regression to predict the value of observation point. Moreover, We combine multi-objective genetic algorithm with support vector machine to improve the performance of support vector regression and reduce the time cost by selecting input variables.

- *We test the proposed schemes on large amounts of real-world datasets consisting of a variety of meteorological elements*

  We measure performances and time costs of the proposed schemes and compared them with those of the existing schemes on real-world data. The datasets we use contain over 4,000,000,000 values on 8 kinds of weather elements. We also investigate the influences of parameters through the experiments.

The remainder of this thesis is organized as follows:

- In Chapter 2, we explain characteristics of meteorological data and techniques including artificial neural network, decision tree, support vector regression and genetic algorithm, which are used in this thesis.

- In Chapter 3, we propose machine learning-based methods to correct abnormal values. They are tested on real-world weather data. Traditional methods are introduced and compared to the proposed methods.

- In Chapter 4, we propose a spatial quality control scheme using support vector regression. Also, a multi-objective genetic algorithm to

decide input variables for support vector regression is proposed. We compare them with previous methods on real-world datasets.

- In Chapter 5, we draw our conclusions.

Chapter 2

Preliminary

2.1 Meteorological Data

2.1.1 Automatic Weather Station

Measurement of the meteorological elements at the earth's surface is required by any application of remote sensing to studies of biosphere of the earth. Automatic weather stations (AWS) is an automated system that allows a computer to observe and collect numerical values of multiple meteorological elements, which include temperature, wind speed, wind direction, humidity, atmospheric pressure, cloud height, visibility, precipitation, depth of snow and solar radiation. AWSs are aimed to precisely measure and record standard meteorological elements over long term, at relatively low cost. Generally, AWSs should have meteorological sensors offering an electronic signal, electronic part to convert the sensor signal to a digital value, storage media to collect the data on site, and hardware to transmit the digital values. Additionally, the mast and mounting hardware or protective housings and power supply are important components to the system [Tan90]. Power supplies in lower power AWSs use solar panels and rechargeable batteries and others in higher power AWSs use AC power grid. Sensors in most AWSs include thermometer for measuring tempera-

ture, anemometer for measuring wind speed, wind value for measuring wind direction, hygrometer for measuring humidity, and barometer for measuring atmospheric pressure. Ceilometer for measuring cloud height, visibility sensor, rain gauge for measuring precipitation, ultrasonic snow depth sensor for measuring depth of snow, and pyranometer for measuring solar radiation are occasionally equipped too. The Automated Surface Observing System (ASOS) [aso] is a representative example of AWS which is operated by United States. In South Korea, 672 AWSs were operated by Korea Meteorological Administration by 2014 over 100,284 km^2. In these facilities, wind direction, wind speed, temperature, relative humidity, atmospheric pressure, mean sea level pressure, Rainfall occurrence, and hourly precipitation were measured. We used data collected in the KMA's AWSs for our experiments.

The development of AWS has enabled i) real-time information retrieval, ii) reduced maintenance costs, iii) increased accuracy of observations, iv) a larger amount of data, and v) easier weather observations in poorly accessible regions. The progress of technology enables AWS to be smaller and cheaper, and furthermore meteorological data can be collected even from mobilephone [HKI+18, KHY+16, KKY+15].

2.1.2 Quality Control

Collected observational data needs the examination to ensure quality of data. There are three main reasons of quality control: i) to be assured that meteorological data are proper; ii) to find incorrect data possibly bringing a wrong decision making; iii) to identify and recover complication in facility maintenance and sensor calibration [DPJ+00]. Followings are basic quality

control processes which are typically used [EGG11]:

- *Physical limit test or range test*

  If the observed value is higher or lower than the physically possible upper or lower limit, respectively, it is classed as an error.

- *Step test*

  If the difference between the current observation value and the value right before is more than a certain value, it is classed as an error.

- *Persistence test*

  A value is classified as an error when the accumulated change in the observed value within a span of time is smaller than a certain value.

- *Internal consistency test*

  Some meteorological elements are climatologically related. If observations of two elements are contradictory, they are erroneous.

2.2 Decision Tree Learning

Decision tree learning is one of the most widely-used techniques for approximating a target function represented by decision tree. In decision tree, each node specifies a test of attributes and leaves represent predicted outcomes. A solution to an NP-complete problem is required to learn an optimal decision tree [HR76]. Therefore, most algorithms for learning decision trees are heuristics such as ID3 [Qui86] and C4.5 [Qui93], which are based

on greedy search. These algorithms work in top-down manner by selecting the best variable which divide the data points at each step.

Two main categories of decision tree are classification tree and regression tree. A classification tree predicts a value which is the class to which the data belongs while a regression tree, which is suitable for this study, provides a continuous output value. Figure 1 briefly explains how a typical tree construction algorithm works. There are significant criteria in this algorithm:

- When to stop partitioning (termination criterion),
- Which is the best splitting test, and
- Which values to assign to the leaf nodes.

The algorithm chooses the best splitting test by comparing goodness of the partitions created by the test, and stops when the node is sufficiently good. Therefore, for the first and second criteria, impurity or discrepancy representing how good is the node made by each split needs to be measured and compared. Measures like entropy, the Gini index, the twoing rule, maximum difference measure are used in decision tree classification [MS95], while variance or standard deviation is used in decision tree regression as the target values are continuous [Bre17]. The most common class of the data points in the node is assigned to the leaf node in decision tree classification, while the average of the target values of the data points in the node is assigned to the leaf node in decision tree regression. In some algorithms, the probability distribution over the classes or target values is assigned to the leaf node.

input : A set of n data points, $\{ \langle x_i, y_i \rangle, i = 1, 2, \ldots, n\}$
output: A decision tree
if *termination criterion* **then**
    create $leafnode$ and assign it a class or a value;
    return *leaf node*;
else
    find the best splitting test s^*;
    create $node t$ with s^*;
    Left_branch(t) =
    RecursivePartitioningAlgorithm($\{\langle x_i, y_i \rangle : s^*(x_i) = true\}$);
    Right_branch(t) =
    RecursivePartitioningAlgorithm($\{\langle x_i, y_i \rangle : s^*(x_i) = false\}$);
    return *node t*;
end

Figure 1: Recursive partitioning algorithm

The major advantage of using decision trees is that the created model is easy to interpret and explain to executives. Also additional assumptions for statistical models are not needed. Moreover, it is relatively easy to handle missing values in samples over other learning techniques. But interactions between the variables can not be captured as only one variable is dealt during the process. It also has been pointed out that a small change in training set yield a big change in the whole tree

2.3 Artificial Neural Networks

Artificial neural networks (ANN) are computational models inspired by the functioning of biological nervous system. They have been proven to be universal function approximators by Cybenko [Cyb89]. Neural networks can provide good functional models in time-series forecasting particularly

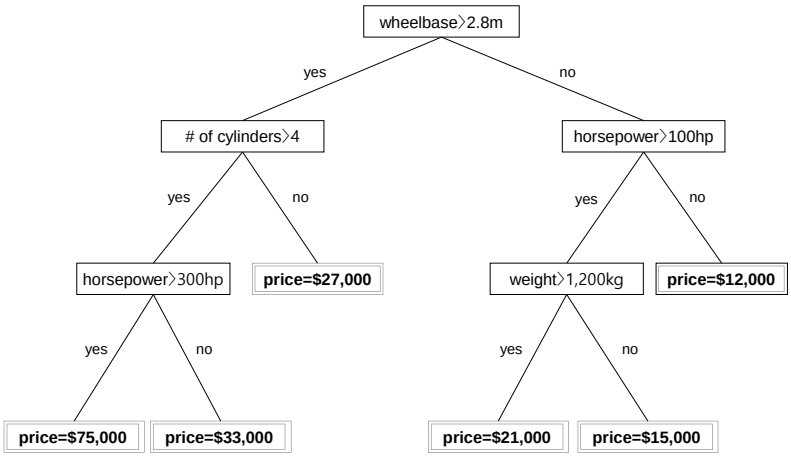


Figure 2: An example regression tree. The leaves are the regression values for the car price

[HOR96].

There are a lot of neural network variations in how to model neurons and their connections. In this study, a multilayer perceptron (MLP) [RHW85] which is one of the most wide-spread neural network models is used. MLPs are feedforward neural networks which are composed of multiple layers of neurons, or nodes. Each node is connected to other nodes at adjacent layer. Activation functions or transfer functions prevent the activation values of the nodes from being too large or too small. One of the most frequently used activation function is logistic function (a.k.a. sigmoid function):

$$f(x) = \frac{1}{1+e^{-x}}.$$

The hidden layers and non-linear transfer functions enable MLPs to represent smooth functional relationship between input and output. An simplified

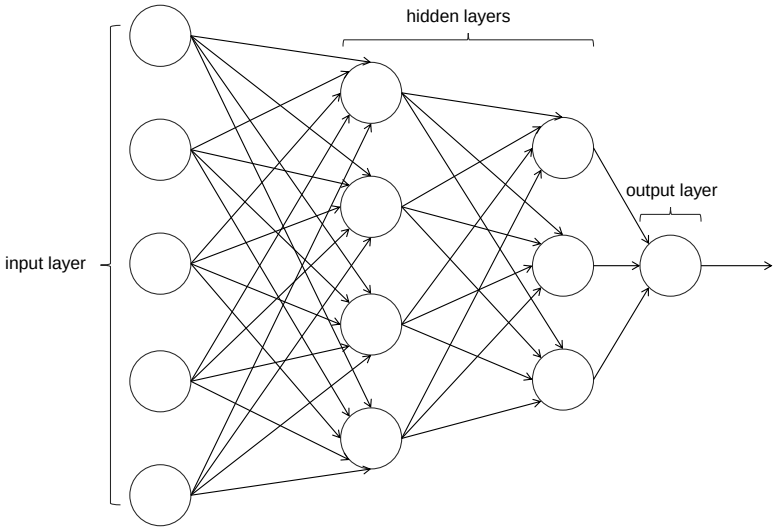


Figure 3: Multilayer perceptron network

structure of MLP with two hidden layer is demonstrated in Figure 3.

To train the networks, MLPs utilize backpropagation algorithm. Backpropagation algorithm is a generalization of the least mean squares algorithm in the linear perceptron. It tries to find the global minimum of the error surface by a gradient descent procedure. Figure 4 describes the pseudo-code of a backpropagation algorithm.

2.4 Support Vector Regression

Support vector machines (SVMs) are supervised machine learning techniques proposed by Vapnik et al. [VL63, Vap00, CV95] at the AT&T Bell Laboratories. In the 1990s, non-linear classification using SVM became popular as an alternative than artificial neural networks [BGV92]. Com-

```
Initialize the weights;
repeat
    foreach d in Data do
        FORWARDS PASS
            Using the instance d, compute the output of every unit at
            each layer;
        end
        BACKWARDS PASS
            foreach unit j in the output layer do
                Compute the error term δ_j of unit j;
            end
            foreach layer k in the hidden layers do
                foreach unit j in the layer k do
                    Compute the error term δ_j with respect to the
                    next higher layer;
                end
            end
            foreach weight w_ij in the network do
                Updates the weight w_ij;
            end
        end
    end
until stopping condition;
```

Figure 4: Backpropagation algorithm

pared to ANNs, SVMs are relatively tolerant to overfitting problem because SVMs are based on the *Structural Risk Minimization principle* while ANNs are based on the *Empirical Risk Minimization principle* [Vap00].

In the SVM, learning proceeds in the direction of maximizing the margin of the support vector, which is a hyperplane that divides each class of the given data. During early research, only linear classification was possible; however, non-linear classification became feasible by mapping data to a higher dimensional space using a kernel function. For example, radial basis function (RBF) transforms the original space into an infinite dimension Hilbert space. The most common kernel functions include:

- Polynomial: $k(x_i, x_j) = (x_i \cdot x_j)^d$,
- Radial basis function: $k(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$,
- Hyperbolic tangent: $k(x_i, x_j) = \tanh(\beta x_i \cdot x_j + c)$, and
- Laplacian: $k(x_i, x_j) = \frac{\theta}{\|x_i - x_j\|} \sin \frac{\|x_i - x_j\|}{\theta}$

Two major applications of SVMs are classification and regression. The main difference between SVMs for classification and those for regression is that the outputs of SVMs for regression are continuous values while those for classification are class labels. A version of an SVM for regression has been proposed in 1997 by Vapnik et al. [VGS97, DBK$^+$97]. This method is called support vector regression (SVR). SVRs performed well especially in estimating time-series data [MSR$^+$97, Kim03].

Given training data $(x_1, y_1), \ldots, (x_N, y_N)$, we want to find a function $f(x)$ which has at most ε deviation from targets y_i for all training data. And

at the same time, we want $f(x)$ to be as less complex as possible. Optimization problem can be formulated as follows:

$$\text{minimize} \quad \frac{1}{2}\|\omega\|^2$$

$$\text{subject to} \quad \begin{cases} y_i - \langle \omega, x_i \rangle - b \leq \varepsilon & \text{and} \\ \langle \omega, x_i \rangle + b - y_i \leq \varepsilon. \end{cases}$$

With a pre-defined ε, this optimization problem might not be feasible. Therefore we may allow some errors using slack variables ξ_i and ξ_i^*. The new optimization problem with slack variables are formulated as follows:

$$\text{minimize} \quad \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle \omega, x_i \rangle - b \leq \varepsilon + \xi_i, \\ \langle \omega, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^*, \quad \text{and} \\ \xi_i, \xi_i^* \geq 0. \end{cases}$$

This approach gives certain penalty for errors which is proportional to the amount by which each point is violating the constraint. The constant $C > 0$ determines the trade-off between the flatness or complexity of the model and the degree up to which deviations larger than ε are tolerated. Figure 5 describes a soft margin support vector regression with slack variables.

Training an SVM requires the solution of a very large quadratic programming (QP) optimization problem. In this study, we used Sequential Minimal Optimization (SMO) [Pla98] as an SVM training algorithm. SMO can solve SVM QP problem rapidly without extra storage and numerical QP

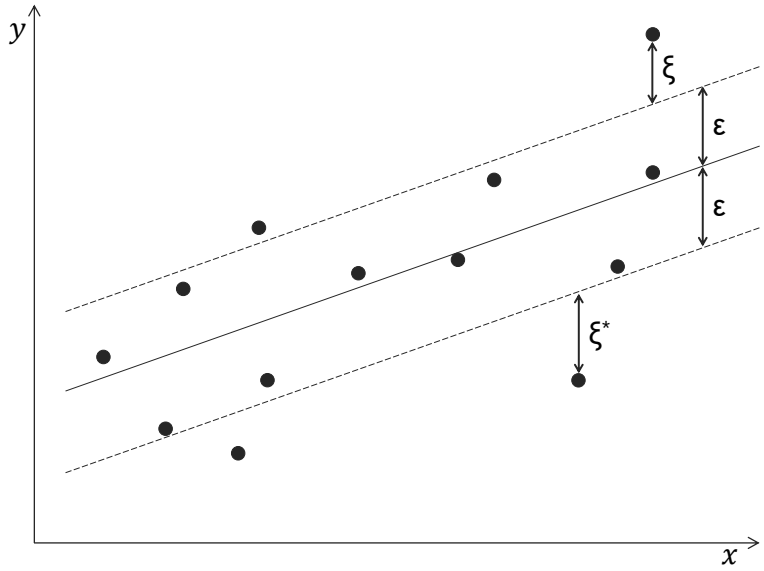


Figure 5: Soft margin loss setting for support vector regression

optimization. The overall QP problem is decomposed into QP sub-problems in SMO. Osuna's theorem [OFG97] ensures the convergence of SMO.

2.5 Genetic Algorithm

The genetic algorithm (GA) is a global optimization technique developed by Holland, which mimics the natural evolution of biological selection [Hol92]. It is used to find a solution with high (or low) fitness while repeating a genetic operation that imitates processes such as selection, crossover, and mutation, which are important elements of evolution. GA is a type of metaheuristic that does not depend substantially on the nature of the problem. It can search all ranges and is less likely to fall into a local optimum. Pure GA is disadvantageous in that it takes a long time to converge. The

hybrid genetic algorithm solves this problem by combining the local optimization algorithm with the GA. The followings are the main components of typical genetic algorithms.

- *Encoding*: In a genetic algorithm, one solution is expressed as a set of genes, or a chromosome. The most widely used representations are strings of binary, integer, and real-value.
- *Fitness function*: This indicates the validity of the solution for a given problem. It measures how good a solution is in terms of satisfying the problem objective.
- *Population*: Population is a set of chromosomes. Chromosomes in the population interact each other to generate new solutions and cull existing solutions.
- *Crossover*: A key operator of the genetic algorithm. In inheriting the features of the parents, we expect that the different advantageous traits combine to produce an offspring chromosome that is superior to the parents.
- *Selection*: This is the operator used to select the parent chromosome for the crossover. To mimic the principle of survival of the fittest in nature, chromosomes with high fitness are selected with high probability.
- *Mutation*: This statistically modifies a portion of the offspring chromosome to increase solution diversity and prevent premature convergence.

- *Repair*: After crossover and mutation, offspring may not meet the constraints of the problem. In that case, the offspring needs to be modified to satisfy the constraints.

- *Local optimization*: Solutions found by a genetic algorithm are not guaranteed to be optimal. They are usually sub-optimal, and sometimes have poor quality. Furthermore, genetic algorithms cost relatively considerable time to get local optima. One way to improve the performance and the consumption of time is combining the genetic algorithm process with a local search algorithm. The most typical local search algorithms to combine are based on greedy search or hill climbing algorithm.

- *Replacement*: To keep the size of the population, some of the existing chromosomes in the population have to be replaced by new offspring. Basically there are two strategies for replacement, i.e., generational replacement and steady-state replacement. Generational GA replace entire population with new chromosomes at each generation, while steady-state GA replace a small fraction of the population during each iteration.

- *Stopping condition*: Termination of the genetic algorithm usually occurs after a pre-specific number of generations. In some implementations, when the diversity of the population is below a certain point, algorithms stop. Or, combination of above two is occasionally used.

Chapter 3

Abnormal Data Correction

Abnormal values in observations lead to an inaccurate analysis, and obstruct making a right decision. Therefore, there is a need to replace abnormalities detected during quality control procedure with values considered to be normal. In this chapter, we present methods to make good substitutions for abnormal values. If observations are time-series data, utilizing observations of before and after is essential to predict the current observation. One of the most frequently used method to estimate values within the range of known values in time-series is interpolation [HMC+01, BC06].

3.1 Traditional Approaches

Suppose that we know the value of a function $f(x)$ at a discrete set of points $x_0, x_1, \ldots, x_n$, where $x_{i-1} < x_i$, for all $1 \leq i \leq n$. Interpolation is the process of estimating $f(x)$ for arbitrary x within the interval $[x_0, x_n]$. This section briefly describes three interpolation techniques used in this study.

3.1.1 Linear Interpolation

Linear interpolation is one of the simplest methods for interpolation. To estimate the value of $f(x)$ within the interval $[x_i, x_{i+1}]$, the linear interpolation utilizes the straight line between the two points $(x_i, f(x_i))$ and

$(x_{i+1}, f(x_{i+1}))$, i.e.,

$$f(x) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} \cdot (x - x_i) + f(x_i). \tag{3.1}$$

In this study, the nearest neighbor interpolation, which assigns the value of the nearest point, is used to estimate the value of $f(x)$ when x does not belong to any interval.

3.1.2 Polynomial Interpolation

Given a set of $n+1$ points $(x_i, f(x_i))$ where $0 \leq i \leq n$, the polynomial interpolation estimates the value of $f(x)$ as a polynomial such that:

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n. \tag{3.2}$$

Substituting the $n+1$ points into Equation (3.2) gives the following $n+1$ equations:

$$\begin{aligned} f(x_0) &= a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n, \\ f(x_1) &= a_1 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n, \\ &\vdots \\ f(x_n) &= a_n + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n. \end{aligned} \tag{3.3}$$

Since there are $n+1$ equations with $n+1$ unknowns, the interpolant $p(x)$ can be constructed by various ways such as Newton's divided difference and Lagrange's interpolation formulas [Atk89]. As Runge's phenomenon

can occur when interpolating using high degree polynomials [FZ07], we restricted the number of points used in the polynomial interpolation to six. When the number of available points is less than four, linear interpolation is used instead.

3.1.3 Spline Interpolation

Spline interpolation is a piecewise polynomial interpolation that uses a spline function, a low degree polynomial, in each interval. The spline functions of degree k are $k-1$ times differentiable such that they fit together smoothly. We used piecewise cubic polynomial functions [Hea96], which is the most commonly used method in the spline interpolation [Atk89]. Since at least four points are needed to find a particular cubic function, linear interpolation is used when the number of available points is less than four.

3.2 Machine Learning Based Approaches

We applied three machine learning techniques, i.e., decision tree regression, artificial neural network, and support vector regression, to model a function estimating observation values to replace. REPTree (reduced error pruning tree) algorithm is used to build regression trees. In this algorithm, reduced error pruning technique is applied to reduce overfitting effect. The minimum number of instances per leaf is set to 2 and the number of folds for reduced error pruning is set to 3. We set the minimum numeric class variance proportion of train variance for split to 0.001. For implementing artificial neural network, multilayer perceptron with one hidden layer which

has $N_{input}/2$ nodes is trained by backpropagation algorithm where N_{input} is the number of input variables. To solve the quadratic programming problem that occurs in training support vector machine, we used SMOreg algorithm, which is a improvement of the sequential minimal optimization (SMO) algorithm [SKBM00]. SMOreg algorithm overcomes an inefficiency problem of SMO algorithm by using two thresholds while SMO algorithm uses only on threshold. The learning rate for the backpropagation algorithm is set to 0.3 and the momentum rate is set to 0.2 We set the number of epochs to train through to 500. For kernel function of SVR, the RBF function was used because the RBF function is better, on average, than the linear or polynomial function. The gamma parameter for the RBF function is set to 0.01. We set the value of epsilon which is the amount to which deviation are tolerated to 0.001. The complexity constant C which determines the balance between the complexity of the model and penalties for unfeasible instances is set to 1.0.

An input of machine learning model consist of three part as follows.

1. Time-series data of target element

   The most fundamental input of machine learning regression model is time-series data, which is the same input as interpolations. The present weather phenomena can be explained from their temporal context. It tends to be influenced by the condition before and after.

2. Observations of other meteorological elements than target element

   Not only the weather element to estimate, but also other elements can help estimating target value. For example, since low air pressure

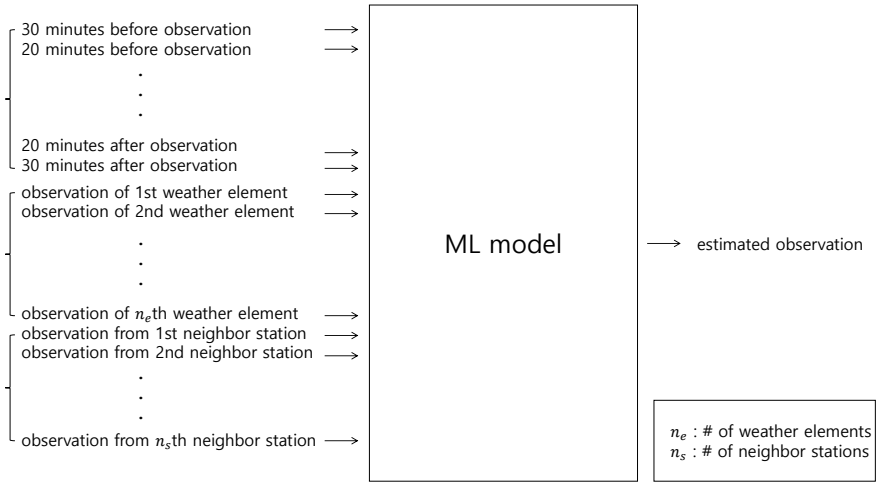


Figure 6: Input and output of the proposed machine learning model

implies high probability of rainfall, air pressure might be used as an input in estimating rainfall occurrence.

3. Observation data of target element from other stations

   Observation data from other stations around target station can be used, because atmospheric phenomena happening in target station has a close relation to that happening in geographically close area.

An output of machine learning model is an estimated observation value to replace an abnormal value. Input and output of the proposed machine learning model are described in Figure 6.

3.3 Datasets

For experiments, we used climatic data which consist of 8 weather elements on 692 AWSs in South Korea with 10-minute intervals from 2007 to

Table 1: Units of AWS data

Weather element	**Unit**
Wind direction	0.1 °
Wind speed	0.1 m/s
Temperature	0.1 °C
Relative humidity	0.1 %
Air pressure	0.1 hPa
MSLP	0.1 hPa
Rainfall occurrence	0 or 1
Hourly precipitation	0.1 mm

2012. Figure 7 shows the locations of the AWSs in South Korea. Collected weather elements include wind direction, wind speed, temperature, relative humidity, air pressure, mean sea level pressure (MSLP), rainfall occurrence, and hourly precipitation. Every value in AWS data is an integer. The units of weather elements are shown in Table 1.

Quality control procedures applied to collected AWS data to filter abnormal values. We ran the following four tests sequentially. Each test has its own target weather elements.

- *Physical limit test*: This test is for all weather elements. The limits and the percentages of the detected abnormal values of weather elements are shown in Table 2.

- *Step test*: Step test is performed on temperature, air pressure, and MSLP. If the difference between the current value and the value observed 10 minutes ago is greater than the critical point, the current value is considered as an abnormal value. Table 3 shows the maximum differences of weather elements and the percentages of the detected

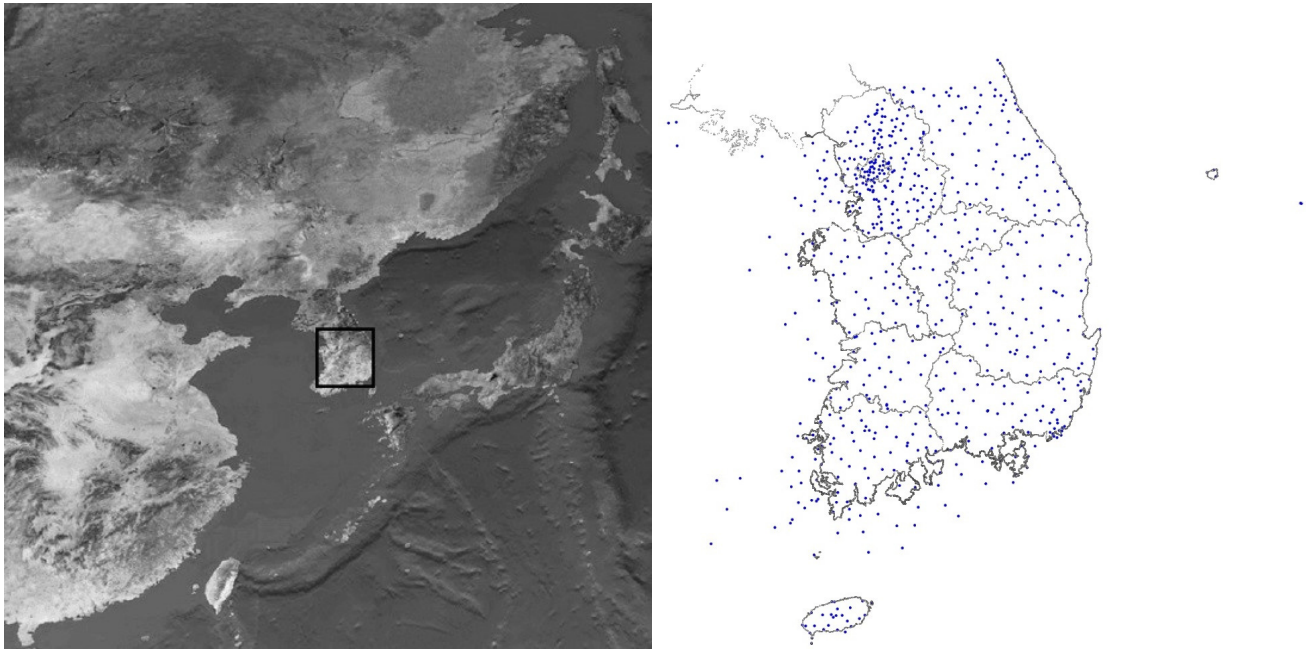


Figure 7: Locations of the 692 AWSs in South Korea [SLK14]

Table 2: Limits and results of absence & physical limit test

Weather element	Lower limit	Upper limit	Abnormality ratio
Wind direction	0.1 °	360 °	8.50 %
Wind speed	0 m/s	750 m/s	8.42 %
Temperature	-80 °C	60 °C	7.38 %
Relative humidity	0.1 %	100 %	8.78 %
Air pressure	500 hPa	1080 hPa	68.83 %
MSLP	500 hPa	1080 hPa	68.92 %
Rainfall occurrence	0	1	8.52 %
Hourly precipitation	0 mm	400 mm	9.29 %

Table 3: Critical points and results of step test

Weather element	Max difference	Abnormality ratio
Temperature	1 °C	7.81 %
Air pressure	2 hPa	69.08 %
MSLP	2 hPa	69.18 %

abnormal values of weather elements.

- *Persistence test*: Persistence test is performed on wind speed, temperature, relative humidity, air pressure, and MSLP. If the variation of a weather element during last 60 minutes is less than the critical point, the values for the duration are considered abnormal. The minimum variations and the results of the test are shown in Table 4.

- *Internal consistency test*: At the last stage of QC procedures, internal consistency test is performed on wind direction, wind speed, rainfall occurrence, and hourly precipitation. If wind direction value or the wind speed value is abnormal, both are considered abnormal. Similarly, if the air pressure value or the MSLP value is abnormal, both

Table 4: Critical points and results of persistence test

Weather element	Min variation	Abnormality ratio
Wind speed	0.5 m/s	12.40 %
Temperature	0.1 °C	8.50 %
Relative humidity	1 %	76.79 %
Air pressure	0.1 hPa	69.51 %
MSLP	0.1 hPa	69.49 %

Table 5: Results of internal consistency test

Weather element	Abnormality ratio
Wind direction	13.50 %
Wind speed	13.50 %
Air pressure	69.62 %
MSLP	69.62 %
Rainfall occurrence	12.30 %
Hourly precipitation	12.30 %

are regarded as abnormal, and if the rainfall occurrence value or the hourly precipitation value is abnormal, both are regarded as abnormal. Additionally, if the rainfall occurrence value is 0 while hourly precipitation is not 0, both are considered as an abnormal values. The results of the test are shown in Table 5.

As results of four validation procedures, the total percentages of the abnormal values in AWS data are shown in Table 6. High abnormality ratios in relative humidity, air pressure, and MSLP are mainly due to the inability of many stations to measure those elements.

Table 6: Overall proportions of abnormal values

Weather element	**Abnormality ratio**
Wind direction	13.50 %
Wind speed	13.50 %
Temperature	8.50 %
Relative humidity	76.79 %
Air pressure	69.62 %
MSLP	69.62 %
Rainfall occurrence	12.30 %
Hourly precipitation	12.30 %
Average	34.52 %

3.4 Experimental Results

For experiments, we made pseudo-abnormal values by deleting observed normal values, which were estimated by each model. Estimated values were rounded into integers as all the values in the AWS system should be integers. The data set consisting of input attributes and a target attribute were prepared using AWS data. The values in input attributes help in estimating the target value. Input attributes are basically composed of values within 30 minutes in the past of the target value and ones within 30 minutes in the future of the target value. But in the later part of this section, results of ML-based approaches with additional input attributes are provided. In the case that the target value is abnormal or all the input values are abnormal, that instance is excluded from the data set. To evaluate the accuracy of each estimation model, we performed a 10-fold cross-validation on the data set. The entire data set was divided into 10 folds, in which 9 folds are for the training set, and the remaining one is for the test set. The ML-based mod-

els were constructed using the training set and verified on the test set. We used libraries including GSL[1] for interpolations and WEKA [HFH+09] for ML algorithms. The root mean square error (RMSE) was used as a measure to compare the accuracy of each method. RMSE is a standard metrics for dealing with errors between model estimated values and observed values in a real environment, including meteorology [CA82, CD14]. If θ is the observed vector, $\hat{\theta}$ is the estimated vector, then the RMSE of $\hat{\theta}$ is calculated as:

$$\mathbf{RMSE}(\hat{\theta}) = \sqrt{\mathbf{E}((\hat{\theta} - \theta)^2)}. \tag{3.4}$$

The lower the RMSE value, the better the model estimate.

3.4.1 Preprocessing Data

Wind direction values in AWS data are measured in degree. Using these values as they are can give rise to calculative errors during estimating process. For example, consider the average of 1° and 359°. Although the desired result is 0° or 360°, the arithmetical result with direct usage of the values is 180°. To overcome this problem, we converted wind direction values into two-dimensional vectors of unit length:

$$\mathbf{v} = (\cos(\theta \cdot \frac{\pi}{180}), \sin(\theta \cdot \frac{\pi}{180})), \tag{3.5}$$

where $\mathbf{v}$ is a converted vector and θ is an original wind direction in degree. Each element of vector is trained and estimated separately. To calculate the error of an estimated direction, we need to convert the vector into the scalar

[1] http://www.gnu.org/software/gsl/

value in degree reversely:

$$\theta = atan2(y,x) \cdot \frac{180}{\pi}, \tag{3.6}$$

where x is the first element of $\mathbf{v}$, y is the second element of $\mathbf{v}$, and $atan2$ is defined as follows:

$$atan2(y,x) = \begin{cases} \arctan(\frac{y}{x}) & x > 0 \\ \arctan(\frac{y}{x}) + \pi & x < 0, y \geq 0 \\ \arctan(\frac{y}{x}) - \pi & x < 0, y < 0 \\ +\frac{\pi}{2} & x = 0, y > 0 \\ -\frac{\pi}{2} & x = 0, y < 0 \\ \text{undefined} & x = 0, y = 0. \end{cases} \tag{3.7}$$

Also, there is an issue in calculating an error of estimated wind direction. For instance, the difference between 1° and 359° has to be 2°, not 358°. We should choose the smaller of d and $3,600 - d$ where d is the difference between estimated direction and observed one (recall that the unit of wind direction in AWS data is 0.1°).

While rainfall occurrence values in AWS data are 0 and 1, they are represented as 0 and 100 respectively in experiments for visual convenience. If a result of the model for rainfall occurrence estimation is greater than 50, the estimated value is 100, and 0 otherwise.

As mentioned in Section 3.3, there are considerable abnormal values in AWS data. When applying estimation models, some attributes of an instance may be abnormal. They need to be handled properly for estimators to work.

In this study, abnormal values are replaced with the nearest value that is not abnormal. If the nearest value in future and that in past are equally near, the average of the two is substituted. However, an attribute in which more than 70% of values are abnormal was not used as an input attribute of estimation models since abnormal values cannot be replaced with appropriate values.

Since AWS data gathered from one weather station for 6 years include about 315,000 instances, a training set and a test set are composed of about 283,500 instances and about 31,500 ones, respectively. Training ANNs and SVRs using all of the instances in the training set takes too much time. Thus, only 20% and 2% of instances in the training set were used in this study for training ANNs and SVRs, respectively.

Table 7 shows the average running time of the estimation models for one weather station and one target weather element. Experiments were conducted on Intel i7 quad-core 2.93 GHz CPUs. Because a test set includes about 31,500 instances, estimating one abnormal value takes very small amount of time. It took relatively long time to train ML-based models. However, training process does not need a real-time response and the time required to train a model is sufficiently small compared to the 10-minute interval of the AWS data.

3.4.2 Results

Table 8 shows the performances of the interpolation methods using two different representations of wind direction. Vector representation reduced RMSE by 34%, 32%, and 31%, respectively.

In Table 9, the performances of the interpolation methods are pre-

Table 7: Average running time of the estimation models

Estimation model	**Time for training (second)**	**Time for test (second)**
Linear interpolation	N/A	0.006
Polynomial interpolation	N/A	0.018
Spline interpolation	N/A	0.023
Decision tree	3.202	0.006
ANN	21.063	0.025
SVR	19.311	0.014

Table 8: Comparison of wind direction representations

Interpolation method	**Representation**	**RMSE**
Linear	Degree	510.349
	Vector	334.920
Polynomial	Degree	527.508
	Vector	360.476
Spline	Degree	521.662
	Vector	357.674

Table 9: Performances of interpolation methods

(RMSE)

Weather element	**Linear**	**Polynomial**	**Spline**
Wind direction	**334.920**	360.476	357.674
Wind speed	**4.093**	4.802	4.665
Temperature	**2.358**	2.700	2.675
Relative humidity	**15.734**	17.880	17.605
Air pressure	**1.174**	1.310	1.286
MSLP	**1.255**	1.409	1.385
Rainfall occurrence	10.285	10.114	**10.046**
Hourly precipitation	1.479	1.492	**1.426**

sented. They used time-series data which consist of values within 30 minutes in the past and the future of the target values as input attributes. Linear interpolation showed the best estimation accuracy in every weather element except rainfall occurrence and hourly precipitation in which spline interpolation performed the best.

In Table 10, the performances of ML-based approaches are presented. They used the same input attributes as the interpolation methods. Decision tree estimated rainfall occurrence well but for the rest of the weather elements, SVR showed the best performance. Compared to the interpolation methods, SVR is preferable for all weather elements except hourly precipitation.

ML-based approaches can use more information beyond the time-series of the target element. Table 11 shows the results of ML-based estimators utilizing all the other weather elements. Although RMSE increased for relative humidity and hourly precipitation, the performance was improved for the other elements.

Table 10: Performances of ML-based models using time-series data

(RMSE)

Weather element	**Decision Tree**	**ANN**	**SVR**
Wind direction	338.842	365.722	**334.766**
Wind speed	4.181	4.629	**4.060**
Temperature	3.194	2.696	**2.343**
Relative humidity	17.129	18.066	**14.492**
Air pressure	2.177	1.306	**1.176**
MSLP	2.303	1.438	**1.254**
Rainfall occurrence	**9.303**	10.218	9.946
Hourly precipitation	2.008	1.761	**1.564**

Table 11: Performances of ML-based models using time-series data and the other elements

(RMSE)

Weather element	**Decision Tree**	**ANN**	**SVR**
Wind direction	**316.671**	363.603	334.596
Wind speed	4.148	4.555	**4.059**
Temperature	3.196	2.708	**2.337**
Relative humidity	17.170	18.474	**14.513**
Air pressure	2.096	1.037	**0.954**
MSLP	2.207	1.122	**1.013**
Rainfall occurrence	**8.335**	10.438	9.944
Hourly precipitation	2.051	1.835	**1.570**

Table 12: Performances of ML-based models using time-series data, the other elements, and three neighbor station data

(RMSE)

Weather element	**Decision Tree**	**ANN**	**SVR**
Wind direction	**315.404**	360.830	333.672
Wind speed	4.144	4.543	**4.054**
Temperature	3.197	2.700	**2.335**
Relative humidity	17.201	18.526	**14.524**
Air pressure	2.094	1.006	**0.938**
MSLP	2.206	1.088	**0.993**
Rainfall occurrence	**8.351**	10.447	9.944
Hourly precipitation	2.071	1.910	**1.575**

The data observed in neighbor weather stations overall helped in estimating unobserved data. We chose k nearest weather stations within a radius of 30 km. Table 12 shows the results when $k = 3$ and Table 13 shows the results when $k = 5$. The estimation accuracy for relative humidity, rainfall occurrence, and hourly precipitation decreased but the other weather elements showed improvement. Increasing the number of neighbor stations improved the performance to some extent.

Table 13: Performances of ML-based models using time-series data, the other elements, and five neighbor station data

(RMSE)

Weather element	**Decision Tree**	**ANN**	**SVR**
Wind direction	**315.340**	360.757	333.607
Wind speed	4.145	4.549	**4.054**
Temperature	3.197	2.698	**2.335**
Relative humidity	17.215	18.688	**14.531**
Air pressure	2.094	1.001	**0.932**
MSLP	2.206	1.007	**0.986**
Rainfall occurrence	**8.352**	10.506	9.945
Hourly precipitation	2.076	1.920	**1.581**

Chapter 4

Spatial Quality Control

Spatial quality control determines whether the observation data of the target station is abnormal based on the values of other observation stations around the target station. It is also referred to as a spatial consistency test [EGG11]. Because this test is based on a large amount of data, it involves more time and resources than basic quality control. Therefore, spatial quality control is often performed in quasi-real-time. Typical spatial quality control process are as follows:

1. Estimate the value of the target station using the values of surrounding observation stations.

2. If the difference between the observed and the predicted value of target station is greater than the pre-specified threshold, the observation is considered as abnormal.

The meteorological elements of the KMA dataset, excluding rainfall occurrence, consist of continuous values; therefore, the predicted value can be estimated naturally via the interpolation or regression model. In the case of rainfall occurrence, it has a value of 0 or 1, so the value should be taken as 0 if the estimated value is less than 0.5, and 1 if the estimated value is 0.5 or more. The acceptable range for the difference between the observed value and the predicted value is generally determined using the standard

deviation of the surrounding stations, which we set to the observation stations within 30 km of the target station. Using the standard deviation as a threshold is based on assuming observations from stations within a close distance are normally distributed. In that case, using three standard deviations as a threshold means 99.73 % of the observations are thought to be normal, where as using two standard deviations means 95.45 % of the observations are thought to be normal. Many statistical outlier detection tests such as Grubbs' test [Gru69] assume a Gaussian distribution for the data. In the spatial quality control procedure suggested by guidelines and operated by institutions including KMA, an observation whose difference exceeds two standard deviations is determined as suspect and an observation whose difference exceeds three standard deviations is determined as warning or errornous [SFA$^+$00, EGG11]. If the standard deviation is 0, because the observation values of all neighboring stations are the same, it is difficult to determine the acceptable range; therefore, the test is not performed. In the KMA dataset, this was often the case for elements such as precipitation and rainfall occurrence, which are always zero during periods of non-rain. Moreover, if there are less than three stations within 30 km, spatial quality control does not proceed because reliable standard deviations cannot be calculated. Also, observations that are missing or identified as abnormal during basic quality control are not considered for spatial quality control.

If the tolerance of the difference between the observed and predicted value is the same, the accuracy of the predicted value estimation will determine the reliability of spatial quality control. In this study, we aim for more accurate spatial prediction and thus improved spatial quality control

performance. Traditional spatial prediction methods include spatial interpolation methods such as the Cressman method [Cre59] and the Barnes method [Bar64]. However, these methods do not reflect the geographical features of each region because they depend only on relative position to estimate the predicted value [Wad87, GKRS88]. Here, we propose a method to improve the accuracy of estimates by overcoming the shortcomings of existing methods by using supervised learning techniques.

4.1 Traditional Approaches

This section describes the spatial interpolation methods used in this study: the Cressman method and the Barnes method. The two methods have been slightly modified by the KMA to detect meteorological anomalies in South Korea. Actual observations are compared with estimates generated by the spatial interpolation methods. If there is a significant difference between observed and predicted values, the observation is classed as 'suspect' or 'error' according to the degree of difference.

4.1.1 Cressman Method

The Cressman method performs spatial interpolation on a two-dimensional distribution of meteorological elements. Meteorological elements at each station are irregularly distributed in two dimensions, and converted into estimated values of the grid points at regular intervals. In this study, the grid interval is 0.2° for both longitude and latitude. The estimated values of the grid points are called the background field, and are calculated with respect

to the effective radius r. The effective radius is the control parameter describing the maximum station distance when estimating each grid point. Let z_i be the observed value of the station i, and d_{ei} denote the distance between the grid point e and the station i. Then, $Z_r(e)$, the estimated value of the grid point e, is the weighted average of the observations within the effective radius r (Figure 8):

$$Z_r(e) = \frac{\sum w_r(i) \cdot z_i}{\sum w_r(i)}, \tag{4.1}$$

where $w_r(i)$, the weight of the station i, depends only on the distance:

$$w_r(i) = \begin{cases} (r^2 - d_{ei}^2)/(r^2 + d_{ei}^2) & \text{if } d_{ei} \leq r \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

To obtain $Z(i)$, the estimated value of a station i, the estimates of the four closest grid points from the station are averaged. After calculating the estimates of all the stations, the background field can be recalculated using the estimates instead of the observations. The estimates of the stations can also be recalculated over the new background field. This process can be repeated as many times as desired. We set the effective radius to 50 km, 30 km, and 10 km and updated the background field and the estimates of the stations.

Let σ_i be the standard deviation of the observations at all stations located within the final effective radius of the station i. If $|z_i - Z(i)|$ is greater

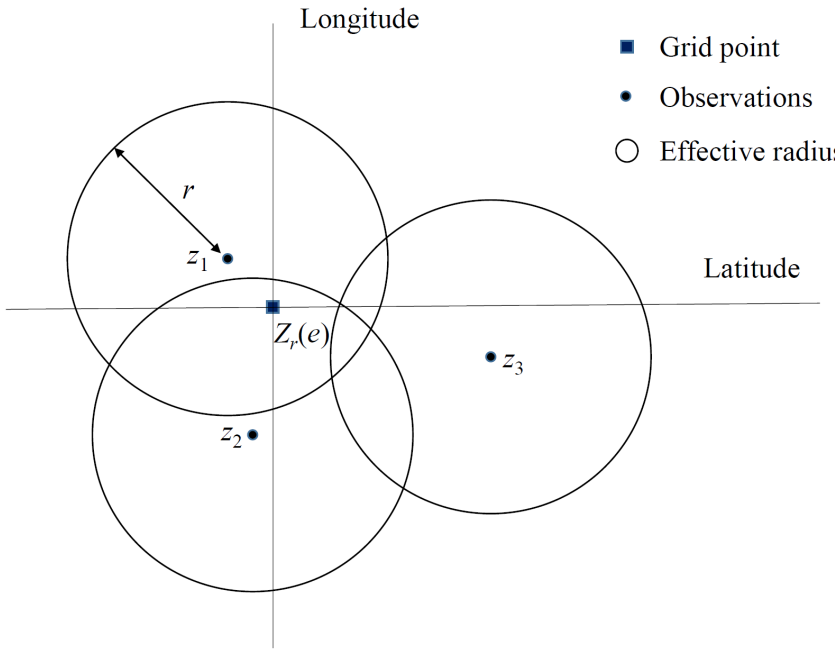


Figure 8: Calculation of the $Z_r(e)$, the estimated value of the grid point e in the Cressman method. Only observations of stations located within the effective radius r are used. In this example, z_1 and z_2 are used to calculate $Z_r(e)$, but z_3 is not used.

than $3 \cdot \sigma_i$, z_i is classified as an error. If $|z_i - Z(i)|$ is greater than $2 \cdot \sigma_i$, z_i is classified as a suspect.

4.1.2 Barnes Method

The Barnes method is a statistical technique that can derive accurate two-dimensional distribution from randomly distributed data in space. It is similar, in many respects, to the Cressman method, but instead uses a Gaus-

sian function in the weight function:

$$w_r(i) = \begin{cases} \exp(-d_{ij}^2/2r^2) & \text{if } d_{ij} \leq r \\ 0 & \text{otherwise,} \end{cases} \tag{4.3}$$

where d_{ij} is the distance between station i and j. The KMA uses observations without using grid points when calculating the estimates by the Barnes method:

$$Z(i) = \frac{\sum w_r(i) \cdot z_i}{\sum w_r(i)}, \tag{4.4}$$

where r is set to 30 km. The process of determining whether or not observations are normal is almost identical to that of the Cressman method. Let σ_i be the standard deviation of the observations at all stations located within 30 km of the station i. If $|z_i - Z(i)|$ is greater than $3 \cdot \sigma_i$, z_i is classified as an error. If $|z_i - Z(i)|$ is greater than $2 \cdot \sigma_i$, z_i is classified as a suspect.

4.2 SVR-based Approach

In this section, we propose a method using support vector regression (SVR) to overcome the spatial prediction limitations of the Cressman and Barnes methods for a target observation station from a spatial quality control perspective. Preliminary study on meteorological elements has shown that the estimation capability of SVR is superior to other machine learning techniques [LMKM14]. In this study, the SVM^{light} [Joa] library was used for C language implementation. Implementation of the learner in SVM^{light}

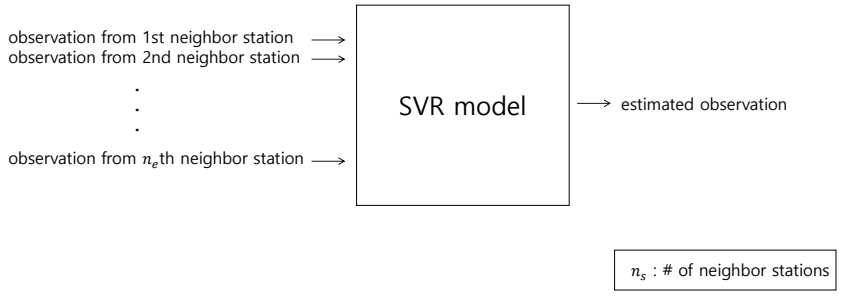


Figure 9: Input and output of the proposed support vector regression model

is described in [Joa98]. We choose the RBF function for the kernel function of SVR and set the gamma parameter for RBF function to 0.01. We set the value of epsilon which is the amount to which deviation are tolerated to 0.1. The complexity constant C which is trade-off between generality and penalties for unfeasible instances is set to $(\frac{\sum x^2}{n})^{-1}$ where x is the input vectors and n is the number of training samples.

The input and the output of the proposed SVR model for spatial quality control are as follows and described in Figure 9.

- Input: observations of stations surrounding the target station
- Output: observation value of the target station

In the input, values that are missing or classified as errors during basic quality control are replaced by the temporally closest values. The wind speed converted into a 2D vector representation was learned and tested by

two separate models for each dimension. Once the model is learned from the values of the target station and the surrounding observation stations in the past, the predicted value of the target station can be estimated for the input that has not been learned. Because past observation values are not labeled as normal or abnormal with respect to spatial quality control, they are learned regardless of normal and abnormal values. Therefore, this approach assumes that most observations are normal and abnormalities are few.

Once the predicted value of the target station is estimated, the process of determining whether the observed value of the target station is normal is the same as the Cressman method or Barnes method. Let z_i and $Z(i)$ be the observations value and the estimated value by SVR of station i, respectively, and let σ_i be the standard deviation of the observations from all stations within a radius of 30 km of station i. If $|z_i - Z(i)|$ is greater than $3 \cdot \sigma_i$, z_i is classified as an error. If $|z_i - Z(i)|$ is greater than $2 \cdot \sigma_i$, z_i is classified as a suspect.

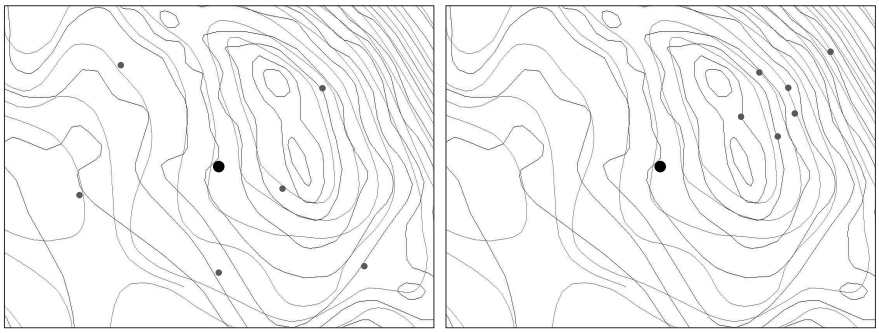
The SVR model can implicitly capture the geographic characteristics of the target station while learning past data. Through this process, the combination of each station and each meteorological element has its own specific model. This is an advantage of SVR over non-ML approaches. However, an approach based on machine learning also has its drawbacks; specifically, that it takes a long time to learn. A method to overcome this is introduced in Section 4.3.

4.3 Selecting Neighboring Stations

The input of the SVR model uses the observations of neighboring AWSs within a certain radius of the target AWS. However, if there are too many neighbors, the learning time of SVR becomes too long. Also, some neighboring stations act as noise instead of helping to estimate the value of the target stations. Therefore, it is necessary to select the best core neighbors to estimate the value of the target station while reducing the number of neighbors used in the input.

4.3.1 Similarity and Spatial Dispersion

Two criteria were applied to select key neighbors. The first considered the similarity of the observations between the target station and the neighbor station. Observations at locations with similar meteorological phenomena are helpful in deriving observations at the target site. The second considered how widespread the neighboring stations were in space. If one constructs a core neighborhood at stations concentrated in a narrow area, the model cannot be flexible to various situations. For example, if there is a peculiar meteorological phenomenon within a narrow area (e.g., a local storm), the estimate will be misled. Spatial dispersion ensures statistical robustness of the model. Figure 10 shows two different choices of neighboring stations. When the amount of rainfall in target station is estimated, the amount of rainfall in neighboring stations is used. If localized heavy rain happens in an area including neighboring stations with low spatial dispersion, the estimated amount of rainfall will be inclined to be very high even though target



(a) Neighbor stations with high spatial dispersion

(b) Neighbor stations with low spatial dispersion

Figure 10: Examples of neighbor selection

station is out of influence of localized heavy rain. On the other hand, it reflects overall surrounding circumstances of rainfall when spatial dispersion of neighboring stations is high."

To measure the similarity of stations according to their meteorological elements, the time series values of the elements are expressed as vectors, and the distance between them is measured in various ways. We used the L_1 distance, L_2 distance, Pearson correlation coefficient, and mutual information to measure the similarity between two vectors. After the distance of all the station pairs was calculated, the smallest value was zeroed and the largest value was normalized to one. The L_1 distance, known as the Manhattan distance or taxicab distance, between two vectors $\mathbf{x}$ and $\mathbf{y}$ was calculated as follows:

$$\|\mathbf{x}-\mathbf{y}\|_1 = \sum_{i=1}^{n} |x_i - y_i|, \tag{4.5}$$

where $\mathbf{x}_i$ is the i-th element of $\mathbf{x}$. We used ($1 - L_1$ distance) as a similarity

measure to ensure that the measurement was as large as the two vectors were similar. The L_2 distance, known as the Euclidean distance, between two vectors **x** and **y** was calculated as follows:

$$\|\mathbf{x}-\mathbf{y}\|_2 = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}. \tag{4.6}$$

We used ($1 - L_2$ distance) as a similarity measure to ensure that the measurement was as large as the two vectors were similar. The Pearson correlation coefficient is used to measure the degree of the linear relationship between two variables. It has a value 1 when there is a perfect positive linear correlation, and -1 when there is a perfect negative linear correlation. The Pearson correlation coefficient is calculated as follows, where $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$:

$$r_{\mathbf{xy}} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}. \tag{4.7}$$

Mutual information measures the mutual dependence between two random variables X and Y. It quantifies the reduction in uncertainty of one of the variables due to knowing the other. Mutual information is calculated as follows, where $p(x,y)$ is the joint probability function of x and y, and $p(x)$ and $p(y)$ are the marginal probability density functions of x and y, respectively:

$$I(X;Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)log(\frac{p(x,y)}{p(x)p(y)}). \tag{4.8}$$

We computed the mutual information from the observed frequency of two

vectors, **x** and **y**, assuming that these vectors constitute an independent and identically distributed sample of (X,Y).

As a measure of spatial dispersion, we used the average of the geographical distance from the nearest station [CE54]. If the set of target stations and selected neighbors is x, and $d_{x_i x_j}$ is the normalized geographic distance between the two stations x_i and x_j, then the spatial dispersion is calculated as:

$$\textbf{dispersion}(x) = \frac{\sum_{x_i \in x} \min_{x_j \in x, x_i \neq x_j} d_{x_i x_j}}{\sum_{x_i \in x} 1}.$$

The larger the spatial dispersion, the better the neighborhood selection. The two criteria of similarity and spatial dispersion often conflict. In general, similarities in climatic characteristics are often due to geographic proximity. Therefore, the key neighborhood screening problem is a multi-objective optimization problem that simultaneously optimizes two or more objectives that are not independent of each other. In this study, we solve the multi-objective optimization problem using genetic algorithms.

4.3.2 Multi-Objective Genetic Algorithm

Several successful attempts have been made to solve multi-objective problems using GA [ZT99, FF98, Coe00, XZ04, KCS06]. Among them, NSGA-II by Deb et al. [DPAM02] is the most well-known. To maximize the function $f_1, f_2, \ldots, f_n$ with n number of objects, if solution x and solution y satisfy the following condition, then it can be said that solution y dominates

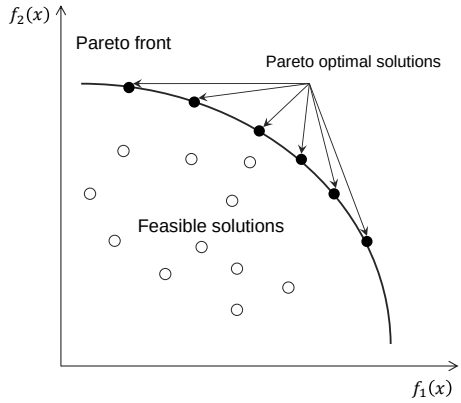


Figure 11: Illustration of Pareto-optimal solutions and Pareto Front in a 2-objective problem

solution x:

$$f_i(x) \leq f_i(y)\ \forall i \text{ and } \exists j : f_j(x) < f_j(y).$$

When a solution is not dominated by any other solution, the solution is called Pareto-optimal. To improve an objective function in a Pareto-optimal solution, one has to sacrifice another objective function. Pareto front is a set of Pareto-optimal solutions. Figure 11 illustrates an example of Pareto-optimal solutions and Pareto front when the problem has two objective functions to minimize. The multi-objective genetic algorithm (MOGA) does not output one solution but several Pareto-optimal solutions. The final solution selection is performed by the decision-maker. In this study, we tested the SVR with several Pareto-optimal solutions for each meteorological element, and selected the best solution on average. Figure 12 shows the structure of the GA used in this study.

```
non-dominated set E ← ∅;
initialize population P;
repeat
    select 2N parents from P;
    create N offspring applying crossover on the parents;
    mutate offspring;
    repair offspring;
    local-optimize offspring;
    P ← offspring;
    update E;
    remove n_E solutions from P;
    add n_E solutions from E to P;
until stopping condition;
return E;
```

Figure 12: The framework of our hybrid multi-objective genetic algorithm

- *Encoding*: One chromosome is represented by a one-dimensional binary string. Each gene corresponds to one station. If the value of the gene is '0', the observation value of the corresponding station is not used as the input of the SVR. If it is '1', it is selected as an input of the SVR. Figure 13 shows an example of neighbor selection represented by one-dimensional binary string.

- *Fitness function*: When the individual objective function is $f_1, f_2, \ldots, f_n$, the fitness value of solution x is calculated as:

$$f(x) = w_1 f_1(x) + w_2 f_2 + \cdots + w_n f_n(x),$$

where $w_1, w_2, \ldots, w_n$ are non-negative and $\sum_{i=1}^{n} w_i = 1$, each weight w_i is randomly set for every generation, not as a fixed value. This

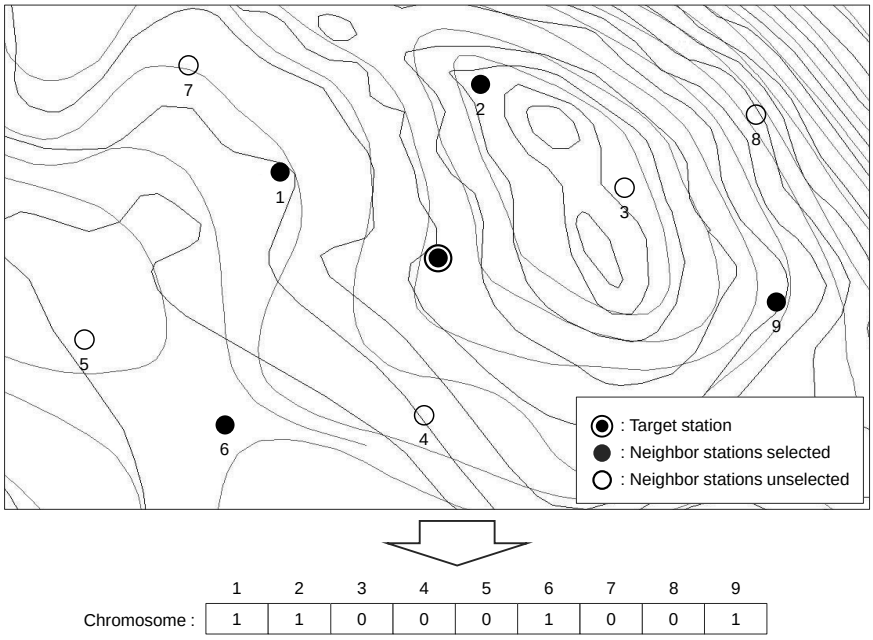


Figure 13: An example of representation of the solution

allows the algorithm to search for various Pareto-optimal solutions [MI95]. This method is more intuitive than the algorithm that uses Pareto ranking-based fitness evaluation [FF93], and easier to be combined with a local optimization algorithm [IM96]. In this problem, $n = 2$, and $f_1(x)$ and $f_2(x)$ correspond to similarity and spatial dispersion, respectively.

- *Population*: In this study, the size of the population was set to 50. The initial population consisted of 50 randomly generated chromosomes.

- *Selection*: roulette-wheel selection, one of the most widely used selection operators, was used. The probability that the best fitness solution will be selected is four times the probability that the lowest fitness solution will be selected.

- *Crossover*: In this study, we used a two-point crossover with two cut points. Figure 14 illustrates a process of two-point crossover.

- *Mutation*: Each gene was flipped with a probability of 10%.

- *Repair*: The number of genes with a value of '1' in the chromosome may be different from the number of stations to be selected after crossover and mutation. If the number of genes with a value of '1' is insufficient, we repeat the process of changing the value of the randomly selected gene among genes with a value of '0' to '1'. On the other hand, if the number of genes with a value of '0' is insufficient, we repeat the process of changing randomly selected genes to '0' among genes with a value of '1'.

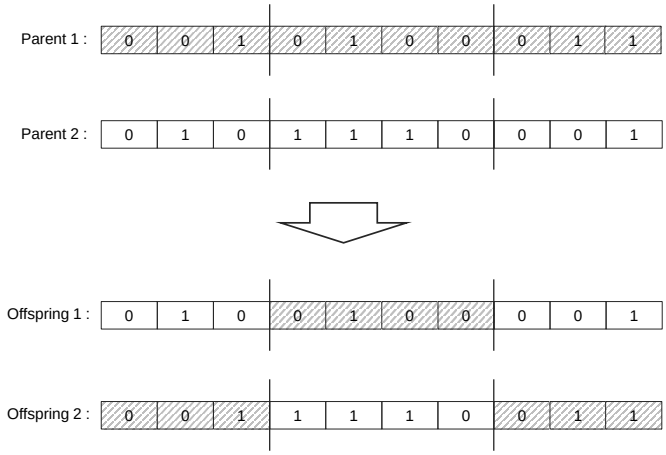


Figure 14: Two-point crossover

- *Local optimization*: This exchanges the values of two genes whose fitness value increases when exchanged. This process is repeated until the exchange of any two gene values can no longer increase the fitness value.

- *Replacement and elitism*: We used a generational GA to generate offspring as large as the size of the population, and replace the entire population. Among the solutions found so far, non-dominant solutions that are closest to the Pareto-optimal are stored in an external archive. This non-dominant solution archive updates every time a new solution is created. In other words, the solution that is dominated by the new solution is removed from the existing non-dominant solution archive, and the new solution is stored in the archive when it is a non-

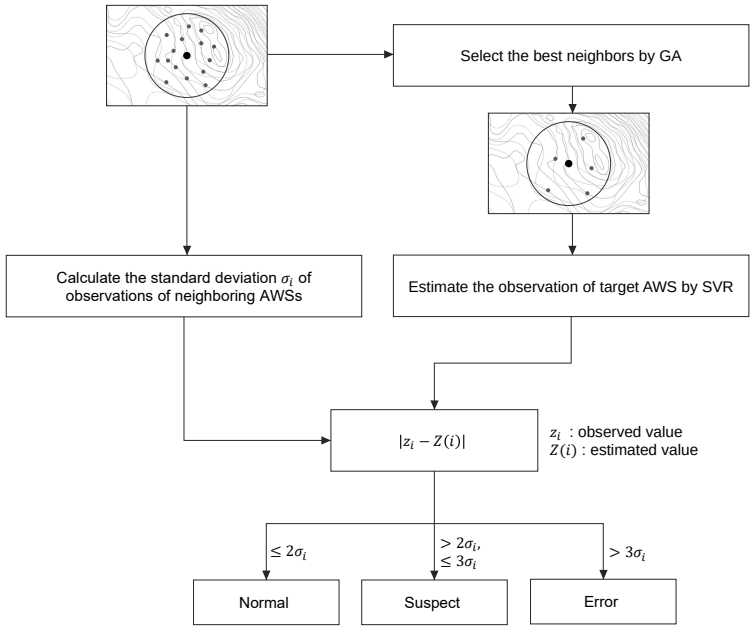


Figure 15: The proposed spatial quality control process

dominant solution. As survival of good solutions within a population can result in a good solution for the next generation, some of the population are replaced by solutions in non-dominant solution archive. In this algorithm, 20% of the entire population was randomly replaced with solutions in non-dominant solution archive.

- *Stopping condition*: The genetic algorithm stops when 1,000 generations have passed.

Table 14: Meteorological elements in automatic weather station (AWS) data

Meteorological element	**Unit**
Wind direction	°
Wind speed	m/s
Temperature	°C
Humidity	%
Atmospheric pressure	hPa
Hourly precipitation	mm
Rainfall occurrence	0 or 1

4.4 Datasets

Experiments for spatial quality control cover meteorological data from 572 AWSs operated by KMA in South Korea. Figure 16 shows the locations of the target AWSs.

Target data includes meteorological information measured every 1 minute from January 1, 2014 at 00:00 to December 31, 2014 at 23:59. In one year, 525,600 pieces of observational data are collected for each meteorological element at each station. We selected seven major meteorological elements for analysis: 10-minute average wind direction, 10-minute average wind speed, 1-minute average temperature, 1-minute average humidity, 1-minute average pressure, 1-hour cumulative amount of precipitation, and precipitation. Table 14 shows the types and units of meteorological elements used in this study.

Wind direction values expressed in degrees were converted into two-dimensional unit vectors, the same as in Section 3.4.1. In the spatial quality control process, the components of the two vectors are processed separately.

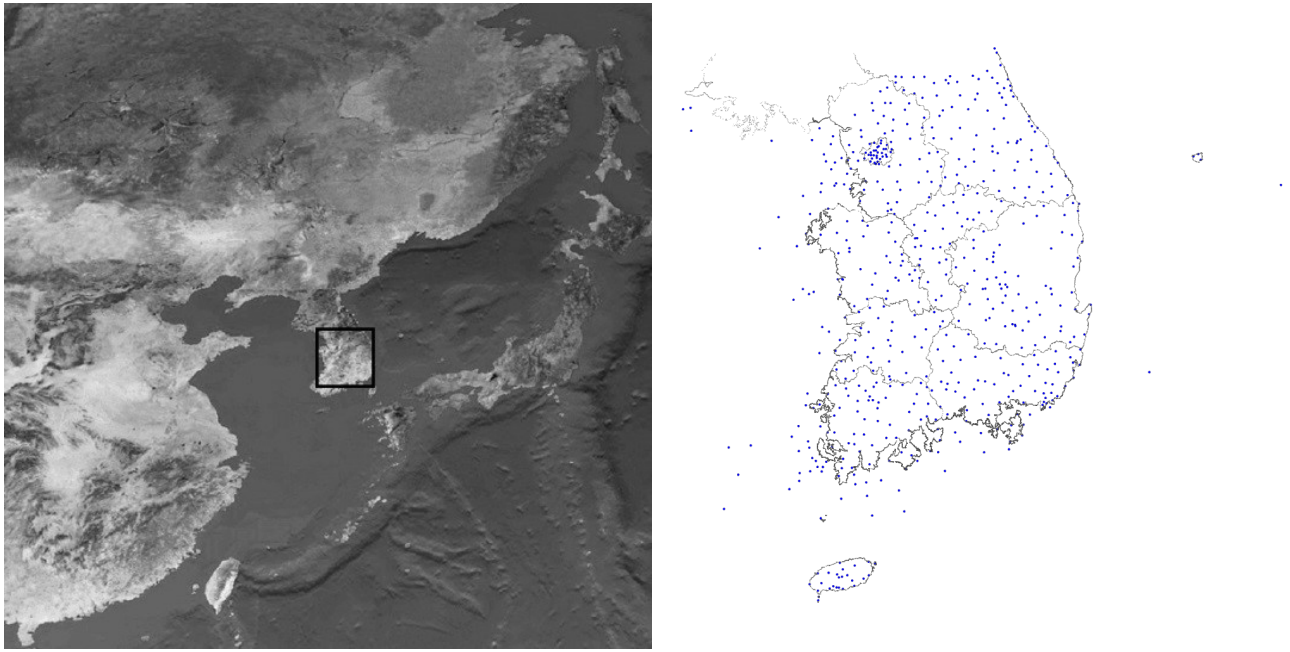


Figure 16: Locations of the 572 automatic weather stations (AWSs) in South Korea [SLK14]

Table 15: Limits for physical limit test

Meteorological element	**Lower limit**	**Upper limit**
Wind direction	0°	360°
Wind speed	0m/s	75m/s
Temperature	-80°C	60°C
Humidity	1%	100%
Atmospheric pressure	500hPa	1080hPa
Precipitation	0mm	400mm
Rainfall occurrence	0	1

When a quantitative comparison of wind direction is required, the wind direction represented by the vector was converted back to degrees.

The data used in this study was first filtered through the following four basic quality control procedures. Each test was performed sequentially. If any test failed, the data was classified as an error, and subsequent tests were not performed. Each test and the numerical criteria are the same as those used by KMA.

- *Physical limit test*: The physical limit test is performed on all meteorological elements. Table 15 shows the physical limits of each meteorological element, which are based on World Meteorological Organization (WMO) standards [Jar08].

- *Step test*: The step test is performed for wind speed, temperature, humidity, and atmospheric pressure. If the difference between the current observation value and the value one minute prior is more than a certain value, it is classed as an error. Table 16 shows the maximum variation of each meteorological element.

Table 16: Maximum amount of change for step test

Meteorological element	**Maximum amount of change**
Wind speed	10m/s
Temperature	1°C
Humidity	10%
Atmospheric pressure	2hPa

Table 17: Minimum amount of change for persistence test

Meteorological element	**Minimum amount of change**
Wind speed	0.5m/s
Temperature	0.1°C
Humidity	1.0%
Atmospheric pressure	0.1hPa

- *Persistence test*: The persistence test is performed for wind speed, temperature, humidity, and atmospheric pressure. A value is classified as an error when the accumulated change in the observed value within 60 minutes is smaller than a certain value. Table 17 shows the minimum variation within 60 minutes for each meteorological element.

- *Internal consistency test*: The internal consistency test is performed for pairs of wind direction and wind speed data, and pairs of precipitation and rainfall occurrence data. If any one of the factors in each pair is determined to be an error in another test, the other factor is also perceived as an error. Also, if the rainfall occurrence value is 0 but the precipitation value is not 0, both values are classed as suspects.

Table 18 shows the percentages of normal, error, and suspect values, re-

spectively, after performing each test on the KMA dataset. If the observed meteorological element is not available due to an absence of observational equipment, or if the observed value is missing, it is classified as uninspected. All subsequent experiments were performed only on data determined as normal after basic quality control.

4.5 Experimental Results

In this section: i) detailed good parameters are selected, ii) the performances of the estimation methods are compared, and iii) the results of the proposed spatial quality control procedure are presented using meteorological data collected by the KMA for a year in 2014. To measure the accuracy of each estimation method, results are evaluated by RMSE. As the accuracy of estimates should be based on normal observations, only observations classed as normal by the model are used to calculate RMSE. When comparing the RMSE of two or more models, only those observations determined as normal by all models were used.

Performance evaluation of SVR estimation models was achieved through 10-fold cross-validation. All data was divided into 10 folds, of which 9 were used as the training set and the other was used as the test set. Learning and test are performed 10 times so that each fold can be used as a test set. Due to there being 7 meteorological elements in 572 AWSs, and 10 models must be learned each time, a total of 40,040 models were created for each experiment. The entire training set consists of 473,040 data sets. Because of the large number of models and the overly long total execution time, we

Table 18: Results of basic quality control

Meteorological element	**Normal**	**Limit Error**	**Step Error**	**Persistence Error**	**Consistency**		**Uninspected**
					Suspect	**Error**	
Wind direction	81.97%	$1.68e^{-3}$%	N/A	N/A	0.00%	2.80%	15.23%
Wind speed	81.90%	$2.85e^{-4}$%	$1.36e^{-3}$%	$8.47e^{-1}$%	0.00%	2.80%	14.45%
Temperature	96.41%	$3.22e^{-3}$%	$7.73e^{-3}$%	$1.02e^{-1}$%	N/A	N/A	3.48%
Humidity	54.47%	$2.23e^{-2}$%	$3.38e^{-3}$%	5.43%	N/A	N/A	40.07%
Atmospheric pressure	38.30%	$1.83e^{-1}$%	$5.72e^{-5}$%	$3.69e^{-2}$%	N/A	N/A	61.48%
Hourly precipitation	93.08%	0.00%	N/A	N/A	1.94%	0.00%	4.98%
Rainfall occurrence	93.08%	$2.34e^{-4}$%	N/A	N/A	1.94%	0.00%	4.98%

N/A : not available.

Table 19: Accuracy of estimates according to wind direction representation

Representation	**RMSE**
Degree	92.17
Vector	**68.28**

sampled 5,000 data sets and used them as training sets for the parameter optimization experiments. We then describe the change in performance and time caused by increasing the size of the training set once the final parameter is determined.

The experiment was performed on an Intel i7 quad-core 2.93 GHz CPU. Each experiment used only one core. Experiments with a long execution time were performed by dividing each of the seven machines by observatories, and the total execution time included the execution time of each machine.

4.5.1 Representation of Wind Direction

Section 4.4 describes the process of converting wind direction expressions from degrees to 2D vectors. Table 19 compares the accuracy of SVR estimates for each wind direction representation. The accuracy of the estimate is much higher when expressed in terms of vector expression than degrees. Thus, all subsequent experiments used a vector expression to represent wind direction.

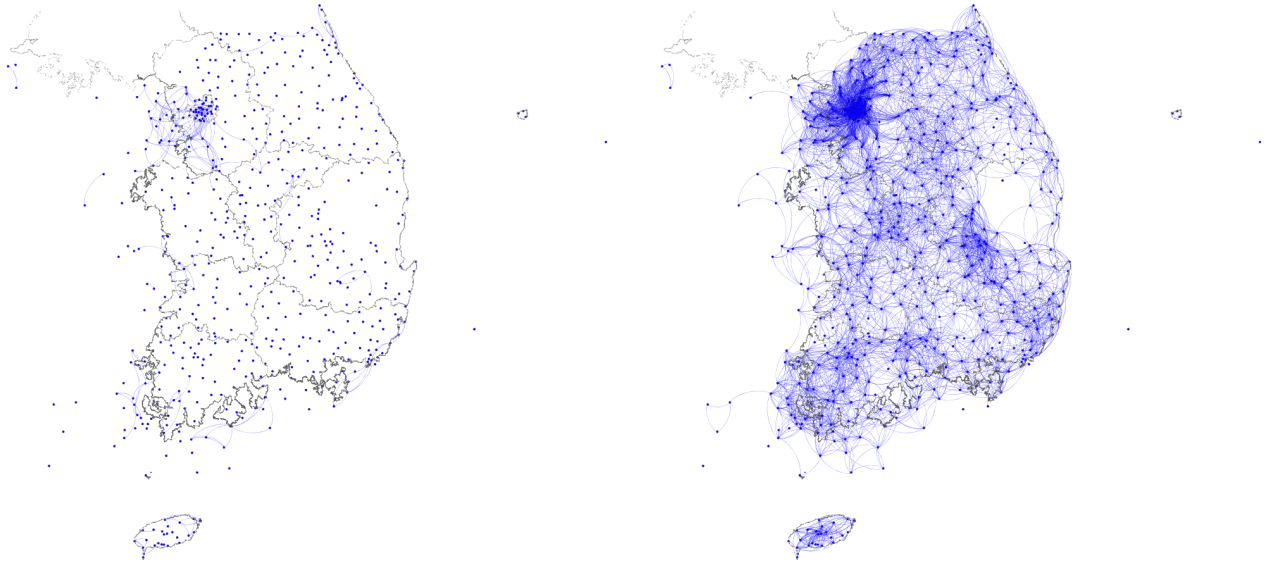
Table 20: Accuracy of estimates for each similarity measure

Meteorological element	**L_1**	**L_2**	**PCC** [1]	**MI** [2]
Wind direction	104.574	105.624	102.786	**101.424**
Wind speed	1.228	**1.224**	1.306	1.317
Temperature	**1.241**	**1.241**	1.327	1.319
Humidity	**8.085**	8.086	8.757	8.829
Atmospheric pressure	**6.497**	**6.497**	8.134	7.256
Hourly precipitation	1.074	**1.065**	1.066	1.155
Rainfall occurrence	**0.151**	**0.151**	0.152	0.157

[1] Pearson correlation coefficient
[2] Mutual information

4.5.2 Similarity Measure

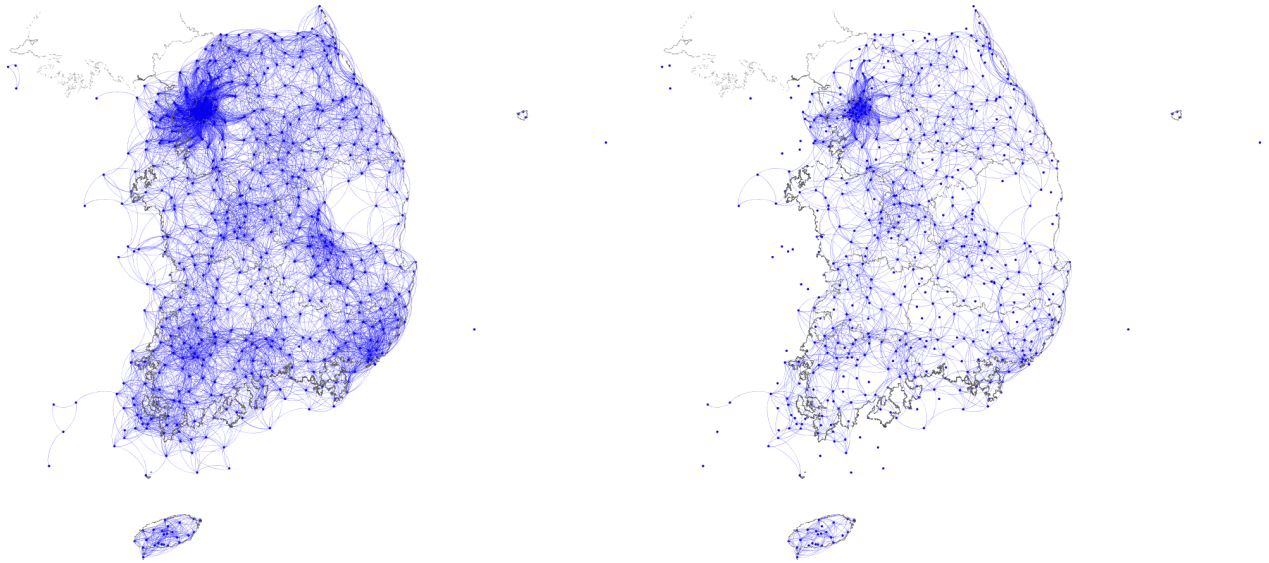
Section 4.3 describes four measures used to calculate the similarity between two observation stations. To compare the usefulness of each measure, the accuracy of the estimates predicted by the Madsen-Allerup method [AMV97] is examined. The Madsen-Allerup technique selects the stations similar to the target station, then uses the observed values of selected stations to obtain the estimate of the target station; therefore, the higher the quality of the similarity measure, the more accurate the estimate. Table 20 shows the estimation accuracy of the Madsen-Allerup method for each similarity measure. In all subsequent experiments, we used the highest quality similarity measure for each meteorological element. Figure 17 shows the connected station pairs with a similarity greater than 0.5.



(a) Wind direction

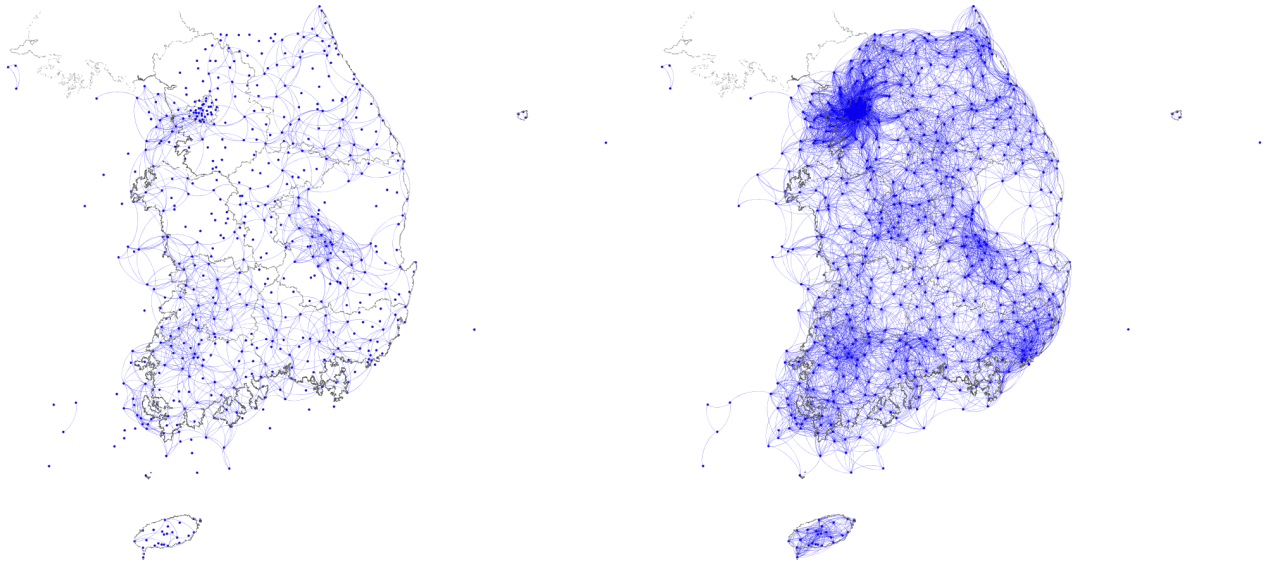
(b) Wind speed

Figure 17: Similarity map for different meteorological elements



(c) Temperature (d) Humidity

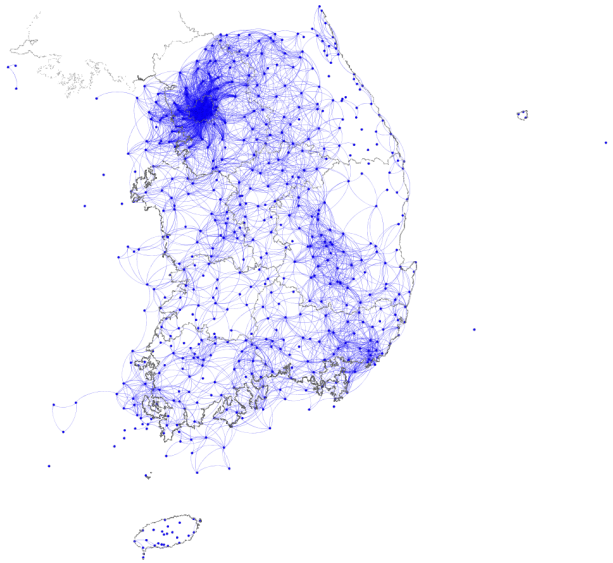
Figure 17: Similarity map for different meteorological elements (cont.)



(e) Atmospheric pressure

(f) Precipitation

Figure 17: Similarity map for different meteorological elements (cont.)



(g) Rainfall occurrence

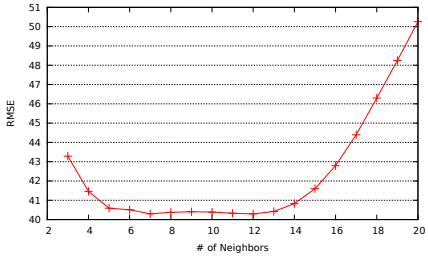
Figure 17: Similarity map for different meteorological elements (cont.)

Table 21: Optimal number of neighboring stations per meteorological element

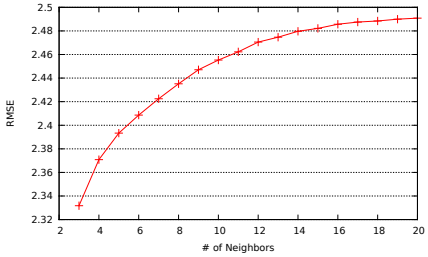
Meteorological element	Optimal # of neighbors
Wind direction	7
Wind speed	3
Temperature	11
Humidity	20
Atmospheric pressure	3
Hourly precipitation	8
Rainfall occurrence	10

4.5.3 Selecting Neighboring Stations

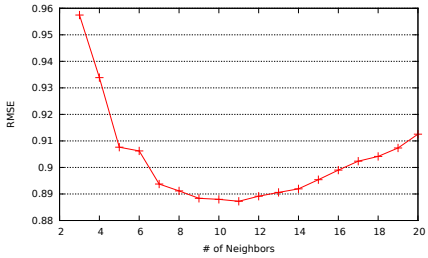
In Section 5, we proposed MOGA to select input variables to improve SVR performance and speed. Figure 18 shows the accuracy of estimates based on the number of neighboring stations selected by MOGA. The greater the number of parameters (over a certain amount), the worse the performance of the SVR, and the longer it takes to train. The optimal number of neighboring stations with the best performance differs with the meteorological element. Table 21 shows the optimal number of neighboring stations according to each meteorological element. All subsequent experiments were fixed using the optimal number of neighbors. Table 22 compares the estimation accuracy of SVR when neighboring stations were selected randomly, with the accuracy of SVR when neighboring stations were selected using MOGA. We confirm that selection of neighbors using MOGA improves the estimation performance of SVR.



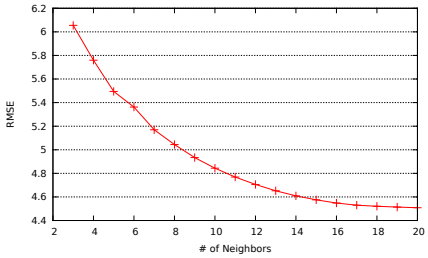
(a) Wind direction



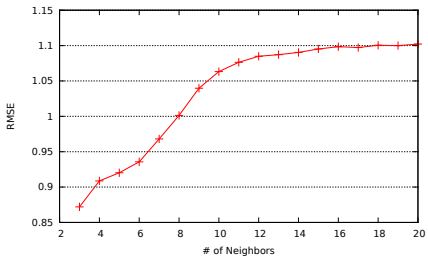
(b) Wind speed



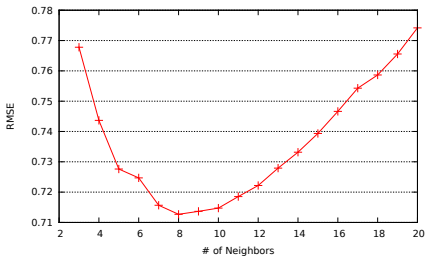
(c) Temperature



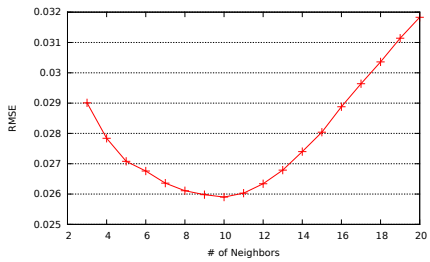
(d) Humidity



(e) Atmospheric pressure



(f) Precipitation



(g) Rainfall occurrence

Figure 18: Accuracy of estimates according to the number of selected neighboring stations

Table 22: Comparison of SVR estimation accuracy with neighboring stations selected randomly or by MOGA

		(RMSE)
Weather Element	**Random**	**MOGA**
Wind direction	50.390	**48.499**
Wind speed	2.523	**2.513**
Temperature	0.970	**0.902**
Humidity	5.216	**5.038**
Atmospheric pressure	1.066	**1.063**
Hourly precipitation	0.847	**0.762**
Rainfall occurrence	0.028	**0.026**

4.5.4 Comparison of Estimation Models

Table 23 shows the accuracy of estimates for each estimation model. Estimation using SVR model is better than that using Cressman or Barnes algorithms. Hourly precipitation does not show much improvement compared to other meteorological elements. Because there are many more days without rain than with rain, there is rather sparse data distribution for rainy days, which results in learning difficulties.

Table 24 shows the execution time of spatial quality control according to each estimation model. The execution time might be considered of little importance as a single process of spatial quality control can be executed in a very small time. But if a quality control process should be performed in a centralized single facility, a large number of meteorological data from every observational station need to be inspected in real time. For example in our test data, there are 572 stations and they collect 7 kinds of meteorological observation data. It takes about 5.77 seconds to inspect every data

Table 23: Comparison of estimation accuracy based on estimation model

(RMSE)

Meteorological element	**Cressman**	**Barnes**	**SVR**
Wind direction	53.568	75.470	**48.341**
Wind speed	2.347	2.315	**2.179**
Temperature	1.180	2.583	**0.880**
Humidity	6.755	12.767	**4.582**
Atmospheric pressure	5.663	11.601	**0.847**
Hourly precipitation	**0.583**	0.833	**0.583**
Rainfall occurrence	0.071	0.137	**0.021**

from every station using the Cressman method and it should be executed in every minute. Moreover, the execution time becomes more important as the number of stations and the kind of data get bigger and the time interval for collecting data get shortened.

Spatial quality control is fastest using the Barnes algorithm, but the accuracy of the estimation is very poor. Spatial quality control using SVR is approximately 6 times faster than that using the Cressman algorithm, but more time is required to learn the SVR model. However, as it does not give weight to more recent data in the learning process, there is no need to learn the model every time the spatial quality control is performed. If the model uses sufficient previous data, the performance of spatial quality control is not adversely affected, even if the learning cycle for model updates are only once a week or a month.

Table 24: Execution time for spatial quality control based on estimation model

Estimator	Average time spent in learning one model (second)	Average time spent in determining one observation (second)
Cressman	—	$1.442e^{-3}$
Barnes	—	$8.427e^{-5}$
SVR	6.839	$2.303e^{-4}$

4.5.5 Size of Training Set

In general, the higher the number of training samples in the SVR, the higher the accuracy of the estimate, but the longer the learning time. Table 25 shows the accuracy of estimates based on the number of training samples. Exceptionally, in the case of wind speed, the performance tends to decrease as the number of training samples increases. Figure 18 also shows that the fewer input variables of SVR, the better the performance with regards to wind speed. In the present model structure, it is difficult to learn wind speed; thus, over-fitting seems to occur if the model becomes overly complicated.

Figure 19 shows the learning time according to the number of training samples, and Figure 20 shows the time taken purely for spatial quality control, excluding learning time. Theoretically, the time taken to test the SVR model is not affected by the size of the training set, but as the training set grows, the complexity of the model becomes larger (e.g., the number of support vectors increases), and the time required for the test also increases. However, as the number of samples increases, the increase in test time gradually decreases. The test time is expected not to increase after the number

Table 25: Accuracy of estimated values based on the size of the training set

(RMSE)

Meteorological element	**5,000**	**10,000**	**15,000**	**20,000**	**25,000**	**30,000**
Wind direction	43.820	42.831	42.298	41.948	41.691	**41.481**
Wind speed	**2.363**	2.365	2.367	2.369	2.369	2.370
Temperature	0.902	0.879	0.870	0.863	0.860	**0.857**
Humidity	4.710	4.330	4.130	3.998	3.904	**3.831**
Atmospheric pressure	0.871	0.837	0.817	0.807	0.797	**0.785**
Hourly precipitation	0.763	0.746	0.736	0.732	0.727	**0.724**
Rainfall occurrence	0.026	0.025	**0.024**	**0.024**	**0.024**	**0.024**

of samples reaches a certain point. Experiments on all observation stations using 30,000 samples took approximately 15 days on seven machines. Due to time limitations, we could not experiment with more samples, but there seems to be room for further performance improvement. In this study, all the stations were analyzed together, but the burden of the learning time would not be as great if each test were conducted separately for each observation station.

4.5.6 Result of Spatial Quality Control

Table 26 shows the results of applying the proposed spatial quality control procedure to actual data. As described above, the spatial quality control applies only to observations that are determined as normal during basic quality control. Therefore, values that did not pass the basic quality control are classed as uninspected during spatial quality control. The high ratio of uninspected observations of humidity and atmospheric pressure is due to the lack of measuring instruments for those elements in many observation stations.

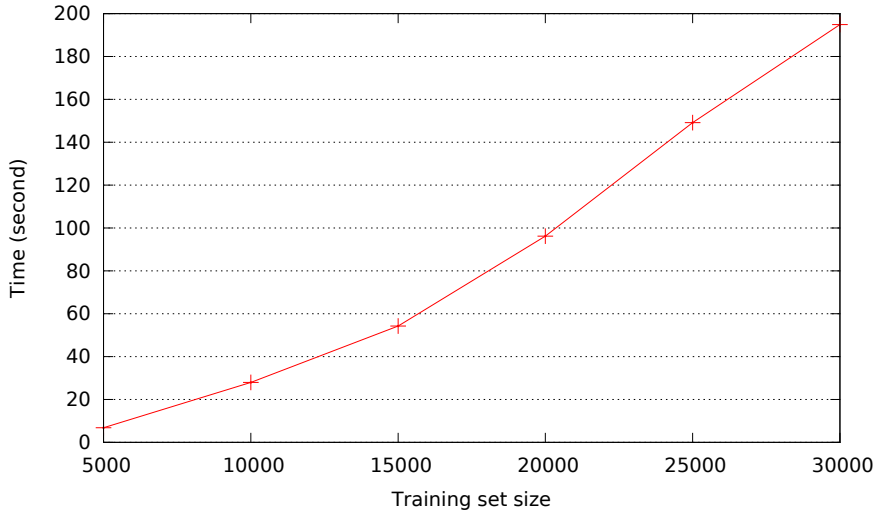


Figure 19: Average time spent on learning one model depending on the size of the training set

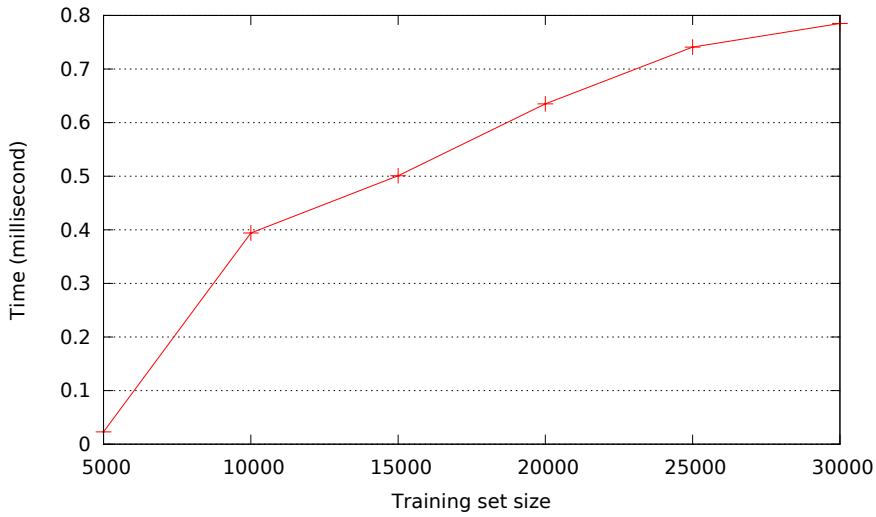


Figure 20: Time spent on determining one value depending on the size of the training set

Table 26: Results of the proposed spatial quality control method

Meteorological element	**Normal**	**Suspect**	**Error**	**Uninspected**
Wind direction	72.9%	6.32%	$6.31e^{-1}$%	20.2%
Wind speed	75.7%	3.49%	$6.56e^{-1}$%	20.2%
Temperature	93.8%	$3.98e^{-1}$%	$9.08e^{-2}$%	5.67%
Humidity	52.8%	$2.08e^{-1}$%	$5.33e^{-2}$%	47.0%
Atmospheric pressure	36.4%	$4.75e^{-4}$%	$1.60e^{-4}$%	63.6%
Hourly precipitation	87.1%	$8.73e^{-1}$%	2.99%	9.04%
Rainfall occurrence	89.2%	1.38%	$4.16e^{-1}$%	9.04%

Chapter 5

Conclusions

In this thesis, we proposed Machine learning based approaches to deal with abnormalities in observation data. The subject includes how to detect abnormalities and how to get proper values to substitute for the detected abnormalities. The experiments on the large volume of real-world observational data, that is, meteorological data, showed that our approaches outperformed the traditional approaches based on interpolations.

We presented three ML-based approaches to correct abnormal values in observational data. We compared them with three interpolation methods: linear interpolation, polynomial interpolation and spline interpolation, using the same input attributes. Furthermore, we used additional information about other elements beyond the target element for better estimation. Also, the data from neighbor observational points were employed to give support to ML-based approaches. We tested proposed methods on automated weather station data consisting of wind direction, wind speed, temperature, relative humidity, air pressure, mean sea level pressure, rainfall occurrence and hourly precipitation. Support vector regression (SVR) outperformed the interpolation methods for all weather elements except hourly precipitation. Decision tree showed the best performance over all the other approaches in estimating wind direction and rainfall occurrence. Experimental results show that additional information improved the estimation accuracy.

However, hourly precipitation was hard to be estimated by ML-based approaches. For hourly precipitation, the more input attributes are, the worse the performances of models become. Traditional interpolations still worked well on estimation of hourly precipitation.

Also, we proposed a method to detect the spatial abnormality of observational data using SVR. First, the value of the corresponding point was predicted using observations made in the surrounding area, then any abnormality was detected by checking whether the observation differs from the predicted value outside of a predetermined range. SVR was used to create a model to predict the value of observational point. In addition, we used multi-objective genetic algorithm to select SVR input variables to improve model performance and to reduce computation time. Experiments on actual weather data, comprising wind direction, wind speed, temperature, humidity, atmospheric pressure, hourly precipitation and rainfall occurrence, show that using SVR is more accurate than the existing Cressman or Barnes methods for estimating the value of an observation station. Therefore, more accurate anomaly detection is possible through more accurate predictions. If the model can be learned in advance for a fixed cycle rather than learning the model every time, the proposed method has an acceptable execution time. A limitation of the method is that pre-accumulated data is required, but it was confirmed through experiments that data collected over approximately one year provides sufficiently high performance.

As the proposed method are not designed to treat a specific data, it can be applied to other observational data such as sea surface temperature, radiation level, sunshine duration and cloud height. Other valuable research could

examine whether state-of-the-art learning techniques such as deep learning can yield more accurate predictions than machine learning techniques we used, which was not attempted here due to limitations of the system environment. In addition to accurate predictions, additional studies are required on the acceptable difference between the observation and the estimate which we set using the standard deviation during spatial quality control. Furthermore, it will be interesting to compare the anomaly detection technique with unsupervised learning technology as opposed to that based on prediction using supervised learning. Although our methods were successful at most meteorological elements, both of the detection and the correction on hourly precipitation was hard to achieve by machine learning. Results displayed a tendency that the performances got worse as we used more information as input variables. Therefore, additional studies are needed to overcome a sparsity of data and to prevent overfitting issues. There are different types of abnormal values: missing values, consistently biased values, fluctuant values, and so on. It is expected that, if we classify them well, we can detect or recover them more successfully than in this study, by using various methods tailored to fit the classes to which they belong.

Bibliography

[AMV97] P. Allerup, H. Madsen, and F. Vejen. A comprehensive model for correcting point precipitation. *Hydrology Research*, 28(1):1–20, 1997.

[aso] National weather service: automated surface observing system. http://www.nws.noaa.gov/asos/. Accessed: 2014-04-14.

[Atk89] K. Atkinson. *An Introduction to Numerical Analysis*. Wiley, 2nd edition, 1989.

[Bar64] S. L. Barnes. A technique for maximizing details in numerical weather map analysis. *Journal of Applied Meteorology*, 3(4):396–409, 1964.

[BC06] J.-C. Baltazar and D. E. Claridge. Study of cubic splines and Fourier series as interpolation techniques for filling in short periods of missing building energy use and weather data. *Journal of Solar Energy Engineering*, 128(2):226–230, 2006.

[BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

[BL94] V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley & Sons, 1994.

[Bre17] L. Breiman. *Classification and Regression Trees*. Routledge, 2017.

[CA82] R. Carbone and J. S. Armstrong. Note. evaluation of extrapolative forecasting methods: results of a survey of academicians and practitioners. *Journal of Forecasting*, 1(2):215–217, 1982.

[CBK09] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[CD14] T. Chai and R. R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)?–arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3):1247–1250, 2014.

[CE54] P. J. Clark and F. C. Evans. Distance to nearest neighbor as a measure of spatial relationships in populations. *Ecology*, 35(4):445–453, 1954.

[CM07] P. Cortez and A. d. J. R. Morais. A data mining approach to predict forest fires using meteorological data. In *New trends in artificial intelligence : proceedings of the 13th Portuguese Conference of Artificila Intelligence*, pages 512–523. Associação Portuguesa para a Inteligência Artificial (APPIA), 2007.

[Coe00] C. A. Coello. An updated survey of GA-based multiobjective optimization techniques. *ACM Computing Surveys (CSUR)*, 32(2):109–143, 2000.

[Cre59] G. P. Cressman. An operational objective analysis system. *Monthly Weather Review*, 87(10):367–374, 1959.

[CV95] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[Cyb89] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[DBK+97] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik, et al. Support vector regression machines. *Advances in Neural Information Processing Systems*, 9:155–161, 1997.

[DGD+04] C. Daly, W. Gibson, M. Doggett, J. Smith, and G. Taylor. A probabilistic-spatial approach to the quality control of climate observations. In *14th AMS Conference on Applied Climatology*, pages 13–16, 2004.

[DPAM02] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[DPJ+00] P. Doraiswamy, P. Pasteris, K. Jones, R. Motha, and P. Nejedlik. Techniques for methods of collection, database management and distribution of agrometeorological data. *Agricultural and Forest Meteorology*, 103(1):83–97, 2000.

[EGG11] J. Estévez, P. Gavilán, and J. V. Giráldez. Guidelines on validation procedures for meteorological data from automatic weather stations. *Journal of Hydrology*, 402(1):144–154, 2011.

[FF93] C. M. Fonseca and P. J. Fleming. Multiobjective genetic algorithms. In *IEE colloquium on Genetic Algorithms for Control Systems Engineering (Digest No. 1993/130)*. UK : IEE, 1993.

[FF98] C. M. Fonseca and P. J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 28(1):26–37, 1998.

[FHQ04] S. Feng, Q. Hu, and W. Qian. Quality control of daily meteorological data in china, 1951–2000: a new dataset. *International Journal of Climatology*, 24(7):853–870, 2004.

[FZ07] B. Fornberg and J. Zuev. The runge phenomenon and spatially variable shape parameters in RBF interpolation. *Computers & Mathematics with Applications*, 54(3):379–398, 2007.

[Gan88] L. S. Gandin. Complex quality control of meteorological observations. *Monthly Weather Review*, 116(5):1137–1156, 1988.

[GDE04] D. Y. Graybeal, A. T. DeGaetano, and K. L. Eggleston. Improved quality assurance for historical hourly temperature and humidity: development and application to environmental analysis. *Journal of Applied Meteorology*, 43(11):1722–1735, 2004.

[GKRS88] N. Guttman, C. Karl, T. Reek, and V. Shuler. Measuring the performance of data validators. *Bulletin of the American Meteorological Society*, 69(12):1448–1452, 1988.

[Gru69] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[HA04] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004.

[Hea96] M. T. Heath. *Scientific Computing: An Introductory Survey*. McGraw-Hill Higher Education, 2nd edition, 1996.

[HFH+09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[HGS+05] K. Hubbard, S. Goddard, W. Sorensen, N. Wells, and T. Osugi. Performance of quality assurance procedures for an applied climate information system. *Journal of Atmospheric and Oceanic Technology*, 22(1):105–112, 2005.

[HKI+18] J.-H. Ha, Y.-H. Kim, H.-H. Im, N.-Y. Kim, S. Sim, and Y. Yoon. Error correction of meteorological data obtained with mini-AWSs based on machine learning. *Advances in Meteorology*, 2018, 2018.

[HMC+01] N. S. Holter, A. Maritan, M. Cieplak, N. V. Fedoroff, and J. R. Banavar. Dynamic modeling of gene expression data. *Proceedings of the National Academy of Sciences*, 98(4):1693–1698, 2001.

[Hol92] J. H. Holland. *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT press, 1992.

[HOR96] T. Hill, M. O'Connor, and W. Remus. Neural network models for time series forecasts. *Management Science*, 42(7):1082–1092, 1996.

[HR76] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[Hub01] K. Hubbard. Multiple station quality control procedures. *Automated Weather Stations for Applications in Agricul-*

ture and Water Resources Management, pages 133–138, 2001.

[IM96] H. Ishibuchi and T. Murata. Multi-objective genetic local search algorithm. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 119–124. IEEE, 1996.

[Jar08] M. Jarraud. Guide to meteorological instruments and methods of observation (WMO-no. 8). *World Meteorological Organisation: Geneva, Switzerland*, 2008.

[Joa] T. Joachims. SVM^{light}: Support vector machine. http://svmlight.joachims.org.

[Joa98] T. Joachims. Making large-scale SVM learning practical. Technical report, Technical report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund, 1998.

[Kal00] S. A. Kalogirou. Applications of artificial neural-networks for energy systems. *Applied Energy*, 67(1):17–35, 2000.

[KCBH13] M. Kubik, P. J. Coker, J. F. Barlow, and C. Hunt. A study into the accuracy of using meteorological wind data to estimate turbine generation output. *Renewable Energy*, 51:153–158, 2013.

[KCS06] A. Konak, D. W. Coit, and A. E. Smith. Multi-objective optimization using genetic algorithms: a tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

[KHY+16] Y.-H. Kim, J.-H. Ha, Y. Yoon, N.-Y. Kim, H.-H. Im, S. Sim, and R. K. Choi. Improved correction of atmospheric pressure data obtained by smartphones through machine learning. *Computational intelligence and neuroscience*, 2016:4, 2016.

[Kim03] K.-J. Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1):307–319, 2003.

[KKY+15] N.-Y. Kim, Y.-H. Kim, Y. Yoon, H.-H. Im, R. K. Choi, and Y. H. Lee. Correcting air-pressure data collected by MEMS sensors in smartphones. *Journal of Sensors*, 2015, 2015.

[KY16] Y.-H. Kim and Y. Yoon. Spatiotemporal pattern networks of heavy rain among automatic weather stations and very-short-term heavy-rain prediction. *Advances in Meteorology*, 2016, 2016.

[LMKM14] M.-K. Lee, S.-H. Moon, Y.-H. Kim, and B. R. Moon. Correcting abnormalities in meteorological data by machine learning. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 888–893. IEEE, 2014.

[MI95] T. Murata and H. Ishibuchi. MOGA: Multi-objective genetic algorithms. In *IEEE International Conference on Evolutionary Computation*, volume 1, pages 289–294. IEEE, 1995.

[MS95] K. V. S. Murthy and S. L. Salzberg. *On Growing Better Decision Trees from Data*. PhD thesis, Citeseer, 1995.

[MSR+97] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In *Proceedings of the 7th International Conference on Artificial Neural Networks*, volume 1327 of *Lecture Notes in Computer Science*, pages 999–1004. Springer, 1997.

[OFG97] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *Proceedings of*

the 1997 IEEE Workshop on Neural Networks for Signal Processing VII, pages 276–285. IEEE, 1997.

[OK02] J. Oerlemans and E. J. Klok. Energy balance of a glacier surface: analysis of automatic weather station data from the Morteratschgletscher, Switzerland. *Arctic, Antarctic, and Alpine Research*, 34(4):477–485, 2002.

[Pla98] J. Platt. Sequential minimal optimization: a fast algorithm for training support vector machines. Technical report, April 1998.

[Qui86] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[Qui93] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[RDO92] T. Reek, S. R. Doty, and T. W. Owen. A deterministic approach to the validation of historical daily temperature and precipitation data from the cooperative network. *Bulletin of the American Meteorological Society*, 73(6):753–762, 1992.

[RHW85] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[SBCR09] G. Sciuto, B. Bonaccorso, A. Cancelliere, and G. Rossi. Quality control of daily rainfall data with neural networks. *Journal of Hydrology*, 364(1):13–22, 2009.

[SBS+04] P. Svensson, H. Björnsson, A. Samuli, L. Andresen, L. Bergholt, O. E. Tveito, S. Agersten, O. Pettersson, and F. Vejen. Quality control of meteorological observations:

description of potential HQC systems. *met.no Report*, (10), 2004.

[SFA+00] M. A. Shafer, C. A. Fiebrich, D. S. Arndt, S. E. Fredrickson, and T. W. Hughes. Quality assurance procedures in the oklahoma mesonetwork. *Journal of Atmospheric and Oceanic Technology*, 17(4):474–494, 2000.

[SH03] A. Stoppa and U. Hess. Design and use of weather derivatives in agricultural policies: the case of rainfall index insurance in Morocco. In *International Conference "Agricultural Policy Reform and the WTO: Where are we heading?"*, 2003.

[SK12] J.-H. Seo and Y.-H. Kim. Genetic feature selection for very short-term heavy rainfall prediction. *Convergence and Hybrid Information Technology*, pages 312–322, 2012.

[SKBM00] S. K. Shevade, S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy. Improvements to the SMO algorithm for svm regression. *IEEE Transactions on Neural Networks*, 11(5):1188–1193, 2000.

[SLK14] J.-H. Seo, Y. H. Lee, and Y.-H. Kim. Feature selection for very short-term heavy rainfall prediction using evolutionary computation. *Advances in Meteorology*, 2014, 2014.

[Slu09] R. Sluiter. Interpolation methods for climate data: literature review. *KMNI Intern Rapport*, (4):1–24, 2009.

[Tan90] B. D. Tanner. Automated weather stations. *Remote Sensing Reviews*, 5(1):73–98, 1990.

[Vap00] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2000.

[vdWGvdB+05] R. van de Wal, W. Greuell, M. van den Broeke, C. Reijmer, and J. Oerlemans. Surface mass-balance observations and automatic weather station data along a transect near Kangerlussuaq, West Greenland. *Annals of Glaciology*, 42(1):311–316, 2005.

[VGS97] V. Vapnik, S. E. Golowich, and A. Smola. Support vector method for function approximation, regression estimation, and signal processing. *Advances in Neural Information Processing Systems*, pages 281–287, 1997.

[VL63] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780, 1963.

[Wad87] C. G. Wade. A quality control program for surface mesometeorological data. *Journal of Atmospheric and Oceanic Technology*, 4(3):435–453, 1987.

[XZ04] L. Xiujuan and S. Zhongke. Overview of multi-objective optimization methods. *Journal of Systems Engineering and Electronics*, 15(2):142–146, 2004.

[YHG08] J. You, K. G. Hubbard, and S. Goddard. Comparison of methods for spatially estimating station temperatures in a quality control system. *International Journal of Climatology*, 28(6):777–787, 2008.

[YLB03] H. Yang, L. Lu, and J. Burnett. Weather data and probability analysis of hybrid photovoltaic–wind power generation systems in Hong Kong. *Renewable Energy*, 28(11):1813–1824, 2003.

[Zah04] I. Zahumenskỳ. Guidelines on quality control procedures for data from automatic weather stations. *World Meteorological Organization*, 2004.

[ZT99] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

국문 초록

관측 시스템에서 수집되는 관측 자료는 여러 현상을 예측하고 분석하는데 중요한 역할을 한다. 그러나 관측 자료에는 여러가지 이유로 상당한 양의 비정상 값이 존재한다. 이런 비정상 값을 찾아내고 처리하는 일은 매우 중요하다. 가장 대표적이고 중요한 관측 자료 중 하나는 기상 관측 자료이다. 본 논문에서는 비정상 값을 탐지하고 보정하기 위해서 기계 학습을 기반으로 한 새로운 방법을 제시하고, 다양한 종류의 실제 기상 관측 자료에 테스트했다.

기상학에서는 비정상 값을 찾는 과정을 품질 관리라고 부른다. 품질 관리 과정에서 발견된 비정상 값을 보정하기 위해서 기계 학습 기법을 이용한 세 가지 추정 모델을 제시했다. 우리는 제시된 모델을 기존의 추정 모델, 보간법과 비교했다. 목표가 되는 기상 요소만 사용하는 보간법과는 달리, 제안한 모델은 관련된 다른 기상 요소들과 주변의 기상 관측 지점의 자료도 사용한다. 신뢰할만한 기관에서 수집된 실제 자료에 대해서 실험해본 결과, 제안한 방법은 보간법에 비해서 RMSE를 8.35% 감소시켜, 더 정확하게 목표값을 추정할 수 있음을 보였다. 다시 말해, 우리가 제시한 방법은 예전 방법들보다 더 적절하게 비정상값들을 대체할 수 있다.

또한 우리는 공간적인 관점에서 관측 자료 중에 비정상 값을 찾아내기 위한, 향상된 품질 관리 기법을 제시한다. 관측값을 예측하기 위해서 지지 벡터 회귀가 사용되었다. 예측된 값과 실제 관측값의 차이를 통해서 관측값이 정상인지 비정상인지를 판별한다. 또한 지지 벡터 회귀의 성능을 향상시키고 수행 시간을 줄이기 위해서, 지지 벡터 회귀의 입력 변수

를 선별한다. 선별 과정에서 유사도와 공간성 다양성이라는 두 가지 목적 함수를 동시에 최적화하기 위해, 다목적함수 유전 알고리즘이 사용되었다. 실제 자료를 사용한 실험에서 지지 벡터 회귀를 이용한 추정은 기준이 되는 방법들에 비해서, 경쟁력 있는 수행 시간을 유지하면서 RMSE를 45.44%만큼 감소시켰다.